

CAPÍTULO 1

TRABAJOS PREVIOS

En este trabajo se plantea la obtención de un *modelo de agentes y su ambiente* que puede aplicarse en situaciones de comercio electrónico donde los *agentes* interactúan entre ellos considerando que los elementos del lenguaje que utilizan para comunicarse pueden estar sujetos a ambigüedad y es necesario encontrar algún tipo de equivalencia porque algunos tienen una organización de conceptos (*ontología*) diferente. La elección de papeles que le permiten a un agente alcanzar sus propósitos se hace mediante un módulo de planeación. Considerando esto, iniciaremos este capítulo exponiendo los trabajos previos de los temas relacionados con el trabajo que se desarrolla en esta tesis.

Desde épocas remotas los seres humanos han utilizado el intercambio de bienes para subsistir. La integración de personas con diferentes culturas y el desarrollo de los medios de comunicación han propiciado el fenómeno de la Globalización y esta a su vez favorece el intercambio comercial entre diferentes puntos del mundo. Uno de los más recientes paradigmas en las transacciones comerciales es el *comercio electrónico*, inicialmente popularizado con el intercambio electrónico de datos (EDI) y actualmente tomando diferentes modalidades de acuerdo a los participantes, como el caso de B2B (negocios entre empresas) o B2C (venta a consumidores).

Los participantes de las transacciones comerciales son entre otros: vendedores, compradores, ofertadores, demostradores, cobradores, repartidores. Cada participante se comporta de acuerdo a su papel que tiene en el escenario específico, por ejemplo un comprador de maíz que fabrica tortillas en un momento dado toma el papel de comprador y en otro contexto el de vendedor.

La caracterización de los elementos del comercio electrónico y las etapas de las transacciones comerciales se indica en la sección 1.1. En nuestro trabajo enfocamos algunos problemas existentes en la etapa de búsqueda de una transacción comercial. Entre ellos, la necesidad de interactuar con los proveedores del bien que se necesita y que en un momento dado se puede delegar para que la búsqueda la realice un agente en vez de una persona física. De aquí el interés por el tema de *agentes* expuesto en la sección 1.2.

En la comunicación entre participantes de una transacción comercial, es necesario contar con un lenguaje común con el que se entiendan los participantes y así llegar a acuerdos para cerrar la transacción. En el comercio electrónico un problema se deriva de la falta de presencia física por una parte, y donde surgen problemas como son la autenticación de los participantes, la seguridad y la entrega remota de los bienes, en parte, el problema de la entrega de bienes puede aminorarse en

empresas que cuentan con sucursales ya que si se solicita un producto en un punto geográfico distante, en forma automática se enruta el pedido hacia la sucursal más cercana al punto de solicitud. Para nosotros estos problemas son motivo de otra línea de investigación por lo que es de nuestro interés los problemas relacionados con la reducción de la ambigüedad en los mensajes que intercambian los participantes de una transacción, por lo que exponemos en la sección 1.3 el tema de la representación del conocimiento, en particular presentamos el tema de *ontologías* y algunos trabajos relacionados sobre esto.

Un agente se caracteriza porque su funcionamiento está orientado a alcanzar algún propósito en forma autónoma y proactiva, para esto es conveniente que cuente con un plan que le permita elegir las acciones más apropiadas. Las decisiones de las acciones que realiza o va a realizar también están influidas por el ambiente en que se encuentra y por los recursos y características del mismo. Las técnicas de *planeación* (sección 1.4) permiten a un sistema deliberar sobre las acciones más propicias para alcanzar un objetivo.

El motivo de la sección 1.5 es plantear la problemática involucrada en los eventos inesperados que son sucesos que se sabe pueden ocurrir pero no se tiene certeza de cuando será, por lo que se debe contar con la capacidad de reaccionar de la manera más favorable cuando se presenten esto.

En la sección 1.6 exponemos las conclusiones derivadas de los problemas y trabajos previos expuestos en las secciones de este capítulo.

1.1 COMERCIO ELECTRÓNICO

En esta sección se trata el tema del comercio electrónico, indicando su definición, la caracterización de productos que se intercambian y los elementos tecnológicos que lo soportan. Son de particular interés las etapas de una transacción comercial dado que en éstas se encuentran varios problemas que se inician desde que un cliente busca un producto para satisfacer una necesidad o bien, cuando un proveedor quiere promover un producto.

La búsqueda de un producto en un contexto de operaciones comerciales implica la interacción con diferentes proveedores, con quienes se verifican las características deseadas contra las que se ofrecen además de intentar minimizar el costo de adquisición, al mismo tiempo que se intenta contar con el bien en el tiempo más breve para utilizarlo.

Una alternativa de solución para ahorrar tiempo en la búsqueda de productos es la estandarización de los mismos. El problema con esto es, que en ocasiones los clientes requieren productos con algunas variantes en volúmenes significativos que resultan convenientes para negociarlas con los fabricantes.

Un cliente puede intentar encontrar el o los productos que estima satisfacen sus necesidades visitando directamente a los proveedores; esto es común en mercados locales. En otros casos se intenta localizar los bienes de interés mediante la búsqueda en medios electrónicos como por ejemplo utilizando el teléfono, el fax, correo electrónico o mediante la navegación en páginas web de algunos proveedores.

El *comercio electrónico* [Nonel 1999] es el concepto *que abarca las transacciones comerciales transmitidas electrónicamente, usando para ello las redes telemáticas y alguna forma de pago (como el dinero electrónico* [Parunach 1996] *o algún convenio de pago a la entrega)*. Esto incluye entre otros: intercambio de bienes, servicios, información digital, publicidad, licitaciones, concursos, subastas digitales, investigación de mercados,

integración del sistema de transferencia de información electrónicamente (Electronic Data Interchange, EDI) como por ejemplo el envío de una factura con validez jurídica.

El comercio electrónico es una etapa de la evolución del comercio, considerado desde el antiguo intercambio de bienes (trueque), seguido por la invención de la moneda (o artículo común de intercambio, como el cacao), los créditos, los bancos, las tarjetas de débito y crédito y recientemente el dinero electrónico.

En el mercado digital se considera la existencia de dos tipos de productos, tal y como lo define Nicholas Negroponte en su libro "Being Digital":

- a) Productos de bits, aquellos que pueden distribuirse a través de la red, por ejemplo, programas de computadora, música, libros, servicios, etc. Estos presentan el inconveniente de poder ser duplicados por el cliente sin perder su calidad.
- b) Productos de átomos, aquellos que necesitan ser distribuidos por medios de transporte tradicionales (aviones, barcos, camiones, etc.), por ejemplo, plantas, mesas, computadoras, azulejos, etc. En estos productos se presentan problemas en su traslado como el caso de artículos de cristal o componentes electrónicos, otros problemas son en el transporte de productos perecederos como alimentos debido a sus requerimientos de refrigeración y manejo.

El lugar en donde se localiza una empresa en Internet se conoce como su sitio y se identifica por una dirección llamada localización universal del recurso (Universal Resource Location, URL). Una empresa con presencia en Internet [Amor 2000, pp 61-107] debe contar infraestructura que incluye elementos como sistemas de información, servidores, equipo de telecomunicaciones y convenios con empresas de servicios, es decir:

- a) Un *servidor* de cómputo con un procesador lo más veloz posible, varios Gigabytes de memoria principal (de ser posible), discos duros con la mayor capacidad posible y conexión a Internet mediante accesorios de comunicaciones del mayor ancho de banda al que pueda acceder la empresa.
- b) Una *dirección electrónica* única (IP address) y un dominio propio registrado⁶ (URL) (generalmente la nomenclatura es www. Empresa.com), donde Empresa es la razón social o un acrónimo relativo a la misma.
- c) Se necesita una infraestructura compleja de diferentes *servidores, protocolos, manejadores* (entre ellos de base de datos, impresión, páginas web), *soporte para pagos electrónicos*. Estos se integran en las *aplicaciones* que brindan servicio a los clientes y a las áreas internas que lo requieren. Las bases de datos (pueden estar distribuidas y/o fragmentadas, ser una o varias). Se pueden tener programas que dan diferentes servicios y que son operados por diferentes usuarios del sistema mismos que pueden estar en diferentes puntos geográficos. Para su funcionamiento se considera la interacción entre diferentes servidores como los de páginas web, base de datos, de intercambio de información, de impresión; estos dependiendo de la funcionalidad de que consista el sistema, además se consideran los elementos de seguridad que acompañan a estas. Generalmente las etapas para realizar una aplicación implican la realización de un análisis para la detección de

⁶ Un sitio donde se puede registrar es <http://www.domainbank.com>

requerimientos, diseño para especificar los componentes y sus interacciones, implantación en donde se desarrollan los componentes e interfaces hacia los distintos usuarios del sistema tanto internos como externos; la liberación del mismo involucra capacitación y mercadotecnia para los usuarios externos de la empresa y que generalmente son los que obtendrán la funcionalidad del mismo.

- d) *Convenio* con empresas bancarias y de mensajería (por ejemplo: FedEx, DHL, Estafeta, etc.), para que realicen la distribución y entrega de los productos, en el caso de que no se tenga la infraestructura.
- e) Una unidad para recepción de comentarios, quejas, sugerencias y felicitaciones sobre los productos que distribuye la empresa, en algunos casos llamado call-center (centro de llamadas) o unidad de *servicios al cliente*.

La Navidad del 2000 [Aldunante 2000] fue especialmente aleccionadora para muchas empresas que venden productos en Internet porque debido a la demanda que se dio, las empresas *punto com* fueron incapaces de procesar físicamente la enorme cantidad de pedidos navideños que les solicitaron, aún trabajando 24 horas al día fue imposible evitar los retrasos en el envío y por lo tanto en las entregas, además que en ocasiones enviaron un producto de una calidad diferente a la esperada y que se presentaron errores en la entrega. Esta lección manifiesta que cuando se incursiona en los negocios electrónicos además de la infraestructura tecnológica es conveniente contar con alianzas estratégicas y suficiente personal para soportar los mercados Globales y pedidos masivos en periodos críticos.

El comercio electrónico (publicidad, ventas, compras, promociones, licitaciones, subastas, intercambios) puede tomar varios nombres, de acuerdo con los participantes, es decir:

- a) B2C (business to consumer) comercio de negocio a consumidor, es el tipo de comercio electrónico más popular. En Internet existen muchos sitios en este rubro, por ejemplo www.amazon.com, www.bn.com, www.palm.com. El caso de C2C lo consideramos como una forma de esta categoría debido a que es una empresa la que funge como intermediaria entre dos personas para realizar la transacción comercial. Algunas empresas de C2C son: www.deremate.com y www.subastando.com.
- b) B2B (business to business) [Tian 1999] comercio de negocio a negocio, es una de las formas de comercio electrónico con mayor crecimiento en los importes económicos manejados, un sitio de ejemplo de B2B es www.eficentrum.com.
- c) e-Gobierno, representado en múltiples formas, por ejemplo con sitios que ayudan en funciones de gobierno, por ejemplo, el pago de impuestos en Chile [Zerda 2001], el sistema de compras en México (el cual es el sistema electrónico de contrataciones con el gobierno federal, www.compranet.gob.mx). Por otra parte existen algunas propuestas para mejorar la democracia a través de Internet [Grönlund 2001] [Watson 2001] [Becker 2001] [Snellen 2001] incluyendo las votaciones [Mohen 2001] [Phillips 2001].

El comercio electrónico desde finales de los 60s mediante EDI [Chan 1998] contribuyó a mejorar el nivel de servicio de las empresas hacia los clientes, reduciendo los costos y el tiempo de procesamiento de una transacción comercial. Por ejemplo, en el caso de la empresa de telecomunicaciones Bell Atlantic [Sivori 1996] (la cual genera aproximadamente 1.4 millones de facturas al año), al implantar EDI con sus principales proveedores y clientes, logró ahorros de 350 mil dólares al año. Kambil [Kambil 1997] menciona ahorros de entre 300 mil y 700 mil dólares de acuerdo a las investigaciones realizadas en varias empresas que han utilizado el comercio electrónico para mejorar sus negocios. Empresas como Amazon, Cisco, Onsale, BidnAsk y VirtualVineyards son pioneros en nuevos modelos de negocios usando comercio electrónico. Otras empresas han utilizado el Internet para potenciar sus puntos fuertes como es el caso de Bimbo [Stok 2001], esta empresa obsequió equipo de computación y comunicaciones además de capacitación a sus principales clientes para que estos realicen sus pedidos y pagos a través de la Internet, logrando con esto importantes ahorros y evitando que su personal de distribución (principal activo) transporte valores o nueva productos sin demanda.

Las empresas que tienen necesidad de ingresar a Internet tienen dos opciones, generar una infraestructura dedicada a soportar las operaciones a través de Internet o adaptar sus sistemas existentes (llamados sistemas legados) para integrarlos a la nueva forma de brindar los productos o servicios sin descuidar los canales existentes (la forma tradicional en que los brinda).

El auge del comercio electrónico a través de Internet, en los últimos años ha estado creciendo como lo muestran los datos del Business Week [Tygar 1996] (véase la figura 0.1 en la Introducción en este documento) en donde las *proyecciones para el 2005 indican que aproximadamente un cuarto del comercio mundial se realizará a través de Internet*, aunque esto está sujeto a variaciones ocasionadas por eventos sociales que ocurran en el mundo durante el periodo considerado, hasta ahora y a pesar de los problemas suscitados, existen indicadores que permiten mantener esta creencia.

Regularmente la primera forma de comercio electrónico que realizan las empresas es colocando un catálogo de productos con fotografías y precios fijos [Lomuscio 2000], aunque en Internet con sus capacidades de procesamiento de datos es posible personalizar los precios de acuerdo a quien realiza las compras, por ejemplo asignar descuentos a clientes frecuentes, o sugerirles algún producto que puedan requerir con base a los que típicamente se adquieren. Estas características de personalización se han estudiado e implementado en algunos sistemas y la tendencia es hacia facilitar el comercio electrónico mediante la plataforma de Internet [Parkes 1999] aprovechando las ventajas de los sistemas distribuidos y otras tecnologías de software.

1.1.1 ESTRUCTURA DE LAS TRANSACCIONES DE COMERCIO ELECTRÓNICO

Cada transacción de comercio electrónico debe llevarse en su totalidad o bien cancelarse pero debe tener un inicio y un fin, de otra forma no se puede considerar como una transacción. Una *transacción de comercio electrónico* se puede definir como *un conjunto de operaciones que se realizan entre un cliente y un proveedor en donde se intercambia un producto (que ofrece el proveedor) por un valor que aporta el cliente*.

Las transacciones de comercio electrónico (figura 1.1) se realizan en diferentes etapas que van desde que al cliente le surge una necesidad e identifica un satisfactor que puede cubrirla hasta que lo recibe, el cual puede estar bien o mal y para lo cual se recurre a los servicios al cliente para emitir su queja o su beneplácito (aunque esté último se refleja si el cliente continua realizando compras). Un producto se entiende aquí como un bien, servicio o información que tiene una utilidad concreta para

el cliente. Las etapas intermedias entre estos dos puntos inician en la búsqueda de un producto, luego la recolección de información sobre su costo, realización del pago y finalmente la recepción del mismo que puede ser que el cliente recoja su producto o bien que alguien se lo entregue. Tygar [Tygar 1996] resalta la importancia de considerar atómicas a las transacciones de comercio electrónico (en forma similar a lo establecido en base de datos) es decir, se llevan a cabo completamente las operaciones, desde que el cliente solicita y realiza el pago hasta que lo recibe y en caso de existir alguna falla o anomalía en alguna etapa del proceso, la transacción debe aparecer como si no hubiese existido.

Una transacción comercial consiste de operaciones físicas y electrónicas. Esto trae como consecuencia que en ocasiones se considere una transacción terminada al concluir las operaciones electrónicas (esto es cuando se realiza el pago electrónico). Algunos productos (los de bits) pueden entregarse por medios electrónicos y propiamente concluye la transacción, aunque en los productos de átomos esta debiera concluir hasta que el cliente recibe su producto (mismo que en ocasiones es robado al encontrarse en un buzón y donde esto queda en la responsabilidad del cliente).

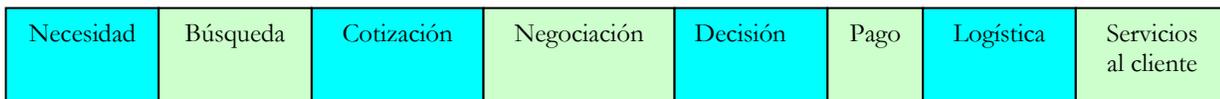


Figura 1.1 Etapas de una transacción de comercio electrónico

Durante las diferentes fases de la transacción se generan algunos documentos (ordenes de compra, facturas, notas de crédito, autorizaciones de pago por parte de un banco) los cuales, pueden ser impresos y enviados; o bien transmitidos electrónicamente. Es importante que el cliente y el proveedor se puedan entender durante las diferentes etapas de la transacción, desde la búsqueda, dado que pueden estar utilizando diferentes idiomas, o uno mismo pero, cada uno con diferente especialidad como el caso de un químico con un plomero.

