

Plataforma para el uso de Mapas Conceptuales para la Representación de Conocimiento

Perla Señas – Norma Moroni
Departamento de Ciencias de la Computación
Universidad Nacional Del Sur - Argentina
psenas@cs.uns.edu.ar - moroni@criba.edu.ar

Resumen

La Informática Educativa debería trascender la mera simplificación de las tareas en los procesos de enseñanza-aprendizaje, para consolidarse como herramienta efectiva en la construcción del conocimiento. Es de interés la creación de ambientes computacionales especiales dentro de los cuales los estudiantes puedan realizar tareas de aprendizaje y con facilidades para tal fin. Actualmente han cobrado especial interés aquellos ambientes cuya arquitectura responde al modelo de una sociedad de agentes que trabajan en forma cooperativa. Entre ellos, se destacan aquellos entornos que permiten visualizar las representaciones de conocimiento proporcionando experiencias de aprendizaje muy ricas para los alumnos. La selección de una representación capaz de adecuarse a los requerimientos de los aprendices humanos y de los agentes de software es uno de los puntos claves para el diseño de los sistemas de aprendizaje mixtos. En este trabajo se presenta una extensión del ambiente MCH existente, orientado al trabajo sobre la representación de conocimiento.

Palabras claves

Informática Educativa - Ambientes de Aprendizaje - Mapas Conceptuales Hipermediales - Representación de Conocimiento

1. Introducción

En el ámbito de las investigaciones de Informática Educativa existe consenso en considerar que un uso racional que saque partido de las potencialidades de la tecnología computacional puede modificar sustancialmente la manera de enseñar y de aprender. La tendencia es usar esta tecnología para trascender la mera simplificación de las tareas en los procesos de enseñanza-aprendizaje, convirtiéndose en herramienta efectiva para la construcción del conocimiento.

El sentido de la incorporación de la computadora a los procesos de aprendizaje debe identificarse con el desarrollo de las actividades de pensamiento. Para ello es de interés la creación de ambientes especiales dentro de los cuales el educando pueda realizar tareas de aprendizaje. En el área del software educativo los entornos de aprendizaje inteligentes trascienden la aproximación de entrenamiento y práctica, pues se los diseña para asumir la responsabilidad de enseñar empleando un diálogo de iniciativa mixta. En este tipo de entornos, el software y los alumnos que integran el sistema de aprendizaje son tratados como procesadores de conocimiento. Existen dos enfoques bien diferenciados para abordar su diseño: el tradicional y el de agentes. En el primero la arquitectura se presenta en forma modular (base de conocimiento, interface, modelo de alumno, módulo de estrategias de enseñanza, módulo de control y módulo para resolución de problemas) y en el segundo la arquitectura responde al modelo de una sociedad de agentes que trabajan en forma cooperativa.

En este trabajo se centra la atención en los agentes pedagógicos, es decir en aquellos agentes autónomos que apoyan el aprendizaje humano integrando junto con estudiantes, ambientes de aprendizaje interactivos. Ellos son capaces de un espectro amplio de interacciones instruccionales

efectivas con los educandos que participan en el entorno de aprendizaje, incluyendo diálogo multimodal. Pueden colaborar con estudiantes y con otros agentes, integrando acción con instrucción. Esto contrasta con los típicos sistemas inteligentes de instrucción que sólo aportan desde un lado y que son capaces de interactuar con un solo estudiante a la vez. Los agentes pedagógicos son capaces de proveer retroalimentación continua a los alumnos durante su trabajo en el entorno, tienen capacidad de presentarse dando la sensación de estar vivos y de inducir en los estudiantes los mismos tipos de respuestas afectivas que generan otros tipos de entes vivos. El desarrollo de estas capacidades en una entidad artificial requiere de una representación de conocimiento y razonamiento sofisticada.

Esta propuesta se centra en el uso de los Mapas Conceptuales Hipermediales (MCH) [Señ96] y de los Mapas Conceptuales a la Sowa (MC^S) [Señ99] como esquemas de representación de conocimiento en agentes inteligentes. Los MCH se basan en los Mapas Conceptuales de Novak (MC) e incorporan la flexibilidad y riqueza que permite la tecnología hipermedial. En el área educativa, ambos esquemas han sido probados con éxito como potentes estructuras capaces de contribuir con la construcción de aprendizajes significativos en las personas. Se destaca el valor del recurso hipermedial, no sólo en el aspecto operacional sino en los planos relacionados con la percepción y la abstracción [Zan98]. Se detectan falencias al querer extender el modelo a los agentes de software. Por esa razón, se realiza una extensión de los MC incorporando elementos de los Grafos Conceptuales de Sowa. Se definen entonces, los MC^S y una arquitectura para la representación de la base de conocimiento de un Agente Inteligente apuntando al aprendizaje, es decir, a la generación y modificación de los MC^S que representan el conocimiento del mencionado agente.

Cuando se examina el límite computacional sobre el razonamiento automatizado y su efecto sobre la representación del conocimiento, se advierte que no se razona correctamente y con igual facilidad sobre los distintos lenguajes de representación. Además, generalmente el grado de dificultad aumenta en forma paralela con el poder expresivo de los lenguajes. El objetivo de este trabajo se centra en un esquema de representación de conocimiento lo suficientemente flexible para el manejo de los alumnos y lo suficientemente riguroso para poder realizar razonamiento automatizado. Se presenta una extensión a la Plataforma de desarrollo de MCH, con el propósito de incluir el trabajo con MC^S.

