

# Un guide sur la toile pour sélectionner un logiciel de tracé de graphes

Bruno Pinaud, Pascale Kuntz

► **To cite this version:**

Bruno Pinaud, Pascale Kuntz. Un guide sur la toile pour sélectionner un logiciel de tracé de graphes. Veille Stratégique scientifique et technologique (VSST), Sep 2004, Toulouse, France. pp.339–347, 2004. <inria-00335952>

**HAL Id: inria-00335952**

**<https://hal.inria.fr/inria-00335952>**

Submitted on 3 Nov 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Un guide sur la Toile pour sélectionner un logiciel de tracé de graphes

Bruno Pinaud(\*), Pascale Kuntz(\*\*)

{bruno.pinaud, pascale.kuntz}@polytech.univ-nantes.fr, bruno.pinaud@knowesis.fr

(\*) Knowesis SAS, Atlanpôle La Fleuriaye, BP 40703, Carquefou Cedex, France

(\*\*) Ecole Polytechnique de l'université de Nantes, Rue Christian Pauc, BP50609 44306  
Nantes Cedex 3

## Résumé

Les graphes permettent d'une part, en tant qu'objets combinatoires, de modéliser et d'analyser des systèmes de relations complexes entre des entités et d'autre part, de représenter ces relations sur des supports visuels afin de les rendre accessibles à un utilisateur non spécialiste. Si ces derniers sont rapidement devenus des outils privilégiés pour de nombreuses problématiques applicatives comme la découverte de relations non explicites en veille technologique, le choix d'une méthode de visualisation efficace reste encore très souvent une question ouverte pour l'utilisateur. Afin de guider ce dernier dans sa sélection, nous dressons une typologie succincte des principaux modes de représentation à savoir le tracé statique, le tracé interactif et le tracé des très grands graphes, et nous finissons par la présentation d'un nouveau site Web, dédié au référencement des logiciels de tracés de graphes. Le site est consultable à l'adresse : <http://hulk.knowesis.fr/GVSR>. Son originalité réside en particulier dans une présentation homogène des informations pertinentes mises en œuvre au travers d'un ensemble de fiches codées en XML.

## Mots-clés

Site web, annuaire, logiciels de tracé de graphes, XML

## Keywords

Website, directory, graph drawing software, XML

## Palabras clave

Sitio web, directorio, softwares de diseño gráfico, XML

# 1 Introduction

*Il faut imaginer dans sa tête des trucs qu'on appelle sommets, et pour toute paire de sommets soit une arête qui les joint, soit une non-arête qui les laisse sans joint : ceci est un graphe selon Berge [21].* Il s'ensuit de cette définition de l'un des plus célèbres maîtres du domaine, que les graphes sont les outils privilégiés pour modéliser un système de relations entre des entités. Ils permettent de caractériser précisément des propriétés d'un tel système via un arsenal combinatoire sophistiqué [2] tout en facilitant l'accès au profane à des structures complexes via notamment des représentations visuelles adaptées. Berge proposait de voir le dessin dans sa tête [21] ! Mais heureusement, depuis les premiers travaux de Knuth [13] dans les années 60, la représentation visuelle des graphes sur des supports plus partageables dans de nombreux domaines d'applications est devenue un domaine de recherche à part entière, animée par la communauté Graph Drawing, à l'intersection de la combinatoire [7] et de la visualisation de l'information [10].

Si il est en général aisé de justifier de l'utilisation d'un modèle de graphe dans le cadre de travaux de recherche, il est bien souvent difficile de choisir une méthode de tracé. L'ouvrage collectif récent dirigé par Mutzel et Jünger [16] peut certainement faciliter cette tâche en répertoriant une quinzaine de logiciels actuels de visualisation et de navigation parmi les plus performants. Cependant, ces logiciels puissants n'offrent évidemment pas tous une prise en main immédiate. De plus les fonctionnalités proposées peuvent dépasser de loin les besoins de l'utilisateur pour des applications pointues telles que celles rencontrées en veille technologique (*e.g.* [8,12]). Différents sites de la Toile présentent également des logiciels généraux ou spécialisés (*e.g.* [24], [25]). Mais, il s'agit souvent de listes de pointeurs peu organisées qui nécessitent une investigation importante pour l'analyse de leur contenu.