Las etapas consideradas en una transacción comercial son:

- a) *Necesidad*, es cuando un cliente requiere de algún satisfactor para cubrir una necesidad, por lo que al identificar sus características, lo traduce a un conjunto de productos que considera, pueden contribuir, o bien resolverla totalmente. Un problema en esta etapa es la dificultad para detectar los problemas o necesidades. Otra situación es cuando es difícil describir con claridad el producto que se requiere como solución ante la necesidad detectada.
- b) *Búsqueda*, una vez que el cliente ha identificado los productos que necesita, debe considerar los sitios en donde puede hallarlos. Aquí se presenta el problema de dónde realizar la búsqueda. En ocasiones puede recurrir a amigos o personas que le hacen algunas sugerencias, en otros casos visita muchos sitios probables con el consiguiente gasto en tiempo. En los casos en que le es posible encontrar un sitio donde se encuentran productos similares a los que busca, el problema es encontrar exactamente lo que requiere, porque es posible que encuentre algo parecido y tenga que modificarlo para ajustarlo a sus necesidades, como en el caso de algunos muebles de oficina. Un cliente puede encontrar ayuda consultando catálogos de productos, el problema también es que en ocasiones no se publica toda la información sobre un producto y resulta necesario

consultar al proveedor para conocer los detalles de interés del cliente que faltan en dicho catálogo. En otros casos las palabras utilizadas en los catálogos no son completamente comprensibles para un cliente, sea porque están en otro idioma o porque resultan muy especializadas. El problema en la elaboración de catálogos es que algunos clientes requieren de mayor información mientras que otros requieren menos. Llevando esto al contexto del comercio electrónico, se encuentran problemas similares pues la metáfora del catálogo se ha mantenido, aquí es posible contactar al proveedor (que puede estar físicamente en otro lugar diferente a donde se ubica el cliente) mediante correo electrónico, aunque si el producto no satisface las necesidades se tiene el inconveniente de la demora en tiempo por concepto de logística.

- c) *Cotización*, una vez identificados los lugares donde se encuentran los productos de interés, se trata de obtener el costo de los mismos. Cuando esto no se encuentra dentro de la información que el cliente ha obtenido, entonces se solicita al proveedor y se espera recibir un documento con el costo del producto, las condiciones de pago y los servicios de transporte. En muchos sitios de Internet esto se hace mediante la metáfora de un carrito de autoservicio («shopping cart») el cual se puede visualizar en la pantalla de la computadora. Actualmente los mercados electrónicos («marketplaces»⁷) brindan mejores opciones a los clientes porque manejan varios proveedores ubicados en diferentes lugares y ofrecen mejores características de servicio considerando su ubicación geográfica.
- d) *Negociación*, esta etapa consiste en tratar de obtener las mejores condiciones (generalmente para el cliente) en una transacción comercial, por ejemplo, entrega de productos en un lugar diferente al esperado, muestras gratis, algún descuento, por volumen o por ser un cliente asiduo u obtener una mejor calidad por el mismo precio. Los problemas que se involucran son referentes al intercambio de mensajes.
- e) *Decisión*, una vez que el cliente tiene la información de los productos y cotizaciones se debe tomar una decisión sobre la adquisición del o los productos. Aunque esto es realizado por el cliente, esto afecta al proveedor porque si la transacción continua debe preparar el producto para enviarlo al cliente; de otra forma la transacción concluye. La decisión que favorece a un proveedor tiene mayor probabilidad de realizarse cuando los productos que el cliente adquiere satisfacen sus necesidades en vez de ser superfluos (que únicamente aumentan las cifras de ventas). Muchos clientes se inclinan por productos que son estándares en el mercado y dudan respecto a productos “novedosos” o sin una marca que los respalde.
- f) *Pago*, cuando tanto el cliente como el proveedor han aceptado los productos y las condiciones de pago y entrega se procede a realizar el pago. Hay que notar que si las condiciones son COD (cóbrese o devuélvase) esto se hará una vez que el producto llegue a manos del cliente con el inconveniente de que si el producto es rechazado, ya se ha incurrido en gastos de transportación. Cuando el pago es con tarjeta de crédito, la transacción puede hacerse en línea y casi de inmediato, este se utiliza en muchos sitios de Internet (portales, mercados electrónicos). Es importante notar que el auge del comercio electrónico ha creado formas alternativas de dinero [Panurach 1996], como el dinero electrónico («e-cash»), transferencia electrónica de fondos (en ocasiones utilizando cuentas bancarias referenciadas) y dinero digital (como las tarjetas prepagadas, en las cuales se va descontando el importe de las compras que realiza su poseedor).

⁷ Sitios como <http://www.amazon.com>, <http://www.bn.com>, <http://www.todito.com>

- g) *Logística*, es la etapa referente a la distribución y entrega de los productos que le sirven al cliente para satisfacer sus necesidades. Los productos de información (bits) como música, programas, artículos, etc. pueden enviarse a través de Internet. Los productos de átomos generalmente se pueden enviar mediante servicios propios de entrega de la empresa o por convenio con empresas como FedEx, DHL, Estafeta o algún otro. Generalmente el cliente puede monitorear el flujo de su pedido desde que lo realiza hasta que se le entrega por medio de una clave de rastreo («tracking») proporcionada al realizar su compra.
- h) *Servicios al Cliente*, también llamados centros de atención («call-center»), es un componente de una transacción de comercio electrónico que le permite al cliente notificar al proveedor si algún producto llegó con defectos que se le atribuyen a la fabricación o mal manejo durante el envío. También sirve para notificar cuando el cliente recibió productos adicionales a los esperados. Es a través de este medio que el cliente y el proveedor pueden negociar medidas compensatorias convenientes para ambas partes. Esta etapa también sirve para realizar cruces de información de los clientes de una empresa pues se puede evaluar si varios clientes que han comprado un producto tienen correlación comprando otro, lo cual se puede aprovechar para ser ofrecido a clientes similares

El interés de este trabajo consiste en enfocar la etapa de búsqueda de una transacción comercial por los problemas que involucra tanto por la cantidad de posibles sitios para encontrar un producto (se ha hecho usando agentes) y por los problemas de comunicación que surgen entre los compradores y vendedores.

1.1.2 BÚSQUEDA EN EL COMERCIO ELECTRÓNICO

La búsqueda en el comercio electrónico es una etapa de una transacción comercial, donde el cliente ya cuenta con una idea del o los productos que pueden satisfacer sus necesidades.

El primer problema que puede enfrentar un cliente es determinar el lugar en donde iniciará la búsqueda de su producto. Dicho lugar determinará la forma de interacción que el cliente tenga con el proveedor; ya que si el producto es posible encontrarlo en un mercado local lo más probable es que haga una visita directamente a los posibles proveedores. Si el sitio es con algún proveedor distante se puede acceder a los catálogos del mismo. Si los involucrados son empresas también existe la posibilidad que tengan algún convenio mediante un sistema de consulta de productos en línea en donde el cliente puede realizar su búsqueda. En caso que el producto se tenga que obtener en otro país, es posible que la búsqueda deba realizarse en Internet visitando los sitios de los posibles proveedores.

En base a lo anterior pueden surgir problemas de comunicación, en algunos casos debido al lenguaje que se utiliza, el cual puede superarse mediante traductores. Pero aun cuando se encuentre un lenguaje que ambas partes entienden, surgen problemas en cuanto a la comprensión mutua de los objetos a los que se refiere cada cual. Esto porque tradicionalmente las palabras pueden conducir a ambigüedades, por ejemplo, si una persona requiere un perico puede estarse refiriendo a un ave o bien a una herramienta. Otro problema relativo a la comunicación es el grado de detalle que pueden estar manejando tanto el cliente como el proveedor, como por ejemplo en el caso de que un cliente quiera comprar maíz y en donde un proveedor tenga distintas variedades del mismo, como azul, blanco, amarillo, rojo y entonces el cliente le indica que solamente quiere maíz que no le importa de

cual le surta (el proveedor puede darle del que tenga menor demanda). En ocasiones puede suceder que aspectos como el precio, condiciones de pago o de entrega intervienen para determinar si un sitio es adecuado para ubicar un producto, por ejemplo, en el caso de un producto que pueda adquirirse en dos sitios en Internet y del cual uno ubicado en un país vecino tenga condiciones de entrega en tiempo y costo mejores respecto a un proveedor ubicado en el mismo país.

Además durante la realización de la búsqueda es posible que existan imprevistos como por ejemplo cuando se ha contactado a un proveedor y un amigo del cliente le comenta que tiene un producto exactamente como el que está buscando y que se lo puede proporcionar sin ningún costo.

Es importante notar que esta etapa puede implicar que el cliente dedique mucho tiempo y para un proveedor puede ocurrir que le dedique tiempo y recursos a un cliente que finalmente no adquiera algún producto.

Algunos factores que pueden propiciar que el cliente opte por un proveedor en vez de otro son el hecho de que el segundo maneje marcas reconocidas, productos estandarizados de calidad reconocida, los consejos de amigos, experiencia con productos similares.

En muchas ocasiones lo más probable es que se tengan varios proveedores candidatos que pueden proporcionar el producto deseado y que en etapas posteriores de una transacción de comercio electrónico se puede determinar cual es el más apropiado.

Las contribuciones de esta tesis son particularmente útiles en esta etapa del comercio electrónico al permitir entre otros establecer el mapeo entre conceptos de ontologías diferentes para hacer posible que alguien que vende “chabacanos” pueda comercializarlos con alguien que se necesita “albaricoques”.

La búsqueda de un producto puede realizarse en diferentes formas.

- a) *Directa*, cuando una persona interactúa con otra persona. En este caso algunos de los problemas que pueden darse son relativos al uso de un lenguaje común aunque a pesar de esto puede suceder que el significado de algunos de los conceptos a los que se hagan referencia puedan requerir palabras diferentes para cada uno de los participantes. Los problemas se dan porque los participantes no pueden compartir sus representaciones internas de lo que uno quiere y de lo que el otro puede proveer. Si bien esta etapa puede hacerse mediante la presencia física, las etapas subsecuentes de la transacción comercial pueden realizarse en forma electrónica como el pago con tarjeta de crédito, monitoreo de la transportación de los bienes a través de Internet o la de servicios al cliente vía fax (Figura 1.2).

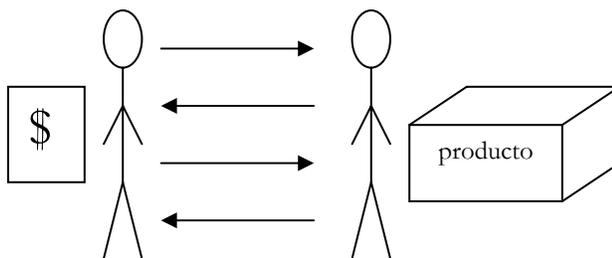


Figura 1.2 Interacción directa en la búsqueda de un producto

- b) *En Línea*, es cuando una persona tiene acceso a alguna forma de catálogo de productos mediante un canal electrónico (posiblemente Internet) (Figura 1.3). Se ha visto favorecida por el auge de las líneas de comunicación y las redes. Se da por personas con acceso en línea al proveedor o entre empresas que se encuentran enlazadas mediante algún tipo de conexión electrónica. El acceso a información de los productos vía telefónica o mediante fax también se consideran dentro de este rubro. Algunos de los problemas que surgen es que la retroalimentación de algunos datos faltantes o que se quieren aclarar debe realizarse sin las ventajas de una situación presencial en donde es más sencillo intercambiar información en el momento en que ocurre alguna duda. En este caso es importante que el cliente sea capaz de identificar los datos que se encuentran en la información que le proporciona el proveedor y que cuente con alguna forma de retroalimentación en caso que sea necesario aclarar algunas de las características del producto que no se proporcionan explícitamente en la información accesada. Otros problemas están relacionados con la disponibilidad de la línea de comunicación, el ancho de banda para transmitir diferentes tipos de datos como imágenes o sonidos, en donde se afecta la calidad de información que requiere anchos de banda grandes. Además las personas con acceso a estos proveedores está restringida a únicamente aquellos que acceden a la tecnología para establecer comunicación con el proveedor. Otro problema adicional consiste en el manejo de diferentes husos y horarios: una empresa de México que quiere encontrar un producto en línea o solicitar información adicional de un proveedor por ejemplo de India, hallará en su horario de trabajo que aquel está indisponible en ese momento y que el mejor para localizarlo es en la madrugada.

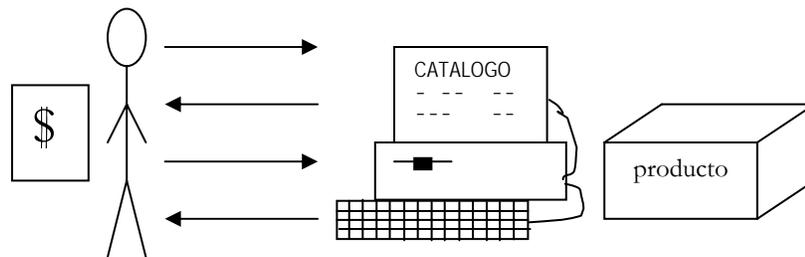


Figura 1.3 Búsqueda en Línea de un producto

- c) *Indirecta*, es cuando se utiliza algún tipo de programa para que se haga cargo de localizar el producto de interés. Esto se puede dar mediante programas que en Internet ayudan a una persona o empresa a localizar algunos productos o bien mediante el uso de agentes de software (descritos en la sección 1.2) (Figura 1.4). En este tipo de situaciones es importante que ambas partes acuerden sobre la estructura de la información que comparten y el tipo de respuestas que se darán.

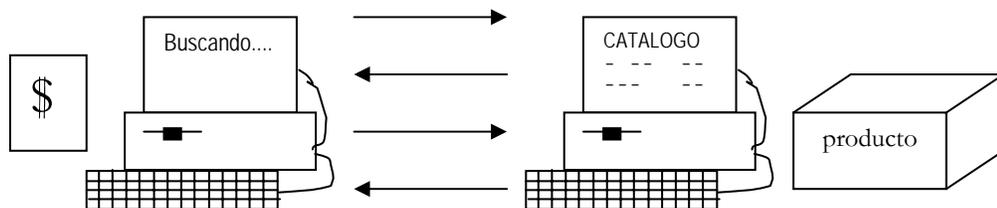


Figura 1.4 Búsqueda Indirecta de un producto

Cada una de las formas de búsqueda que se han mencionado pueden involucrar procesos de automatización y requerir establecer acuerdos como el formato en que se envía la información o los datos que se desean. El mayor grado de automatización ayuda a las partes involucradas a reducir el tiempo que una persona dedica a la búsqueda de un producto determinado. Generalmente cuando un usuario hace una búsqueda directa utiliza más tiempo que si le encarga esta labor a un programa, en el segundo caso solamente requiere indicarle los requerimientos y revisar una vez que se obtengan los resultados.

1.1.2.1 Búsqueda Indirecta

La búsqueda indirecta se realiza mediante un mecanismo automático como por ejemplo algún programa (recientemente llamados agentes de software). La principal ventaja es un ahorro en tiempo que de otra forma utilizaría una persona humana. Un problema es que se requiere estructurar los sitios de información de una manera previamente establecida entre el que consulta y el que tiene la información. La necesidad de que las partes compartan formatos tiene como consecuencia que varios clientes estarán fuera de la negociación debido a que no cuentan con el formato que se utiliza para realizar las transacciones.

En cuanto a la descripción de las características de un producto deseado es posible que esté limitado a solamente los acuerdos previos que se hayan establecido.

Desde hace varios años el Intercambio Electrónico de Datos («Electronic Data Interchange», EDI) (figura 1.5) se ha utilizado para que las empresas soliciten información entre ellas mediante mensajes que han acordado previamente. Tanto para solicitar información como para recibir las especificaciones de pedidos. El problema de este enfoque es el costo del equipo de comunicación y la dificultad para expresar requerimientos con variantes. Actualmente con la llegada de Internet se sigue utilizando el formato de comunicación de EDI, en algunos casos usando el protocolo SMTP (Simple Mail Transfer Protocol).

```
ST*840*159
BTQ*00*¢47391*820430
N1*SE*X, Inc.
N1*BY*Y Co.
P01*1*30000*EQ*0.42*PN*74355*PD*Circuit Network
SCH*10000*EQ****002*820604
SCH*20000*EQ****0002*820709
CCT*1*30000
SE*9*159
```

Figura 1.5 Un mensaje de EDI

Con el propósito de mejorar la expresividad y sensibilidad a pequeñas alteraciones en los mensajes de EDI, algunos investigadores han propuesto lenguajes alternativos. Covington [Covington 1997] propone el Language for Electronic Commerce («Lenguaje para Comercio Electrónico», LEC) (figura 1.6) el cual es más robusto a pequeños cambios en la estructura del contenido, al mismo tiempo que tiene más capacidad de que las partes puedan inferir algunos datos que no estén explícitos en la información intercambiada.

