Herramientas para la construcción de conocimiento en ambientes de aprendizaje abiertos: Construcción y Visualización del Grafo Integrador de un MCH

Sergio R. Martig Perla Señas
Dpto. de Ciencias de la Computación
Universidad Nacional del Sur – Bahía Blanca
{smartig,psenas @cs.uns.edu.ar}

Resumen:

Desde una postura constructivista, aprender significa construir estructuras de conocimiento. Para promover situaciones de aprendizaje constructivista los ambientes deben poseer facilidades para la construcción de conocimiento y para la participación activa del alumno en el desarrollo de ese proceso. La construcción del esquema gráfico de las representaciones de conocimiento proporciona experiencias de aprendizaje muy ricas para los estudiantes. Para realizarlo dentro de un Ambiente de Mapas Conceptuales Hipermediales (MCH) resulta de sumo interés poder visualizar todas las vistas del MCH en forma integrada. Para ello se le asocia al mapa un grafo denominado Grafo Integrador (GI_{MCH}). Resolver la visualización de dicho grafo no es un problema trivial, si se tiene en cuenta el elevado número de nodos que tendrá en la mayoría de los casos. Se presenta en este trabajo la definición, justificación desde lo pedagógico y la solución para la construcción y visualización de un GI_{MCH}.

Palabras Claves:

Tecnología Avanzada en Educación - Aprendizaje Significativo - Mapas Conceptuales Hipermediales - Visualización de Grafos.

1 Introducción:

Los Mapas Conceptuales Hipermediales (MCH) constituyen un recurso poderoso para la representación de las ideas, su uso es beneficioso para investigadores y profesionales de distintas disciplinas, quienes pueden contar con una herramienta específica para la representación de las ideas. Desde lo didáctico, suman la riqueza de los MC y el poder de la hipermedia, constituyéndose en una moderna tecnología educativa para el trabajo sobre meta-aprendizaje y meta-conocimiento.

Las representaciones pictóricas de grafos han sido una herramienta de modelado tradicional para una variedad de aplicaciones en ingeniería de software, bases de datos, visualización, administración de proyectos, redes de computadoras, etc. Dichas representaciones fueron y son usadas intensivamente, incluso en forma manual. En la actualidad la generación automática de los dibujos de grafos encuentra muchas aplicaciones: diagramas de flujo de datos, jerarquías de clases orientadas a objetos, diagramas de entidad relación, cartas de estructura de sistemas, redes de Petri, diagramas de transición de estados, esquemas de circuitos, representación del conocimiento, entre otras tantas y en particular se los propone en este trabajo para la representación del Grafo Integrador de un MCH (GI_{MCH}).

2 Ambientes de aprendizaje abiertos con Mapas Conceptuales Hipermediales

Los MCH al igual que los Mapas Conceptuales de Novak (MC), sobre los cuales se basan, son esquemas de representación de conocimiento. En síntesis los MCH suman la riqueza de los MC con el potencial de la tecnología hipermedial, todo esto apoyado por una plataforma que hace flexible su diseño y mantenimiento, lo que los constituye en una poderosa herramienta para la construcción de

conocimiento. En el área educativa, ambos esquemas han sido probados con éxito como potentes estructuras capaces de contribuir con la construcción de aprendizajes significativos [Zan98].

Los MC son una herramienta cuya potencialidad ha sido probada en el logro de aprendizajes significativos, para poder asumir la propia elaboración de significados, en definitiva para poder "aprender a aprender". En los últimos años se ha incrementado su uso en las aulas, no sólo como herramienta de aprendizaje sino también como técnica de evaluación. Sin embargo, se ha detectado que existen aspectos operacionales que dificultan la realización de estos mapas, si se piensa en su construcción mediante elementos tradicionales, especialmente cuando el mapa cuenta con un elevado número de conceptos, o cuando es necesario hacer refinamientos del mismo o cuando se desea interconectar mapas ya existentes. La aplicación de tecnología informática hipermedial facilita la construcción de los MC, salvando las dificultades antes mencionadas. Además y principalmente, en los MCH, se favorece sustancialmente el trabajo de selección y de jerarquización de conceptos, enriqueciendo de esta manera su valor pedagógico. Siguiendo esta motivación se crearon los MCH, una metodología de desarrollo y una plataforma específica que permite la creación, mantenimiento, lectura e interconexión de estos mapas.

Un MCH es un documento hipermedial; cada nodo de la hipermedia contiene una colección de conceptos relacionados entre sí por palabras enlace. A cada uno de estos nodos se lo denomina *vista* del MCH. Cada vista puede ser visualizada en una ventana y es caracterizada por un color y por un nombre, por defecto el del concepto propio más abarcativo en dicha vista. En una vista pueden distinguirse los siguientes elementos:

Elementos		**Representación**
Vistas		Ventana
Conceptos Propios	Terminales	Elipse Rotulada
	No Terminales	Botón Elíptico Rotulado
Conceptos Importados		Botón Rectangular Rotulado
Relaciones	Internas	Arco Rotulado
	Externas	Arco Rotulado entre rectángulo y elipse
Apariencias	Sonido – Texto Video –Gráfico	Elipse Marcada
Ejemplos		Palabras

Con el propósito de poder alcanzar los beneficios formativos del trabajo con MC enunciados por Novak [Nov85] y de poder aprovechar las ventajas de su representación hipermedial, es importante contar con una metodología de desarrollo de MCH y con una plataforma ad-hoc para la creación, lectura e interconexión de estos mapas [Señ96b] [Mor96].