2. Esquema de la Base de Conocimiento de un Agente Pedagógico

Al intentar usar a los MCH como esquema de representación en ambientes de aprendizaje mixtos, es decir, que incluyen agentes pedagógicos y humanos, se observa que si bien es una representación probada con éxito entre agentes humanos, se detectan falencias al querer extender el modelo a los agentes de software. Dichas falencias se centran en los siguientes puntos:

- Jerarquización de los conceptos
- Definición de clases e individuos
- Manejo en la aridad de las relaciones

Se adopta como solución un modelo que consiste en un esquema de clases y un esquema de representación de proposiciones [Señ99]. El esquema de clases es un reticulado representado por medio de un MCH basado en el modelo de los MC de Novak. Para la representación de las proposiciones se migra a un modelo fundamentado en los GC de Sowa [Sow84], para los cuales ya están resueltos los problemas anteriormente planteados. Se elige el modelo de los GC por ser intuitivo, por su notación gráfica, su impacto visual y por la lógica subyacente. Los GC forman una base fuerte para el razonamiento lógico, se pueden usar las relaciones y conceptos resultantes y mantener la consistencia. Se definen así, los MC^S y se presentan reglas canónicas y operaciones lógicas para la formación de nuevos MC^S a partir de otros existentes. Se logra una representación que es equivalente a la notación del cálculo de predicados y que permite razonar con más facilidad.

2.1. Esquema de clases

La herencia es una herramienta natural para representar el conocimiento en forma taxonómicamente estructurada. Esta organización garantiza que todos los miembros de una clase hereden las propiedades adecuadas, asegurando consistencia con la definición de las clases. Con esta estrategia se reduce el tamaño de la base de conocimiento, y se permite la implementación de valores por defecto y excepciones. Los valores por defecto se heredan simplemente desde las superclases apropiadas.

Un modelo que es capaz de representar aquellas jerarquías que permiten una multiplicidad de clases padre es más expresivo. Aunque estas jerarquías de herencia múltiples pueden introducir dificultades en la definición de los lenguajes de representación, sus beneficios son grandes en relación con estas desventajas [Lug97]. Los reticulados constituyen una forma común para el caso de herencia múltiple. Se establece un orden parcial en el conjunto de las clases, indicado por el símbolo \subseteq (\subseteq representa la inclusión entre clases). Se definen los conceptos de subclase y superclase, y como se trata de un reticulado, las clases pueden tener múltiples padres y múltiples hijos. Sin embargo, cada par de clases debe tener una superclase común mínima y una subclase común máxima. La superclase común mínima de una colección de clases es el lugar apropiado para definir las propiedades que son comunes sólo a esas clases. Para resolver el problema que se presenta cuando hay clases que no tienen superclases o subclases comunes naturales, se incorporan dos clases especiales que cubren esas funciones. Se logra así que la \subseteq de clases sea un verdadero reticulado.

En esta propuesta la jerarquía de clases es representada por un MCH que cuenta con dos clases estándares: una clase Universal como superclase de todas las clases y una clase Absurda como subclase de todas las clases. Siguiendo las convenciones de MC, si C_2 es una subclase de C_1 , C_1 aparece en la representación en un nivel superior al de C_2 . Las clases quedan vinculadas a través de la relación “es un”, por lo tanto es necesario el dibujo explícito de una flecha, como se muestra en la figura 1.

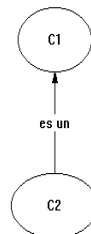


Figura 1

Para jerarquías de gran número de clases, el uso de MCH marca una diferencia importante en lo operacional. Por otra parte, la posibilidad de incorporar apariencias extras (textuales, animaciones gráficas o sonidos) se presenta como un complemento interesante para la comprensión de los agentes humanos.

2.2. Representación de las proposiciones

Cada proposición simple se representa por medio de un MC^S que es un grafo dirigido finito caracterizado por:

Los nodos del mapa representan conceptos, gráficamente los conceptos se dibujan como elipses rotuladas.

Todas las relaciones son binarias. Se mantiene la representación tradicional para las relaciones de los MC, es decir mediante arcos etiquetados con el nombre de la relación.

Los nodos representan objetos del universo de discurso; pueden ser concretos o abstractos. Los

conceptos concretos incluyen conceptos genéricos y conceptos específicos.

Las proposiciones verbales se representan de la siguiente manera: el concepto verbo es raíz del MC^S que representa la proposición. Por ejemplo, para la proposición *El oso toma agua*, la relación *agente* vincula el concepto *toma* con el concepto *oso* y la relación *objeto* vincula el concepto *toma* con el concepto *agua* como puede observarse en la figura 2.

En las proposiciones nominales, el concepto al que se le asocia una propiedad es el concepto raíz del MC^S. Por ejemplo para la proposición *Pájaro color azul* el MC^S asociado es el de la figura 3.