En la Universidad de Carolina de Sur, el equipo del Dr. Michael Huhns [Huhns 2001] se encuentra realizando un proyecto que consiste en establecer una metodología para coordinar un grupo de agentes que mejore los procesos de una cadena de valor. La *cadena de valor*⁸ se refiere al conjunto de etapas que conforman y unen la cadena de suministro y demanda de cualquier compañía. Se refiere a los procesos que permiten generar la demanda de servicios y productos basada en las necesidades reales de los clientes. Por otro lado la cadena de suministro cubre los procesos que permiten brindar dichos productos y servicios de una manera eficiente y efectiva.

```
lec(
  dialect(1.25,full),
  from="Y Co.",
  to="X, Inc.",
  content:
    we request:
      you inform us:
        price_of:
          item_1:= (part_no:747355,description:"Circuit Network"),
          action=
            deliver(from=you,to=us,item=item_1,qty=20000,date(82/06/04)&
            deliver(from=you,to=us,item=item_1,qty=20000,date(82/07/09))
)
```

Figura 1.6 Ejemplo de un mensaje en LEC

Un aspecto importante dentro del proyecto de Huhns es la identificación de documentos de negocios, expresados mediante DTD y XML. En el CIC-IPN, Sergio Noriega [Noriega 2001] automatiza transacciones e-comerciales agente-a-agente, usando e-documentos que cumplen con sus respectivos formatos. Por ejemplo, un agente recibe un e-documento (un pedido) y, con el auxilio de su descripción se accesa a la base de datos (del proveedor), generando la orden de embarque (otro e-documento) y la e-factura. El accesador-generador está parametrizado por el meta-esquema de la base de datos, de tal suerte que se usa el mismo accesador-generador para cualquier base de datos (del proveedor).

Los documentos descritos por el formato de Definición de Documento Tipo («Document Type Definition», DTD) [González 2001], se basan en el Lenguaje Extendido de Marcas («eXtended Markup Language», XML) y actúan como la definición de las normas que regulan la estructuración de un documento.

Mediante DTD-XML se cumplen dos objetivos:

- a) Determinar si el *documento es válido* de acuerdo al tipo de documento que se quiere revisar. Esto es porque se pueden tener varios e-documentos y se necesita ubicar el que se quiere analizar.
- b) Una vez que se sabe cual es el documento que se tiene, entonces se procede a validar que la estructura interna y los componentes que contiene sean los apropiados, considerando que hay *elementos obligatorios y opcionales*.

⁸ <http://www.sintec.org>

Una DTD especifica las características del contenido de un documento que puede expresarse en XML, se utiliza tanto para estructurar el contenido de un archivo como para validar la estructura. La principal ventaja que ofrece este sistema es la de ahorrar trabajo tanto de análisis como de programación a los desarrolladores, al mismo tiempo que brinda una plataforma de documentos estándar para el intercambio entre los interesados.

Existen dos tipos para especificar el contenido de un documento:

- a) *internas*, las cuales se incluyen en el mismo archivo de contenido (figura 1.7).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE coleccion [<!ELEMENT libro (#PCDATA)>])

<coleccion>
<libro>Guía Práctica de XML</libro>
<libro>Ulises</libro>
<libro>El juego de Ender</libro>
</coleccion>
```

Figura 1.7 Una DTD interna. En la parte superior se especifica la estructura del documento de la parte inferior

- b) *externas*, en estas la descripción del documento se hace en un archivo separado del contenido (figura 1.8). La ventaja radica en que al estar separada la descripción del contenido, la descripción se puede compartir entre diferentes usuarios para intercambiar e-documentos con un formato similar.

coleccion.dtd	coleccion.xml
<pre><!ELEMENT coleccion (pelicula)+> <!ELEMENT pelicula (titulo, actor+, director+)> <!ELEMENT titulo (#PCDATA)> <!ELEMENT actor (#PCDATA)> <!ELEMENT director (#PCDATA)></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE coleccion SYSTEM "coleccion.dtd"> <coleccion> <pelicula> <titulo>Blade Runner</titulo> <actor>Harrison Ford</actor> <actor>Sean Young</actor> <director>Ridley Scott</director> </pelicula> <pelicula> <titulo>Con faldas y a lo loco</titulo> <actor>Jack Lemmond</actor> <actor>Tony Curtis</actor> <director>William Wilder</director> </pelicula> </coleccion></pre>

Figura 1.8 Una DTD externa, *coleccion.dtd* es el descriptor y *coleccion.xml* es el contenido

La necesidad de automatizar las etapas de las transacciones comerciales ha propiciado el desarrollo de herramientas que tomen la iniciativa y realicen tareas en forma autónoma (esto es, con una menor supervisión constante de un elemento de control).

Hasta ahora se han propuesto diversos sistemas en donde se aplican agentes en el comercio electrónico, algunos trabajos relacionados con este tema se presentan en la sección 1.2 en este documento.

Se ha propuesto un estándar basado en XML para los negocios electrónicos llamado e-business XML (ebXML⁹), este propone una especificación para crear un mercado global electrónico sencillo donde las empresas de cualquier tamaño y localización geográfica pueden realizar negocios a través del intercambio de mensajes basados en XML. La especificación se enfoca en definir un protocolo de comunicación neutral para intercambiar mensajes electrónicos de negocios. Definiendo las envolturas que dan confiabilidad y seguridad a la información que se envía. La especificación se puede utilizar con HTTP y SMTP. La funcionalidad central consiste de:

- a) una *especificación de mensajes* que pueden enviarse a través de HTTP o SMTP,
- b) *extensiones* para utilizarse con SOAP (Simple Object Access Protocol),
- c) *manejo de errores*, una descripción de como reporta errores un servicio de mensajes de ebXML,
- d) *seguridad*, proporciona la especificación de la semántica de seguridad para los mensajes de ebXML,
- e) *respuesta síncrona*, indica si se regresa o no una respuesta con sincronía.

ebXML y los Web Services continúan en desarrollo y todavía no se pueden plantear como un sustituto a la tecnología de EDI, como lo indica el artículo “*Web Services and ebXML need to show that they can outperform EDI. It won't be easy*” de Alan Kotok publicado en Internet¹⁰ el 11 junio de 2002: “A few years ago, the XML glitterati predicted an early death for electronic data interchange (EDI). The XML Handbook (Prentice-Hall, 1998), for example, said on page 103 ‘Traditional’ EDI is based on outdated principles that will cause it to fade into technological obscurity, as it becomes embraced and replaced by the ‘New’ EDI. Traditional EDI refers to the use of rigid transaction sets with business rules embedded in them. This model simply does not work in today's rapidly changing business environment.”

1.2 AGENTES

El aumento de las transacciones electrónicas, requiere para su manejo automático, de herramientas con la capacidad de intercambiar información sin ambigüedad pero con un nivel de detalle apropiado para realizar las operaciones comerciales. Los agentes, que hasta ahora se han implantado como

⁹ <http://www.ebxml.org>

¹⁰ <http://www.webservices.org/index.php/article/articleview/447/1/24>

versiones de laboratorio se pretende que en el futuro se ofrezcan como productos para realizar este tipo de operaciones.

Los antecedentes de los Agentes se encuentran en las áreas de Computación Distribuida e Inteligencia Artificial Distribuida.

La *Computación Distribuida* (Distributed Computing, DC) [Flores 1999] es una *disciplina que estudia y desarrolla teorías y técnicas para integrar componentes informáticos heterogéneos*, procurando dos objetivos: autonomía de cada componente y capacidad de colaborar entre ellos. Los componentes pueden ser computadoras, servidores de base de datos, impresoras, u otros equipos o programas.

El área de *Inteligencia Artificial Distribuida* (Distributed Artificial Intelligence, DAI) [Erceau 1993] [Flores 1999] [Nwana 1996], nace a principios de los 70s, con el propósito de poder integrar en un mismo sistema las competencias de diferentes sistemas especializados¹¹. La DAI se define como la *ampliación de la Inteligencia Artificial clásica mediante su distribución entre varios agentes especializados*. Una premisa utilizada en la distribución del conocimiento es, que es más fácil modelar la pericia de varios individuos en forma separada que globalmente.

Desde la perspectiva de un agente el ambiente es el conjunto de recursos, propiedades y agentes que se encuentran en el sistema. Wooldridge [Wooldridge 1999] caracteriza a los ambientes de agentes en base a los criterios siguientes:

- a) *Accesibilidad*, cuando el agente tiene el control completo sobre el ambiente, esto es, que puede acceder todos los componentes que lo forman, se dice que el ambiente es accesible al agente. El dual son ambientes inaccesibles, donde existe al menos un elemento que el agente no puede acceder. Internet es un ambiente no-accesible.
- b) *Determinismo*, si en un ambiente una acción tiene un efecto único y específico, es decir, no hay incertidumbre acerca del mismo, se dice que el ambiente es determinístico. El caso contrario es cuando alguna acción realizada tiene un resultado que puede ser imprevisto, por ejemplo, al modelar una mesa donde se tienen figuras geométricas que un agente puede mover, al colocar una esfera sobre una mesa, el resultado final puede ser que la esfera caiga al suelo y que no se encuentre en el lugar donde fue colocada originalmente.
- c) *Ocurrencia de los eventos*, cuando las acciones de un agente se consideran como un episodio sin relación con otros escenarios o tiempos se le llama episódico, por ejemplo un sistema para ordenar una lista de correo electrónico es un ambiente episódico, mientras que uno no episódico es aquel donde se tenga que razonar hacia episodios futuros como la compra de una camioneta, la cual requerirá de aceite, gasolina, pago de tenencia, entre otros, en un tiempo futuro y por lo tanto el costo de mantenimiento debe considerarse al momento de la compra, porque si quien la compra se acaba su dinero en la compra inicial será muy difícil que pueda darle el mantenimiento mínimo requerido.
- d) *Forma de los cambios*, cuando el ambiente permanece sin cambio excepto por las acciones que realiza el agente, se dice que es *estático*. Es *dinámico* aquel en el que ocurren procesos y operaciones que provocan variaciones fuera del control del agente. El mundo físico es altamente dinámico.

¹¹ Un antecedente notorio es el sistema de Pizarrón (blackboard) de Raj Reddy, donde varios agentes (él no utilizó este término) interaccionaban para lograr un fin común: entender una oración hablada

- e) *Cantidad de variables*, si hay un número finito de acciones y percepciones del ambiente se dice que es discreto, por ejemplo un juego de ajedrez es discreto, mientras que el ambiente de una ciudad donde se conduce un taxi es continuo.

El mundo real se considera como inaccesible, no determinístico, dinámico y continuo.

El ambiente puede estar formado por un agente (*uniagente*) o varios (*multiagente*) en un mismo sitio o en movimiento constante (*agentes móviles*) [Rus 1997a].

Diferentes autores han dado conceptos sobre los agentes, de estos se pueden extraer algunas propiedades comunes:

- a) *Guiado por un propósito u objetivo*, esto es que un agente se crea para realizar algún tipo de acciones específicas, en algunos casos puede concluir su ejecución cuando considera que ha alcanzado sus propósitos u objetivos, entre otros permanece activo esperando actuar cuando se vuelva a presentar la oportunidad de alcanzar su objetivo, como un agente que elimina correos electrónicos indeseados.
- b) *Autonomía* respecto del usuario [Wooldridge 1999], en ciertos casos el agente puede actuar y tomar decisiones por cuenta propia, de las cuales es responsable el propietario del agente (generalmente una persona humana).
- c) *Proactividad* [Wooldridge 1999], es la capacidad de actuar sin la invocación explícita del usuario, esta característica le permite en algunos casos anticiparse y hacer propuestas en vez de esperar órdenes, esta es una diferencia con los sistemas basados en objetos, que esperan que alguien los invoque mediante un método.
- d) *Adaptabilidad*, es la habilidad de enfrentar exitosamente situaciones no familiares, novedosas o ambientes con cambio. Al realizar negociaciones automáticas, se adapta porque permite interactuar (“conversar”) con otro agente cuya estructura de conocimientos no es idéntica a la suya.
- e) *Sociabilidad*, es la capacidad para comunicarse, cooperar o negociar tanto para intercambiar información como para resolver conflictos o alcanzar objetivos comunes en un grupo de agentes. En esta característica es importante considerar a los lenguajes de programación de agentes, de comunicación y el manejo de ontologías mixtas.
- f) *Sentido común*, característica que le permite intercambiar información no estructurada de una forma eficiente.
- g) *Personalidad*, le permite a los agentes tener características que le hacen predecible en su comportamiento tanto con otros agentes como con usuarios humanos. Por ejemplo, un agente que se dedique a manejar pedidos a través de correo electrónico será poco probable que controle un robot para explorar volcanes.

El estudio de los agentes se ha realizado en dos diferentes niveles [Wooldridge 1999] [Huhns 1999], el primero se refiere a los aspectos *macro* donde existen varios agentes interactuando, conocido como Sistemas Multi Agente (SMA, «Multi Agent Systems»). El otro, se refiere a los aspectos *micro*,

esto es, las características internas de un agente. Lo más frecuente es encontrar que un agente tiene propósitos que intenta alcanzar en forma autónoma. El aspecto macro enfoca a las interacciones interagente mientras que el micro a las intraagente. En los SMA las propiedades más comunes son las relativas a la sociabilidad.

Para programar sistemas de agentes se han propuesto diferentes lenguajes motivados en la necesidad de pasar de los modelos formales a la implementación de sistemas. Algunos de los lenguajes desarrollados son: AgentSpeak(L), 3APL, GOLOG, CONGOLOG, METATEM.

En *AgentSpeak* [Inverno 1998] los agentes están basados en el modelo de creencias, deseos e intenciones. Cada creencia o meta está representada como un evento. Los agentes responden a cambios en sus metas y creencias como resultado de sus percepciones; para responder ante estos cambios, un agente selecciona algún plan de un depósito de planes. Las acciones de cada agente se basa en las operaciones siguientes:

- a) Si tiene *eventos* a procesar, se selecciona uno.
- b) Toma de la biblioteca de *planes* aquellos que corresponden con el evento que ocurrió.
- c) Selecciona un plan generado como una *instancia* del mismo.
- d) Adiciona la instancia del plan a las *intenciones del agente*. Las intenciones son aquellos compromisos que el agente esta comprometido a realizar.
- e) *Selecciona una intención* y considera el siguiente paso en su plan general. Si es una acción, se ejecuta y si es una meta, se agrega un evento al conjunto de eventos a procesar.
- f) Si el *plan general se completa*, se considera un siguiente plan, y si las intenciones están vacías se puede remover del conjunto de intenciones.

El lenguaje *METATEM concurrente* [Fisher 1994] tiene como propósito la ejecución directa de formulas de lógica temporal. El lenguaje consiste de un modelo de objetos concurrentes, lógica temporal ejecutable y paso de mensajes múltiples («broadcasting»). El tiempo se modela como una secuencia discreta infinita de estados con un punto de inicio llamado ‘el inicio del tiempo’ y un futuro infinito. La lógica que se utiliza es temporal de primer orden. Los objetos son entidades independientes que escuchan mensajes de otros agentes. Un ejemplo de un objeto que maneja una pila («stack») es el siguiente (se indica únicamente su declaración):