La realización de nuevos aprendizajes, implica en algunos casos, una revisión de los conocimientos anteriormente adquiridos y en todos los casos una integración de lo nuevo con lo previamente incorporado. "Para trabajar en este sentido, resulta de interés, un ambiente para desarrollar interconexiones de MCH. La principal motivación para esta tarea está dada por la necesidad de integrar los nuevos conocimientos con los conocimientos previos y por el deseo de relacionar distintos puntos de vista de un mismo tema. Se trata de una propuesta que trasciende la mera simplificación de las tareas en los procesos de enseñanza-aprendizaje, y que se presenta como una herramienta efectiva para la construcción del conocimiento, de acuerdo con una psicología del aprendizaje constructivista" [Sen99]. El ambiente de aprendizaje que comprende los MCH y la plataforma para su manejo, incluye la posibilidad de la interconexión de mapas, ampliando de este

modo, la riqueza de esta técnica como recurso valioso y comprobado para la representación de las ideas [Zan98].

A modo de ejemplo se incluyen en la figura 1 cinco vistas de un mapa conceptual hipermedial sobre el tema de los MCH. Ha sido generado utilizando la plataforma desarrollada para tal fin [Mor96]

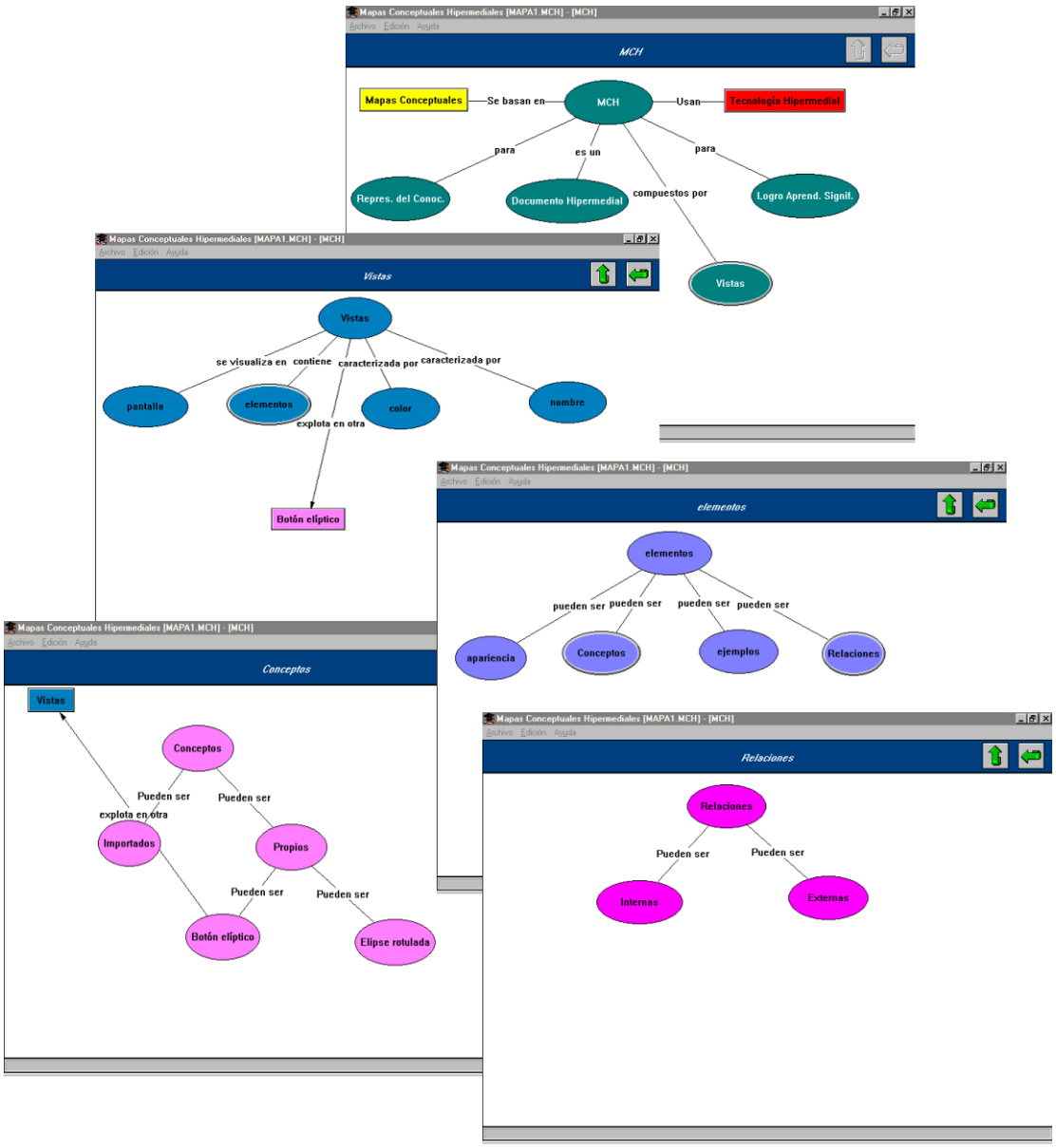

Figura 1: Cinco vistas de un Mapa Conceptual Hipermedial sobre el tema de los MCH.

2.1 Grafo Integrador del MCH

Para el trabajo de construcción de conocimiento dentro de un Ambiente MCH resulta de sumo interés poder visualizar todas las vistas de un MCH en forma integrada, como si fuese un MC tradicional al que se le adiciona el código cromático y diferentes apariencias en la representación de los conceptos. Con ese propósito se define el GI_{MCH}, que se obtiene como síntesis de todas las vistas del MCH original (ver figura 2 sobre MCH de cinco vistas).