Ces limitations nous ont conduit à développer un site Web appelé GVSR (Graph Visualization Software References) qui répertorie sous une forme standard les logiciels accessibles sur la Toile. Ce site, qui naturellement se veut évolutif, présente actuellement une cinquantaine de logiciels très divers répartis en quatre catégories principales.

L'objectif de cette communication n'est évidemment pas de rentrer dans les détails des algorithmes mis en œuvre, mais plutôt de proposer sous une forme synthétique des points d'entrée pour un chercheur confronté au problème, lui évitant peut-être ainsi de redévelopper lui-même une antépénultième heuristique pas nécessairement parmi les plus efficaces. La section 2 rappelle les problèmes principaux soulevés par la représentation visuelle des graphes. La section 3 présente le site développé.

## 2 Les problèmes de tracé

Etant donné un graphe  $G=(V,E)$  avec un ensemble  $V$  de sommets et un ensemble  $E$  d'arêtes, le problème générique de tracé le plus classique -qualifié de tracé statique- consiste à dessiner  $G$  sous une forme "intelligible" sur un support standard bidimensionnel voire depuis peu tridimensionnel [14]. Plus récemment, l'évolution du recueil dynamique des données et la croissance de leur volume ont stimulé l'intérêt accordé respectivement à deux problèmes : le tracé dynamique où, à chaque instant  $t$ , un graphe  $G_t$  doit être dessiné en tenant compte du tracé du graphe  $G_{t-1}$  proposé à l'instant précédent, et le tracé des très grands graphes qui nécessitent

de nouvelles approches, les supports de tailles standardisées ne permettant plus de dessiner ces graphes lisiblement dans leur intégralité. En complément, les logiciels de dessin qui considèrent les graphes comme des substrats de visualisation de l'information ont conduit au développement de nouvelles fonctionnalités visant à faciliter la navigation et la recherche d'informations.

## 2.1 Le tracé statique

La qualité du dessin est décisive pour l'appropriation de la représentation par l'utilisateur [17]. Pour préciser cette notion délicate, qui reste *in fine* subjective, on retient généralement quatre concepts de base détaillés dans la monographie de Di-Battista et al. [7] :

- la convention de tracé, souvent inhérente aux pratiques du domaine d'application, qui spécifie les règles géométriques de lecture du tracé ; par exemple, dans une représentation polygonale, chaque arête est représentée par une ligne polygonale ;
- les contraintes du support et de l'oeil humain qui imposent notamment des écarts minimums à respecter entre les entités géométriques (points, boîtes, ...) représentant les sommets et les courbes représentant les arêtes ;
- les critères « esthétiques » qui définissent les propriétés à satisfaire pour faciliter l'intelligibilité. Ces critères sont définis par des contraintes combinatoires : minimisation du nombre de croisements d'arêtes, minimisation de la somme des longueurs des arêtes ou de la longueur de l'arête maximale, minimisation des coudes dans certains tracés, ... La plupart de ces critères ne peuvent cependant pas être satisfaits simultanément. Des travaux récents en psychologie cognitive ont montré que la réduction des croisements est le critère prépondérant pour la lisibilité et la mémorisation [18].
- les contraintes « sémantiques » associées à l'interprétation des composantes du graphe ; par exemple, des proximités sémantiques doivent être respectées dans le positionnement des sommets.

## 2.2 Le tracé dynamique

Lorsque l'on introduit des fonctionnalités permettant l'ajout ou la suppression dynamique de sommets et d'arêtes, la lecture des dessins dans un contexte évolutif conduit à des contraintes additionnelles.