```
stack(pop, push)[popped, stack-full]
```

donde pop y push son los mensajes que reconoce el objeto. popped y stack-full son los mensajes que puede emitir el mismo.

GOLOG [Levesque 1997] es un lenguaje de programación lógica cuyo interprete mantiene automáticamente una representación explícita del mundo con situaciones dinámicas. El mundo se modela sobre la base de un estado inicial y las acciones que ocurren en el mismo. Cada acción considera las precondiciones del mismo y los efectos antes de reflejarlos en el ambiente. La representación del mundo se hace basándose en el calculo situacional que consiste en predicados de primer orden con una función do descrita como $do(\alpha, s)$ que denota la situación siguiente a una situación dada s resultado de realizar la acción α . Las acciones pueden parametrizarse, por ejemplo

$put(x, y)$ significa que se coloque el objeto x sobre el objeto y , en cuyo caso $do(put(A, B), s)$ indica la situación resultante de colocar A sobre B cuando el mundo se encontraba en la situación s . Una ventaja en este lenguaje es para representar acciones con símbolos adicionales a los lógicos como el `if`, `while` y definición de secuencias llamadas procedimientos.

CONGOLOG [Giacomo 2000] es una extensión al lenguaje GOLOG en el sentido de permitir acciones incompletas, para esto se define el predicado `Trans`, además de otros elementos para priorizar la ejecución de procesos concurrentes, interrumpir la ejecución cuando ciertas condiciones se hacen verdaderas y tratar con acciones exógenas.

El predicado `Trans` se define como:

$$\text{Trans}(\delta, s, \delta', s')$$

donde δ es un programa, s es una situación, s' es una situación resultante y δ' es un programa incompleto resultado de la ejecución de alguna acción.

También se introduce el predicado:

$$\text{Final}(\delta, s)$$

que significa que la computación se realizó completamente y que no hubo ninguna parte del programa δ sin ejecutar al pasar de la situación s a la s' .

El lenguaje *3APL* [Inverno 2000] se diseñó para soportar la construcción de sistemas de agentes complejos a través de conceptos como creencias, metas y planes. Consta de un conjunto de primitivas para programar agentes, estas son, un conjunto de creencias, metas y reglas de razonamiento. Las creencias representan los elementos con los que el agente debe tratar, mientras que las metas permiten al agente enfocarse en lo que debe lograr y para representar la forma en la que puede lograrlo. Las creencias son fórmulas de primer orden formadas por términos variables y constantes. Las acciones se definen también como fórmulas de primer orden, por ejemplo, `ins_agenda(meeting, may5th12:00, 60min, [john, peter], utrecht)` con la que se inserta en una agenda manejada por un agente una cita de una hora de duración con John y Peter en Utrecht. Se pueden definir funciones auxiliares para regresar conjuntos de variables en un átomo, una creencia o una acción. La arquitectura del lenguaje está basada en el ciclo de pensar-actuar, el cual se divide en dos partes, en la primera se hace el razonamiento utilizando las reglas existentes para esto y en la segunda se ejecuta alguna acción.

Además de los lenguajes para programar agentes se han definido otros lenguajes para describir su *interoperación y comunicación*. El grupo formado por el Knowledge Sharing Effort (KSE) patrocinado por ARPA, ASOFR, NRI y NSF, tiene tres subgrupos dedicados a este tipo de proyectos [Finin 1995]:

- a) *Interlingua*, quienes definieron el Formato de Intercambio de Conocimiento (Knowledge Interchange Format, KIF).
- b) *Compartición y Reuso de Bases de Conocimiento*, su propósito es facilitar el consenso sobre el contenido de las bases de conocimiento.
- c) *Interfases externas*, están interesados y hacen propuestas para lograr la interacción entre los sistemas basados en conocimiento y otros módulos. Uno de los resultados de este grupo

es la propuesta del protocolo de comunicación Knowledge Query Manipulation Language (KQML).

El lenguaje *KIF* es una propuesta para representar conocimiento. Su intención es proveer de un lenguaje para el intercambio del contenido entre bases de conocimiento. Se propone como un vehículo para expresar conocimiento y meta-conocimiento. En particular puede utilizarse para generar traductores entre lenguajes, por ejemplo se usa un traductor de un lenguaje A hacia KIF y de KIF hacia un lenguaje B y viceversa, haciendo necesario solamente 2n traductores para n lenguajes.

La sintaxis de KIF es la de cálculo de predicados de primer orden en forma prefija con extensiones para soportar el razonamiento no-monotónico y definiciones. Algunos ejemplos del uso de KIF se indican a continuación. La semántica del núcleo de KIF es similar al de la lógica de primer orden.

- a) Representación del contenido de una base de datos:

```
(empleado LOFI-681118-MU4 ventas 6000)
(empleado UOST-700514-LI3 mercadotecnia 9000)
(empleado JITE-640319-53P contabilidad 4000)
```

- b) Mediante expresiones con operadores relacionales, se pueden definir términos más complejos, para indicar que un mueble es más grande que otro se escribe:

```
(> (* (ancho mesa1)(longitud mesa1))
  (* (ancho mesa2)(longitud mesa2)))
```

- c) El manejo de variables para hacer consultas se hace colocando una comilla precediendo la variable deseada junto con el operador “;”:

```
(persona Juan '(empleado ,?x ,?y ,?z))
```

- d) KIF también puede usarse para describir procedimientos de escritura de programas:

```
(programa X (new-line t)
  (print "hola")
  (new-line t))
```

El lenguaje *KQML* fue concebido como un formato de mensajes para intercambiar conocimiento entre agentes. Los aspectos principales de KQML pueden resumirse como sigue:

- a) Los mensajes de KQML son *opacos en su contenido*. Intentan comunicar una intención de un agente a otro.
- b) A las primitivas del lenguaje se les llama *performativas*, ya que definen operaciones que los agentes pueden utilizar para comunicarse entre ellos.

- c) En un ambiente de Agentes que utilizan KQML pueden existir otros, llamados *facilitadores* para proporcionar funciones y servicios como asociación de direcciones físicas con nombres simbólicos o registro de bases de datos. Se puede considerar a estos como intermediarios en los procesos de comunicación entre agentes.

KQML está formado por tres capas:

- a) La capa de *contenido*, indica el contenido de un mensaje codificado en su propio lenguaje de representación, esto significa que KQML no proporciona ningún lenguaje de codificación de contenido.
- b) La capa de *mensaje*, se utiliza para codificar los mensajes enviados entre aplicaciones. Una de sus principales funciones es identificar el protocolo utilizado para que el mensaje sea enviado de un agente a otro.
- c) La capa de *comunicación*, codifica las características que describen los parámetros de comunicación, como la identificación del envió y el receptor; además proporciona un identificador único asociado con la comunicación.

La sintaxis de KQML se basa en una lista de paréntesis balanceados. El elemento inicial de la lista es la performativa, el resto de los elementos son los argumentos escritos como pares de palabras clave/valor. Ejemplo:

```
(ask-one :sender Juan
        :content (PrecioVenta maíz-blanco ?Precio)
        :receiver almacenista
        :reply-with existencias
        :lenguaje Prolog
        :ontology Almacen)
```

En este ejemplo la performativa es `ask-one`, el contenido es `(PrecioVenta ?Precio)`, la ontología es `Almacen`, el receptor del mensaje es `almacenista`, la consulta está escrita en `Prolog`. El valor de `:content` forma la capa de contenido, los valores de `:reply-with`, `:sender` junto con `:receiver` forman la capa de comunicación. Las performativas `:language` y `:ontology` forman la capa de mensaje. En un tiempo dado el `almacenista` le enviara una respuesta a `Juan` como la siguiente:

```
(tell :sender almacenista
      :content (PrecioVenta maíz-blanco 10)
      :receiver Juan
      :in-reply-to existencias
      :lenguaje Prolog
      :ontology Almacen)
```

Las performativas se agrupan en las categorías indicadas en la figura 1.9.

Con el propósito de modelar *sistemas complejos de agentes*, los investigadores del área han realizado extensiones a metodologías existentes en otras áreas, intentando introducir a los agentes como elementos naturales dentro de los sistemas. La idea es construir piezas de hardware y/o software autónomos y proactivos. Las metodologías [Iglesias 1998] a las que han hecho extensiones son las Orientadas a Objetos y las de Ingeniería del Conocimiento.

CATEGORIA	NOMBRE
Basic query (Consultas básicas)	evaluate, ask-if, ask-about, ask-one, ask-all
Multi-response query (Respuestas múltiples)	stream-about, stream-all, eos
Response (Respuesta)	reply, sorry
Generic informational (Información genérica)	tell, achieve, cancel, untell, unachieve
Generator (Generador)	standby, ready, next, rest, discard, generator
Capability-definition (Definición de capacidades)	advertise, subscribe, monitor, import, export
Networking (Redes)	register, unregister, forward, broadcast, route

Figura 1.9 Performativas de KQML por categoría

La ventaja de aprovechar las *Metodologías Orientadas a Objetos* se debe a que algunos investigadores consideran a los agentes como objetos activos con un estado mental. Los aspectos que se toman en cuenta son los estáticos para las estructuras de los objetos y las relaciones estructurales, dinámicos para describir sus interacciones y funcionales para ilustrar el flujo de datos de su metodología.

Las extensiones en las *Metodologías de Ingeniería del Conocimiento* proporcionan una base útil porque los agentes tienen características cognitivas y esas metodologías proporcionan técnicas para modelar el conocimiento que requieren. Se aprovechan entonces las técnicas para el desarrollo de bibliotecas de ontologías y métodos para la solución de problemas.

Las limitantes de las metodologías de la ingeniería del conocimiento comparadas con las orientadas a objetos, son: su falta de extendibilidad y de popularidad de aquellas. Tradicionalmente las metodologías existentes consideran al conocimiento centralizado y por lo tanto no consideran los aspectos sociales o distribuidos de los agentes o su actitud orientada a objetivos.

La metodología *Gaia* [Wooldridge 2000a] se usa para el análisis y diseño orientado a Agentes, es independiente de plataforma y propone un proceso de abstracción con el cual modelar y desarrollar sistemas complejos. Las aplicaciones en donde es apropiado aplicar *Gaia* son:

- a) Agentes con sistemas computacionales que hacen uso de *recursos* (como los de Unix).
- b) Se supone que la meta es obtener un sistema que maximiza una *utilidad global*. No admite la posibilidad de conflictos verdaderos.
- c) Los agentes son *heterogéneos* en el sentido de que pueden implantarse usando diferentes lenguajes de programación sin suponer ninguna plataforma de ejecución final.

- d) La organización del sistema es *estática*, en donde las relaciones interagentes no cambian durante su ejecución.
- e) Las capacidades de los agentes y los servicios que proporcionan son *estáticos*, en el sentido de que no cambian al ejecutarse.
- f) El sistema completo consta de un número *reducido* de agentes (menos de 100).

Gaia le permite a un analista avanzar sistemáticamente de un establecimiento de requerimientos a un diseño que es suficientemente detallado y que puede ser implantado directamente. Consta de dos conceptos principales que se desarrollan para llegar a un sistema en funcionamiento:

- a) *abstractos*,
- b) *concretos*.

Los componentes *abstractos* se usan para conceptualizar el sistema durante la etapa de análisis. Los *concretos* generalmente tienen una contraparte dentro del sistema en ejecución y se utilizan principalmente durante la etapa de diseño.

Los componentes abstractos generalmente son:

- a) *Papeles*, que son descripciones abstractas de la función esperada de una entidad
- b) *Permisos*, que se refieren a los recursos que puede utilizar un papel cuando se lleva a cabo
- c) *Responsabilidades*, que son la funcionalidad esperada de un papel
- d) *Protocolos*, que establecen el intercambio de mensajes entre papeles.
- e) *Actividades*, propiedades activas, propiedades de seguridad.

Los componentes concretos son: *tipos de agentes*, *servicios* y *adquisiciones*. Las relaciones entre los papeles forman el modelo de interacción.

De acuerdo a las *competencias de los agentes* y la forma como interactúan en su entorno, se han establecido dos escuelas para caracterizar a los sistemas de agentes [Erceau 1993]:

- a) *Cognitiva*, donde se tienen agentes que interactúan y cada agente por sí mismo ya es capaz de ejecutar operaciones relativamente complejas.
- b) *Reactiva*, los agentes son más sencillos en cuanto a sus capacidades de decisión y acción pero son más numerosos y activos.

1.2.1 ESCUELA COGNITIVA

La escuela cognitiva inicia en la década de 1970 adoptada por Douglas Lenat en el MIT junto con F. y Barbara Hayes-Roth. Originalmente se proponían realizar sistemas expertos clásicos que se comunicaran y cooperaran. En este caso los sistemas cognitivos comprenden un número reducido de agentes, que disponen de una base de conocimiento que les permite realizar razonamientos relacionados con el campo de aplicación al cual están dirigidos. Cada agente se considera como un sistema experto relativamente complejo en donde para resolver problemas aún más complejos (cuya actual competencia es insuficiente), debe cooperar con otros, para esto, es necesario que se comunique y en ocasiones realice negociaciones. Estos agentes se consideran como granularidad fuerte y las interacciones tienen un aspecto social.

Hasta ahora esta escuela es la que ha conseguido las aplicaciones más avanzadas. Algunos modelos de agentes propuestos bajo este enfoque son: la arquitectura BDI (Belief-Desire-Intention) («Creencia-Deseo-Intención») [Rao 1997] y la Programación Orientada a Agentes [Shoham 1997].

En la *arquitectura BDI* las intenciones se tratan a la par de las creencias y los deseos como parte del estado mental de un agente, la cual determina su comportamiento para alcanzar sus metas. La formalización de las intenciones se basa en un modelo de mundos posibles. Las intenciones son planes de acciones parciales que el agente está comprometido a realizar para alcanzar sus metas.

El mundo se modela usando una estructura temporal con un pasado único, y el futuro es una ramificación de posibilidades llamado árbol del tiempo. Un punto en particular del árbol se llama situación. Una situación se representa mediante un conjunto de proposiciones. Un evento transforma una situación en otra, estos se clasifican en primitivos y no primitivos. Los eventos primitivos son los que el agente puede realizar directamente mientras que los otros son secuencias de eventos primitivos y pueden mapear puntos no adyacentes.

Las ramificaciones son las opciones que tiene un agente. Los eventos se representan en los arcos que unen dos situaciones.

Por ejemplo en la figura 1.10 en la situación del lado izquierdo se tienen los predicados p y f , en caso que se realice el evento $d1$ se pasa a la situación $\sim p$ y $\sim f$, pero si se realiza el evento $d2$ se llega a la situación en donde se tiene $\sim p$ y f .

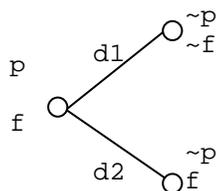


Figura 1.10 Árbol temporal en que se modela el tiempo en los mundos posibles. Las proposiciones que determinan a cada situación son p y f . Los eventos para pasar de una situación a otra son $d1$ y $d2$

Las proposiciones que tienen valores determinados en la situación actual corresponden con las creencias de un agente. Estas se utilizan para determinar los eventos posibles a realizarse.

El agente puede intentar ejecutar algún evento pero puede fallar. Así que se distingue entre ejecución exitosa y fallida.

Además de los predicados de la situación actual, los mundos accesibles desde ese punto tomando en cuenta los eventos que puede realizar el agente pasan a formar parte de sus creencias.

Las creencias de los mundos posibles a partir de una situación dada le permiten a un agente determinar las metas que le es posible alcanzar. Entonces *las metas por alcanzar corresponden a los deseos del agente*. Si bien esto puede generar que dos deseos sean inconsistentes porque los resultados son antagónicos, es decir en uno se puede alcanzar p y en otro $\sim p$, aunque las metas siempre son consistentes dado que el agente solamente alcanzará una de ellas cada vez.

Las *intenciones son los eventos que un agente se ha comprometido a realizar para alcanzar los mundos posibles*. Las intenciones se pueden representar por conjuntos de mundos posibles por alcanzar.

Ejemplo: si un agente en una situación dada tiene las creencias tener hambre (f) y tener dinero (p) (figura 1.11), pero con el deseo de no tener hambre y dos mundos posibles, uno en donde ya no tiene hambre pero tampoco dinero y otro donde continua con hambre pero sin dinero (figura 1.12).



Figura 1.11 Creencias iniciales de un agente que tiene hambre (f) y dinero (p)

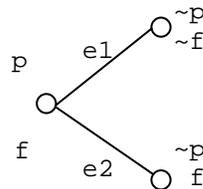


Figura 1.12 Creencias de los eventos que puede realizar el agente. Los eventos que el agente cree posibles son, $e1$ =comprar comida, $e2$ =comprar bicicleta

En este caso y dado que la meta del agente en cuestión es quitar su hambre, su deseo es realizar el evento $e1$ y acceder a la situación $\sim p$, $\sim f$. Por lo tanto, una vez que el agente establece el compromiso de alcanzar la situación deseada se convierte en una intención (figura 1.13).

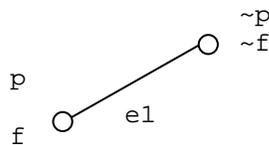


Figura 1.13 Dado que el deseo del agente es quitar su hambre, su intención es el evento $e1$

En este escenario los mundos accesibles de las creencias, intenciones y metas representan diferentes opciones para el agente.

Shoham [Shoham 1990] propone una arquitectura para la *Programación Orientada a Agentes* («Agent Oriented Programming», AOP) en el sentido de proporcionar una especialización de la programación dirigida a objetos. En su modelo, un agente consiste de componentes como creencias («beliefs»), decisiones («decisions»), capacidades («capabilities») y obligaciones («obligations»). Es por esta razón que el estado de un agente se llama estado mental. El estado mental se describe formalmente como una extensión de la lógica porque introduce aspectos temporales. Además se introducen operadores para las obligaciones, decisiones y capacidades. Los agentes se controlan por los programas agentes que incluyen primitivas para comunicarse con otros. En el espíritu de la teoría de los actos del habla, cada primitiva de comunicación es de cierto tipo, como informar, requerir y ofrecer. El paradigma en que se apoya la AOP es en la interacción entre agentes

De acuerdo a Shoham un sistema completo de AOP debe incluir tres componentes:

- a) Un lenguaje formal restringido con sintaxis y semántica claras para describir el estado mental de un agente por modalidades como creencias y compromisos.
- b) Un lenguaje de programación en el cual se puedan definir y programar los agentes con comandos primitivos como REQUEST o INFORM, la semántica del lenguaje debe corresponder adecuadamente a la semántica del estado mental.
- c) Un agentificador que convierta dispositivos neutrales en agentes programables.

El trabajo de Shoham se centra en el segundo aspecto. Los componentes mentales de su modelo son las creencias («beliefs»), decisiones («decisions»), capacidades («capabilities») y obligaciones («obligations»). Además de estas adiciona el componente tiempo que matiza a los componentes mentales en el sentido de que se puede tener una creencia en un tiempo determinado pero en otro tiempo se tiene otra creencia diferente.

Las acciones se representan como predicados calificados con un índice temporal. Por ejemplo en vez de registrar el hecho de que un robot levantó un brazo en el tiempo t , se indica que el predicado $\text{levantoBrazo}(\text{robot})^t$ es verdadero, en donde el índice t , indica el tiempo en que se hace la aseveración.

Las capacidades se indican mediante un operador (CAN) y el hecho de que es capaz un agente de una acción se indica con otro operador (ABLE).

Los operadores para las creencias, obligaciones y decisiones son respectivamente (B), (OBL), y (DEC). Estos también pueden llevar un índice indicando el tiempo en que estos se hacen verdaderos, además pueden estar subindizados por los agentes involucrados.

Con base en los operadores descritos se puede definir un lenguaje de programación y su interprete. Por lo tanto un agente puede tener capacidades, creencias iniciales, y compromisos.

Además se tienen las instrucciones INFORM para solicitar información, REQUEST para solicitar una acción, DO para que un agente ejecute alguna acción, IF para decidir si se realiza o no.

1.2.2 ESCUELA REACTIVA

La escuela reactiva se considera su inicio en la década de 1980 con Robin Brooks en el MIT, L. Steels en Bélgica y en Francia en el Laboratorio de Formas e Inteligencia Artificial (LAFORIA). En este caso, no es necesario que cada agente sea inteligente para conseguir un comportamiento global inteligente; sino contar con algunos mecanismos de reacción ante los acontecimientos, sin tener en cuenta ni una explicitación de los objetivos ni los mecanismos de planificación, de donde, el comportamiento es un efecto emergente. Por ejemplo, si se quiere construir un robot que sea capaz de seguir una pared. En la IA clásica lo que se hace es descomponer el problema en diferentes subfunciones: detectar la pared, ir hacia ella, detenerse a cierta distancia, girar y a continuación desplazarse gestionando convenientemente la distancia a la pared. Con el enfoque de *agentes reactivos* el comportamiento consiste de dos mecanismos simples, considerado cada uno como un agente: el primero hace que el robot sea atraído por los obstáculos y el segundo que sea repelido por ellos. Al interaccionar ambos mecanismos el comportamiento emergente que aparece es la funcionalidad “seguir la pared”. Los agentes que intervienen son de granularidad débil asemejados a agentes del tipo autómeta simple. Estos trabajos son de corte empírico respecto a los de la escuela cognitiva de tendencia más formal.

1.2.3 APLICACIONES DE AGENTES

Desde principios de la década de 1990 se han desarrollado diversas aplicaciones de Agentes. En esta sección se describen algunas de éstas, agrupadas por los rubros más usuales. Las aplicaciones se hacen en diferentes ámbitos: comercial, industrial, investigación o académico. Es de esperarse un aumento en la aplicación de esta tecnología que resulta de la unión de trabajos de Ingeniería de Software e Inteligencia Artificial.

Interfases de Usuario

Las interfaces de usuario y ayudas para el manejo de información son las primeras aplicaciones de los agentes. El sistema *Maxims* [Maes 1994] maneja el correo electrónico de un usuario y sus citas electrónicas con diferentes niveles de autonomía que varían en función del aprendizaje del agente, una vez que ha captado las preferencias del usuario, se le permite decidir y realizar acciones en forma independiente y autónoma.

El sistema *CAP (Calendar APrentice)* [Mitchell 1994] está basado en agentes que se ejecutan en la Web para manejar la agenda electrónica de un usuario, estos agentes se comunican entre ellos y realizan negociaciones para establecer el mejor horario para las humanos involucrados en una cita.

Los sistemas Multiagente (contienen varios agentes) pueden ser similares o bien diferenciados (con funciones específicas). El trabajo de Barret [Barret 1997] es una *arquitectura multiagente* (de agentes con funciones específicas cada uno: monitores, editores, generadores, autónomos) para auxiliar a los usuarios de Internet mediante la personalización de sus requerimientos. Thomas [Thomas 1998] desarrolló el sistema multiagentes *BASAR* (Building Agents Supporting Adaptive Retrieval) con el propósito de brindarle a un usuario asistencia para manejar su información personal dentro de Internet. Con el propósito de generar agentes para organizaciones virtuales Contreras [Contreras 2001] creo una infraestructura multiagentes para facilitar la *interoperabilidad entre agentes* construidos por diferentes desarrolladores al estar basado en estándares de agentes (FIPA compliant).

Personajes Sintéticos

Las aplicaciones orientadas a la interacción entre agentes han propiciado también las interfaces gráficas en donde algunos de los personajes generados en 3D están basados en agentes y responden a movimientos gestuales mediante el reconocimiento de la persona que les da instrucciones, por ejemplo, Maes y su equipo [Maes 1995] crearon un *entorno con un perro (agente)* que responde a las órdenes gestuales de su propietario, el perro está en un mundo virtual.

Rist et. al [Rist 1998] han desarrollado agentes inteligentes para dotar de *movimientos y gestos coherentes* (muy cercanos a la realidad) a sus personajes sintéticos. Proponen el sistema para las interfaces de enseñanza asistida por computadora o para presentaciones en nombre de un usuario humano. Un trabajo similar en donde se realiza la coordinación gestual mediante la tecnología de agentes lo realizó Cassell y su equipo [Cassell 1994] para generar agentes que *sincronizan sus gestos con la reproducción de voz* que les corresponde. Un trabajo aplicado a facilitar el aprendizaje mediante la *generación de gestos apropiados* en los sistemas automatizados lo realizó Towns [Towns 1998]. Wachsmuth [Wachsmuth 1997] implementó el proyecto *Viena* que consta de un agente con movimientos apropiados para facilitar la interacción de un usuario humano con ambientes virtuales.

En algunos casos a los agentes de software se les intenta mostrar con *características antropomórficas*, algunas aplicaciones de investigación incluyen las emociones [Bates 1994], personalidad [Moon 1996] [Lester 1997] [King 1996], y creatividad [Boden 1994]; con lo que los investigadores pretenden acercar a los usuarios una forma amigable y de ayuda en sus sistemas de información, aunque no necesariamente se tienen las capacidades de inteligencia y razonamiento que la apariencia inspira, lo cual ha mostrado que es una desventaja, debido a que los usuarios humanos generan desconfianza y resistencia hacia el uso de estas interfaces por falta de congruencia entre la funcionalidad y la apariencia.

Extensiones a los Sistemas Orientados a Objetos