El GI_{MCH} tiene las siguientes características:

- Es un grafo jerárquico. La ubicación de los nodos respeta la jerarquía original del MCH.
- Aparecen todos los conceptos una única vez como nodos del grafo.
- Se respetan las relaciones del MCH original, quedando plasmadas como arcos etiquetados.
- Se mantiene el código cromático y la asociación de apariencias.

2.1.1 Creación del Grafo Integrador de un MCH

A continuación se desarrollan los algoritmos para la creación del GI_{MCH} a partir de un MCH dado.

Algoritmo de Creación del GI_{MCH}

(M: MCH, $\underline{GI_M}$: Digrafo por niveles)

Acciones

CrearGrafoNulo($\underline{GI_M}$)

ExpandirGrafo(VistaPrincipal(M), 1, Concepto, $\underline{GI_M}$)

IncorporarRelacionesExternas(M, $\underline{GI_M}$)

Algoritmo ExpandirGrafo

(V: Vista, NivelInicialVista: Entero, C: Concepto, GI: Grafo)

Var

G_{vista}: Grafo

Acciones

GenerarGrafoPorNiveles(V, NivelInicialVista, Color(V), $\underline{G_{vista}}$)

Integrar(G_{vista}, C, GI)

Para cada botón elíptico B de V

ExpandirGrafo(Vista(B), Nivel(B), B, GI)

Algoritmo IncorporarRelacionesExternas

(M: MCH, GI: Grafo)

Acciones

Para cada vista V de M

Para cada botón rectangular C_{imp} de V

Para cada elipse rotulada o Botón elíptico C_{propio} de V tal que existe la relación R entre C_{propio} y C_{imp}

Si no existe un arco rotulado con R entre C_{propio} y C_{imp} en GI_M

Entonces

Agregarlo

Algoritmo GenerarGrafoPorNiveles

(V: Vista, NivelInicialVista: Entero, Color: TipoColor, G: Grafo)

Genera un grafo con las siguientes características:

- Es jerárquico
- Dirigido
- Cada nodo tiene los siguientes atributos:
  - Rótulo: Nombre del concepto correspondiente en V.
  - Nivel: Entero según la posición relativa del concepto asociado en la vista.
  - Color: Color de la vista a la cual pertenece como concepto terminal.
- Cada arista (c_i, c_j) se obtiene a partir de la relación R entre los conceptos c_i y c_j de la vista, tomando el rótulo de la misma como atributo.

Algoritmo Integrar(G_{vista}: Grafo, C: Concepto, GI: Grafo)

Acciones

Si GI = GrafoNulo

Entonces GI $\leftarrow$ Gvista

Sino

$V_{GI} \leftarrow V_{GI} \cup V_{Gvista}$

$A_{GI} \leftarrow A_{GI} \cup A_{Gvista}$

Para cada vértice $v \in V_{GI}$

Si $v \in V_{Gvista}$

Entonces

$Color(v_{VGI}) \leftarrow Color(v_{VGvista})$

Observación: Como en los MCH los conceptos terminales aparecen representados una única vez, es posible identificar vértice con rótulo en el GI en formación.

A modo de ejemplo se muestra en la figura 2 el Grafo Integrador del mapa conceptual de la figura 1. En el mismo se indican los niveles asociados a cada vértice

