

# CAPÍTULO 3

## MAPEO DE UN CONCEPTO AL MÁS SIMILAR EN OTRA ONTOLOGÍA

En este capítulo se plantea el problema de la comunicación entre dos agentes que tienen internamente su propia organización de conceptos y externamente expresan sus mensajes utilizando un lenguaje común entre ellos. Para lograr la comprensión entre ambos ellos desarrollamos un algoritmo donde se intercambian palabras que en algunos casos son ambiguas porque se refieren a conceptos diferentes, pero utilizadas en pares (refiriéndose al concepto y a su padre respecto a una taxonomía jerárquica) permiten reducir la ambigüedad.

La comunicación de mensajes entre dos entidades (como dos personas, sistemas de información o agentes), expresados mediante un código (lenguaje) que ambas partes comprenden, involucra ocasionalmente fallas en el canal de comunicación o que los involucrados manejen diferentes lenguajes. En este trabajo se considera el caso en que los mensajes siempre llegan y que se utiliza un mismo lenguaje. Entonces el problema a considerar es relativo al entendimiento entre las entidades. Como se mencionó, ocasionado porque dos agentes manejan diferentes conceptos en sus representaciones internas y los tienen organizados en formas diferentes (porque cada uno puede tener una especialidad diferente: médicos, artistas, agricultores, químicos, abogados, etc.)

En los sistemas informáticos la representación del conocimiento es una manera de codificar la realidad en la cual se encuentra inmerso el mismo. Estos estudios se han hecho para obtener modelos del mundo real para manipularse y obtener resultados aprovechables en el contexto donde proviene la información. Existen diferentes formas de estructurar el conocimiento como se explicó en la sección 1.3, esto implica que una entidad puede estar especializada en el manejo de información visual mientras que otra puede serlo en base de datos. En cuanto al contenido, dependerá de los temas que resulten de interés para cada una de las entidades, de esta forma para una persona que trabaje con agricultura, lo más probable es que su conocimiento esté estructurado en conceptos como tierra de hoja, temporal, lechugas, manzanas, peras, ejotes, entre otros. A la forma particular de organizar los conceptos por parte de una entidad le llamaremos *ontología*.

En las transacciones comerciales es necesario intercambiar información y conocimiento para hacer posible la comercialización de productos. Por lo tanto, es necesario que los participantes establezcan

acuerdos acerca de lo que quieren intercambiar. Para esto se toma en consideración que cada entidad tiene su propia ontología y utiliza un lenguaje común mediante el cual comunica sus intereses. Los conceptos son independientes del agente que los maneja y se consideran únicos, esto es, el concepto maíz-cereal será el mismo para todos los agentes aún cuando algunos utilicen las palabras maíz, corn, milpa u otras para referirse al mismo.

A pesar de que se utiliza un lenguaje común, los conceptos que se establecen cuando se intenta la comprensión entre agentes, no necesariamente corresponden a los conceptos a que se refieren las entidades en comunicación, esto se debe a que cada entidad representa de manera propia sus conceptos y utiliza diferentes palabras para referirse a los mismos. Si se utilizan palabras únicamente puede ocurrir que dos entidades se estén refiriendo a dos cosas distintas, por ejemplo si alguien pregunta por una manzana, una entidad puede referirse a la manzana de una colonia mientras que otra a una fruta. Por esto, diferenciamos entre palabras ordinarias (que pueden ser ambiguas al hacer referencia a conceptos) y conceptos que son únicos.

En otros términos, se intenta resolver el problema entre dos agentes que intercambian palabras en un lenguaje común, y nos interesa la comprensión que los participantes logran alcanzar (la manera en que llevan las palabras intercambiadas a su representación interna cada uno). Como ya se mencionó en la sección 1.3 cuando nos referimos a un concepto usamos palabras minúsculas subrayadas y las palabras que lo referencian en minúsculas sin subrayar. En ambos casos las palabras se expresan en singular. Los conceptos se organizan en una jerarquía, el nodo superior es el concepto más general respecto a los de nivel (la raíz del árbol es el concepto más general) (figura 3.1). Los conceptos están unidos por relaciones es-un u otra similar, por ejemplo en la figura 3.1 cosa agrupa a planta y animal donde cada uno es-un cosa, a su vez el concepto planta agrupa a planta-comestible y planta-ornato y ambos tienen la relación es-una, un nodo es una instancia de su nodo superior.