Los agentes en la aplicación de sistemas de información han sido propuestos como una *extensión a la actual tecnología orientada a objetos* [Amandi 1997] quien propone una arquitectura de dos niveles para dotar de componentes de agentes a sistemas de objetos tradicionales. En el primer nivel se tienen los objetos de percepción, comunicación y conocimiento, en el segundo los de reacción y deliberación. Genesereth [Genesereth 1994] establece que existen diferentes formas de *adicionar las propiedades de agente a sistemas legados* (sistemas construidos sin el paradigma de agentes), mediante:

- a) Transductores, son módulos de agentes que interactúan con los sistemas tradicionales para dotar al segundo de una apariencia de agente (figura 1.14a).
- b) Envoltente, esto es un código que se convierte en la nueva interfaz del sistema para poder interactuar con otros agentes (figura 1.14b).
- c) Reescribir el sistema cuando hacer lo anterior (a, b) es muy difícil y se necesitan las propiedades de agentes en el modulo de interés (figura 1.14c).

Enseñanza

En la enseñanza apoyada por agentes hay diversos trabajos. Adelson [Adelson 1992] hace un estudio sociológico para facilitar el aprendizaje por parte de un estudiante del *idioma francés*, el sistema esta

desarrollado en dos versiones, una en donde el alumno debe captar las expresiones que se dirigen hacia él y otra en donde un agente de software representa al alumno y entonces el estudiante puede “imitar” las expresiones que realiza el agente. En este caso la mayoría de los estudiantes prefirió y obtuvo un mayor aprendizaje utilizando la versión basada en agentes.

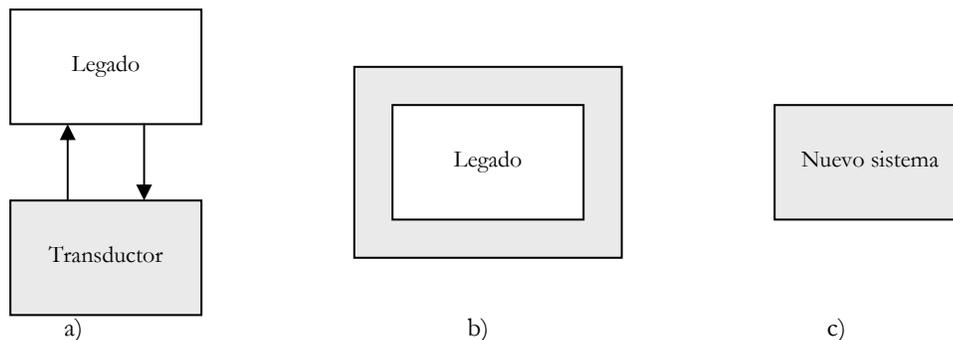


Figura 1.14 Estrategias para convertir sistemas legados en aplicaciones basadas en agentes
a) adicionando un módulo transductor, b) envolviendo la aplicación legada, c) reescribiendo el código

Selker [Selker 94] desarrolló el sistema *Coach* orientado a facilitar el aprendizaje de lenguajes de programación, el sistema contiene tres ventanas en una pantalla, en una ventana el estudiante escribe su código y puede ejecutarlo, la salida se visualiza en una segunda ventana y en la tercera, el agente (que tiene un modelo del estudiante a través de los errores y aciertos en los diferentes programas que realiza) hace sugerencias. Es importante notar que además del agente, el estudiante puede recurrir a la ayuda normal con que cuentan los ambientes de programación. El estudiante puede o no aceptar la ayuda o sugerencias del agente.

Bibliotecas Digitales

En la *Universidad de Michigan* el equipo de Durfee [Durfee 1997] ha desarrollado una arquitectura para adicionar servicios (basados en agentes) en bibliotecas digitales, los más destacados son los de búsqueda y notificación al usuario cuando el conocimiento solicitado ya se encuentra disponible. Sánchez [Sánchez 1997] también trabaja en la *adición de agentes* a las bibliotecas digitales para mejorar los servicios al usuario.

Ayala [Ayala 1998] ha desarrollado un modelo colaborativo como parte de la evolución natural de los sistemas que han pasado desde la instrucción asistida por computadora, la instrucción asistida por sistemas y tutores inteligentes, ambientes de aprendizaje inteligentes, soportes de aprendizaje colaborativo hasta los sistemas basados en agentes para auxiliar en el aprendizaje. El sistema *Gracile* consta de dos tipos de agentes: de dominio y mediadores. Los agentes de *Gracile* son deliberativos, esto es, toman decisiones sobre los aspectos que pueden contribuir al aprendizaje del estudiante. *Gracile* se utiliza para la enseñanza del idioma Japonés. Los agentes de dominio asisten a los estudiantes en la construcción y análisis de oraciones mediante la aplicación de reglas expresadas en Prolog. Los mediadores por su parte trabajan en forma cooperativa para encontrar oportunidades de

aprendizaje para el estudiante, generando un modelo representando e intercambiando creencias (de las capacidades de su estudiante) para obtener propuestas sobre prácticas que pueden serle útiles y así, facilitar y mejorar su aprendizaje. El modelo es un conjunto de creencias que obtiene el agente mediador con respecto a los objetivos de aprendizaje; sus compromisos y capacidades son actualizables en cualquier momento para reflejar lo más fielmente a los miembros del grupo.

Plataformas de Agentes

Los agentes móviles tienen la posibilidad de “moverse” de un equipo hacia otro copiándose junto con los datos que pudiera contener. El Agent TCL de Robert Gray (presentado como tesis doctoral) [Rus 1997a] es un *lenguaje y una arquitectura de agentes móviles* que les permite el paso de un equipo a otro mediante una instrucción, al mismo tiempo que sus mecanismos censan el tráfico de la red para permitirles la movilidad en forma eficiente y con tolerancia a fallas.

Actualmente existen diferentes plataformas para desarrollar agentes, algunas se indican en el anexo A en este documento. Las aplicaciones de estas plataformas es de naturaleza diversa, por ejemplo, para brindar servicios apoyándose en la web, entre estos se encuentra *Agent Cities* iniciado como una plataforma llamada Web Services (basada en tecnología de software independiente de los agentes).

Comercio Electrónico

El área de comercio electrónico usando agentes ha tenido varios desarrollos en experimentos de laboratorio con posibilidades potenciales de aplicación práctica.

En la actualidad los sistemas que están en funcionamiento, donde algunos utilizan agentes, son los llamados sistemas recomendadores [Scahfer 1999] y son una de las tendencias de la actualidad en los sitios de comercio electrónico en Internet. Estos sistemas son deseables porque hacen sugerencias a los compradores facilitándoles la decisión sobre adquirir o no algún producto.

Algunos de los sitios que cuentan con recomendadores son:

- a) *www.amazon.com*, el cual tiene dos recomendadores, uno que sugiere un libro cuando muchas personas que compraron otro también compraron ese. El segundo envía un correo electrónico al cliente cada vez que se adicionan productos al catálogo de Amazon. Además Amazon cuenta con un lugar donde un cliente puede enviar comentarios sobre un libro que adquirió y de esta forma otros compradores pueden visualizar los comentarios y hacer una mejor decisión.
- b) *www.cdnow.com*, funciona en dos modos, en el primero cuando un cliente pide un álbum de discos, el recomendador sugiere 10 albums relacionados con el artista solicitado. La segunda forma es cuando el cliente llena un formato de aquellos albums de un artista que el ya tiene, aquellos que le gustaría tener y los que no desea. Entonces el segundo recomendador utilizando esta información hace una sugerencia sobre los albums que puede adquirir. Sobre la base de las respuestas, el recomendador hace los ajustes convenientes.
- c) *www.ebay.com*, le permite a compradores y vendedores contribuir con perfiles de retroalimentación sobre las negociaciones que realizan. La retroalimentación consiste en el grado de satisfacción (satisfecho/neutral/insatisfecho) así como comentarios

específicos. La información es utilizada por un recomendador para proporcionarla a los compradores.

- d) *www.levis.com*, recibe recomendaciones sobre artículos de la marca Levi's. Los clientes indican si son hombres o mujeres y proporcionan información sobre algunos aspectos de su preferencia y en base a esta información el sistema les hace las sugerencias.
- e) *www.moviefinder.com*, Le permite a los clientes localizar una película con un tema de sus favoritos o bien ligas en donde puede encontrar información del director o de los artistas principales. El predictor se basa en intereses previos del cliente y de la calificación que ha dado a cada uno de ellos.
- f) *www.reel.com*, es similar a Amazon.com en cuanto a que hace sus recomendaciones sobre la base de información en la que se encuentra con respecto a la página de cada película.

La recomendación puede ser automática o por solicitud del cliente. Las estrategias utilizadas para recomendar productos se resumen en la figura 1.15.

SÍTIO	ESTRATEGIA DE RECOMENDACIÓN
www.amazon.com	Artículos similares Notificación por correo electrónico Comentarios de los compradores anteriores
www.cdnow.com	Artículos similares Los más solicitados
www.levis.com	Comentarios de los compradores anteriores
www.moviefinder.com	Artículos similares Los más solicitados
www.reel.com	Artículos similares Navegación

Figura 1.15 Estrategias para recomendar artículos en algunos sitios de comercio electrónico

Con el comercio electrónico es posible proporcionar precios individuales a cada transacción, de donde resulta conveniente contar con alguna forma de negociación automatizada. Una de las ventajas es que, es posible dar un trato personalizado a cada uno de los clientes en función del volumen de compras o algún otro criterio que puede encapsularse en un agente del cliente; a uno frecuente además de recomendarle un artículo en función de su perfil se le pueden ofrecer promociones especiales, esto es más difícil en otros enfoques por la cantidad de información que debe procesarse constantemente. En algunas ocasiones es un conocimiento no explícito pues lo tienen los gerentes de cuenta de cada cliente. Este tipo de servicios puede realizarse por cliente o por producto.

Algunas ventajas de utilizar agentes en el comercio electrónico son: a) permiten a las personas usar el tiempo que gastarían realizando las negociaciones en alguna otra actividad; b) al ser sistemas computarizados, tienen acceso a un mayor número de sitios de interés al ser parte misma del ambiente de Internet.

Dado el auge del comercio electrónico y previendo la necesidad de contar con ayudas para la atención de clientes por una parte y para poder hacer promociones y negociaciones en forma autónoma, han surgido los *agentes promotores (push)* [Berghel 1998] son útiles para distribuir información sin el requerimiento explícito del consumidor. Esto nos brinda una forma de envío de mensajes multidestino (multicast), teniendo la posibilidad de mandar los datos de promociones u ofertas a un gran número de destinatarios.

La contraparte de los agentes promotores son los *extractores (pull)*, que se encargan de tomar información, sin la solicitud explícita de un usuarios usando estos, se pueden mantener actualizados los programas en una máquina, dado que el agente extractor se encarga de detectar cuando exista una versión nueva en un servidor y entonces la distribuye a los sitios que la usan.

Las tecnologías de agentes promotores y extractores tienden a ser utilizadas como parte de las estrategias de comercio electrónico para llegar a número grande de clientes o para mantener actualizados los sistemas sin tener la necesidad de llevar a cabo las acciones de actualización.

El proyecto *Sardine* [Morris 2000] es un sistema basado en agentes para comprar boletos de avión a precios variables. El vendedor debe determinar dinámicamente si da un asiento o no dependiendo de la capacidad de los aviones y en función de las múltiples pujas que le llegan. En el sistema únicamente se evalúa la simulación de tres tipos de subastas, a la baja, a la alza y precios constantes. Los resultados que se obtuvieron es que el enfoque de precios a la alza es más conveniente para una subasta desde la perspectiva del proveedor.

Leyton et. al [Leyton 2000] propone que para que los agentes que participan en *subastas electrónicas* realicen en conjunto con un coordinador una pre-subasta que permita identificar mediante la participación cooperativa de los agentes aquel que tiene mayor posibilidad de ganar de tal forma que al entrar a la subasta principal el que puede ganar haga una oferta más conservadora mientras que otros pueden esperar otra oportunidad o intentar participar pero, teniendo en cuenta que es muy probable que gane otro agente. Este es un ejemplo de sistema en donde únicamente se toma el precio como factor de decisión.

El proyecto *Kasbah* [Chavez 1997] es un sistema multiagentes que funciona en la Web donde el usuario puede definir agentes para realizar compras o ventas en nombre del primero. El propósito principal de Kasbah fue ayudar al usuario a ahorrar tiempo en realizar las negociaciones típicas de las operaciones de compra y venta. Al lugar en que los agentes interactúan se conoce como el “lugar de mercadeo” («marketplace»). Aquellos que participan en el sistema deben soportar el protocolo de interacción entre agentes, aunque en sus principios el sistema realizaba operaciones de compra y venta relativamente simples. Los agentes no son inteligentes, solamente realizan negociaciones utilizando las funciones predefinidas en el sistema y utilizando los valores de los parámetros indicados por el usuario propietario del agente. En Kasbah un agente es como un anuncio de una oferta o de la solicitud de un producto, pero con la ventaja de que ahora el anuncio es activo, es decir, puede buscar a su complemento. Por ejemplo si un usuario introduce al sistema un agente que compra, entonces este buscará a otros que realicen operaciones de venta. Si un usuario quiere vender un producto debe indicar algunos parámetros, entre ellos la fecha en que desea tener vendido el producto, el precio deseado y el menor precio aceptable además de una curva de reducción del precio en caso necesario, la cual dirige la estrategia de negociación. La curva de reducción de precio puede ser una línea recta, una parábola o alguna otra curva que refleje el interés del usuario. Si un usuario quiere comprar un producto, los parámetros que debe proporcionar son, la fecha en que desea haber comprado el producto, el precio que desea pagar y el máximo precio aceptable además

de una curva de incremento que en caso necesario utilizará para negociar con los agentes vendedores. Cuando un agente ha encontrado un producto, en ocasiones para cerrar el trato se consulta a su usuario para la decisión final, lo cual puede hacer en línea o mediante un correo electrónico. En otros casos el mismo agente cierra el trato. Para asegurar que los tratos se hagan de manera correcta, se ha propuesto un conjunto de agentes reguladores que se dedican a evitar acciones ilegales por parte de los agentes.

El prototipo de Kasbah utiliza algunos métodos para comprar y vender entre los agentes:

- a) `accept-offer(agent, from-agent, offer)` se usa para preguntarle a un agente si acepta o no una oferta, el agente regresa “acepto” o “rechazo”.
- b) `what-is-price(agent, from-agent)` este método sirve para que un agente (`from-agent`) pregunte el precio a otro agente (`agent`).
- c) `what-is-item(agent, from-agent)` este método lo usa `from-agent` para preguntarle a `agent` sobre el artículo que intenta vender o comprar.

Otros métodos permiten conocer cuales son los agentes compradores o vendedores, adicionar uno nuevo al sistema como un comprador o un vendedor y terminar o indicar que ya se ha realizado una compra o venta. Estos métodos están predefinidos y no es posible utilizar otros para ampliar las interacciones entre los agentes. Únicamente pueden existir compradores o vendedores. Por otra parte no se mencionan los aspectos referentes a la ontología que maneja cada agente, por lo que un requisito es que los nombres de los productos que se compran y venden deben coincidir. En caso de que un agente tenga alguna falla no se menciona que hacer al respecto. El lenguaje de comunicación entre agentes es propio y en futuras implementaciones se consideraría utilizar algún lenguaje estándar como KQML o alguno parecido.

El proyecto *Mari* [Tewari 2000] (*Multi-Attribute Resource Intermediary*) («*Intermediario de Recursos Multi-Atributos*») propone mejorar los mercados en línea que involucran compradores y vendedores de bienes y servicios intangibles. *Mari* es una arquitectura intermedia que pretende generalizarse como una plataforma para la especificación y distribución de bienes y servicios heterogéneos. Es un intento para superar a los sistemas que únicamente toman en cuenta el precio como factor de decisión para que un cliente elija un producto; a través de considerar información adicional sobre el producto que se intercambia. *Mari* integra características de los proyectos *Market-Market* y de *Tête-à-Tête* para modelar las preferencias del usuario y las negociaciones en cuanto al precio. *Mari* intenta soportar múltiples compradores y vendedores con múltiples productos utilizando una ontología común para que sea posible intercambiar productos entre los múltiples compradores y vendedores. Para participar en *Mari* un vendedor (humano) crea un agente y un comprador (humano) crea otro. Cada usuario selecciona funciones de incremento de precios preespecificadas que aparecen en el lado derecho de una pantalla diseñada en HTML.

El proyecto *FishMarket* [Noriega 1997] emplea el concepto de agentes para modelar una subasta de pescado, en donde se considera que las interacciones son relativamente sencillas y que los procesos de deliberación de un agente son complejos y en donde el tiempo juega un papel importante, porque las decisiones se toman en un periodo de tiempo pequeño. Principalmente da cuenta de los aspectos de diálogos entre los agentes participantes, requiriendo que estos se comuniquen en una ontología común. Los diálogos involucran múltiples participantes quienes intercambian mensajes en lenguajes complejos. Los artefactos monológicos clásicos no son adecuados para enfrentar situaciones de este grado de complejidad. El trabajo de Noriega propone que los protocolos de los diálogos sean

obligados y no solamente deseables. Los protocolos de interacción es un aspecto de los sistemas multiagentes que para el proyecto FishMarket se ha encontrado particularmente significativo, tanto desde la perspectiva teórica como de aplicación. Para que los diálogos se realicen en forma adecuada se tienen dos tipos de reglas que permiten regular las interacciones entre los agentes: obligatorias estructurales y discrecionales. Las obligatorias estructurales permiten manejar las colisiones mientras que las discrecionales se aplican a través de la participación de los miembros de la administración. Algunos ejemplos son:

- a) *Reglas obligatorias.* 1.- Los turnos para hacer pujas tienen un tiempo fijo de un segundo. 2.- Todas las señales de apropiación (mío) en una puja se reconocerán. 3.- Si se recibe más de una señal “mío” en un turno de puja, entonces se declara una colisión. 4.- Si se declara una colisión, el lote subastado se vuelve a subastar pero a un precio con un incremento. 5.- El incremento de precio es una constante 20%.
- b) *Reglas discrecionales.* 6.- Si un comprador excede su límite de crédito, se inhibe para seguir participando en la subasta. En este caso el comprador puede negociar con la administración que se le permita seguir participando. 7.- Si un comprador es inhibido y solicita una actualización en su crédito, puede seguir participando en la subasta.