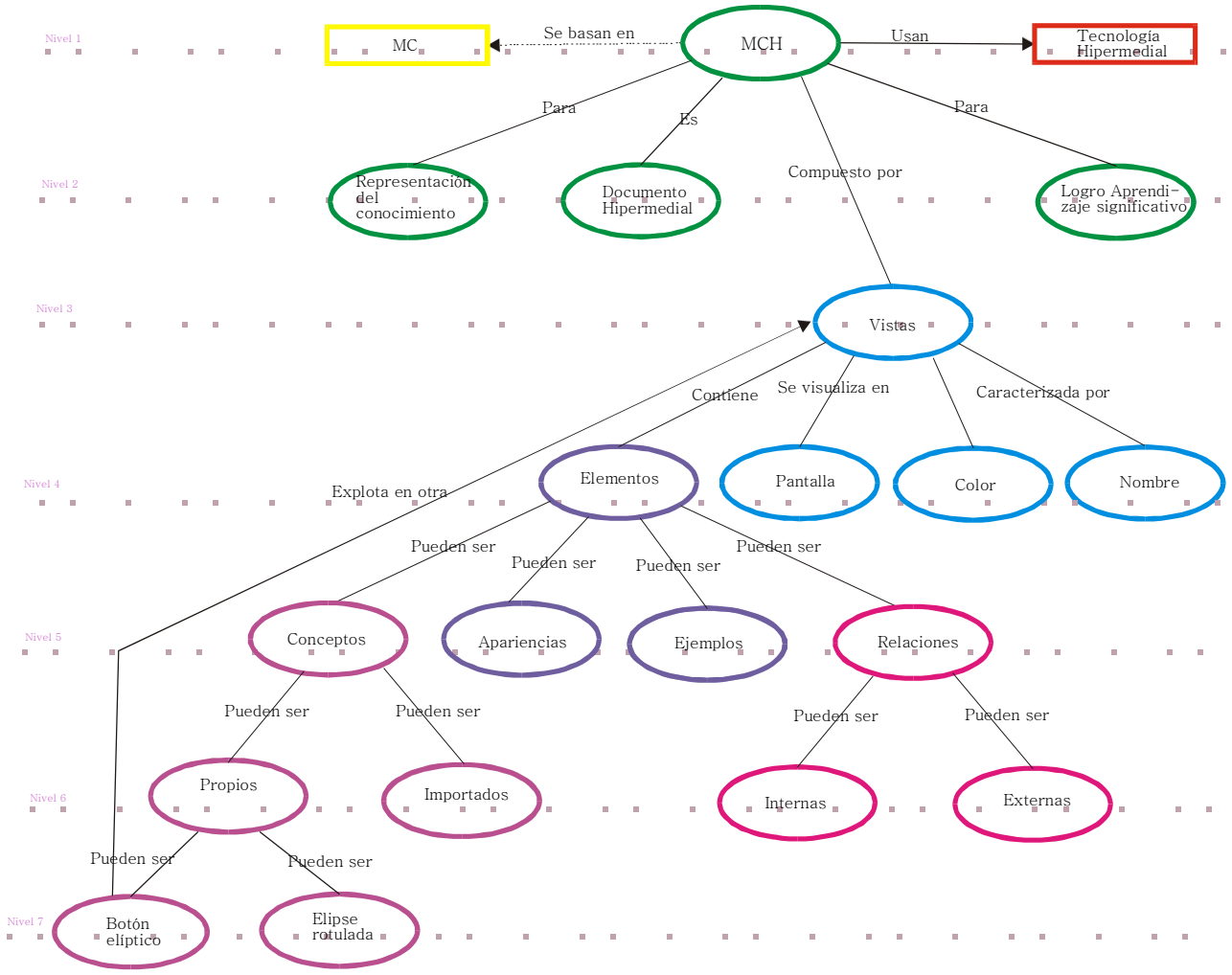

Figura 2: Grafo Integrador del MCH de la figura 1

3 Visualización de Grafos

Los grafos son estructuras abstractas que pueden ser utilizadas para representar información que pueda ser modelada como objetos y conexiones entre los mismos. Estas características y el hecho de ser fáciles de leer y entender hacen que sean una herramienta muy útil en diferentes dominios, en particular en el área de representación de conocimiento.

La visualización de grafos encara el problema de construir representaciones geométricas de los mismos respetando determinados criterios estéticos y restricciones impuestas por el usuario. Hay varios criterios estándar que se utilizan en la graficación de grafos. Usualmente los vértices son representados por símbolos tales como cajas o puntos, y las aristas por curvas abiertas de Jordan conectando los símbolos que representan a los vértices asociados. Sin embargo esos estándares pueden variar según la aplicación. Además un grafo tiene infinidad de formas de ser dibujado. La utilidad de una representación dada depende de su legibilidad, es decir de la capacidad de transmitir el significado de ese grafo en forma clara y rápida. Los aspectos de legibilidad pueden ser expresados por medio de los criterios estéticos, tales como la minimización de cruces entre aristas o la muestra de simetrías. En este marco varios problemas de visualización de grafos pueden ser

formalizados como problemas de optimización de objetivos múltiples, por ejemplo obtener una representación con un mínimo número de cruces y puntos de inflexión y mínima área.

3.1 Técnicas Generales para el dibujo de grafos:

Las nociones de conectividad y planaridad juegan un rol muy importante en el dibujo de grafos. En particular para el caso de los grafos planares: [DIB99]

- Los cruces de aristas disminuyen la legibilidad [PCJ96] [PUR97].
- Hay un fuerte desarrollo teórico sobre los grafos planares, dentro de la teoría de grafos, que puede ser usado para simplificar consideraciones topológicas que de otra manera serían engorrosas de probar.
- Los grafos planares son ralos: La fórmula de Euler [BON76] implica que un grafo planar simple de n vértices tiene como máximo 3n - 6 aristas.

Antes de describir las distintas familias de algoritmos para generar dibujos de grafos, analicemos cuales son los datos de entrada que se deberán proveer.

El dato de entrada fundamental para un algoritmo de visualización de grafos es el propio grafo G que se quiere visualizar.

Dado que la percepción humana de un mismo dibujo varía de persona en persona, y que aplicaciones diferentes pueden requerir tipos de representaciones diferentes, surge que otro parámetro esencial en el dibujo de grafos es el del dominio de aplicación. Estos requerimientos que contribuirán a caracterizar el dibujo a obtener, pueden ser expresados en términos de los siguientes aspectos:

a) *Convenciones del dibujo:* Son las reglas básicas que el dibujo debe respetar para ser admisible. Algunas de las convenciones más usadas son las siguientes: Dibujos con líneas poligonales, dibujos con líneas rectas, dibujos con líneas ortogonales, dibujos con coordenadas enteras, dibujos planos.

b) *Estética:* Los criterios estéticos especifican un conjunto de propiedades del gráfico que deberemos aplicar, tanto como sea posible, para mejorar su legibilidad. Los criterios comúnmente adoptados son: Minimizar el número de cruces entre aristas, minimización del área de dibujo, minimización de la suma de las longitudes de las aristas, longitud uniforme de las aristas, máximo número de inflexiones, uniformidad de inflexiones, resolución angular, aspect ratio y simetrías.

c) *Restricciones:* Mientras que tanto las convenciones como los criterios estéticos se aplican a todo el grafo, las restricciones comúnmente se refieren a subgrafos específicos. Algunos ejemplos de restricciones son: Ubicación relativa de un conjunto de vértices dentro del dibujo, mantener un determinado subconjunto de vértices cercanos entre sí, Dibujar un determinado paso en una dirección, Dibujar un determinado subgrafo con una forma dada.