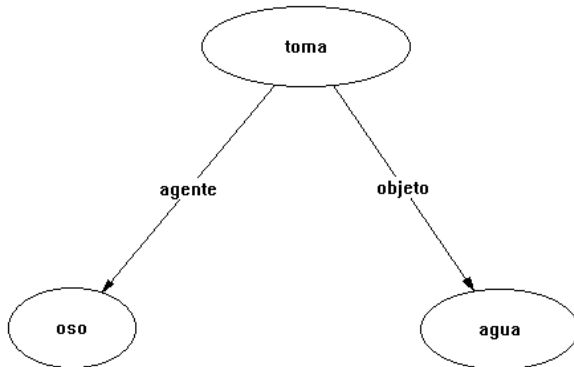


Figura 2

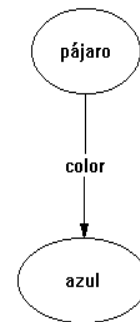


Figura 3

En los MC^S que representan las proposiciones, siguiendo la propuesta de los GC, cada concepto sustantivo es un individuo único de una clase particular. Puede tratarse de un individuo genérico o de un individuo específico. La notación para los distintos casos en las respectivas elipses es la siguiente:

individuo genérico	sin marcador genérico	< Nombre de clase >
	usando marcador genérico	< Nombre de clase > : *
individuo específico	usando nombre	<Nombre de clase> : <Nombre de individuo>
	usando marcador	<Nombre de clase> : # <Número de individuo>

Cada individuo en el mundo del discurso tiene asociado un único token, llamado marcador numérico, que lo identifica plenamente. Esto permite indicar individuos específicos pero sin nombre.

Los MC^S permiten el uso de variables con nombre. Éstas son representadas por un asterisco seguido del nombre de la variable (por ejemplo *X). Esto es útil si dos elipses distintas indican el mismo individuo, pero se trata de un individuo no especificado. El mapa de la figura 4 representa la afirmación *El niño apoya la frente sobre sus rodillas*. Aunque no se especifica cuál es el *niño* al que se refiere la proposición, la variable *X indica que la *frente* y las *rodillas* pertenecen al mismo *niño*.

Los MC^S permiten nodos proposicionales para representar proposiciones subordinadas o coordinadas. De tal forma, además de usar los MC^S para definir relaciones entre objetos del mundo se puede también definir relaciones entre proposiciones. Un nodo proposicional se representa como un nodo del mapa que está etiquetado con un MC^S que representa una proposición, es decir se indica como una elipse que contiene otro MC^S. Por ejemplo la sentencia *Juan cree que el pájaro es azul* se representa por el MC^S que muestra la figura 5. En este caso *cree* es una relación que toma como argumento una proposición.

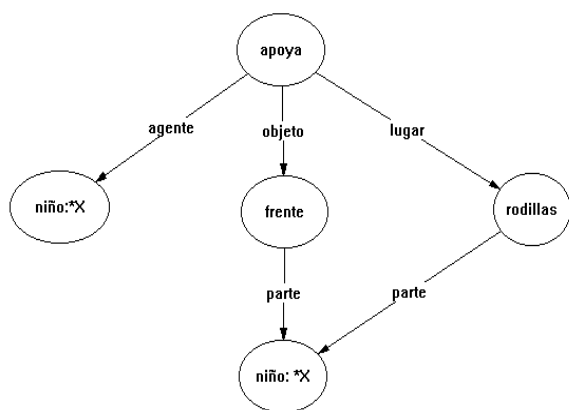


Figura 4

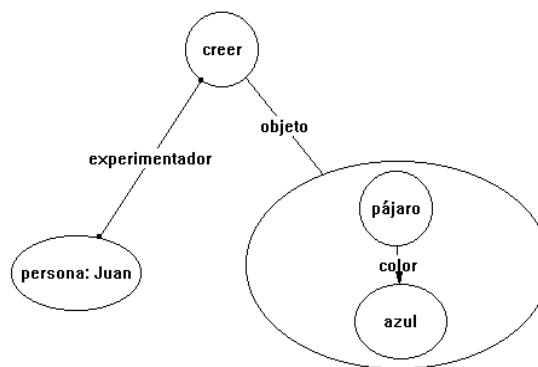


Figura 5

Cada MC^S representa una proposición simple. Los MC^S pueden ser arbitrariamente complejos, pero son siempre finitos. Una base de conocimiento típica contendrá un cierto número de estos mapas, además del MCH que representa el esquema de clases.

Los conceptos proposicionales pueden ser usados con relaciones apropiadas para representar conocimiento acerca de proposiciones. Se muestra así cómo los MC^S con nodos proposicionales pueden ser usados para expresar los conceptos modales de conocimiento y creencia.

2.3. Creación de nuevos MC^S

Para crear nuevos MC^S a partir de MC^S existentes se incluyen operaciones que permiten tratar la generalización y la especialización (reglas de formación canónica) y operaciones lógicas. En el primer grupo se encuentran las operaciones copiar, restringir, unir y simplificar y en el segundo, negación conjunción y disyunción.

2.3.1. Reglas de formación canónica

Dados los MC^S m_1 y m_2 , el resultado de aplicar cada una de las reglas da como resultado un nuevo MC^S como se indica en la figura 6.