En effet, contrairement au tracé statique où les contraintes liées au tracé sont connues à l'avance et ne changent pas dans le temps, le tracé dynamique doit prendre en compte le fait que l'utilisateur connaît le tracé présenté à l'instant  $t-1$ . Ce tracé doit donc être pris en compte lors de la conception du nouveau tracé. Donc, à chaque instant  $t$ , le nouveau tracé doit non seulement être intelligible selon les critères énoncés à la section précédente mais il doit, de plus, permettre une transition aisée avec le tracé présenté à l'instant précédent. Ceci permet à l'utilisateur d'éviter de perdre son temps et une énergie cognitive importante pour la découverte du nouveau tracé.

Parmi les nombreuses méthodes adaptées au tracé dynamique, une des plus simples à mettre en œuvre pour conserver un tracé intelligible, et faciliter la transition interprétative entre les instants  $t$  et  $t-1$ , consiste à simplement animer cette transition sur l'écran de l'utilisateur. L'inconvénient de cette méthode est qu'elle est très coûteuse en temps de calcul, et doit être suffisamment lente pour permettre à l'utilisateur de voir et d'enregistrer les différents changements opérés sur le graphe.

Pour assurer un certain intérêt au tracé dynamique, une bonne interactivité avec l'utilisateur est nécessaire. Contrairement à la méthode précédente, qui permet une interactivité assez pauvre du fait de la vitesse réduite des animations, une autre approche consiste à préserver au mieux la stabilité des tracés. Ceci passe par la préservation de la vision du graphe par l'utilisateur à  $t-1$ , c'est à dire sa carte mentale, en limitant les perturbations apportées sur le nouveau tracé par rapport aux précédents [6,9]. La stabilité est une notion complexe qui dépend des caractéristiques géométriques et combinatoires du tracé, mais aussi des facultés de perception et de mémorisation de l'utilisateur. Cependant, deux facteurs prédominants semblent se dégager :

- les positions des sommets doivent changer le moins possible. Leur stabilité semble plus importante que celle des arêtes : les sommets servent de repères spatiaux alors que les arcs sont essentiellement utilisés pour découvrir des relations entre des sommets déjà localisés ;
- l'ordre relatif des composantes du graphe dans le repère géométrique adopté doit être conservé tant que cela est possible car l'utilisateur repère les composants les uns par rapport aux autres.

Pour rendre opérationnelles ces contraintes de stabilité et ainsi mesurer le degré de conservation de la carte mentale de l'utilisateur entre les tracés de deux graphes successifs  $G_{t-1}$  et  $G_t$ , différentes métriques basées sur des indices de similarité ont été proposées [5]. La difficulté du tracé dynamique est qu'il faut en permanence trouver un tracé réalisant un bon compromis entre la préservation de la carte mentale de l'utilisateur, la lisibilité du tracé et le temps de calcul (en général, il faut éviter d'avoir plus de 200ms entre deux tracés successifs). Cependant, si des heuristiques ont été proposées pour les grandes familles de tracés (orthogonaux, en niveaux, ... voir [4] pour une synthèse récente), le problème reste encore très ouvert et peu de logiciels intègrent actuellement la gestion dynamique de façon efficace.

