

# **Construction automatique de mondes virtuels représentant le comportement dynamique d'un réseau.**

P. Abel(\*), P. Gros(\*), D. Loisel(\*), J.P. Paris (\*\*), C. Russo Dos Santos(\*)

(\*) Institut Eurécom, Multimedia Communications Dpt.

(\*\*) CNET France Télécom.

{abel,gros,loisel,russo}@eurecom.fr,  
jeanpierre.paris@cnet.francetelecom.fr

## **Résumé**

Ce papier se concentre sur les problèmes posés par la construction automatique et dynamique de mondes virtuels à partir d'informations de gestion de réseau collectées en temps réel. Le système décrit a été conçu dans le cadre du projet CyberNet. Il s'agit d'un outil interactif de visualisation 3D destiné à démontrer l'apport la réalité virtuelle dans la visualisation des informations, notamment celles du management de réseau. Dans cet article nous exposons les principes de structuration des informations collectées aboutissant à la définition de services. Ensuite nous présentons la méthode d'association de ces services à des composants graphiques. Ce papier met également en évidence les problèmes liés à la dynamique des informations.

## **1. Introduction**

Aujourd'hui, la supervision d'un réseau ne doit plus être limitée à la visualisation topologique ou géographique du réseau et de ses éléments. La supervision doit souvent se faire dans le cadre de l'administration de services offerts par le réseau (routage, mail, application client/serveur, etc.). L'état d'un service peut généralement être décrit à l'aide d'un ensemble de données distribuées sur différents éléments du réseau. Pour contrôler le bon fonctionnement ou la qualité d'un service, les opérateurs ne disposent actuellement que d'outils d'interrogation des différents éléments, matériels ou logiciels, constitutifs du réseau et ils sont contraints d'analyser un gigantesque flot continu d'information. Le rôle de l'opérateur est alors de synthétiser/extraire les informations pertinentes en fonction du service supervisé. Cette tâche est d'autant plus complexe que ces données sont dynamiques.

A partir de ce constat, nous pensons que l'utilisation de la réalité virtuelle peut apporter une vision synthétique et globale de l'information et doit permettre à l'utilisateur d'appréhender les états et les évolutions du système qu'il observe. L'idée de ce projet est d'exploiter à la fois le nombre important de paramètres visuels disponibles (forme, position, couleur, texture, éclairage des objets,...) et les capacités d'animation en temps réel de ces différents paramètres afin de représenter dans un monde virtuel cohérent un vaste volume d'informations dynamiques. Le monde virtuel représentant le service étudié doit être construit automatiquement et évoluer en fonction des données récoltées.

Dans un premier temps, nous étudions les principes de structuration de l'information collectée sur le réseau. Ensuite, nous décrivons les différents composants graphiques et leurs paramètres visuels exploités pour visualiser l'information. Pour terminer, le processus de construction automatique des mondes virtuels à partir des informations structurées est présenté.

## 2. Collecte et structuration des informations

Notre objectif est de construire des représentations de services qui seront ensuite visualisées. Nous présentons ci-après la méthode utilisée pour structurer les données collectées et construire la représentation du service. Pour illustrer ce mécanisme, nous utiliserons l'exemple très simple de la supervision des processus et des utilisateurs d'une station de travail.

### 2.1. Collecte

Les informations brutes sont collectées de façon continue sur le réseau en interrogeant les logiciels, les machines et les éléments actifs utilisés. Les méthodes de collecte utilisées sont les protocoles standards (SNMP, MIB) qui utilisent un modèle « pull ». Afin de présenter des informations reflétant l'état actuel du réseau et de ne pas surcharger le réseau avec de trop nombreuses interrogations, nous avons développé des agents, utilisant les protocoles susnommés, qui implémentent un modèle « push » [9]. Ces agents, placés près des éléments à observer, sont chargés d'envoyer les données à leurs abonnés selon des politiques préétablis .

### 2.2. Entités

Les abonnés sont chargés de structurer les informations qu'ils reçoivent sous la forme d'**entités** logiques correspondant aux différents éléments (physiques ou conceptuels) que l'on désire analyser. Ces entités sont définies, typées et enregistrées pour être utilisées ultérieurement par les services. Elles contiennent une collection d'informations en rapport avec leurs types. Dans notre exemple de supervision des processus et des utilisateurs d'une station de travail , nous devons créer des entités logiques de type utilisateur, machine et processus (cf. figure 1). Les abonnés mettent continuellement à jour les champs des entités à partir des informations qu'ils reçoivent.

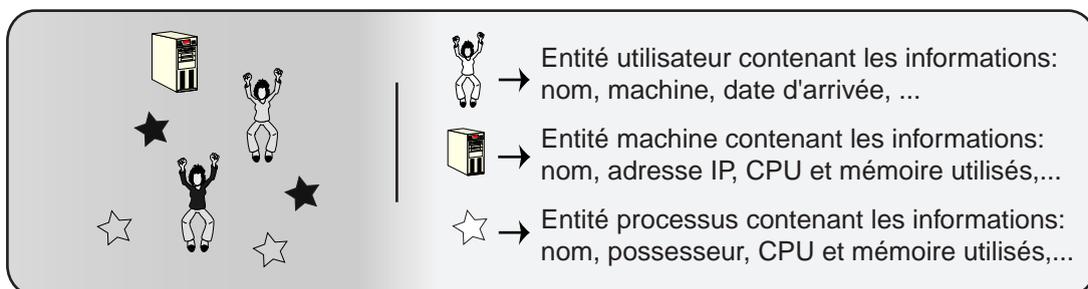


Figure 1 : Entités créées pour la supervision des utilisateurs et des processus d'une machine.