La regla de restricción puede usarse para hacer que aparezca una correspondencia entre dos conceptos y así se pueda aplicar luego la regla *Unir*. Las reglas *Restringir* y *Unir* juntas permiten la implementación de la herencia. El reemplazo de un marcador genérico por uno individual implementa la herencia de una de las propiedades de la clase a un individuo. Por ejemplo en el MC^S m_3 , el profesor *Juan* hereda la propiedad de *nacionalidad argentina* originalmente definida en m_1 para un individuo genérico de la clase *profesor*. El reemplazo de la etiqueta de una clase por la etiqueta de una subclase define la herencia entre una clase y una subclase. Es el caso de la propiedad heredada por la subclase *profesor* en m_4 desde la clase *persona* en m_2 .

Uniéndolo un MC^S con otro y restringiendo ciertos conceptos, se puede implementar herencia de una variedad de propiedades.

Como los MC^S se basan en el modelo de los GC, también se puede aplicar a ellos unión y restricción para implementar supuestos plausibles que juegan un rol importante en la comprensión del lenguaje común, por ejemplo de la sentencia *María y Tomás salieron juntos a comer pizza* puede ser modelado con MC^S .

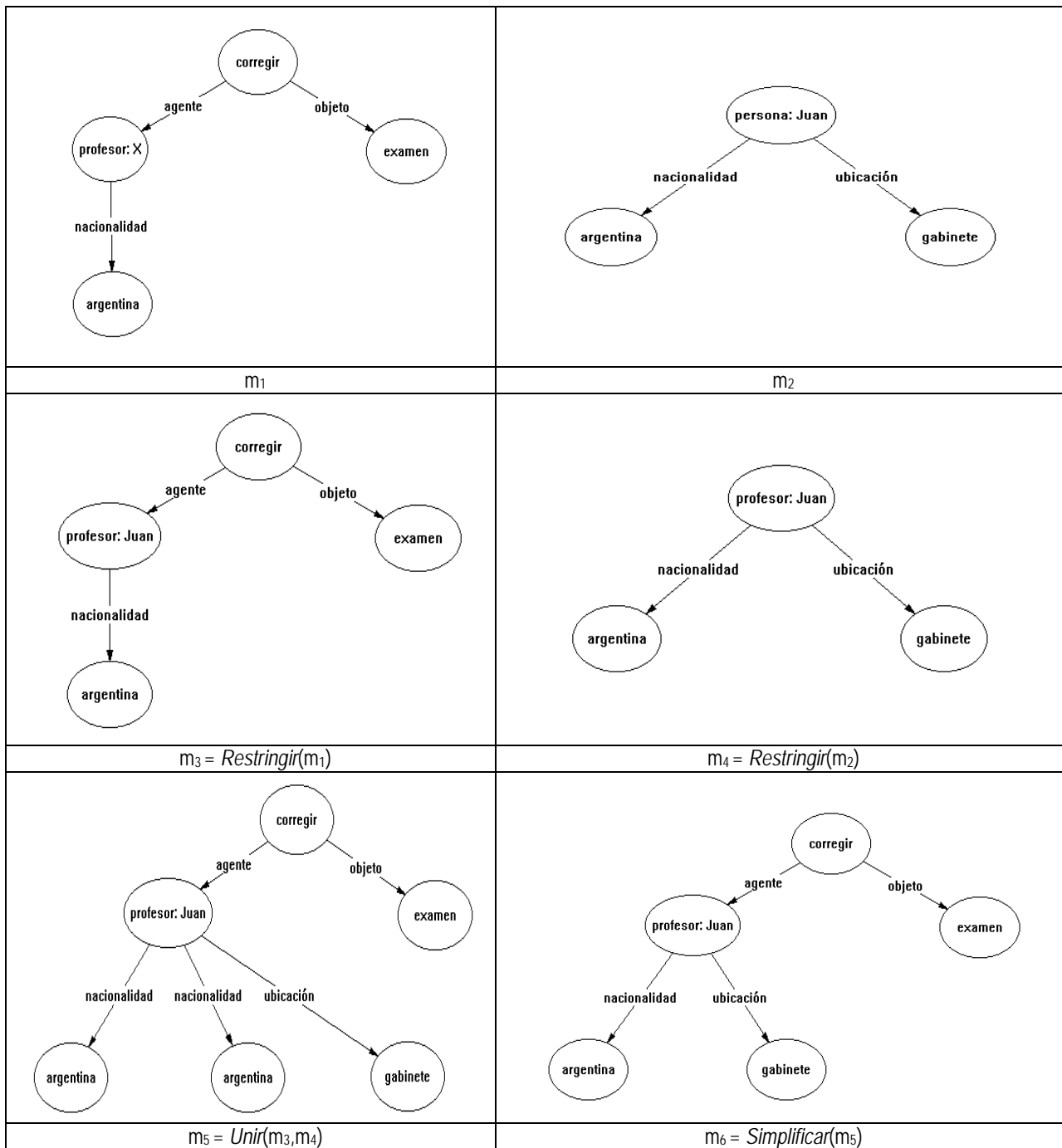


Figura 6

Como en el caso de los GC, la unión y la restricción de MC^S son reglas de especialización. Ellas definen un orden parcial sobre el conjunto de los MC^S derivables. Si un MC^S m_1 es una especialización de m_2 entonces se puede decir que m_2 es una generalización de m_1 . Como Luger ha mencionado, las jerarquías de generalización son importantes en la representación de conocimiento; ellas, junto con la provisión de bases para herencia y otros esquemas de razonamiento del sentido común, se usan en muchos métodos de aprendizaje. Obviamente, no se trata de reglas de inferencia, ellas no garantizan que desde MC^S verdaderos se derivarán siempre MC^S verdaderos. En la restricción del mapa m_3 de la figura 6 el resultado podría no ser verdadero, por ejemplo si *Juan* no es un *profesor*. Otro ejemplo representativo de no preservación de la verdad lo constituye m_5 en la