La investigación en el dibujo de grafos se ha enfocado casi exclusivamente en el aspecto algorítmico, donde el dibujo del grafo es generado de acuerdo a un conjunto de criterios estéticos y convenciones prefijadas. Este enfoque no es muy flexible a la adopción de restricciones impuestas por el usuario. Hay trabajos que muestran que los algoritmos existentes no tienen una gran capacidad para satisfacer restricciones y que es muy dificultosa su modificación para lograrlo. [TAM88][DIB92]

En los últimos años se han realizado varios intentos en el desarrollo de lenguajes para la especificación de restricciones y en el diseño de técnicas para dibujar grafos basadas en sistemas de restricciones [KAM89][MAR91][DEN93]. Eades y Lin [LIN95] combinan métodos algorítmicos y

declarativos en el dibujo de árboles. Brandenburg presenta un intento basado sobre una gramática para grafos. [BRA95]

3.1.1 Paradigmas en los Algoritmos para Dibujar Grafos:

Sea C una clase de Grafos y sea C' una subclase de C. Un algoritmo que pueda aplicarse para dibujar cualquier grafo perteneciente a C, puede también ser aplicado para hacer lo propio con cualquier grafo perteneciente a C'

Dado que la metodología para dibujar grafos puede verse como un pipeline de pasos funcionales relativamente independientes, y que

- cada paso ejecuta una única tarea que servirá de entrada para el próximo paso
- la entrada y la salida de cada uno de los pasos del pipeline son grafos, no necesariamente perteneciendo a la misma clase, pero enriquecidos con características que hacen a su posterior representación

se puede concluir que es posible sintetizar nuevas metodologías concatenando pasos funcionales pertenecientes a otras metodologías y optando por una estrategia para abordar el problema de dibujar un grafo.

A continuación se describen brevemente los métodos más populares para abordar el problema de dibujar un grafo:

3.1.1.1 Topología-Forma-Métrica (Topology-Shape-Metrics)

Este método originalmente propuesto en [BAT86] [TAM87] y [TAM88] fue pensado para construir dibujos en grillas ortogonales, y permite el tratamiento homogéneo de un amplio rango de criterios estéticos y restricciones.

La idea básica del método es la caracterización de un dibujo ortogonal mediante tres propiedades fundamentales *Topología, Forma y Métrica*, definidas en término de las clases de equivalencia que los autores establecen entre los dibujos ortogonales de un mismo grafo. Los conceptos dados no sólo pueden utilizarse para caracterizar dibujos ortogonales, sino también dibujos de líneas poligonales en general.

La relación jerárquica existente entre los conceptos de topología, forma y métrica sugieren una generación paso a paso del dibujo, donde en cada paso se produce una representación intermedia. El método consiste en los siguientes pasos: En una primer etapa se planariza el grafo de entrada, luego se obtiene una representación ortogonal del mismo y por último se determinan las coordenadas de cada vértice y de los puntos de inflexión de las aristas. El objetivo normalmente es mantener mínima el área total del dibujo. También en este paso se eliminan los vértices ficticios insertados en el paso 1.

Las ventajas de los métodos de este tipo son su rapidez, su aplicabilidad en caso de tener que dibujar arcos rectos, ortogonales y aún poligonales, soportados por resultados teóricos sobre dibujado plano y extensible aún a grafos con gran cantidad de nodos. Sin embargo como desventajas cabe destacar que su implementación es realmente compleja, que es muy dificultoso extenderlo a 3D y que las transformaciones topológicas pueden alterar el mapa mental que tiene el usuario del grafo. La limitada capacidad de satisfacción de restricciones puede mirarse como ventaja y desventaja al mismo tiempo.

3.1.1.2 Fuerza-Dirigida

Los algoritmos pertenecientes a esta familia constituyen métodos intuitivos para crear dibujos de líneas rectas de grafos. Estos son bastante populares porque sus versiones básicas son fáciles de entender y codificar. Estos algoritmos simulan un sistema de fuerzas definido sobre el grafo de entrada y produce una configuración de energía localmente mínima.

También en este caso, la limitada capacidad de satisfacción de restricciones puede mirarse como ventaja y desventaja al mismo tiempo. Las restricciones pueden especificarse tanto por el posicionamiento de los vértices del grafo como a través de las fuerzas.

Este método es muy fácil del implementar, se le pueden adicionar heurísticas de manera sencilla, la evolución lenta del dibujo del grafo a su estado final preserva el mapa mental que tiene el usuario del grafo, puede extenderse fácilmente a 3D y en la práctica trabaja bien para grafos pequeños con estructura regular. Como desventajas caben citar su lentitud en ejecución y su dificultad para ser extendido en el dibujado de grafos que representarán sus arcos con líneas ortogonales y poligonales.

3.1.1.3 Jerárquico

Esta metodología propuesta en [SUG81],[CAR80] y [WAR77] es intuitiva y fue diseñada para representar dígrafos acíclicos con convenciones de dibujos con líneas poligonales y orientadas hacia abajo. Podemos distinguir los siguientes pasos: Se genera un dígrafo en niveles propios (proper layered digraph) en el que los vértices de G tienen asignadas capas o niveles N_1, N_2,..., N_h, tal que si (u,v) $\in$ A con u $\in$ L_i y v $\in$ L_j, entonces i = j +1. Luego se procede a minimizar el número de cruces entre las aristas de las distintas capas y por último se procede a la asignación de las coordenadas horizontales a los vértices preservando el ordenamiento asignado en el paso previo. El generar entonces el dibujo final y las aristas se dibujan como líneas rectas entre los vértices, pudiendo quedar algunas aristas representadas por una poligonal.

4 Dibujo del GI_{MCH}: Una aplicación del método Jerárquico

De las tres grandes familias de algoritmos utilizados para el dibujado de grafos (Paso de Planarización, Jerárquica y Sistemas de Fuerzas), teniendo en cuenta las características del GI_{MCH} se opta por la segunda.