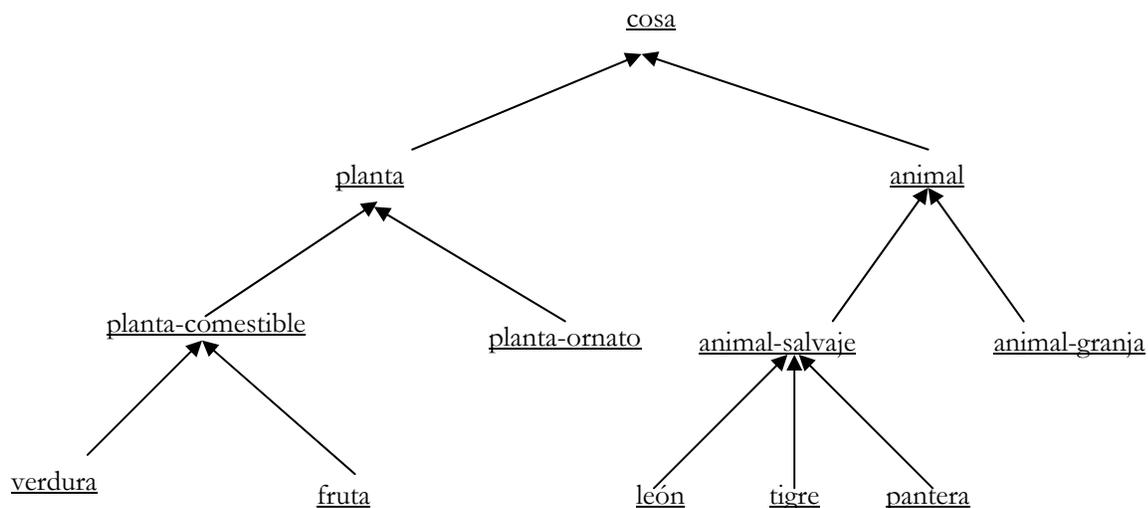


Figura 3.1 Organización de una Ontología

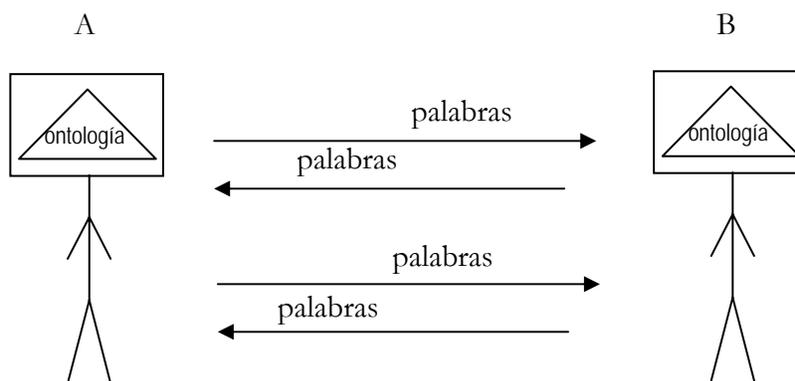
La relación entre palabras y conceptos puede estar sujeta a ambigüedad, debido a que una palabra puede mapear a muchos conceptos, muchas palabras a un concepto o una palabra mapea a un concepto. Algunos ejemplos de esto son:

- una a muchos*, la palabra perico se puede referir al concepto perico-animal o a perico-herramienta;
- muchas a uno*, las palabras melocotón y chabacano se refieren al concepto chabacano;
- una a uno*, una palabra mapea a un concepto único, por ejemplo en el caso de:  $\pi$  (la palabra  $\pi$  mapea al concepto  $\pi$ ), 5, Mar Caspio, polinomios como  $(x + 4)^2$ , posiciones geográficas como (19°N, 90°W), un dibujo, un diagrama arquitectónico.

Existe también la posibilidad que una entidad no maneje algunas palabras porque no tiene conceptos con quien asociarlas.

Por ejemplo si una persona utiliza la palabra perico se tiene que revisar el sentido de la misma porque puede referirse al concepto perico-animal, (ave verde), o al concepto perico-herramienta (especie de llave), o quizá perico-apodo (sobrenombre de una persona). De esta forma cuando un agente quiere comprar un perico (animal), al utilizar el concepto y ubicarlo en su ontología se utiliza la información del concepto y de su padre (en este caso el concepto animal) para reducir la ambigüedad. *Cuando un agente es capaz de llevar a su representación interna las palabras relacionando un concepto con las palabras que se le indican, decimos que el agente entendió el mensaje.*

Los agentes que consideramos, intercambian palabras ordinarias para comunicar sus conceptos de interés (figura 3.2).



**Figura 3.2** Intercambio de palabras (ambiguas) para encontrar los conceptos entre dos agentes

Las ontologías que consideramos se componen de nodos organizados en una jerarquía formada por relaciones *es-un*, donde el nodo de nivel superior corresponde con la clase más general y a partir de ella se van obteniendo nodos que son subconjuntos del nodo de nivel anterior en clases más específicas. De esta forma el padre de un nodo se considera un concepto más general, mientras que un nodo hijo se considera como una especialización del primero.

Cada concepto tiene una lista de propiedades expresadas como pares en donde el primero es el atributo y el segundo su valor (ambos expresados como conceptos) (atributo valor), en estas listas se aplica la recursividad, por ejemplo: (maíz (color azul) (tamaño ((valor 1) (unidad cm))) (dureza duro-macizo)).

Estas relaciones de padre a hijo se utilizan para encontrar el concepto al que se refiere un agente al utilizar palabras [Guzmán 2000] [Guzmán 2000a], por ejemplo, para dos agentes que se comunican, sus conceptos no necesariamente están organizados de la misma manera.

Considerando la forma en que las personas llevan a cabo el proceso de interacción se observa que inicialmente crean un contexto de la información de interés para ambos y comienzan con un intercambio de mucha información (palabras), luego cuando ya se entienden necesitan menos información para que quede claro lo que quieren comunicarse, finalmente, llegan a un punto en donde utilizan palabras clave y se comunican una gran cantidad de conocimiento de interés para ambas partes [Guzmán 2002], por ejemplo puede ser típico un mensaje de la forma “necesito maíz como siempre”, lo cual se puede interpretar como “necesito una tonelada de maíz en forma de elotes entregado en la ciudad de Mérida a finales de octubre y el pago se realizará en efectivo a contraentrega”.

Al considerar el caso del entendimiento entre dos agentes que intercambian palabras para intentar mapear los conceptos que tienen en su representación interna existen diferentes alternativas desarrolladas hasta ahora como se comentó en la sección 1.3, una de ellas consiste en que utilicen la misma ontología y se intercambien los mismos conceptos, lo cual implica que los agentes que participan en las interacciones comparten la misma representación interna y utilizan elementos de la misma (como apuntadores a nodos). Otra alternativa es crear una ontología de nivel superior o utilizar alguna existente donde se mapean hacia la superior los conceptos de las ontologías de nivel inferior. En general, no se puede esperar que dos agentes compartan una misma ontología, principalmente, al considerar que existen muchas empresas que realizan comercio con otras tantas y solamente para algunas transacciones, por lo que el costo de hacer coincidir sus ontologías implica un costo mayor al esperado para realizar operaciones de comercio electrónico. Por esto se requiere de alguna herramienta con la que realice el mapeo de conceptos entre dos ontologías respetando la estructuración individual.

Cada agente en el modelo presentado en el capítulo 2 tiene una representación interna (llamada ontología) y un conjunto de palabras que representan a cada concepto dentro de la ontología, a partir de esto se hace el mapeo de los conceptos sin necesidad de integrar los conceptos que forman las dos ontologías. La ventaja de esto es que, solamente se mapean entre dos y diez conceptos y no 30 mil o 50 mil que contiene una ontología; tampoco se necesita integrar la ontología con otras 4 o 5 ontologías. Lo que estamos proponiendo es que si se tiene una ontología compleja y únicamente interesa que un agente se ponga de acuerdo sobre algunos conceptos de interés para llevar a cabo una transacción comercial, únicamente se haga el mapeo de estos conceptos aprovechando que los agentes utilizan palabras de un lenguaje común y que cada concepto es único dentro del sistema de agentes. Esto está apoyado en el hecho de que un agente no le puede copiar su representación interna a otro agente porque si esto fuera posible el problema ya estaría resuelto, de la misma manera que un ser humano no le puede copiar parte de su cerebro a otro para darse a entender, situación similar en muchos sistemas informáticos, donde tampoco es posible transferirle el conocimiento a otro programa porque cada uno maneja sus estructuras de datos propias.