misma figura, ya que el profesor que está en el gabinete podría ser una persona distinta del que corrige el examen. Estas operaciones no preservan verdad pero tienen la importante propiedad de preservar la condición de significatividad, es decir las reglas de formación canónica no permiten formar MC^S sin sentido desde otros que sí lo tienen, lo que constituye una propiedad importante. “Aunque ello no suena a reglas de inferencia, las reglas de formación canónica forman las bases para muchos de los razonamientos plausibles realizados en comprensión del lenguaje natural y razonamiento del sentido común” [Lug97].

2.3.2. Operaciones Lógicas

– *Negación*: La existencia de los nodos proposicionales en los MC^S hace que se pueda implementar con facilidad la negación de una proposición. Se define una operación llamada *neg* que toma como argumento un concepto proposicional y afirma ese concepto como falso. Para la representación gráfica, se muestra la proposición que se quiere negar como un nodo proposicional, y para establecer la negación se usa un nodo ficticio desde el que parte la relación *neg* hacia el nodo proposicional. El uso de ese nodo ficticio es al solo efecto de tratar a la operación *neg* como binaria.

– *Conjunción*: se pueden formar MC^S que representen aserciones disyuntivas. Si cada uno de los MC^S que representan las proposiciones a coordinar tiene un nodo raíz, se puede establecer la relación “y” vinculando ambos nodos raíz; de lo contrario puede representarse cada una de las proposiciones a vincular por medio de un nodo proposicional y luego relacionarlos con “y”.

– *Disyunción*: De acuerdo con las reglas de la lógica, usando negación y conjunción se pueden formar MC^S que representen aserciones disyuntivas. Para simplificar esto también se puede definir una relación “o” la cual toma dos proposiciones y representa su disyunción en forma análoga a como se representa la conjunción.

2.3.3. Cuantificación de variables

Se asume que en los MC^S los conceptos genéricos están existencialmente cuantificados. Por ejemplo en el caso del MC^S de la figura 3, el concepto genérico *pájaro* representa una variable existencialmente cuantificada. Este MC^S se corresponde con la expresión lógica:

$$\exists X \exists Y (pájaro(X) \wedge color(X, Y) \wedge azul(Y))$$

Se puede representar cuantificación universal mediante el uso de negación y cuantificación existencial. Por ejemplo, Para el MC^S que representa la negación de la proposición *El pájaro es amarillo* se tiene la siguiente expresión lógica:

$$\forall X \forall Y (\neg(pájaro(X) \wedge color(X, Y) \wedge amarillo(Y)))$$

Un MC^S que hace referencia a un individuo particular, por ejemplo el que representa la proposición *El oso Simón es de color marrón*, se corresponde con la siguiente expresión del cálculo de predicados

$$\exists X_1(oso(Simón) \wedge color(Simón, X_1) \wedge marrón(X_1))$$

3. Poder expresivo de los MC^S

Como los ejemplos precedentes sugieren, existe una correspondencia directa desde los MC^S hacia la notación del cálculo de predicados. Los MC^S resultan equivalentes al cálculo de predicados en su poder expresivo. El siguiente algoritmo permite obtener la expresión del cálculo de predicados equivalente a un MC^S dado.

Algoritmo Expresión Lógica Equivalente

Entrada: $MC^S m$

Salida: expresión del cálculo de predicados equivalente a m

Pasos:

- 1- Asignar una única variable X_1, \dots, X_n a cada uno de los n conceptos genéricos en m .
- 2- Asignar una única constante a cada concepto individual en m . Esta constante puede simplemente ser el nombre o el marcador usado para indicar el referente del concepto.
- 3- Representar cada nodo concepto por un predicado unario con el mismo nombre del tipo de ese nodo y cuyo argumento es la variable o constante asignada a ese nodo.
- 4- Representar cada relación conceptual en m como un predicado binario cuyo nombre es el mismo que el de la relación. Esto permite que cada argumento del predicado sea la variable o la constante asignada al correspondiente nodo concepto vinculado a tal relación.
- 5- Tomar la conjunción de todas las sentencias atómicas formadas en los puntos 3 y 4. Éste es el cuerpo de la expresión del cálculo de predicados. Todas las variables en la expresión son existencialmente cuantificadas.

Es importante destacar que aunque los MC^S , así como los GC, pueden reformularse usando la sintaxis del cálculo de predicados, ellos soportan un número de mecanismos de inferencia de propósito especial tales como unión y restricción que no son normalmente parte del cálculo de predicados.