En la práctica, la aplicación de algunas reglas puede involucrar un diálogo posterior entre el subastador y el comprador; por ejemplo, cuando un agente se excede en su límite de crédito y requiere solicitar una ampliación. La caracterización de diálogos, excluye soliloquios y argumentación monológica, por la virtud de la condición de multiplicidad. Un proceso dialógico debe satisfacer las condiciones siguientes:

- a) Debe haber al menos *dos participantes*.
- b) Los participantes tienen *creencias*.
- c) Los participantes *intercambian locuciones*.
- d) Los intercambios de locuciones están sujetos a un *protocolo de interacción común*.
- e) *Las creencias de los participantes pueden cambiar* debido a los diálogos.

En FishMarket hay dos clases de agentes: externos (compradores y vendedores) e internos (administradores). Los vendedores llevan bienes a una subasta y los compradores llevan dinero. A través de los procesos dialógicos los bienes y el dinero cambian de manos. Pero los compradores y los vendedores nunca se hablan directamente, sino que únicamente interactúan a través de los administradores del FishMarket. Los procesos dialógicos ocurren en una localidad específica e involucran participantes que toman un papel instanciable por cualquier agente. A esto se le llama una escena. A un conjunto de escenas cuya precedencia temporal y causal se indica como a una gráfica de ejecución se le llama estructura de ejecución. En FishMarket se han creado estructuras de ejecución, los agentes que pueden tomar los papeles de esa estructura ya están prefijados, y los propósitos de los agentes son fijos. Los bienes que pueden adquirir se encuentran en un Catálogo. Y pasan a través de diferentes etapas durante una subasta. Primeramente los registra el vendedor con el admisor de bienes que lo inscribe en un catálogo de “bienes disponibles” luego, estos pasan hacia el subastador quien asigna el estado no-vendido, entonces una vez que pasa por la subasta se le asigna el estado “vendido” o “apartado”. Conforme el bien cambia su estado, información nueva o diferente se le

asigna, un número de catálogo, un precio inicial, un precio de reserva, un precio de venta, y el precio actual en el tiempo t , su vendedor y su comprador.

1.3 REPRESENTACIÓN DEL CONOCIMIENTO

Un aspecto que le permite a los agentes interactuar con otros y su ambiente se apoya en una representación de las características, recursos y entidades de su entorno. Se han desarrollado diferentes formas de representar el conocimiento utilizadas en sistemas de información, pero aún no existe un consenso para tener una forma universal de esto.

El conocimiento se ha representado en los sistemas de información utilizando diversos formalismos. En el área de software de sistemas ha utilizado principalmente gramáticas y algunas estructuras de datos, la Inteligencia Artificial es un área donde se tiene una mayor diversidad de formas para representar el conocimiento. La intención de esta última es contar con estructuras de datos y procedimientos eficientes para captar, representar, almacenar y recuperar el conocimiento dotando a los sistemas de capacidades que se pueden atribuir a los seres humanos.

Algunas de las formas que se han utilizado para representar el conocimiento son (figura 1.16):

- a) *Estructuras de datos*, principalmente representan objetos o estructuras y sus componentes.
- b) *Lógica matemática*, representa los hechos en forma declarativa y a partir de los cuales se pueden hacer deducciones. Sobre este particular se han desarrollado a su vez lenguajes de programación especializados como Prolog y sus variantes.
- c) *Procedimientos*, es una forma de representar el conocimiento mediante los procesos que indican como usarlo.

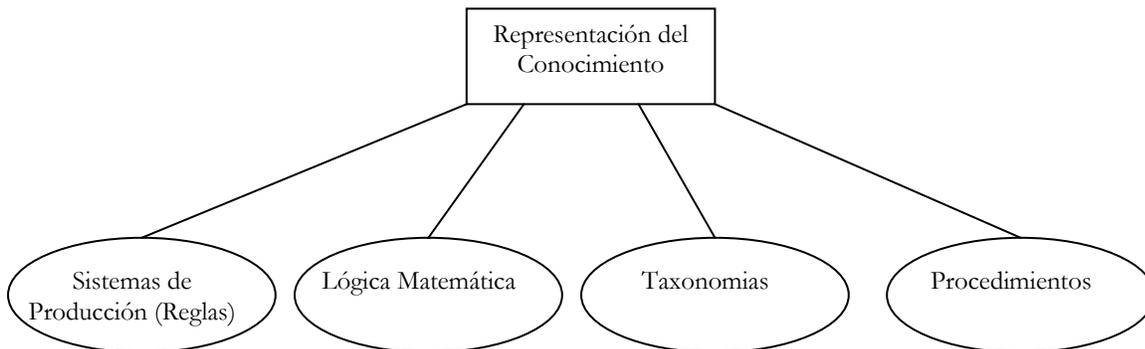


Figura 1.16 Algunas formas de Representación del Conocimiento

En las estructuras de datos utilizadas en la representación del conocimiento se encuentran diferentes tipos como los marcos, guiones, redes semánticas y taxonomías.

Las *Redes Semánticas* [Quillian 1968] son de las primeras formas de representación de conocimiento basadas en estructuras. La intención original fue la de proveer de una forma para representar los significados de palabras en Inglés. La información se representa como un conjunto de nodos

conectados unos con otros mediante arcos etiquetados que ilustran relaciones entre ellos. Las relaciones pueden ser de diferente naturaleza como por ejemplo: tiene, come, es-un. Y la estructuración es la de una red. Por lo tanto la inferencia implica hacer una búsqueda en la red. En [Olivares 1991] se plantea un mecanismo para la *construcción automática de una red semántica* a partir de oraciones declarativas en lenguaje natural restringido y distribución lingüística; la recuperación de conocimiento se realiza mediante oraciones imperativas.

Los *Marcos* se utilizan para describir colecciones de atributos de un objeto desde diferentes puntos de vista, por ejemplo, se pueden indicar las propiedades y los usos que posee normalmente una mesa. Intenta capturar la experiencia previa que se tiene cuando se analiza un objeto y a partir de la cual se pueden construir nuevas estructuras de conocimiento o para describir situaciones. Un marco generalmente describe una clase de objetos y consiste de un conjunto de ranuras que representan aspectos del objeto. Las ranuras pueden llenarse con otros objetos o con condiciones que deben cumplirse en él; o bien, tener un valor por omisión que indica la propiedad típica y que se utiliza en caso de ausencia de información y se pueda utilizar para asumirla.

En los *Scripts* («guiones») se representan secuencias de actividades de situaciones comunes (en forma estereotipada). Un guión consta de un conjunto de ranuras. Con cada ranura se asocia algún tipo de información en forma similar a como ocurre en un marco, pero en este caso, la información indica las acciones que se realizan. Este tipo de estructuras es útil para registrar la causalidad de eventos del mundo real. La descripción de situaciones típicas se utilizan para que en los casos de ausencia de información se puedan tomar acciones que se consideran características de lo que se está representando. Por ejemplo el guión para comprar semillas como maíz, en donde el cliente pregunta por el producto, el proveedor le informa los tipos de semilla que puede venderle, el precio, condiciones de pago y lugar de la entrega. Luego el cliente delibera y emite una decisión y solicita el producto que requiere, efectúa su pago. Entonces el proveedor le entrega o envía el producto que el cliente recibe y concluye el guión. Este tipo de estructura es en la que se inspira la descripción de los papeles en el modelo de interacción de agentes propuesto.

Una *Taxonomía* se usa para descomponer objetos en otros más simples. Una de las formas más utilizadas es la relativa a relaciones es-un o es-parte-de. Por ejemplo se puede tener el árbol de relaciones es-un de la figura 1.17 o las relaciones es-parte-de para componer a un objeto complejo.

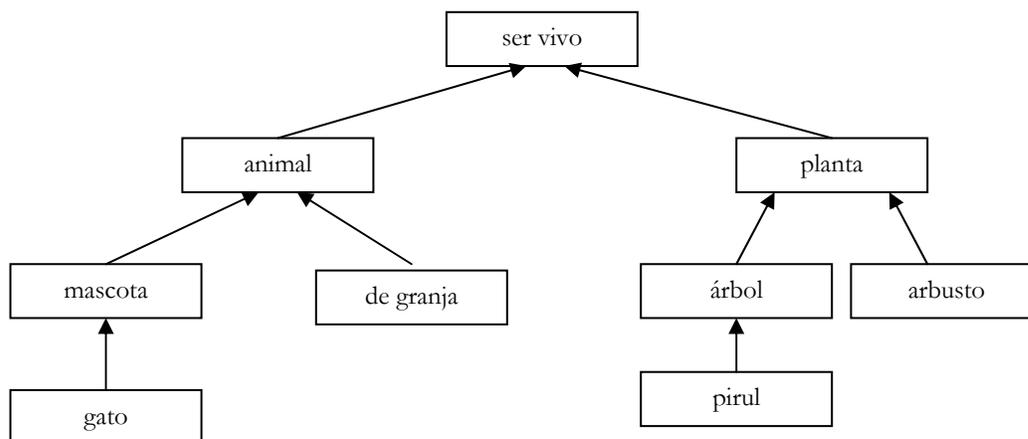


Figura 1.17 Organización de objetos mediante una taxonomía. Las flechas de abajo hacia arriba indican la relación es-un, esto es el nodo inferior es una instancia del nodo de nivel superior.: pirul es un árbol.

La *Dependencia Conceptual* es una teoría de cómo representar el significado de oraciones de lenguaje natural. Se forma por una estructura de nodos de información y un conjunto de primitivas mediante las cuales, se pueden formar piezas de información con significado. Por medio de las primitivas se construyen piezas con significado en donde se registra la temporalidad, esto es si es algo que ocurrió en el pasado o sucederá en el futuro. La inferencia de conocimiento se aplica utilizando como base las primitivas que expresan las relaciones entre los nodos.

Las representaciones basadas en la *Lógica Matemática* utilizan una notación para representar los hechos del mundo real que pueden ser falsos, verdaderos o tener valores intermedios entre estos dos; además se usan reglas de inferencia para llevar a cabo deducciones de otros hechos. Por ejemplo si se tiene un hecho que indica que si un ser es humano entonces es mortal y el hecho de que Emilio es hombre, mediante una regla de inferencia llamada Modus Ponens se puede deducir que Emilio es mortal. Para generar inferencia se han desarrollado diversos métodos que van desde resolución hasta mecanismos especializados como la inferencia usando redes bayesianas; los predicados se han representado en diferentes lógicas entre las que se encuentran las modales, la difusa, la temporal y deóntica entre otras.

Los *Procedimientos* representan secuencias de acciones agrupando la funcionalidad como una caja negra donde no es posible en determinar con precisión el conocimiento específico que se está aplicando. Por otra parte es la forma más difundida de tener conocimiento de cómo-hacer.

1.3.1 ONTOLOGÍAS

Las ontologías son una forma de representación de conocimiento basado en taxonomías, permiten estructurar en forma inambigua los conceptos de un dominio específico y así determinar su validez en cierto dominio de conocimiento como: electrónica, medicina, mecánica, etc.

Los trabajos que se han realizado sobre ontologías los clasificamos como:

- a) *Especificación de ontologías*, se hacen procesos de ingeniería de conocimiento para encontrar los conceptos y la forma de estructurarlos. Algunos trabajos sobre esto son los de CyC [Lenat 1990] y las especificaciones de FIPA.
- b) *Integración de ontologías*, en donde se pretende integrar ontologías mediante la unión de dos o más ontologías. Sobre esto existe el trabajo de Ontolingua [Farquar 1997] y *Observer* [Kashyap 1998].
- c) *Mapeo de ontologías*, se refiere a encontrar equivalencias entre ontologías utilizando información que puede resultar ambigua y que se propone en este trabajo como alternativa de solución al tratamiento de ontologías.
- d) *Aplicaciones de ontologías*, se tienen en diversos lugares, por ejemplo en la estructuración de conceptos en una biblioteca digital como el caso de la Biblioteca Digital de la Universidad de Michigan [Weinstein 1997] o el aprovechamiento de intercambio entre bases de datos realizadas por diferentes grupos de personas [Huhns 1997].

En el proyecto *CyC* [Lenat 1990] se utiliza una ontología organizada alrededor del concepto de categoría, también referida como clase o colección. Las colecciones se organizan en una jerarquía de generalización/especialización es decir superconjuntos y subconjuntos. En esta jerarquía el conjunto universal se llama cosa («thing») mismo que se particiona en dos subconjuntos, *cosaRepresentadaInternamente* y *cosaRepresentada*. Las instancias de la primera incluyen el número '5', la cadena "terrible", etc. las cosas que pueden representarse en Lisp, dado que es el sustrato de CycL. CycL es el lenguaje de expresiones de Cyc. La otra parte de 'cosa' incluye clases como *objetoIndividual* y *colección*. Los objetos individuales son cosas como Alfredo, *palacioBellaArtes*, 24feb1990. Las instancias de *Colección* son cosas que agrupan cosas como Mesa (el conjunto de todas las mesas), Cena (el conjunto de todos los eventos de cena), y así sucesivamente. La organización continua sucesivamente hacia abajo. Esta colección de conceptos se utiliza para establecer las características de aquellos objetos de interés para una persona y poder validar sus componentes. Actualmente el proyecto *CyC* continua en la empresa CycCorp que fundó Douglas Lenat.

Farquar [Farquar 1997] supone que en el mundo existen ontologías bien diseñadas y modulares, por lo que, la construcción de una ontología nueva es una cuestión de ensamblar algunas existentes como lo describe el servidor *Ontolingua*, el cual proporciona diferentes tipos de operaciones para combinar ontologías: inclusión, restricción y refinamiento polimórfico. Por ejemplo la inclusión de una ontología en otra tiene el efecto de que la ontología compuesta consiste de la unión de las dos ontologías (sus clases, relaciones y axiomas). El servidor *Ontolingua* permite reutilizar ontologías existentes para tareas como integración de información y desarrollo de bases de conocimiento. Consiste de un conjunto de herramientas y servicios para soportar el proceso de integración de ontologías. Se espera que las ontologías sirvan para que otros grupos de personas puedan ensamblar una ontología nueva a partir de las existentes. El servidor de ontologías está disponible a través de Internet¹².

Las ontologías se usan para organizar el conocimiento en diferentes sistemas de información. Una *ontología se entiende* [Guzmán 1998] *como una especificación de conceptos en forma de árbol formando una taxonomía jerárquica con relaciones de subconjunto de padres a hijos* (figura 1.18). Los conceptos son independientes del lenguaje de comunicación entre agentes, se refieren a lo que un agente entiende de la realidad, cada concepto es único en la ontología. Cada nodo del árbol representa exactamente un concepto.

La definición de Ontología de Gruber [Gruber 1993] *es una especificación explícita de una conceptualización*. En donde una conceptualización es una visión simplificada (abstracta) del mundo que se quiere representar para algún propósito. Una visión del mundo incluye conceptos, objetos y otras entidades en donde existen relaciones entre estos.

Para especificar una ontología [Gruber 1993] Gruber propone los criterios siguientes:

- a) *Claridad*, en una ontología se debe comunicar el significado de los elementos definidos con objetividad. Las definiciones deben ser independientes del contexto social o computacional. Cada definición debe estar definida por elementos necesarios y suficientes. Es conveniente también documentar cada definición mediante lenguaje natural.

¹² <http://ontolingua.stanford.edu>

- b) *Coherencia*, significa que al menos los axiomas del concepto deben ser coherentes. En contraste si una inferencia parte de axiomas que contradicen una definición o un ejemplo dado informalmente, entonces la ontología es incoherente.
- c) *Extensibilidad*, el diseño debe anticipar usos del vocabulario compartido. En otras palabras, se debe ser capaz de extender y especializar la ontología en forma monotónica, esto es sin requerir revisar las definiciones existentes.
- d) *Mínima guía de codificación*, la conceptualización se debe especificar en un nivel de conocimiento que minimice la dependencia sobre la codificación utilizada. La razón de esto se debe a que los usuarios de una ontología pueden implementarse en diferentes sistemas y estilos de representación.
- e) *Compromisos ontológicos mínimos*, debe tener el menor apego a una realidad específica, esto significa que la ontología debiera instanciarse y especializarse libremente para diferentes necesidades.

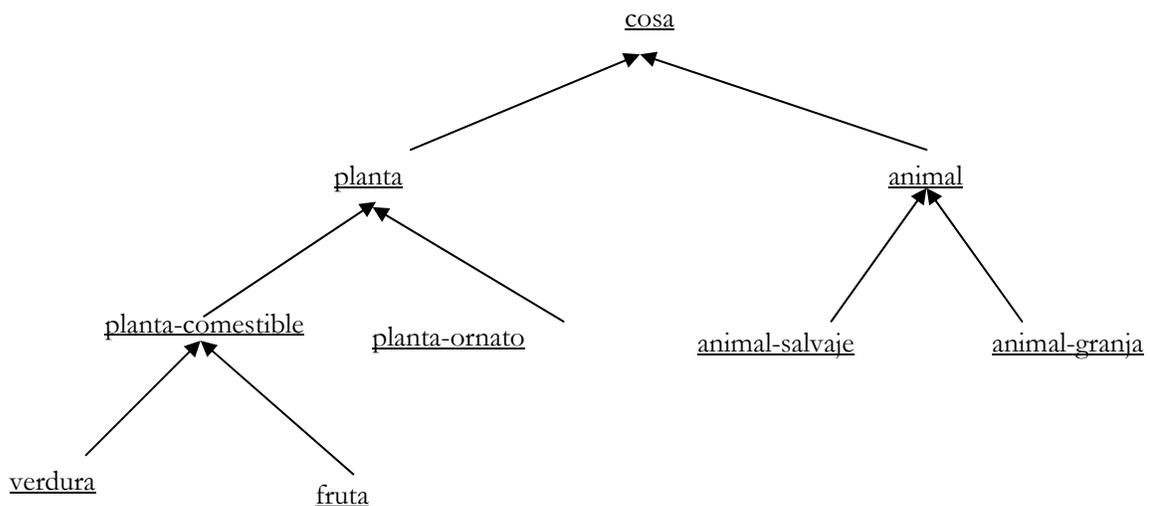


Figura 1.18 Una ontología es una taxonomía jerárquica, las flechas indican relaciones es-un de hijos a padres

Los elementos para considerar el manejo de ontologías mixtas son las palabras y los conceptos [Guzmán 1997] “La diferencia entre una palabra y un concepto radica en que, una palabra o vocablo puede representar varios conceptos o acepciones. Un concepto en cambio (por definición) no es ambiguo”, por ejemplo, la palabra mano se mapea a los conceptos: mano-humana (parte del cuerpo humano), mano-majador (la mano del mortero, del metate), mano-etapa (una mano de pintura). En el resto del documento escribiremos los conceptos con letras minúsculas subrayadas y las palabras que los referencian con minúsculas únicamente.

Una ontología permite captar los conceptos que pertenecen a un dominio específico, por ejemplo electrónica, medicina, mecánica, etc.

Cuando dos agentes tienen diferentes ontologías, el problema es encontrar para un concepto en una ontología, los conceptos más aproximados en la otra ontología. Si por ejemplo se tienen dos agentes, uno de ellos quiere comprar un florero como el que se muestra en la figura 1.19 (en este caso es el concepto que desea comunicar al otro agente) entonces el agente comprador trata de describirle al vendedor lo que desea. Los *conceptos corresponden a representaciones internas* propias de cada sistema, las *palabras representan unidades de información que pueden compartir y que mapean conceptos*. Dado que los conceptos pertenecen a la representación interna no es posible (en general) copiarla en otro sistema y obtener el mismo significado. De esta forma, las palabras que si son comunes entre estas entidades, se utilizan para establecer los conceptos a los que se refieren.

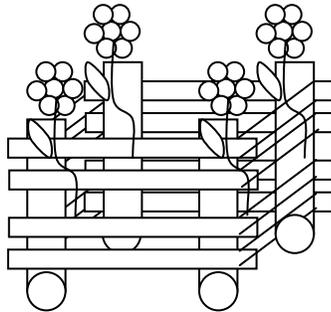


Figura 1.19 Un objeto para describirlo mediante una ontología y mapearlo en otra

Por ejemplo cuando un comprador le describe un objeto deseado a un vendedor, lo hace:

- a) En *Español y por teléfono*, “se trata de una canasta cuadrada de palma con cuatro flores grandes en las esquinas de la canasta”. Aquí aún existe la posibilidad que la canasta sea más pequeña o más grande de lo esperado, colores equivocados, material inaceptable.
- b) Enviándole un *dibujo* en CAD-CAM¹³ al vendedor indicándole las medidas y el tipo de flor que se espera. En este caso es posible que indicándole las medidas y una imagen de lo que se desea exista menos ambigüedad.
- c) Con una *interacción* frente al vendedor, aquí el comprador puede tratar de explicarle con ejemplos que tenga a la mano, aunque esto no garantiza la completa comprensión por parte del vendedor de lo que quiere el comprador. También puede suceder que el vendedor tenga algo parecido a lo que busca el comprador y entonces este identifica el artículo que quiere y solamente le comunica al vendedor que es ese artículo el que quiere.
- d) Llevándole un *ejemplo* de lo que desea al vendedor, aquí la ambigüedad se elimina pero aún falta que el vendedor tenga o pueda conseguirle el artículo deseado al comprador.

El problema en la referencia a conceptos de dos ontologías, es que se hace mediante palabras ambiguas, por ejemplo, en la figura 1.20 se muestran dos agentes intentan referirse a un concepto común, en este caso maíz. El problema es que el agente de la ontología 1 usa la palabra ‘maíz’ para referirse al concepto maíz mientras que el de la ontología 2 utiliza la palabra ‘corn’.

¹³ Programa de diseño gráfico

Se presentan algunos problemas en los mensajes que intercambian dos agentes, como:

- El uso de lenguajes diferentes, por ejemplo, uno utiliza Inglés (corn) y otro Español (maíz).