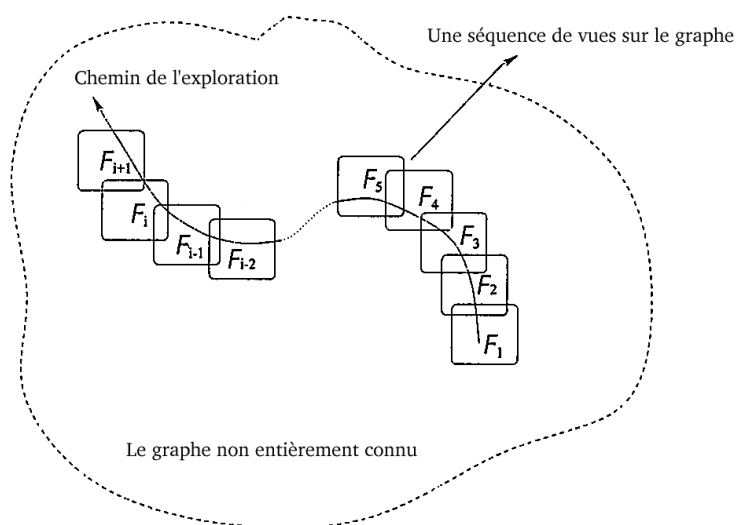
## 2.3 Le tracé de très grands graphes

L'explosion de la quantité de données à traiter conduit à de nouveaux problèmes de représentation. Deux exemples, associés à des problématiques de veille sur la Toile, permettent de concrétiser les nouveaux ordres de grandeur mis en jeu. Les graphes de contacts ou de co-citations peuvent porter maintenant sur des centaines de milliers de sommets [26]. Pour estimer des paramètres décrivant la structure des relations sur la Toile, Barabasi *et al.* travaillent sur des graphes échantillons de 300 000 documents et 1 500 000 liens (estimé à environ 0.3% de la Toile) [19]. De telles structures ne peuvent pas être affichées en entier sur un écran statique tout en restant lisible et compréhensible par l'utilisateur.

Plusieurs familles de méthodes de visualisation de ces vastes structures existent et se distinguent par la connaissance du graphe *a priori*. Si le graphe est entièrement connu la

technique des « fisheyes » [22] consiste à dessiner un sous-graphe le plus clairement possible en se focalisant sur un sommet et son voisinage puis à appliquer une distorsion sur l'affichage du reste du graphe pour permettre à l'utilisateur d'en voir la globalité. L'utilisateur a donc des informations sur la position de ce sous-graphe par rapport à l'ensemble du graphe et peut exploiter au mieux les informations intéressantes pour lui sur la partie clairement dessinée. Une autre grande famille de méthodes exploite les techniques de la recherche d'information qui utilise souvent le leitmotiv de Shneiderman [23] comme référence : « *Overview first, zoom and filter, then details -on demand-* ». Cette approche consiste tout d'abord à proposer à l'utilisateur une vue synthétique de la globalité des données, puis à lui permettre de se focaliser sur un sous-ensemble (par clustering ou fragmentation bien souvent), et enfin de rendre accessible les caractéristiques précises d'une donnée sélectionnée. Et ce, dans un processus répétitif qui s'arrête lorsque l'utilisateur a acquis suffisamment d'informations pour répondre à des besoins non nécessairement spécifiés préalablement. La difficulté est ici d'appliquer des transformations qui restent compréhensibles par l'utilisateur pour éviter de trop perturber sa carte mentale. D'autres méthodes comme des représentations en 3D (par exemple les « *cone trees* » [20]) permettent une approche sensiblement différente mais la densité d'informations visibles par l'utilisateur à chaque instant augmente sensiblement.

Sur des très gros graphes (100 000 éléments au minimum jusqu'à quelques millions), les méthodes précédentes ne sont pas applicables car il peut arriver que toute la structure ne soit pas entièrement connue (mémoire insuffisante, complexité des calculs à effectuer, ...). Huang et al. [11] ont introduit en 1998 une technique appelée « *Online Graph Drawing* » qui permet à l'utilisateur de se concentrer sur un sous-graphe entièrement connu appelé « *logical frame* » et de naviguer dans le grand graphe en déplaçant cette fenêtre de visualisation (voir figure 1). Ces différentes fenêtres de visualisation respectent bien évidemment les contraintes de tracé statique et les transitions entre les fenêtres sont conçues pour respecter au mieux la carte mentale de l'utilisateur.