La metodología de dibujo en capas o jerárquica es altamente intuitiva y tiene el atractivo que puede ser aplicada a cualquier dígrafo, independientemente de sus características teóricas.

Esta estrategia consta de los siguientes tres pasos:

- *Asignación de capas o niveles*: Asigna los vértices a capas horizontales, es decir en un sistema de ejes cartesianos *x:y* les asigna una coordenada *y*.
- *Reducción de cruces*: Ordena los vértices dentro de cada capa para minimizar el número de cruces entre los arcos.
- *Asignación de la coordenada horizontal*: Determina para cada vértice una coordenada x.

En caso de que el dígrafo de entrada contenga ciclos, es necesario realizar un paso previo de remoción de ciclos, invirtiendo temporalmente las aristas necesarias para que el dígrafo sea acíclico.

4.1 Asignación de Capas

El objetivo de este paso es asignar una coordenada-y a cada vértice. En otras palabras es transformar un dígrafo acíclico en otro por niveles. Esa transformación debe considerar los siguientes puntos:

- El dígrafo por niveles debe ser *compacto*. Esto es, su ancho y su alto deben ser lo más pequeños posible. Dado que la distancia entre capas es constante, el límite inferior de la altura está dado por el número de vértices en un paso desde un vértice fuente, a un sumidero.
- El dígrafo por niveles debe ser *propio*, esto es fácilmente logrado insertando 'vértices ficticios' a lo largo de las aristas largas de la siguiente manera: Se reemplaza cada arista (u,v) que se expanda a más de un nivel con el paso (v_1,v_2,…,v_{k-1},v_k) con v_1= u y v_k = v, siendo v_2,…,v_{k-1} vértices ficticios. La necesidad de esto está planteada por el paso de reducción de cruces, que se complica demasiado si tratamos con aristas que se expanden en más de un nivel.

— El número de vértices ficticios debe ser pequeño por varias razones, entre otras podemos citar:

- El tiempo que insumen los pasos siguientes es proporcional a la cantidad total de vértices.
- Los puntos de inflexión en las aristas ocurrirán en las coordenadas asociadas a este tipo de vértices. Dado que las inflexiones de las aristas disminuyen la legibilidad del dibujo es conveniente mantener su número bajo.
- La cantidad de vértices ficticios en un paso tiene relación directa con su longitud en término de las coordenadas-y. Esta probado que es más claro seguir visualmente el trazado de las aristas cortas que el de aquellas que superen una cierta longitud.

Para el caso particular del grafo integrador de un MCH la pertenencia de los vértices a un determinado nivel está dada por la jerarquía de los conceptos que representa cada vértice dentro del mapa. Por lo tanto el dígrafo de entrada ya tiene asignado a cada vértice un nivel o coordenada-y obtenido durante la etapa de construcción del mismo. Es importante notar que el nivel de un nodo (concepto) está dado por la ubicación física que el autor del MCH le dió a ese concepto dentro de la vista correspondiente

Como el dígrafo obtenido de esta manera ya tiene una asignación de niveles, lo que resta hacer en este paso es llevarlo a la forma de un dígrafo por niveles propio. Para lograr este objetivo debemos analizar cada arista y reemplazarla temporalmente por el paso conteniendo la cantidad de vértices ficticios necesarios para limitar las expansiones a 1.

4.2 Reducción de cruces:

En esta instancia lo que deseamos es construir un dibujo del dígrafo por niveles propio generado en el paso anterior con un número de cruces aceptable.

El número de cruces de las aristas entre dos niveles consecutivos, no depende de la ubicación precisa de cada vértice dentro de ese nivel, sino del orden relativo de los vértices dentro del mismo. Es decir es un problema de combinatoria (decidir el ordenamiento de los vértices en una capa) y no de geometría (asignarle la coordenada-x a cada vértice). Aún así, sigue siendo un problema extremadamente complejo. El problema de minimizar el cruce de aristas en un dígrafo por niveles es NP-Completo, incluso si hay solamente dos capas [Ead94]. El problema de reducir el número de cruces entre las distintas capas o niveles un dígrafo por niveles, es una generalización del *problema de reducir el número de cruces entre dos capas* L_i y L_{i+1}. Este problema ha sido denominado como el problema de cruces entre dos capas (two layer crossing problem) y se ha trabajado en extenso sobre el mismo [Ead86, Cat88, Jün97].

Para facilitar el abordaje del problema trabajaremos con un dígrafo de dos capas únicamente (two-layered digraph), $D=\{C_1, C_2, A\}$ donde los conjuntos C_1 y C_2 de vértices son disjuntos y el conjunto de aristas $A \in C_1 \times C_2$. Notaremos con x_I, y lo llamaremos ordenamiento de la capa i, al conjunto de coordenadas x asignadas a los vértices de la capa i. Además notaremos con *cruces*(D, x_1 , x_2) al número de cruces que aparecen en un dibujo del dígrafo D si a los vértices de las capas C_1 y C_2 se les asignan las coordenadas x según x_1 y x_2 respectivamente.

Podemos expresar al número de cruces en un dibujo de un dígrafo con dos capas, ordenadas según x_1 y x_2 respectivamente, de la siguiente manera:

$$cruces(D, x_1 , x_2) = \sum_{x_2(u)<x_2(v)} c_{uv}$$

donde c_{uv} es el número de cruces que se producen entre las aristas incidentes en u, con las aristas incidentes en v, cuando $x_2(u)<x_2(v)$ con $u, v \in C_2$.