Muchas veces la razón de utilizar diferentes estructuras es porque cada programa fue hecho por un grupo de desarrollo diferente, cada uno con una filosofía, idiosincrasia o intención diferente y difícilmente estos se pueden pegar de una forma simple. Dado entonces que en el mundo abunda una heterogeneidad de desarrolladores es que se justifica el que tratamos de aprovechar el conocimiento que ya esta codificado de alguna forma en ontologías existentes.

El algoritmo que presentamos en 3.1 y del cual damos ejemplos en 3.2, tuvo como antecedente otro algoritmo en el cual se ubica el concepto candidato enviando sus palabras al segundo agente, ante lo cual, este busca los conceptos con los cuales asocia estas palabras, (en este caso únicamente se enviaba las palabras del concepto de interés, en el nuevo algoritmo también se incluyen las palabras del padre del concepto). Lo que encontramos es que varios conceptos mapeaban a las palabras, por lo que en caso de duda el agente enviaba las palabras correspondientes a los padres de cada uno, mismos que a su vez al llegar al primer agente generaba un mapeo múltiple que requería mucho más interacciones para determinar cual concepto es el referido. Sobre la base de este proceso de confirmar y refutar nos dimos cuenta que se podría simplificar enviando desde la primera vez tanto las palabras que corresponden con el concepto de interés como las de su padre, por lo tanto es esta la versión que se expondrá en 3.1.

El comparador de ontologías mixtas (COM) es un modulo en el cual se considera la estructuración de una onde una ontología con su diccionario donde se asocian las palabras con los conceptos. Estos elementos los utiliza el algoritmo `hallar`, el cual toma un concepto de una “ontología A” busca sus palabras asociadas, toma al padre del mismo concepto y sus palabras asociadas y con estos conjuntos de palabras busca el conceptos más similar en una “ontología B”.

Un fragmento de un diccionario de palabras-concepto se muestra a continuación, en el mismo se observa que el concepto `cosa` tiene dos palabras asociadas, la palabra `maíz` se refiere a los conceptos `maiz_planta` y `maiz_semilla`.

Palabra	Concepto
cosa	cosa
algo	cosa
fantasma	fantasma
espanto	fantasma
animada	animada
viva	animada
animal	animal
vegetal	vegetal
planta	vegetal
fruta	fruta
comida	fruta
fruta_caliente	fruta_tropical
fruta_templada	fruta_templada
fruta_fria	fruta_fria
vegetal	verdura
verdura	verdura
mamey	mamey
mango	mango
manzana	manzana
durazno	durazno
melocoton	durazno
pera	pera
tejocote	tejocote
maiz	maiz_planta
maiz	maiz_semilla
frijol	frijol
trigo	trigo
cereal	cereal
semilla	cereal

### 3.1 ALGORITMO HALLAR

Como se mencionó en la sección anterior, el algoritmo *hallar* es el resultado de varios experimentos realizados para encontrar el concepto más similar de una ontología A en otra B. En la última versión en este algoritmo se consideran cuatro casos con base a los elementos que se involucran en la búsqueda: el concepto y su padre (o algún ancestro significativo); se considera posible construir algoritmos más complejos en donde se intercambie una sección mayor de una ontología como por ejemplo el concepto sus hijos y los ancestros, esto es motivo de una línea de investigación diferente a la que seguimos. A continuación explicamos la solución que hemos encontrado:

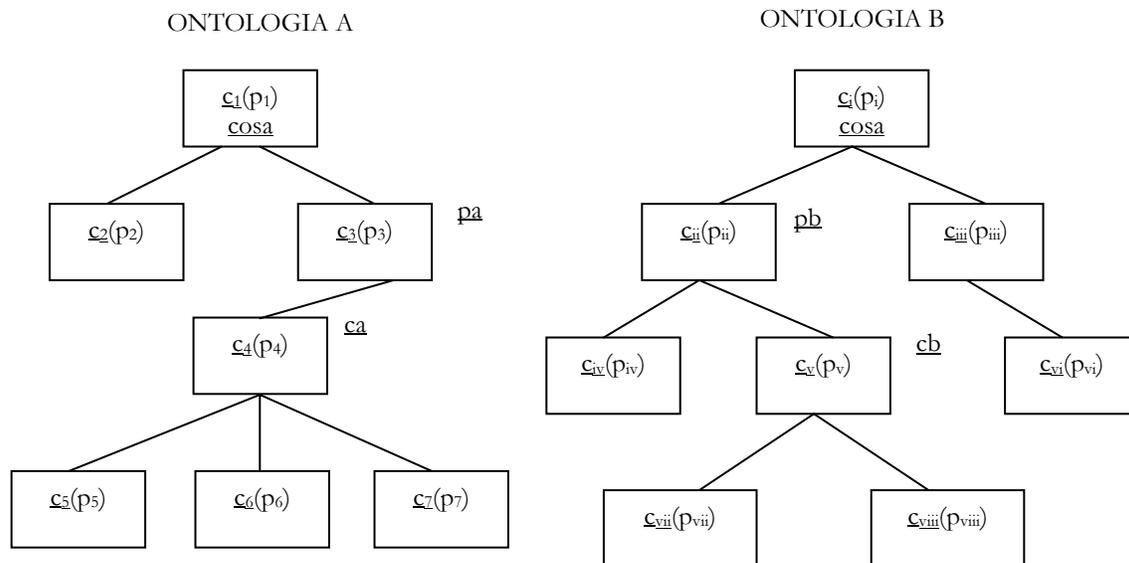
Considerando el concepto y su padre (ancestro) se tienen cuatro casos:

- Cuando *coinciden* las palabras de los *conceptos* y las respectivas de su *ancestro*.
- Cuando *coinciden* las palabras de los *conceptos* pero *no las de su ancestro*.
- Cuando *coinciden* las palabras de su *ancestro* pero *no del concepto*.
- Cuando no coinciden *ni las palabras de su ancestro ni las del concepto*.