**Figure 1. Parcours d'un graphe non entièrement connu avec la méthode des "logical frames".**

Les logiciels permettant de correctement traiter le problème de visualisation des très grands graphes considèrent souvent conjointement un problème de navigation au sein du graphe et un autre de filtrage d'information. Par exemple, dans Nicheworks, Wills [27] utilise des propriétés intrinsèques aux éléments du graphe pour en effectuer un affichage incrémental en ne montrant d'abord que les éléments les plus importants puis en affinant petit à petit la figure. Munzner [15] dans ses différentes réalisations préfère spécialiser ses algorithmes en fonction des données à traiter. Dans Tulip, Auber [1] préfère calculer une métrique sur les sommets, qui est indépendante des données représentées et en fonction de cette métrique, il réalise un affichage incrémental du graphe. Cela permet de toujours avoir un tracé visuellement proche du tracé complet du graphe même avec un nombre très réduit d'éléments.

### 3 Un annuaire de logiciels sur la Toile

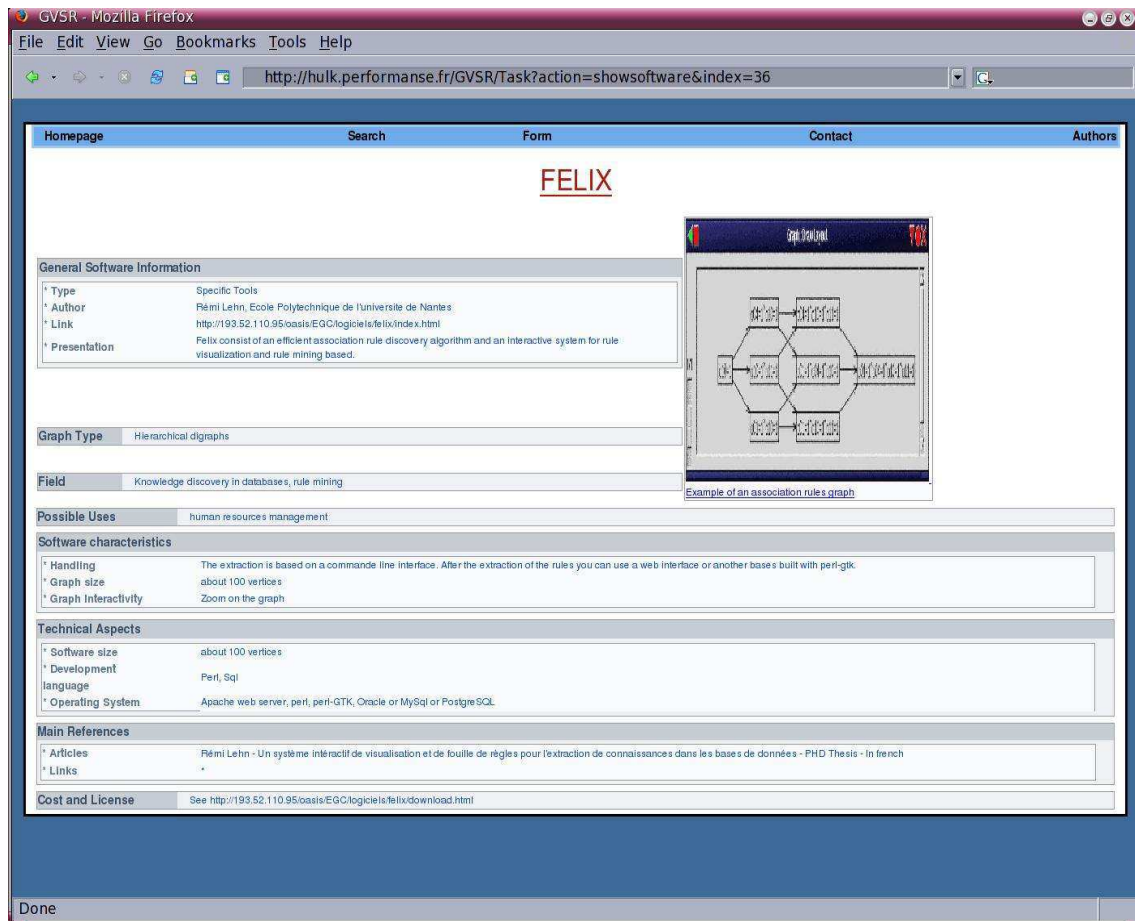
Pour guider l'utilisateur dans la sélection d'un logiciel spécifiquement adapté à ses besoins, nous avons développé un annuaire disponible sur la Toile. Cet annuaire, hébergé par la société Knowesis SAS<sup>1</sup>, disponible à l'adresse <http://hulk.knowesis.fr/GVSR>, permet d'une part d'accéder aisément à la consultation via une page d'accueil catégorisant les logiciels recensés et d'autre part de créer et de stocker dynamiquement de nouvelles fiches-logiciels au format XML de façon native. Cet objectif allié à des besoins de maintenance évolutive justifie notre choix d'un système de gestion de base de données natives XML et d'un serveur d'application permettant de générer les pages dynamiquement.