4. Plataforma para la creación de la Base de Conocimiento

4.1. Creación del sistema de clases

Con el propósito de crear y visualizar MCH se creó una plataforma específica con las siguientes características: Se distinguen dos modos de trabajo: un modo lector y un modo autor. En el modo lector la tarea básica es la inspección de un MCH ya creado, visitando uno o más niveles del mismo. En el modo autor se crean efectivamente los MCH. Ambos modos tienen características de trabajo que los diferencian. En el modo autor se encuentran los menús básicos de cualquier aplicación que disponga de la capacidad de edición: manejo de archivos (abrir, salvar, salvar como, imprimir, etc.), opciones de edición (cortar, pegar, deshacer, mover, seleccionar, etc.) y los recursos gráficos necesarios para la creación del hipertexto.

Además de las características descritas, están disponibles las siguientes herramientas gráficas específicas para la creación de MCH: elipse para rotular (para crear conceptos propios), arco para rotular (para establecer relaciones internas o externas), botón elíptico para rotular (para crear un concepto propio que explota en otra vista), botón rectangular para rotular (para crear un concepto importado), paleta de colores (para establecer el código cromático de las vistas), rectángulo transparente (para asociar ejemplos a un concepto) y opción para asociar apariencias de distinto tipo a los conceptos.

El diseño de esta plataforma, presentado en [Mor96], permite conformar un ambiente que centra la atención en la construcción del conocimiento del usuario y no en los aspectos operacionales. La implementación de la plataforma sigue fielmente la especificación propuesta [Gia96]. En todos los casos la plataforma automatiza lo meramente operacional, liberando la atención del usuario con el propósito de poder centrarla en la diagramación del mapa (procesos de selección, jerarquización, relación y representación)

4.2. Extensión para la representación de proposiciones

La plataforma debe contar con todos los recursos para el ingreso de nuevos MC^S y para la visualización de MC^S creados por la aplicación de reglas de formación canónica o de operaciones lógicas. Además debe poder mostrar en cada caso la expresión lógica equivalente.

Los recursos para el ingreso de nuevos MC^S estarán disponibles en una barra de herramientas específicas, integrada por:

– *Elipse para rotular*: permite representar un concepto del MC^S . Al seleccionarse esta herramienta se arrastra una elipse hasta el lugar de la pantalla que indique el autor. Una vez ubicada la elipse en la hoja de trabajo el cursor queda en su interior, hasta tanto se complete el rótulo del concepto, adaptándose automáticamente el tamaño de la elipse a su contenido (de esta manera se mantiene la consistencia con las opciones para crear MCH). El mencionado rótulo ingresa un nombre de clase, y si se trata de un individuo específico, ingresa el marcador numérico y opcionalmente el nombre. Se realiza automáticamente la actualización y el control de consistencia de la base de datos. Si aparece alguna inconsistencia, la plataforma debe informarla y aceptar el ingreso de las modificaciones necesarias.

– *Arco para rotular*: permite insertar la relación que une a dos conceptos de un MC^S . Se le debe asociar el rótulo con el nombre de la relación. Una vez elegida esta herramienta, se señalan los dos conceptos de la hoja de trabajo que se intentan unir y a partir de allí automáticamente aparecerá el arco dibujado y la plataforma esperará por el nombre que debe ser ingresado. Si el autor mueve las elipses por distintos lugares de la pantalla, la plataforma actualiza automáticamente el dibujo de los arcos.

– *Nodo proposicional*: Al seleccionarse esta herramienta se arrastra una elipse hasta el lugar de la pantalla que indique el autor, se abre automáticamente una ventana auxiliar donde se puede dibujar el MC^S asociado al nodo. Cuando se cierra la ventana auxiliar, el MC^S creado queda incluido en la elipse, sin necesidad de que el autor tenga que preocuparse por los ajustes de tamaño de los elementos que componen el mapa.

Los recursos para crear y visualizar nuevos MC^S mediante la aplicación de reglas de formación canónica o de operaciones lógicas estarán disponibles en el menú Operaciones, integrado por:

– *Copiar*: Cuando se aplica esta opción al MC^S de la ventana activa, se crea otro mapa igual que se visualiza en otra ventana, se abre un cuadro de diálogo para permitir el ingreso del nombre del nuevo mapa.

– *Restringir*: Al aplicarse esta opción al MC^S de la ventana activa se abre un cuadro de diálogo para permitir el ingreso del nombre del concepto que se usará para la restricción. Pueden darse dos casos, el primero cuando un concepto es etiquetado con un marcador genérico, y dicho marcador genérico es reemplazado por un marcador individual de la misma clase, y el segundo, cuando una etiqueta de clase sobre un concepto es reemplazada por la etiqueta de una de sus subclases mientras ésta sea consistente con el referente del concepto. El autor debe ingresar la nueva etiqueta del concepto y si la Plataforma detecta inconsistencia, la informa para que se realicen las modificaciones del caso.

– *Unir*: La regla unir permite integrar dos MC^S en uno solo. Al seleccionarse esta opción la Plataforma abre un cuadro de diálogo en el que muestra la lista de pares de MC^S que tienen un nodo concepto idéntico. Entonces el autor selecciona el par que desea unir, a continuación la plataforma solicita el nombre para el nuevo mapa y abre una nueva ventana en la que muestra el MC^S resultante

– *Simplificar*: Esta opción se aplica cuando un MC^S contiene dos relaciones duplicadas, y por lo tanto una de ellas debe ser borrada. Las relaciones duplicadas aparecen frecuentemente como resultado de una operación de unión.