Sean A y B dos ontologías, para encontrar un concepto  $\underline{cb}$  en B lo más cercano a otro  $\underline{ca}$  en A se utiliza el algoritmo *hallar* en donde se indica un concepto de A que se quiere encontrar en B mediante el envío de las palabras que describen al concepto de A, el concepto más similar de B se devuelve o se indica que no se encontró, el resultado es único, por lo que el algoritmo elige el mejor concepto. El algoritmo usa dos ontologías distintas A y B (figura 3.3). La invocación al algoritmo *hallar* es:

$$\underline{cb} = \text{hallar} (\underline{ca}, A, B)$$

que significa: hallar el concepto  $\underline{cb}$  en la ontología B más cercano al concepto  $\underline{ca}$  de la ontología A.



**Figura 3.3** Dos ontologías diferentes. Cada concepto  $c_x$  se referencia por palabras comunes a ambas ontologías. Se intenta encontrar dos conceptos padre-hijo (pb-cb) en la ontología B que correspondan con el par (pa-ca)

En forma abreviada el llamado a `hallar` lo escribimos como:

$$\underline{cb} = \text{hallar}(\underline{ca})$$

El algoritmo `hallar` utiliza las palabras que se relacionan con cada concepto, es decir no los conceptos directamente. En principio envía las palabras correspondientes al concepto de interés y de su padre. Luego en la ontología B se buscan conceptos que tengan alguna relación ancestro-nodo para encontrar al posible padre y concepto equivalente. Si la información es insuficiente se utiliza recursivamente el algoritmo para encontrar algunos ancestros apropiados. El propósito de `hallar` es encontrar en B los conceptos `cb` y `pb` que corresponden a los conceptos `ca` y `pa` de A, o cuando menos, `hallar cb`.

El algoritmo `hallar` consta de los pasos siguientes:

- a) En B se buscan dos nodos (conceptos) `pb` y `cb`, de manera que las palabras de `pb` coinciden con la mayoría de las palabras (recibidas) de `pa` y las palabras de `cb` coinciden con la mayoría de las palabras (recibidas) de `ca`; `pb` es el padre, abuelo o bisabuelo de `cb`. Con esto se considera el concepto `cb` encontrado como el más cercano a `ca` y por lo tanto la respuesta es `cb` y termina el algoritmo (figura 3.4).

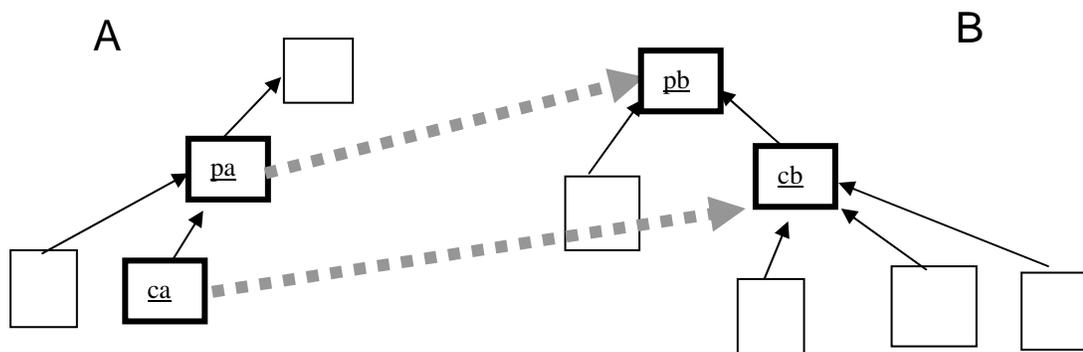
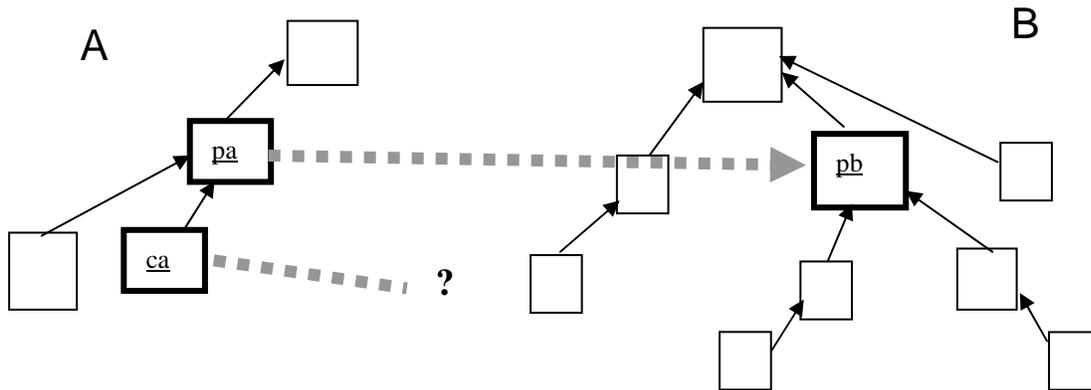


Figura 3.4 Caso a) (`ca`, `pa`) en A hallan un par (`cb`, `pb`) en B que son hijo y ancestro

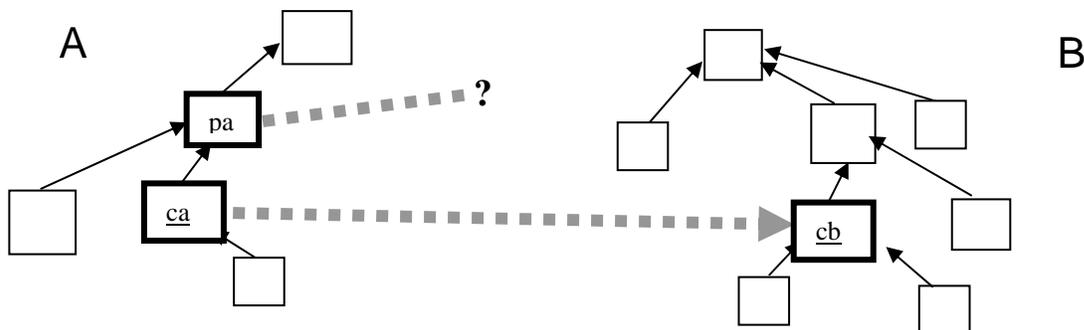
- b) Si se encuentra `pb` pero no `cb`. En este caso, se llama recursivamente al proceso `hallar` de la forma `pb' = hallar(pa)` para confirmar que `pb` es ancestro del concepto de interés (`ca`). Si el `pb'` hallado es `cosa`, es decir, el concepto de más alto nivel, el algoritmo concluye sin éxito; si no, se busca en B para cada hijo de `pb`, aquél, cuya mayoría de propiedades y valores, coincidan (recursivamente vía `hallar`) con las correspondientes de `ca`. Es decir, se busca en B al hijo de `pb` que tenga la mayor cantidad de las propiedades (y valores correspondientes) del concepto `ca`. Se rechazan hijos de `pb` que coinciden sólo en unas cuantas propiedades y en unos cuantos valores con las de `ca`. Si el candidato `cb'` analizado tiene además hijos, se ve si éstos coinciden (usando recursivamente `hallar`) razonablemente con los hijos de `ca`. Si se encuentra un `cb'` con las propiedades deseadas, el algoritmo concluye con éxito indicando el concepto buscado. En otro caso, se intenta hallar el `cb'` entre los hijos del padre (en B) de `pb`, es decir, entre los hermanos de `pb`; en caso necesario, entre los hijos de los hijos de `pb`, o sea, entre los nietos de `pb`. Sino se encontró un `cb'`, entonces el más cercano a `ca` es `pb`, por lo que `hallar` devuelve la anotación