#### 3.1 Description du contenu

La page d'accueil du site regroupe les logiciels dans 4 catégories : les éditeurs de graphes spécialisés dans un domaine particulier qui permettent de construire soi-même interactivement des graphes (« *Specific Tools* »), les logiciels de tracés généralistes (« *Visualization Tools* »), les logiciels spécialisés dans les ontologies (« *Ontology software* »), et les bibliothèques permettant de se développer facilement et rapidement son propre logiciel en utilisant les implémentations d'algorithmes de tracé présentés dans la bibliothèque (« *Libraries* »). Un menu est toujours disponible dans la barre bleue située en haut de la fenêtre. Elle permet par exemple en cliquant sur « *Form* » d'accéder à un formulaire permettant de proposer un nouveau logiciel. Une fois le formulaire rempli au mieux et validé, l'administrateur du site sera averti du dépôt d'une nouvelle fiche et devra la valider avant que la nouvelle fiche-logiciel ne soit accessible aux utilisateurs du site.

---

<sup>1</sup> Knowesis SAS : <http://www.knowesis.fr>



**Figure 2 : Capture d'écran d'une fiche-logiciel.**

Pour l'utilisateur, la spécificité de l'annuaire GVSR repose sur une lecture homogène de la description de chaque logiciel (voir l'exemple d'une fiche sur la figure 2). Ainsi, chaque fiche-logiciel se décompose en huit parties (neuf pour les librairies) auxquelles s'ajoute une capture d'écran permettant de visualiser sur un exemple les caractéristiques majeures du tracé proposé. Les différentes parties sont :

- *General software information* : une description des informations générales sur le logiciel, à savoir sa catégorie, les différents auteurs (entreprises ou personnes), un lien vers le site web associé et une brève présentation générale.
- *Graph type* : le (ou les) type(s) de graphe(s) manipulé(s).
- *Field* : quelques champs d'applications possibles.
- *Possible uses* : quelques exemples connus d'applications.
- *Software characteristics* : les caractéristiques détaillées du logiciel avec le principe succinct d'utilisation du logiciel, l'ordre maximal des graphes et les fonctions interactives de manipulation des graphes tracés.



- *Technical aspects* : les aspects techniques tel que la taille du logiciel, les langages de développement utilisés et l'environnement d'utilisation (systèmes d'exploitations, logiciels annexes nécessaires, type d'ordinateur nécessaire, ...).
- *Main references* : une liste de pointeurs vers les principales références connues (articles et sites web).
- *Cost and license* : le coût éventuel et la licence d'utilisation.
- *Main applications* : réservé uniquement pour les librairies. Ensemble de liens vers des applications basées sur cette librairie.