4.2.1 Algoritmos para reducir el número de cruces

Dado que el problema del número de cruces es NP mucho se ha hecho para su resolución. Se han

desarrollado algoritmos que garantizan encontrar la solución óptima, que si bien no se asegura que terminen en tiempo polinomial han demostrado una perfomance aceptable en grafos de tamaño pequeño o mediano, considerando como tales a los que presenten hasta 15 nodos en su capa más chica [Jün97] [Mut97]. Otro aspecto a tener en cuenta es realmente cuál es el objetivo, obtener un dibujo con un mínimo número de cruces o la obtención de un dibujo legible, con algunos cruces más.

En respuesta a lo anterior se han desarrollado varios métodos heurísticos para resolver el problema en cuestión, entre los cuales encontramos [Dib99]:

4.2.1.1 Métodos de ordenamiento

Una característica común a este tipo de algoritmos es el cálculo previo de los números de cruces de cada capa. Dentro de ésta categoría podemos nombrar a:

- **Intercambio de Adyacentes**: El cual partiendo de un ordenamiento inicial para los vértices de una capa, los recorre intercambiando aquellos pares de vértices adyacentes (u, v) tal que $c_{uv} > c_{vu}$. Repite el proceso hasta que el número de cruces no varía.
- **Partir**: La idea aplicada en éste método es análoga a la usada en el QuickSort, se selecciona un vértice p que actúa como pivote dentro de cada capa y se clasifican los restantes a la izquierda o derecha de p según sea su número de cruces. Luego se invoca recursivamente con cada uno de los conjuntos de vértices obtenidos.

4.2.1.2 El método del Baricentro y el de la Mediana

Uno de los métodos más comunes es el denominado Baricentro, en el cual la posición de cada vértice u es calculada como el promedio de las posiciones de sus vecinos (en la capa C_1)

$$\text{Prom}(u) = 1 / \text{grado}(u) \sum_{v \in N_u} x_1(v)$$

En el caso de que dos vértices tengan el mismo valor Prom(u) se les asigna arbitrariamente algún valor próximo. No debemos olvidar que no se calcula un valor exacto de coordenada, sino que las posiciones relativas de los vértices dentro de cada capa.

El método de la Mediana es muy similar al anterior, la diferencia radica en que la posición asignada a un vértice u es determinada por la posición del vecino central de u. En el caso de haber dos vértices con mismo valor med(u)=med(v) se ubica a la izquierda el que tenga grado par, si los dos tuviesen grado par, se resuelve arbitrariamente.

Es fácil probar que ambos métodos dan ordenamientos con cero cruces cuando éste es posible. Sin embargo cuando una solución de cero cruces no existe, ninguno de los dos devuelve el ordenamiento óptimo. [Dib99]

4.2.1.3 Asignación de la coordenada horizontal

En esta instancia ya tenemos un grafo por niveles propio, y dentro de cada nivel un orden relativo de los vértices pertenecientes al mismo.

El objetivo de este paso es asignar a cada vértice un valor para su coordenada horizontal x, manteniendo el ordenamiento dentro de cada capa, pero también logrando que las aristas sean lo más rectas posibles. Es decir, se trata de asignar a cada nodo una coordenada x que permita reducir los ángulos de quiebre de las aristas a valores aceptables. Dado que los puntos de inflexión ocurren en los vértices ficticios introducidos para transformar el GI_{MCH} en un dígrafo por niveles propio, el objetivo de este paso se puede establecer como el siguiente problema de optimización: Dado un paso dirigido $p=(v_1,v_2,...,v_{k-1},v_k)$ siendo v_2, ..., v_{k-1} vértices ficticios. Si el paso fuera dibujado con una línea recta, entonces para $2 \leq i \leq k-1$, la coordenada x de un vértice v_i , $x(v_i)$, debería satisfacer

$$x(v_i) - x(v_1) = i - 1 / k - 1 \, (x(v_k) - x(v_1))$$

De donde el alejamiento de la coordenada $x(v_i)$ del ideal que lo colocaría sobre la recta $x´(v_i)$ lo podemos expresar de la siguiente manera:

$$d(v_i) = (x(v_i) – x´(v_i) = x(v_i) – [\ i – 1 / k – 1\ (x(v_k) - x(v_1)\ ) + x(v_1)]$$

Luego para lograr que la arista correspondiente a un paso p sea lo más recta posible deberíamos minimizar

$$\sum_{i=2}^{k-1} d(v_i) = \sum_{i=2}^{k-1} (x(v_i) – [\ i – 1 / k – 1\ (x(v_k) - x(v_1)\ ) + x(v_1)])^2$$

Esto lo podemos generalizar a todos los pasos que involucren vértices ficticios. Una cuestión importante a tener en cuenta es que para asegurar que no se perturbará el orden relativo obtenido en el paso de minimización de cruces de los vértices dentro de un mismo nivel, se deberá imponer algún tipo de restricción que deberán satisfacer todo par de vértices v, w con w a la derecha de v

$$x(w) – x(v) \leq \delta$$

Estamos frente a un problema de programación cuadrática que puede ser resuelto por métodos estándar.

5 Conclusiones

Esta presentación se trata de una propuesta de Informática Educativa que trasciende la mera simplificación de las tareas en los procesos de enseñanza-aprendizaje, y se presenta como una herramienta efectiva para la construcción del conocimiento, de acuerdo con un enfoque constructivista.