– *Negación*: Cuando se aplica esta opción al MC^S de la ventana activa, la plataforma primero incluye dicho MC^S en un nodo, luego crea un nodo ficticio y por último establece la relación *neg* desde el nodo ficticio hacia el nodo proposicional.

– *Conjunción*: Al seleccionarse esta opción la Plataforma abre un cuadro de diálogo en el que

muestra la lista de MCS existentes. Entonces el autor selecciona el par sobre los que desea operar y a continuación la Plataforma solicita el nombre para el nuevo mapa y abre una nueva ventana en la que muestra el MCS resultante.

– *Disyunción*: Se resuelve de manera análoga a la opción anterior

Además se integra a la barra de menús la opción MCS con las siguientes posibilidades:

– *Nuevo*: para abrir una ventana preparada para recibir un MCS.

– *Eliminar*: para sacar de la base de conocimiento el MCS que se visualiza en la ventana activa.

– *Nombre*: para nombrar (o renombrar) al MCS que se visualiza en la ventana activa.

– *Expresión lógica*: para calcular y mostrar la expresión lógica equivalente al MCS que se visualiza en la ventana activa.

La figura 7 muestra gráficamente la interface de la Plataforma extendida.

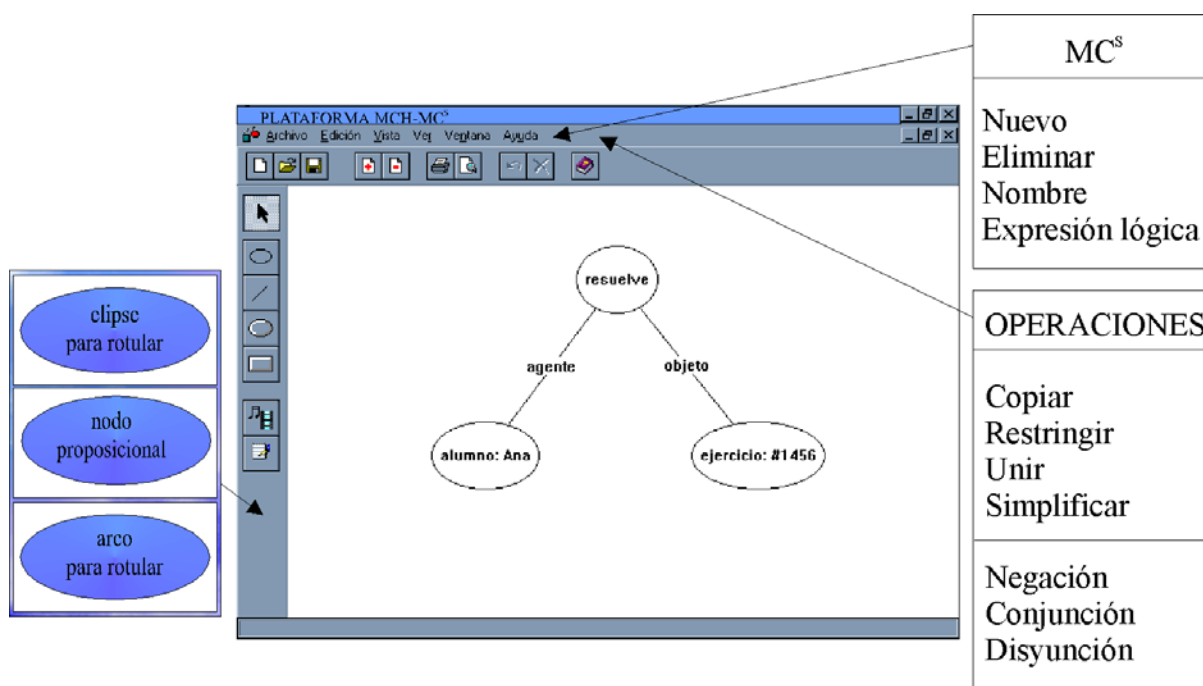


Figura 7

5. Conclusiones

Cuando un educando pone en marcha una estrategia, debe disponer de recursos alternativos, para que en función de las demandas de la tarea de aprendizaje que se le presenta, pueda elegir aquellos que considera mejores. Sin una variedad de recursos no es posible actuar estratégicamente. Se ha comprobado el valor de los MCH como estrategia de aprendizaje y se ha podido observar y evaluar la interacción de alumnos con la plataforma diseñada especialmente para trabajar con estos mapas. Se ha manifestado una diferencia sustancial y favorable con respecto a los MC tradicionales en los planos motivacional, operacional y de la comprensión.

Por otra parte, un punto fundamental en las posturas constructivistas considera que aprender significa construir estructuras de conocimiento. Para promover situaciones de aprendizaje constructivista los ambientes deben poseer facilidades para la construcción de conocimiento. El interés por los micromundos indica una preocupación en estudiar los aspectos cualitativos del aprendizaje y no sólo los aspectos cuantitativos. Otro principio del constructivismo señala que el alumno debe aprender haciendo. El proceso de construcción del esquema gráfico de las

representaciones de conocimiento proporciona experiencias de aprendizaje muy ricas para los alumnos. La selección de una representación capaz de adecuarse a los requerimientos de los aprendices humanos y de los agentes de software es uno de los puntos claves para el diseño de los sistemas de aprendizaje mixtos. En este trabajo se presenta una extensión del ambiente MCH existente, orientado al trabajo sobre la representación de conocimiento.