### 3.2 Description technique

Les fiches-logiciels sont représentées en utilisant le langage XML qui grâce à son format semi-structuré, nous permet de pouvoir facilement faire évoluer la structure d'une fiche. Pour le stockage de ces fiches, nous utilisons la version 1.0 du système de gestion de bases de données (SGBD) « *Xindice* » de la fondation *Apache*<sup>2</sup> développé en Java. Ce SGBD est un des rares systèmes gérant correctement les documents XML de façon native. Il nous permet de directement stocker sous formes de collections tout document XML bien formé. Comme pour tous les SGBD, il est possible de faire des requêtes pour trouver des documents en utilisant le langage d'interrogation de XML : XPATH<sup>3</sup>. Une implémentation du langage XUPDATE<sup>4</sup> permettant de mettre à jours les documents XML de façon automatique est aussi disponible.

Pour transformer les fiches XML en HTML nous avons utilisé le serveur d'applications de la fondation apache : *Tomcat*<sup>5</sup>. Développé en Java, *Tomcat* permet une interaction aisée avec *Xindice* et grâce à la technologie « *Java Servlet Pages* » (JSP), l'ensemble des pages du site est dynamiquement généré sur le serveur. Le client ne reçoit que du HTML très simple. L'utilisation de XML devient complètement transparente pour le client et on s'assure d'un bon fonctionnement du site sur la majorité des navigateurs du marché.

L'administration du site (gestion des fiches, des catégories, du contenu des fiches, ...) s'effectue aussi en utilisant des pages spécifiques qui permettent d'éviter de manipuler directement les fiches au sein du SGBD.

## 4 Perspectives

Le site Web, consacré au référencement des logiciels de visualisation de données par des graphes, que nous avons développé est basé sur les technologies les plus utilisées actuellement. Outre évidemment la mise à jour du site par l'ajout de nouvelles références et l'actualisation de celles existantes, des évolutions de la version actuelle du site sont prévues à court terme ; elles concernent d'une part, l'élargissement des thématiques visées, et d'autre part, des améliorations technologiques.

---

<sup>2</sup> Apache-Xindice : <http://xml.apache.org/xindice>

<sup>3</sup> <http://www.w3.org/TR/xpath>

<sup>4</sup> <http://xmldb-org.sourceforge.net/xupdate/>

<sup>5</sup> Apache-Tomcat : <http://jakarta.apache.org/tomcat>

Au niveau des références, nous nous sommes ici volontairement focalisés sur les représentations bidimensionnelles, tout en notant que certains logiciels référencés proposent également des représentations tridimensionnelles performantes. Ce domaine de recherche étant en plein essor actuellement en visualisation de l'information, les nouveaux logiciels en cours de développement devront être intégrés. En parallèle, l'accessibilité récente des outils de développement en réalité virtuelle conduisent à l'émergence de nouveaux supports pour la représentation de graphes : les relations ne sont plus représentées par des « courbes » comme dans les schémas classiques mais par des métaphores qui ont, en outre, pour objectif de faciliter la navigation [3].

Au niveau technologique, pour faciliter l'utilisation du site et la recherche d'un logiciel, un module effectuant des recherches au sein des fiches-logiciels basé sur le langage XPATH est en projet. La croissance des références devra aussi conduire très certainement à une amélioration de l'ergonomie du site par l'intégration de représentations cartographiques adaptées.