hijo\_de\_pb y el algoritmo concluye. Por ejemplo, A puede enviar a B el par (kiwi, fruta)<sup>15</sup>, B puede tener el concepto fruta pero no el concepto kiwi ni alguna fruta que se le parezca, en este caso, kiwi (la palabra kiwi, enviada por A y recibida por B) se traduce por B a hijo\_de\_fruta, o “alguna fruta que no conozco, que no tengo en mi ontología”, lo más cercano en B a kiwi en A (figura 3.5).



**Figura 3.5** Caso b) Se halla en B al padre pb que corresponde a pa en A, pero no se halla concepto cb en B que corresponde a ca en A

- c) Si se halla cb pero no pb, se busca si el abuelo en B de cb es similar a pa, o si el bisabuelo de cb en B es similar a pa (esto ya fue comentado en el caso b). Si se halla, entonces el concepto en B más similar a pa es el abuelo o el bisabuelo de cb y concluye el algoritmo. Si no se halla, se verifica si la mayoría de las propiedades (y sus valores correspondientes) de ca coinciden con las de cb y si la mayoría de los hijos de ca coinciden (usando `hallar`) con la mayoría de los hijos de cb; si las propiedades y los hijos coinciden, entonces la respuesta es cb y concluye el algoritmo aunque no se haya hallado en B al pb que corresponda al concepto pa en A. Si solamente una parte de propiedades e hijos son similares entonces la respuesta es `probably_cb` y concluye el algoritmo. Si ninguna propiedad ni hijos son similares la respuesta es `no_hay` y concluye el algoritmo (figura 3.6).



**Figura 3.6** Caso c) se encuentra el concepto cb en B similar a ca en A, pero no se encuentra pb en B similar a pa

<sup>15</sup> Son dos palabras, no dos conceptos y un agente no puede enviar conceptos a otro agente (representación interna)

d) Si no se encuentra cb ni pb, entonces la respuesta es no\_hay y concluye el algoritmo (figura 3.7).

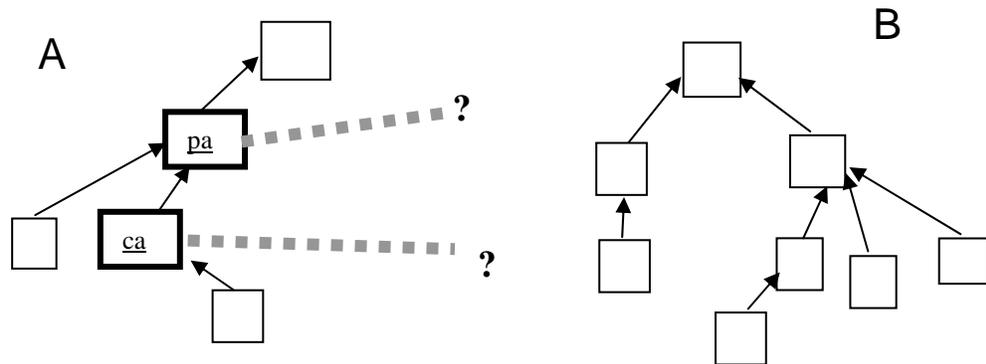


Figura 3.7 Caso d) no se encuentra ni cb, ni pb en B similares a ca y pa en A

### 3.2 EJEMPLOS

Una ontología generalmente consiste de cientos o miles de conceptos, pero como la intención aquí es ilustrar el funcionamiento se usan secciones de ontologías donde únicamente se tienen los conceptos de interés tomando en cuenta esto.

En el caso a), donde se encuentra el mapeo (usando palabras ambiguas) entre el concepto y su padre de una ontología A con el concepto y su padre de una ontología B (figura 3.8).

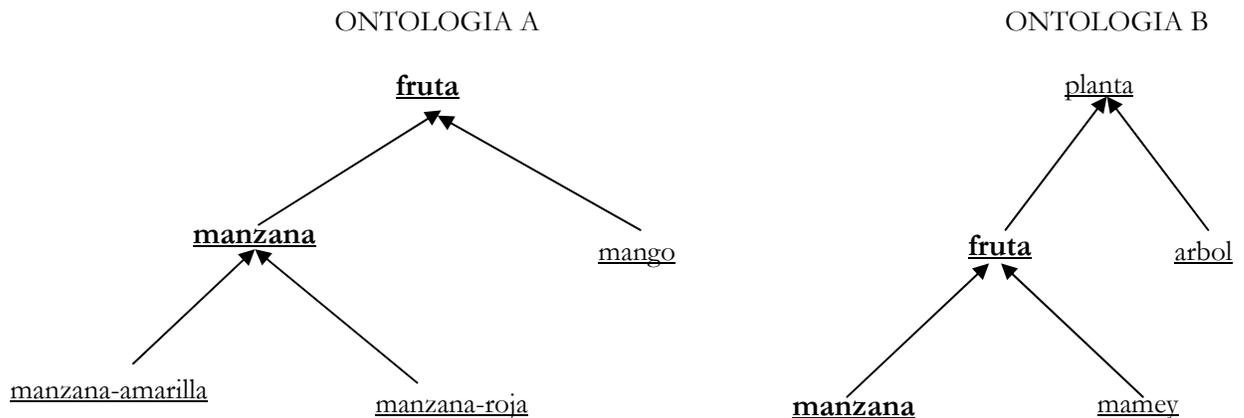


Figura 3.8 Ejemplo del caso a) donde se encuentra el mapeo entre los conceptos y sus padres

En el comparador de ontologías implementado (figura 3.9) se le indican las ontologías que se utilizan mediante los botones Load Ontology A y Load Ontology B presionando el botón Ontology Match se activa el algoritmo hallar implementado. En el campo Result aparece el resultado del mapeo entre las ontologías en cuestión.