- Falta de precisión en el significado que se da a las palabras que se utilizan, por ejemplo si alguien emite la frase en un idioma comprensible entre ambas partes (vender perico 15cm), el problema es que no se sabe que significa (concepto al que se refiere) la palabra 'perico', si es un perico-herramienta o un perico-animal.

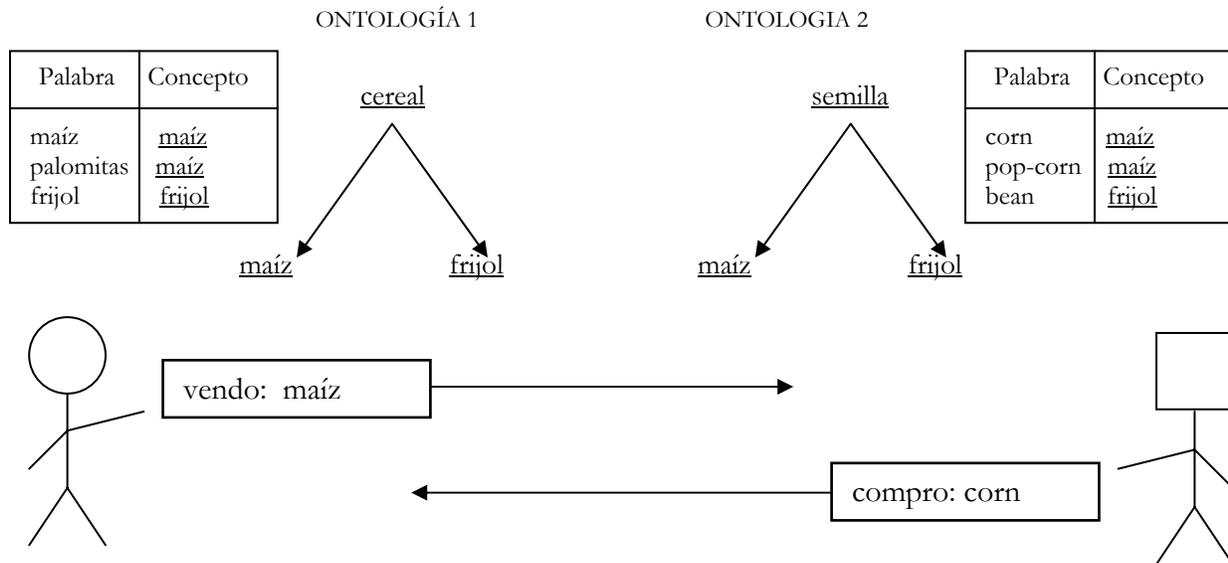


Figura 1.20 Agentes que manejan ontologías diferentes

Para superar la primera forma de ambigüedad es importante que las unidades léxicas se empleen en un idioma común o se utilice alguna herramienta para generarlo.

Para la segunda forma de ambigüedad es conveniente contar con alguna forma de interacción entre las partes para que, en caso de que alguna de ellas, requiera más información tenga la posibilidad de solicitarla, si se presenta la situación de falta de información, entonces se solicite. Nuestra propuesta es reducir la ambigüedad mediante un comparador de ontologías mixtas, en donde se intenta encontrar los conceptos similares entre dos ontologías vía el intercambio de palabras que pueden ser ambiguas (por mapear una palabra a diferentes conceptos, muchas palabras al mismo concepto). Nuestro algoritmo para que dos agentes con ontologías diferentes se "entiendan" sin necesidad de establecer una ontología de nivel superior o de la integración de las ontologías involucradas se desarrolla en el capítulo 3.

1.4 PLANEACIÓN

Cuando existe una o varias entidades donde cada uno tiene varios propósitos que pretende alcanzar debe realizar diferentes acciones. Aquí nombramos a las entidades como agentes. Los agentes interactúan con el ambiente y con otros de ellos utilizando recursos y tratando de realizar algunas acciones u obtener otros recursos para considerar como alcanzados sus objetivos o propósitos. Tradicionalmente el encontrar una secuencia de acciones dada una situación inicial para alcanzar una meta se conoce como el problema de planeación. El *problema de planeación* [Sánchez 1999] *consiste en dada una situación y un conjunto de operadores, encontrar la secuencia adecuada para pasar de una configuración inicial a una deseada (objetivo, propósito o meta)*. Por ejemplo, para el dominio de mover bloques etiquetados, se tiene una configuración inicial, se pretende llegar a una configuración final o deseada (figura 1.21). Para lograrlo se aplican algunas acciones sobre la situación inicial moviendo algunos cubos, subiéndolos o bajándolos o bien apoyándolos en la mesa. En este caso el plan consiste de una serie de movimientos para colocar bloques sobre la mesa o sobre otro bloque.

En este caso únicamente una entidad es la que se encarga de las acciones de mover los cubos.

Al aplicar la planeación es importante distinguir entre el tipo de problema particular y los valores involucrados en el problema, por ejemplo, en el problema de los bloques se mueven algunos para llegar a una configuración deseada y una instancia particular a partir de una configuración inicial como se observa en la figura 1.21. Una instancia de un problema de planeación se refiere a una configuración particular del ambiente.

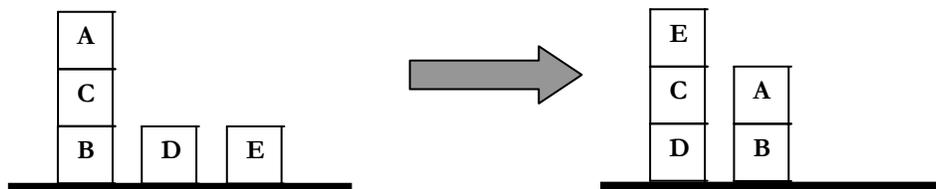


Figura 1.21 Representación gráfica de un problema con bloques

Cualquier instancia de un problema de planeación tiene al menos tres componentes:

- a) Una descripción del *estado inicial* del ambiente.
- b) Una descripción del o los objetivos (propósitos, metas o *estado final*) a alcanzar
- c) Una descripción de las *acciones* aplicables por el agente para pasar del estado inicial al final.

La descripción del mundo de bloques es como sigue: se tiene una mesa plana de tamaño suficiente para colocar los bloques de madera que son de un mismo tamaño, los bloques se pueden colocar uno encima de otro, se tiene un brazo robot que mueve los bloques apilandolos, desapilandolos o moviéndolos de un punto de la mesa a otro. Las acciones que puede realizar el robot son:

Bajar(*a*, *b*) toma un bloque *a* de su posición sobre *b* y lo coloca sobre la mesa; en este caso el brazo debe estar libre para tomar el bloque *a*.

$\text{Subir}(a, b)$ coloca el bloque a y lo posiciona sobre b ; el brazo de estar tomando el bloque a y la superficie de b estar libre.

$\text{Tomar}(a)$ toma un bloque a y lo mantiene sostenido; previo el brazo debe estar libre y a debe estar sin bloques arriba.

$\text{Soltar}(a)$ coloca el bloque a sobre la mesa; el brazo debió estar sosteniendo el bloque a .

La especificación de las condiciones en las que se encuentra el mundo de los bloques se indica mediante los predicados:

$\text{Encima}(a, b)$ el bloque a está sobre el bloque b .

$\text{Mesa}(a)$ el bloque a se encuentra sobre la mesa.

$\text{Libre}(a)$ el bloque a no tiene bloques en la parte superior.

$\text{Sostiene}(a)$ el brazo sostiene el bloque a .

$\text{BrazoVacio}()$ el brazo no sostiene ningún bloque.

Utilizando los predicados anteriores el estado inicial de la figura 1.21 se describe como:

$$\text{Mesa}(B) \wedge \text{Mesa}(D) \wedge \text{Mesa}(E) \wedge \text{Encima}(C, B) \wedge \text{Encima}(A, C)$$

El estado final de la instancia del problema se describe como:

$$\text{Mesa}(D) \wedge \text{Mesa}(B) \wedge \text{Encima}(C, D) \wedge \text{Encima}(E, C) \wedge \text{Encima}(A, B)$$

Para pasar del estado inicial al estado final se aplican las acciones en cierta secuencia. Las acciones se describen como operadores tipo STRIPS, en donde se tienen los *requisitos* que son predicados que se deben de cumplirse para que una acción se aplique, como consecuencia de esto se *borran* los predicados que dejan de ser válidos y se *adicionan* los predicados que son consecuencia de la acción aplicada, las acciones para este problema son:

$\text{Subir}(a, b)$

Requisitos:

$$\text{Libre}(b) \wedge \text{Sostiene}(a)$$

Borrar:

$$\text{Libre}(b) \wedge \text{Sostiene}(a)$$

Adicionar:

$$\text{BrazoVacio}() \wedge \text{Encima}(a, b)$$

$\text{Bajar}(a, b)$

Requisitos:

$$\text{Encima}(a, b) \wedge \text{Libre}(a) \wedge \text{BrazoVacio}()$$

Borrar:

$$\text{Encima}(a, b) \wedge \text{BrazoVacio}()$$

Adicionar:

$$\text{Sostiene}(a) \wedge \text{Libre}(b)$$

$\text{Tomar}(a)$

Requisitos:

$$\text{Libre}(a) \wedge \text{Mesa}(a) \wedge \text{BrazoVacio}()$$

Borrar:

$$\text{Mesa}(a) \wedge \text{BrazoVacio}()$$

Adicionar:

$$\text{Sostiene}(a)$$

$\text{Soltar}(a)$

Requisitos:

$$\text{Sostiene}(a)$$

Borrar:

$$\text{Sostiene}(a)$$

Adicionar:

$$\text{Mesa}(a) \wedge \text{BrazoVacio}()$$

Para aplicar las acciones en una instancia de un problema, se parte de la situación inicial y se verifica cuales acciones unifican sus requisitos con los predicados. Cuando son varias las acciones que unifican se dejan indicadas las listas en donde se adicionan los predicados resultantes. Se repite el proceso de aplicar las acciones que unifiquen con las situaciones que se obtienen en cada rama. Con esto se genera un árbol de búsqueda que tiene complejidad exponencial en el peor de los casos. Los nodos de alguna de las ramas con las que se llega al estado final forman el plan.

Un *plan* es una secuencia de acciones (*parcial o totalmente ordenadas*) que transforma un estado inicial en un estado final sin contradicción entre las acciones, de manera que cuando se alcanza el estado final se satisfacen los objetivos. Un estado, es una representación del mundo en un tiempo particular. El estado está dado por los valores que toman un conjunto de variables que representan a los objetos en ese mundo (predicados). Las acciones del plan pueden ser parciales o totalmente ordenadas en el tiempo, esto significa que pueden ocurrir en secuencia una tras de otra o ejecutarse en paralelo. Cuando la ejecución es en paralelo las acciones deben ser compatibles entre ellas.

Existen diferentes estrategias para sintetizar planes por computadora, algunos toman en cuenta los aspectos del ambiente, otros el número de agentes que efectuarán las operaciones. Una clasificación [Sánchez 1999] de diferentes tipos de planeación que se han desarrollado:

- a) *Planeación clásica*, en donde un agente tiene acceso a todos los recursos de un ambiente, él es el único que produce cambios en el mismo, las acciones se consideran instantáneas. La realización de un plan se hace en dos fases, en la primera se construye el plan y en la segunda se ejecuta. Algunos planificadores de este tipo son Strips y Planex. Generalmente se obtiene una secuencia lineal de acciones. Los planes generados se pueden obtener como un espacio de estados o un espacio de planes. En el espacio de estados cada uno representa un estado del mundo y la solución al problema de planeación es una trayectoria que va del estado inicial al estado meta. En el espacio de planes cada nodo representa un refinamiento del plan hasta alcanzar la meta deseada.
- b) *Planeación con información imperfecta*, se aplica en situaciones donde el ambiente es cambiante y el planificador no puede predecir todos los cambios en el mismo (recuérdese el *frame problem*). El planificador puede conocer algunas cosas imprecisamente o con incertidumbre. Por ejemplo al hacer un plan para preparar una malteada de fresa puede ocurrir que falte lecho o que no haya energía eléctrica así que al ejecutar el plan deben tomarse otras decisiones como conseguir leche. Algunas de las técnicas desarrolladas para planear con información imperfecta son la planeación probabilística, condicional y reactiva. La planeación probabilística construye una secuencia de acciones que se ejecutan en base a las acciones con mayor probabilidad de alcanzar las metas deseadas; algunas técnicas utilizadas son las redes bayesianas y los procesos de decisión de Markov. La planeación condicional se utiliza cuando el ambiente es predecible en alguna media, algunas técnicas de planeación clásica pueden adaptarse para producir planes con etapas condicionales que intentan prever todas las contingencias posibles. La planeación reactiva se utiliza en ambientes muy cambiantes y por lo tanto el agente genera una acción a realizar en cada momento dependiendo de las condiciones que encuentra en el ambiente; este tipo de planeación es muy utilizada en los agentes reactivos.
- c) *Planeación en ambientes dinámicos*, en estos casos es conveniente representar el tiempo en forma explícita para hacer razonamientos. Para esto se ha desarrollado la Planeación

temporal, que aprovecha a la lógica temporal para seleccionar las acciones más apropiadas.

- d) *Planeación en ambientes colaborativos*, en este tipo de problemas se considera el plan interno de un agente y el plan global que regula las interacciones entre los agentes. De los ambientes la información disponible puede ser imprecisa o desconocida (ambiente dinámico). Sobre este tema se encuentran en investigación varias técnicas.

La planeación la realiza un agente o varios agentes, el plan resultante se ejecuta por el mismo agente que lo obtuvo o por otro u otros agentes. El estado final consta de uno o varios propósitos. Dado esto, proponemos un método de planeación en donde existen varios propósitos para un mismo agente y las acciones resultantes las realizará el mismo agente en paralelo cuando existe compatibilidad entre ellas.

En el capítulo 4 se desarrolla una estrategia de planeación para agentes que tienen varios propósitos, los cuales intentan alcanzar en paralelo. Para establecer el plan del agente se consideran sus recursos y los que puede obtener del ambiente, por lo tanto, dos agentes que intentan alcanzar un mismo propósito pueden obtener planes diferentes debido a que cuentan con recursos y se encuentran en situaciones diferentes. Dado que en el ambiente pueden ocurrir eventos inesperados resulta conveniente dotar a los agentes de alguna forma de reaccionar ante estos (similar a la planeación reactiva), en este documento el problema se plantea en la sección 1.5 y se desarrolla en el capítulo 5.

1.5 EVENTOS INESPERADOS

Una vez que los agentes tienen su plan y se encuentran realizando las acciones que conducen a alcanzar sus propósitos es importante considerar la ocurrencia de eventos inesperados que pueden afectar la realización lineal de los mismos.

La ocurrencia de estos eventos los llamamos *inesperados porque no es posible prever con precisión su ocurrencia*, existe un número infinito de estos, pero en el caso de una implantación computacional se hace necesario acotarlos para tenerlos en la memoria de la computadora (RAM o disco duro). Los eventos que percibe se indican para cada agente y además un conjunto de papeles con los que puede reaccionar buscando la reacción en base a un árbol de eventos inesperados.

Araya [Araya 1989] usa un conjunto de planes de contingencia para soportar las actividades en una oficina y coordinarlas cuando surgen excepciones y sorpresas. La aplicación de estos planes es apropiada en las actividades de oficinas dado que son conocidas e involucran a una gran cantidad de agentes en los cuales existe comunicación y coordinación. Sobre todo cuando una actividad involucra un periodo de tiempo largo y tiene la posibilidad de perder la secuencia del progreso de las actividades.

Los planes en este contexto son útiles dadas las razones de que: primero, pueden modelar los aspectos orientados a metas, los métodos utilizados para alcanzar los objetivos y los aspectos temporales; segundo, un plan se puede generar combinando operadores de bajo nivel, logrando una representación que puede modificarse cuando ocurren cambios en la actividades que lo requieren; tercero, los planes permiten la reusabilidad del conocimiento entre actividades dado que se componen de entidades de bajo nivel que pueden compartirse por muchos planes, sin embargo, dado

que, las actividades de las oficinas están compuestas por eventos inesperados que requieren acciones remediales pueden afectarse por sorpresas y oportunidades.

El manejo de contingencias se realiza asumiendo que el plan ya está generado y que solamente se debe instanciar mediante ajustes a un plan generado. Los pasos para superar contingencias son:

- a) *Detectar* la contingencia.
- b) *Determinar* el conjunto de posibles acciones correctivas.
- c) *Verificar*, seleccionar e instalar una acción correctiva.

Una contingencia se detecta por una falla que ocurre durante el desarrollo de un plan ordinario. Las contingencias se detectan en tres puntos diferentes: cuando se verifican por las precondiciones de un operador, cuando se obtiene un error de la ejecución de un operador o cuando se encuentra que el objetivo que se pretende alcanzar ya está alcanzado. Para enfrentar las contingencias, se adicionan a los operadores del plan información sobre las posibles fallas y los operadores alternativos para esto.

Una limitante del trabajo de Araya, es que en un momento dado, es posible que un agente no conozca con precisión, los papeles alternativos que existen en un ambiente o se tiene pensado procesar agentes en forma independiente de las interacciones. En este caso las alternativas de acción se encuentran atadas a los operadores ante determinados imprevistos y no se cuenta con la capacidad de utilizar las mismas ante la ocurrencia de otros.

Por ejemplo, si se tiene un operador “enviar un artículo a un revisor” y el revisor pasado algún tiempo, no entrega su trabajo, puede tener en su operador dos posibles contingencias “el revisor no recibió artículo” o bien “el revisor olvidó el artículo” entonces el sistema aplica alguna de las acciones correctivas como por ejemplo “recordar al revisor por teléfono” o bien “recordar al revisor por email” pero si la causa es que el revisor “no quiere participar” se debe entonces abandonar el operador y buscar otra alternativa; en este caso se procura encontrar otro revisor.

Antes de aplicar los operadores, se verifica contra los requisitos, la factibilidad del nuevo operador, por ejemplo en caso que se vaya a enviar un email al revisor se debe verificar si se cuenta con su dirección electrónica o con su dirección postal en caso que sea por correo ordinario. Los ajustes al plan se propagan hacia adelante. Otros investigadores como Wilkins han trabajado también en la recuperación de la ejecución en sistemas de planeación para mantener la consistencia del plan. Hammond propone la reparación del plan, mediante el uso de elementos de reparación que están indexados por la falla en el mismo, por lo tanto, las reparaciones son independientes del dominio.

En los modelos computacionales se estudian a los sistemas interactivos [Wegner 1995] a diferencia de los sistemas que no lo son y donde se encuentra que en los primeros se tienen comportamientos complejos que resultan en ocasiones, imposibles de generar mediante modelos no interactivos. En este caso sus acciones están dirigidas no tanto por un control interno sino en base a las percepciones que recibe del ambiente en que se encuentra y la elección en cada momento se hace dependiendo de las acciones que puede realizar.

Un evento inesperado es una situación que se sabe puede ocurrir pero no se sabe con certeza si lo hará y cuando; además que no se tiene un código específico predeterminado para atender el suceso, sino un conjunto de alternativas posibles. A diferencia de las excepciones en donde se cuenta con un código específico y único para atenderlas, como en el caso de divisiones entre cero. Las alternativas propuestas se desarrollan en el capítulo 5 en este documento.

1.6 CONCLUSIONES DE TRABAJOS PREVIOS

En este capítulo se presenta la problemática involucrada en las transacciones del comercio electrónico debido al aumento en su volumen y la necesidad de contar con herramientas para facilitar estos procesos en forma automatizada, en particular se propone el uso de agentes aprovechando sus capacidades de autonomía dirigidas a alcanzar un propósito.

La implantación de sistemas de agentes se han intentado mediante el desarrollo de distintos lenguajes para definir agentes: METATEM, GOLOG, CONGOLOG y 3APL entre otros y lenguajes para comunicación y colaboración entre agentes: Kif, KQM y organización de conceptos como Ontolingua, Observer, la de Cyc, entre otras. En los lenguajes de definición de agentes se enfatiza la secuencia de acciones que conduce a un agente a alcanzar sus propósitos. En particular en CONGOLOG se proponen secuencias de acciones en forma reactiva para afrontar las situaciones cambiantes que se presentan en el ambiente, pero no se consideran los recursos del agente y el ambiente. Proponemos una variante a los enfoques propuestos en el sentido de dotar a los agentes de la capacidad de establecer un plan a largo plazo (mediante un proceso deliberativo: planeación) obteniendo un conjunto de papeles que un agente debe realizar para alcanzar sus propósitos. Además nuestros agentes tendrán recursos, usarán ontologías mixtas y contarán con acciones para enfrentar eventos inesperados.