## 5 Bibliographie

- [1] D. Auber. *Tulip: A huge graphs visualization framework*. In *Graph Drawing software*, pages 105-126. Springer-Verlag, 2003.
- [2] C. Berge. *Graphes et hypergraphes*. Dunod, 1973.
- [3] J. Blanchard, F. Guillet, and H. Briand. *A user driven and quality-oriented visualization for mining association rules*. In *Proc. of the IEEE Int. Conf. on Data Mining*. IEEE Computer Society Press, 2003.
- [4] J. Branke. *Dynamic graph drawing*. In M. Kaufmann and D. Wagner, editors, *Drawing graphs: methods and models*, pages 228-246. Springer, 2001.
- [5] S. Bridgeman and R. Tamassia. *A user study in similarity measures for graph drawing*. *J. of Graph Algorithms and Applications*, 6(3):225-254, 2002.
- [6] R.F. Cohen, G. Di-Battista, R. Tamassia, and I.G. Tollis. *A framework for dynamic graph drawing*. In *Proc. of ACM Symp. on Computational Geometry*, pages 261-270. ACM Press, 1992.
- [7] G. Di-Battista, P. Eades, R. Tamassia, and I.G. Tollis. *Graph drawing - Algorithms for the visualization of graphs*. Prentice Hall, 1999.
- [8] B. Dousset, T. Dkaki, and J. Mothe. *Veille scientifique et technique sur internet*. In *conférence sur les systèmes d'information élaborée : Bibliométrie - Informatique stratégique – Veille Technologique*, pages 12-17. 1997.
- [9] P. Eades, W. Lai, K. Misue, and K. Sugiyama. *Preserving the mental map of a diagram*. In *Proc. of Compugraphics*, pages 24-33, 1991.
- [10] I. Herman, G. Melançon, and S. Marshall. *Graph visualization and navigation in information visualization: a survey*. *IEEE Trans. on Visualization and Computer Graphics*, 8 (1), 2000.

- [11] M. L. Huang, P. Eades, and J. Wang. *Online animated graph drawing using a modified spring algorithm*. Australian Computer Science Comm.: Proc. Australian Computer Science Conf., ACSC, 20(1):17-28, 1998.
- [12] S. Karouach and B. Dousset. *Visualisation de relations par des graphes interactifs de grande taille*. ISDM: J. of Information Sciences of Decision Making, 6(57):253-264, 2003.
- [13] D.E. Knuth. *Computer draw flowcharts*. Communications of the ACM, 6(9):555-563, 1963.
- [14] B. Landgraf. *3D graph drawing*. In M. Kaufmann and D. Wagner, editors, Drawing graphs: methods and models, pages 172-192. Springer, 2001.
- [15] T. Munzner. *Interactive Visualization of large graphs and networks*. PhD thesis, 2000.
- [16] P. Mutzel and M. Jünger. *Graph drawing software*. Springer Verlag, 2003. ISBN: 3-540-00881-0.
- [17] H. Purchase. *Effective information visualisation: a study of graph drawing aesthetics and algorithms*. Interacting with computers, 13(2):127-145, 2000.
- [18] H.C. Purchase. *Which aesthetic has the greatest effect on human understanding ?* In Proc. Symp. Graph Drawing' 95, pages 248-261. LNCS, 1998.
- [19] A. Reka, H. Jeong, and A.-L. Barabasi. *Diameter of the world-wide web*. Nature, 401:130-131, 1999.
- [20] G.G. Robertson, J.D. Mackinlay, and S.K. Card. *Cone trees: Animated 3D visualizations of hierarchical information*. In SIGCHI, Conf. on Human Factors in Computing Systems, pages 189-194. ACM, 1991.
- [21] P. Rosenstiehl. *Claude Berge, ses graphes et hypergraphes*. Mathématiques et Sciences Humaines, 160:7-12, 2002.
- [22] M. Sarkar and M. H. Brown. *Graphical fisheye views*. Communications of the ACM, 37(12):73-84, 1994.
- [23] B. Shneiderman. *The eyes have it: A task by data type taxonomy in information visualization*. In IEEE Symp. on Visual Languages, pages 336-342. IEEE Press, 1996.
- [24] *Graph drawing tools and related work*, 1997.  
<http://rw4.cs.uni-sb.de/users/sander/html/gstools.html>.
- [25] *Algorithm engineering research group*, 2000.  
<http://www.dia.uniroma3.it/research/ACG.html>.
- [26] B.C. Tjaden and G. Wasson. *The oracle of bacon*. Technical report, <http://www.cs.virginia.edu/misc/news-bacon.html>, 2000.
- [27] G.J. Wills. *Nicheworks: Interactive visualization of very large graphs*. In Proc. of the Symp. Graph Drawing, pages 403-414. LNCS 1353 - Springer Verlag, 1997.