La creación del GI_{MCH} que se propone, amplía la riqueza del trabajo con los MCH como recurso valioso y comprobado para la representación de las ideas. Extiende su potencial al permitir trabajar expresamente sobre la relación correcta entre conocimientos recientemente aprendidos y los conocimientos previos o entre diversos enfoques de un mismo tema y fundamentalmente la posibilidad de poder visualizar con claridad el marco contextual de un determinado conocimiento.

Actualmente está en desarrollo la ampliación de la plataforma existente para el trabajo con MCH para incluir la generación automática del grafo integrador y su visualización. El diseño contempla la actualización automática y permanente del GI_{MCH} durante la etapa de autoría.

Bibliografía

[Aus78] Ausubel, D. P., Novak J. D. *Educational Psychology: A Cognitive View*. 2nd Ed. New York: Holt , Rinerhart and Winston. 1978.

[Bat86] C.Batini, E. Nardelli y R. Tamassia, *A Layout Algorithm for Data Flow Diagrams*, IEEE Trans. Softw. Eng., SE-12, no. 4, 1986.

[Bra95] Brandenburg, F. J. *Designining Graph Drawings by Layout Graph Grammars, Graph Drawing* (Proc. GD ´94), vol. 894 of Lecture Notes Comput. Sci., Springer-Verlag, 1995.

[Bru84] Bruner, J. *Acción, pensamiento y lenguaje*. Madrid. Alianza. 1984.

[Bru94] Bruner, J. *Realidad mental y mundos posibles*. Barcelona. Gedisa. 1994.

[Car80] M. J. Carpano, *Automatic Display of Hierchized Graphs for Computer Aided Decision Analysis*, IEEE Trans. Syst. Man Cybern., SMC-10, no. 11, 1980.

[Cat88] Catarci, T. *The Assigment Heuristic for Crossing Reduction in Bipartite Graphs*, Proc. 26th. Allerton Conference Commun. Control Comp., 1988

[Coll89] Coll, C. *Aprendizaje escolar y construcción del conocimiento* Ed. Paidós. 1989.

[Ded94] Dede, C. *Making the most of multimedia. Multimedia and Learning: A school leader's guide*. Alexandria, VA. NSBA.

[Den93] Dengler, E. M. Friedell, and J. Marks. *Constraint-driven diagram layout*. Proc. IEEE Sympos. On Visual Languages (VL. ´93), 1993.

[DIB92] G. Di Battista, R. Tamassia y I. Tollis, *Constrained visibility representation of Graphs*, Inform. Process. Lett., vol. 41, 1992.

[Dib99] Di Battista, P. Eades, G. Tamassia, R. y Tollis, I. *Graph Drawing: algoritms for the visualization of graphs*, Prentice Hall, 1999.

[Ead86] Eades, P. and D. Kelly. *Heuristic for reducing Crossing in Two-Layer Networks*. Ars. Combin. 1986

[Ead94] Eades, P. and Wormald, N. .*Edge crossings in drawings of bipartite graphs*. Algorithmica, 11, 1994.

[Jün97] Jünger, M. and Mutzel, *P. 2-Layer Straightline Crossing Minimization: Performance of exact and heuristics algorithms*. JGAA, 1, n. 1, 1997.

[Kam89] T. Kamada. *Visualizing Abstract Objects and Relations*. World Scientific Series in Computer Science, 1989.

[Lin95] Lin, T. y Eades, P. *Integration of declarative and algorithmic approaches for layout creation. Graph Drawing* (Proc. GD ´94), vol. 894, Lectures notes in Computer Science, Springer-Verlag, 1995.

[Leh93] Lehrer, R. *Authors of knowledge: Patterns of Hypermedia Design*. 1993. University of Wisconsin-Madison.

[Mar91] Marks, J. *A formal specification for network diagrams that facilitates automatic design.* Journal of Visual Languages and Computing, 1991

[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.

[Mut97] Mutzel, P. *An Alternative Method to Crossing Minimization on Hierarchical Graphs*.

[Nov84] Novak, Joseph - Gowin, D. Bob. *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.

[PCJ96] H. Purchase, R. Cohen y M. James, *Validating Graph Drawing Aesthetics*, Graph Drawing (Proc. GD ´95), Lecture Notes Comput. Sci. vol. 1027 , Springer-Verlag, 1996.

[Pur97] H. Purchase, *Wich aesthetics has greatest effects on human understanding*, Graph Drawing ´97, vol. 1353 of Lectures Notes in Computer Science, Springer-Verlag, 1997.

[Señ96a] 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, Departament of Computer Science an Engineering. New Orleans 1996.

[Señ96b] Señas, P., Moroni, N., Vitturini, M. y Zanconi, M.: *Combining Conceptual Mappings and Hypermedia*. ED-MEDIA96. Boston. 1996.

[Señ99] Señas, P., Moroni, *Computing Environments For Metalearning:Interconnecting Hypermedia Concept Maps*, ED-MEDIA2000. Canadá.

[SUG81] K. Sugiyama, S. Tagawa y M. Toda, *Methods for Visual Understandinf of Hierarchical Systems*, IEEE Trans. Syst. Man Cyber., SMC-11, no. 2, 1981

[Tam88] Tamassia, R., Di Battista, G. y Batín, C. *Automatic Graph Drawing and Readability of diagrams*. IEEE Trans. Syst. Man Cybern., SMC-18(1), 1988.

[WAR77] J Warfield, *Crossing Theory and Hierarchy Mapping*, IEEE Trans. Sys. Man. Cyber., SMC-7, no. 7, 1977.

[Zan98] Zanconi, M., Moroni, N., Vitturini, M., Malet, A., Borel, C. y Señas, P. *Tecnología computacional y meta-apreendizajes*. RIBIE-98. 1998.