En los sistemas mixtos orientados al aprendizaje que incluyen Agentes Pedagógicos, el esquema de representación de conocimiento debe satisfacer los requerimientos de los agentes humanos y de los agentes de software. En la búsqueda de un sistema que sea lo suficientemente flexible para el manejo de los alumnos y lo suficientemente riguroso como para que los agentes de software puedan razonar, se recurre a un modelo conformado por un esquema de clases y un esquema de representación de proposiciones. El esquema de clases se diseñó como un reticulado que se representa por medio de un MCH basado en el modelo de los MC de Novak. Para la representación de las proposiciones se definieron los MC^S, Mapas Conceptuales basados en el modelo de Sowa. Se trata de una representación que reúne los beneficios de una notación gráfica cercana a los MC tradicionales y que proporciona además una base fuerte para el razonamiento lógico (se muestra la equivalencia con la notación del cálculo de predicados). El sistema se completa con un conjunto de reglas canónicas y operaciones lógicas para la formación de nuevos MC^S a partir de otros existentes. Con el propósito de crear y visualizar una base de conocimiento con estas características, se presenta el diseño de una nueva Plataforma, que comprende la existente para el trabajo con MCH y una extensión para el desarrollo y manejo de los MC^S.

Los resultados de este trabajo son alentadores, se advierte cómo desde el área de Ciencias de la Computación puede hacerse un aporte significativo al área de Educación, que va más allá de lo meramente operacional.

6. Bibliografía

- [Aus78] Ausubel, D. P., Novak J. D. "Educational Psychology: A Cognitive View". 2nd Ed. New York: Holt, Rinehart and Winston. 1978.
- [Coll89] Coll, C. "Aprendizaje escolar y construcción del conocimiento" Ed. Paidós. 1989.
- [Gir98] Giraffa, L. "The use of Agents techniques on Intelligent Tutoring Systems". RIBIE-98. Brasil. 1998.
- [Gri95] Grinstein G - Levkowits H. "Perceptual Issues in Visualization", Springer-Verlag, 1995.
- [Joh97] Johnson, W. and Shaw, E. "Using Agents to Overcome Deficiencies in Web-Based Courseware". Proceedings of the AI-ED 97 Workshop on Pedagogical Agents. 1997.
- [Laj93] Lajoie, S. "Computer Environments as Cognitive Tools for Enhancing Learning". 1993. McGill University.
- [Leh93] Lehrer, R. "Authors of knowledge: Patterns of Hypermedia Design". 1993. University of Wisconsin-Madison.
- [Les97a] Lester, J., Converse, S., Stone, B., Kahler, S., and Barlow, T. "Animated pedagogical agents and problem-solving effectiveness: A large-scale empirical evaluation". Eighth World Conference on Artificial Intelligence in Education. IOS Press, Amsterdam. 1997.
- [Les97b] Lester, J. and Stone, B. "Increasing believability in animated pedagogical agents". In Johnson, W. and Hayes-Roth, B. (Eds.). Proceedings of the First International Conference on Autonomous Agents. ACM Press. 1997.
- [Les97c] Lester, J. "Mixed Initiative Problem Solving with Animated Pedagogical Agents". AI-ED97. Eighth World Conference on Artificial Intelligence in Education - Workshop V : Pedagogical Agents. Japan. 1997.
- [Lug97] Luger, G. and Stubblefield, W. "Artificial Intelligence: Structures and Strategies for Complex Problem Solving". Addison Wesley. 1997.

- [Mor96] Moroni, N. - Vitturini, M. - Zanconi, M. - Señas, P. “Una plataforma para el desarrollo de mapas conceptuales hipermediales”. Taller de Software Educativo - IV Jornadas Chilenas de Computación. Valdivia. 1996.
- [Mou96] Moulin, B. and Chaib-Draa, B. “ An overview of Distributed Artificial Intelligence”. En Foundations of Distributed Artificial Intelligence. O'Hare and Jennings, eds. 1996.
- [Nov84] Novak, J. y Gowin, D. “Learning how to learn”. Cambridge University Press. 1984.
- [Nov85] Novak, J. “Metalearning and metaknowledge strategies to help students learn how to learn. Cognitive Structure and Conceptual Change”. New York. Academic Press.1985.
- [Señ96] Señas, P., Moroni, N., Vitturini, M. y Zanconi, M.: “Hypermedial Conceptual Mapping: A Development Methodology”. 13th International Conference on Technology and Education. University of Texas at Arlington, Department of Computer Science an Engineering. New Orleans 1996.
- [Señ99] Señas, P. Tesis de Magíster: “MCH como herramienta para la Representación de Conocimiento en Agentes Inteligentes”. Universidad Nacional del Sur. 1999.
- [Señ00] Señas, P., Moroni, N. “Computing Environments for metalearning. Interconnecting Hypermedia Concept Maps”. ED-MEDIA 2000. Montreal. Canada. 2000.
- [Sow84] Sowa, “Conceptual Structures: Information Processing in Mind and Machine”. Addison Wesley. 1984.
- [Zan98] Zanconi, M., Moroni, N., Vitturini, M., Malet, A., Borel, C. y Señas, P. Tecnología computacional y meta-aprendizajes. RIBIE-98. 1998.