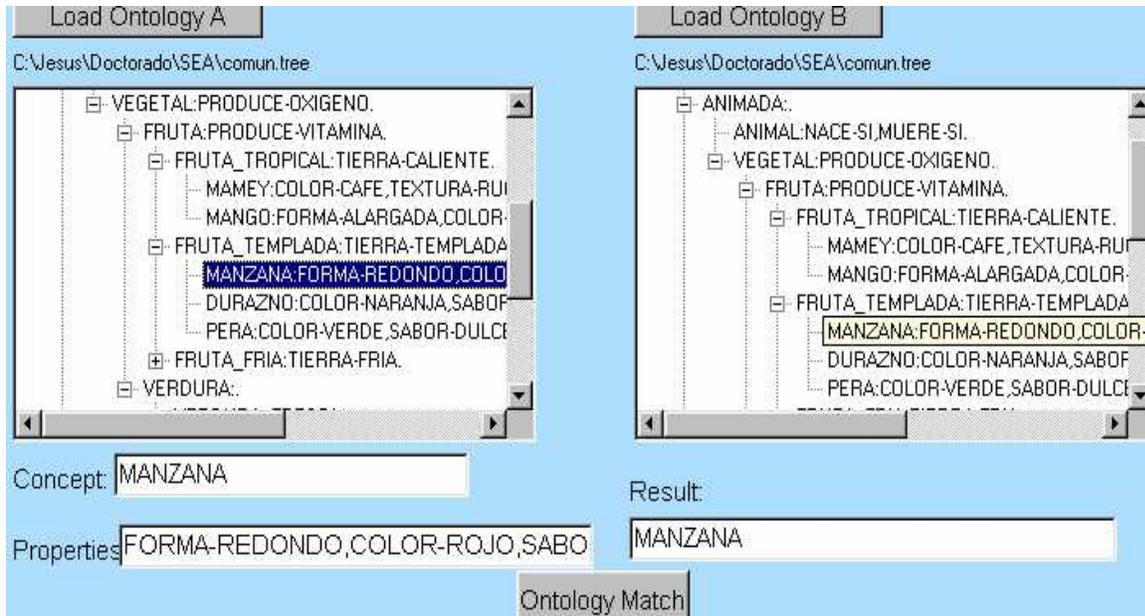


Figura 3.9 Ejecución del COM para el caso a) donde se encuentra la equivalencia de ca y pa con el respectivo cb y pb

El caso b) se encuentra un concepto pero no el padre del mismo por lo que se aplica recursivamente el algoritmo hallar para encontrar otro ancestro equivalente en la ontología del agente A, la figura 3.10 se muestra la sección de las ontologías en donde se presenta esta situación.

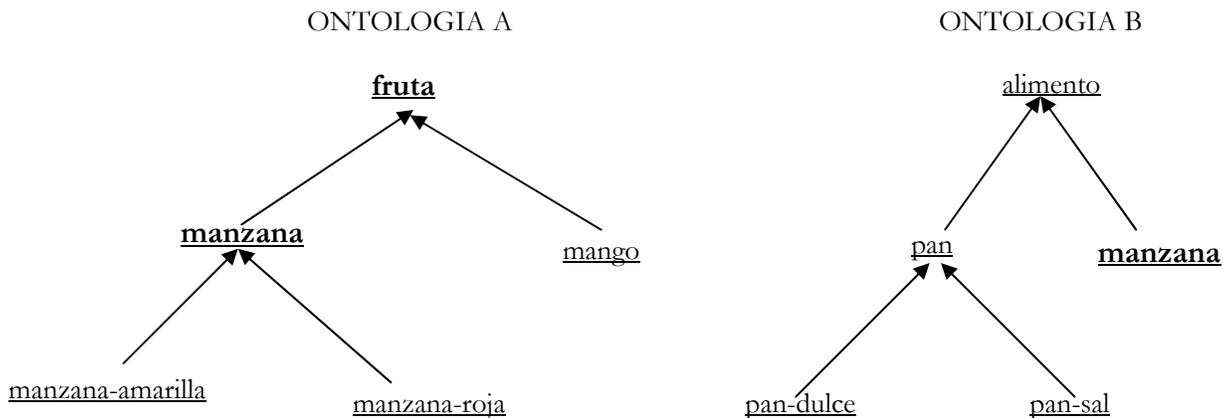


Figura 3.10 Caso b) se encuentra el mapeo entre los conceptos ca de A con cb de B pero no de sus ancestros

En la figura 3.11 se muestra la pantalla del COM en donde se presenta el caso b).

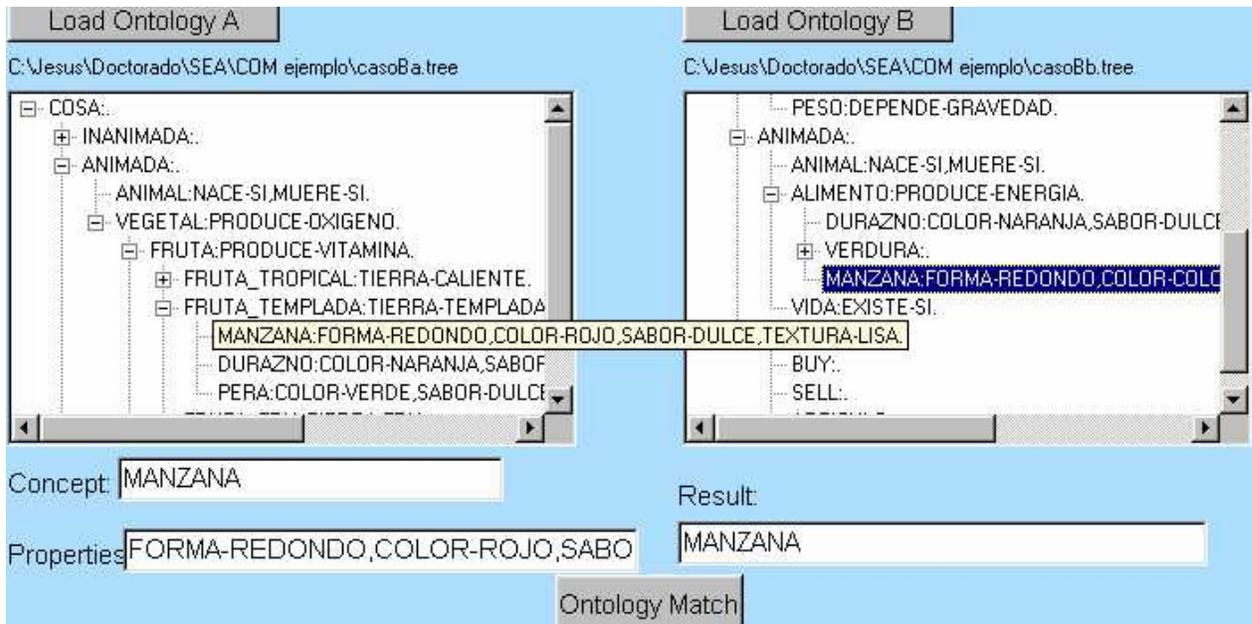


Figura 3.11 Ejecución del COM para el caso b) se encuentra la equivalencia de ca y cb pero no de sus ancestros

Un ejemplo para el caso c) (figura 3.12) en donde se encuentra el mapeo entre los padres del concepto pero no el concepto de interés. En este caso se utilizan las propiedades de los conceptos y sus hijos.

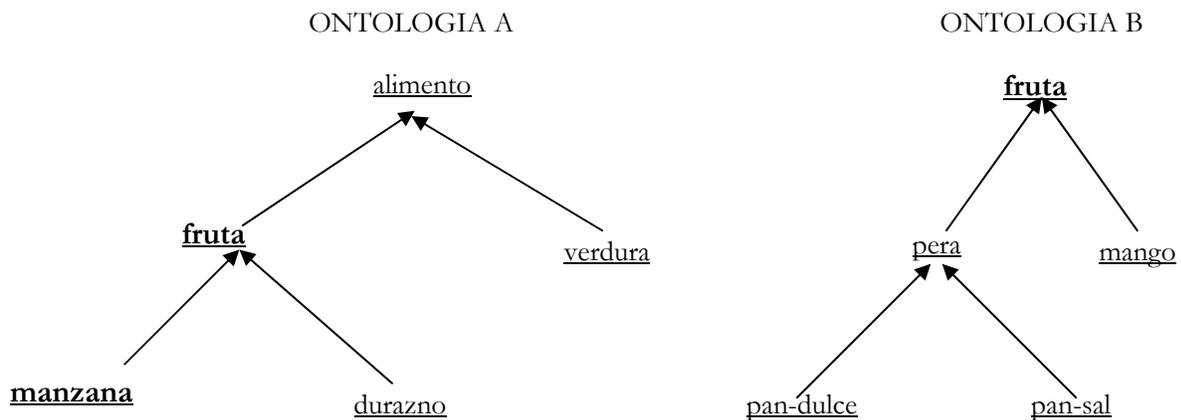


Figura 3.12 Caso c) en este caso se encuentra el mapeo entre los padres pa y pb pero no entre los conceptos

La pantalla del COM donde se muestra un caso c) se muestra en la figura 3.13.

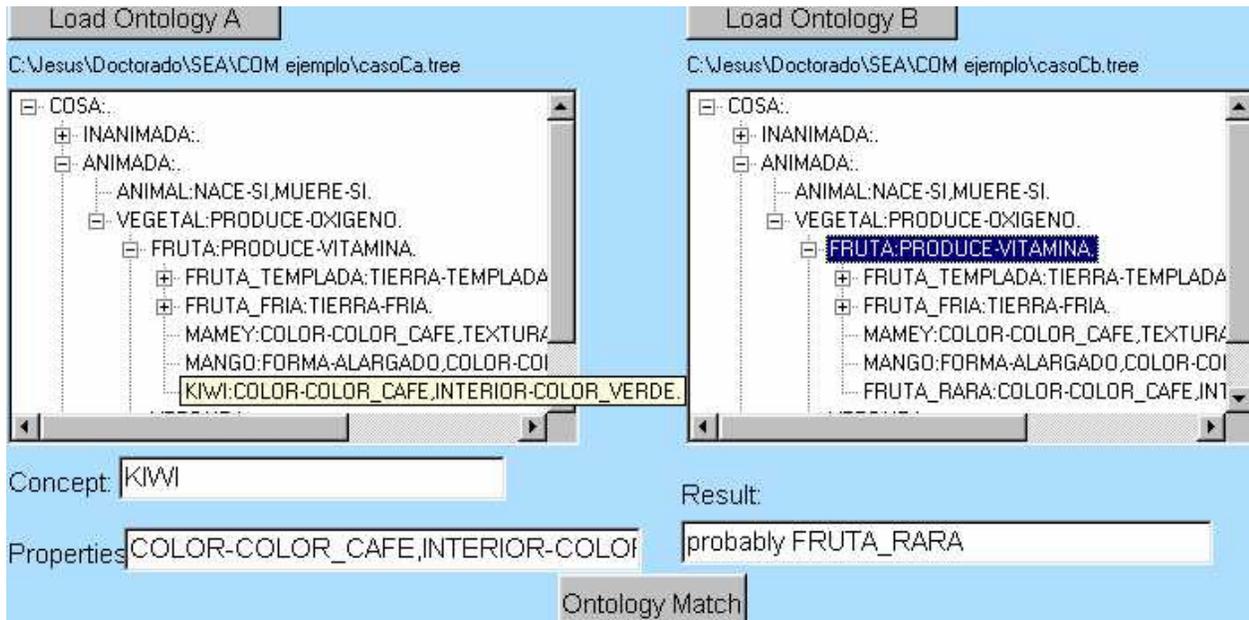


Figura 3.13 Ejecución del COM para el caso c) se encuentra la equivalencia de  $pa$  y  $pb$  pero no de los conceptos  $ca$  y  $cb$

El ejemplo para el caso d) es cuando las ontologías son diferentes y no contienen conceptos comunes, esto es típico cuando los agentes tienen diferentes especialidades, como en el caso de un agricultor y un informático (figura 3.14).

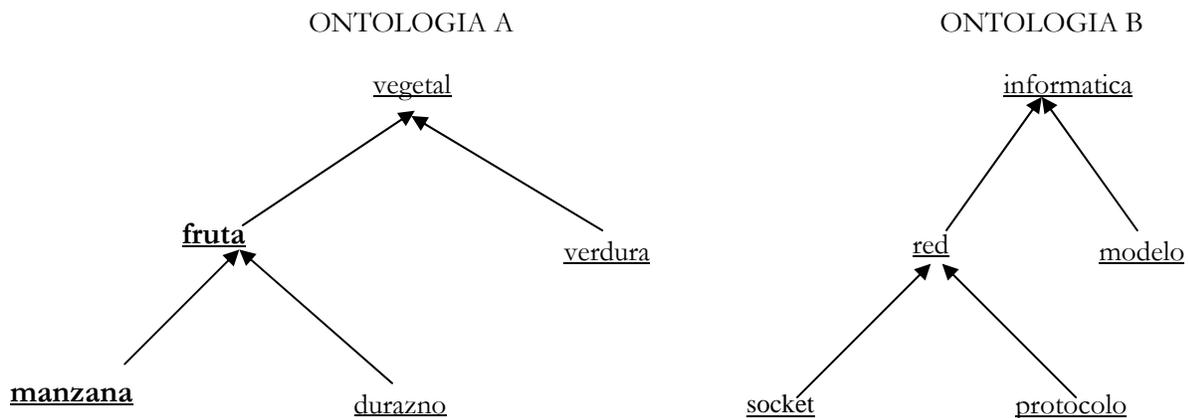


Figura 3.14 Caso d) en donde dos ontologías son diferentes y no existen conceptos en común

La ejecución del COM para el caso d) se ilustra en la figura 3.15.

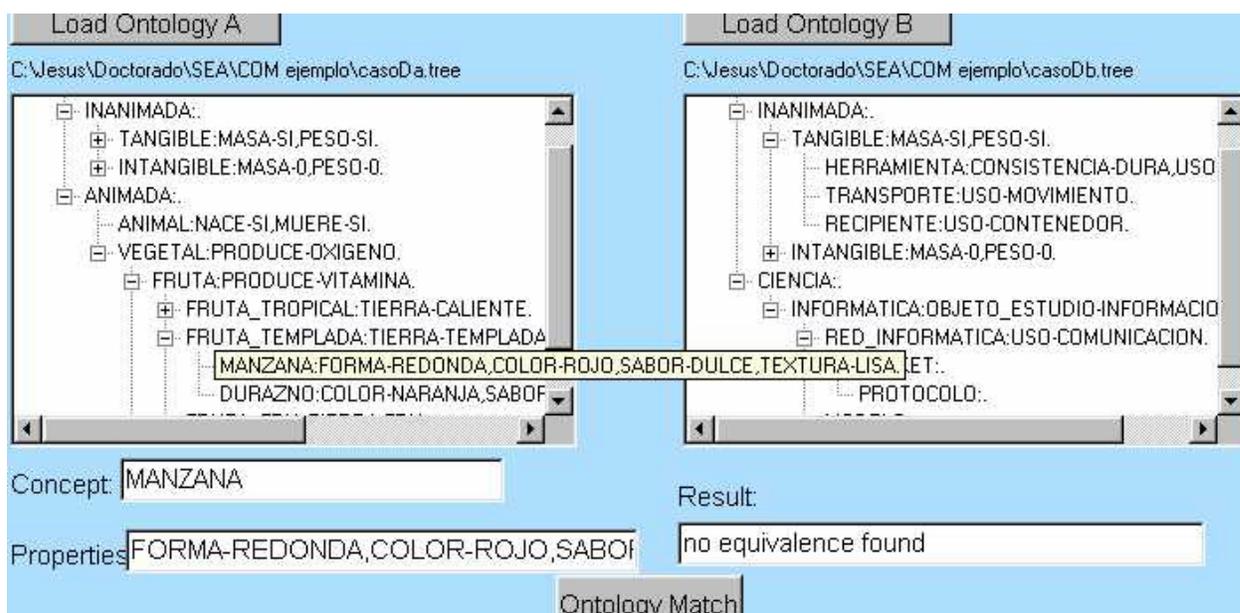


Figura 3.15 Ejecución del COM para el caso d) no existen conceptos comunes

### 3.3 CONCLUSIONES DE MAPEO DE CONCEPTOS

Se propuso en este capítulo una forma de establecer un mapeo de conceptos entre ontologías diferentes, dos a la vez. Específicamente hemos presentado un algoritmo constructivo que halla, para un concepto *ca* en una ontología A, el concepto *cb* más similar de otra ontología B (organizadas ambas en base a una estructura jerárquica) utilizando las palabras que son de un lenguaje común entre agentes pero que el mapeo de palabras a conceptos es ambiguo debido a que una palabra mapea a varios conceptos y que un concepto se puede referir con diferentes palabras. El algoritmo hallar, responde con las palabras del concepto encontrado o con *no\_hay*, *duda\_cb*, *hi\_jo\_de\_pb*.

Nuestra propuesta es una alternativa a los trabajos que se están realizando sobre manejo de ontologías, con la ventaja de que se toman dos ontologías intactas y se trabaja únicamente con los conceptos de interés. Al manejar un mínimo de información es posible entonces aplicar el comparador de ontologías en sistemas de comercio electrónico, bases de datos, base de conocimientos donde muchas veces solamente se requiere realizar pocas transacciones y no se quiere incurrir en el costo que implica generar una ontología común para dos entidades (como empresas). Nuestra propuesta también es útil en la interacción entre agentes ligeros como los móviles o los que se encuentran embebidos en smart cards («tarjetas inteligentes») u otros sistemas distribuidos.

El algoritmo aquí propuesto no usa de una tercera ontología “común” (tipo CYC) para resolver ambigüedades. Empero, el uso de estos recursos de conocimiento (ontologías comunes, tesauros, diccionarios, etc.) podría producir algoritmos más eficientes lo cual, es motivo de otra línea de investigación.

La propuesta que se hace en este capítulo es importante por la cantidad de empresas en el mundo que día con día requieren realizar transacciones entre ellas y necesitan establecer con precisión los productos y condiciones a intercambiar. La forma en que se integra esta propuesta a ontologías existentes es mediante la adición de un diccionario de palabras de un lenguaje común hacia los conceptos descritos en la ontología, con esto establece una independencia entre los conceptos y el lenguaje de comunicación (por ejemplo podrían utilizarse diferentes lenguajes para interactuar con diferentes agentes).