### 2.3. Relations & graphe d'entités

Pour construire notre service nous avons besoin de regrouper les entités constituant celui-ci. Lorsque des liens fonctionnels relient différentes entités (par exemple une relation client/serveur entre deux machines ou plus simplement une relation entre 2 processus s'exécutant sur une même machine), ces liens doivent être modélisés en utilisant des **relations**. Une relation permet donc de modéliser une liaison logique entre deux ou plusieurs entités.

Un graphe d'entités est alors réalisé dont les nœuds sont des entités et les branches des relations. Celui-ci est très complexe car il existe de nombreuses relations entre les multiples entités créées à partir des informations collectées sur le réseau. Puisque le système étudié évolue, la structure de ce graphe évolue dans le temps en fonction de l'apparition ou de la disparition d'entités ou de relations.

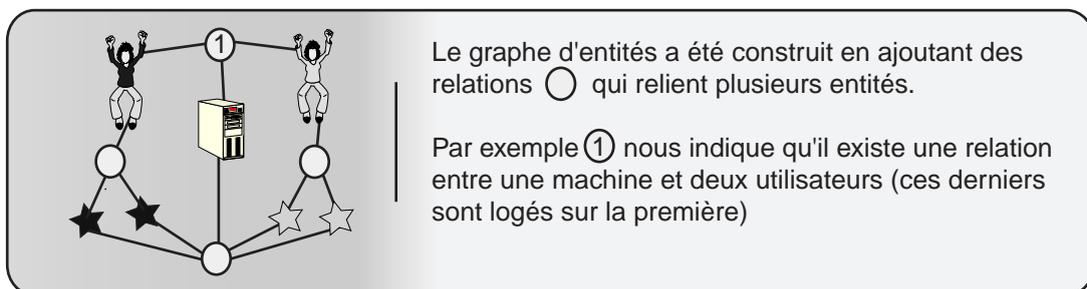


Figure 2 : graphe d'entités

## 2.4. Services

Notre but est de construire des services à l'aide des entités que nous avons créées et organisées. Pour effectuer cette tâche, on extrait des sous graphes du graphe d'entités qui reflètent des services spécifiques que l'on désire visualiser [1]. Cette extraction s'effectue en sélectionnant des entités ayant certains types de relations et sans détruire le graphe initial. La définition de ces services offre un autre avantage : celui de permettre à une entité d'être présente dans plusieurs services mais aussi au sein d'un même service ; ceci est important car une même entité possède souvent plusieurs types de relations et joue donc plusieurs rôles qui doivent être visualisés de façon distincte. Par exemple, dans le service NFS une même machine peut être à la fois client et serveur. Nous verrons par la suite qu'il est nécessaire que ces services soient des graphes non cyclique afin de les visualiser en trois dimensions.

Le caractère dynamique des informations ayant une influence sur le graphe, les évolutions sont répercutées sur les sous graphes extraits reflétant les services.

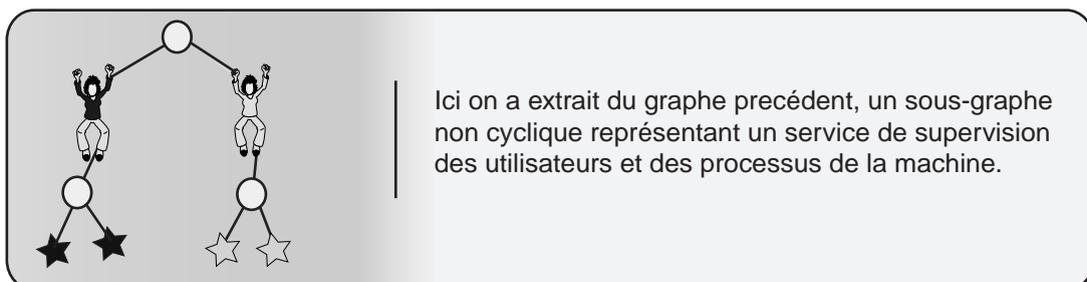


Figure 3 : graphe d'un service

### 3. Composants graphiques

La construction du monde virtuel représentant l'état du service se fait en associant des composants graphiques aux entités et aux relations qui forment le service. Nous avons utilisé deux types de composants graphiques : les glyphes 3D et les gestionnaires de placement.

#### 3.1. Glyphes 3D

Les **glyphes 3D** sont des objets tridimensionnels prédéfinis possédant des paramètres visuels (forme, couleur, taille, animation, etc. de l'objet ou de l'une de ses parties) que l'on peut associer aux différentes informations stockées au sein d'une ou plusieurs entités. Ce sont en quelque sortes des icônes 3D animées. Ce concept a été utilisé pour la première fois en 2D [2]. Toute modification d'une valeur d'une entité entraîne la modification du ou des paramètres visuels liés à cette valeur. La complexité d'un glyphe 3D peut se mesurer en terme du nombre de paramètres visuels qu'il propose et donc du nombre d'information qu'il permet de visualiser. Les glyphes 3D peuvent avoir des niveaux de complexité très variables allant d'un simple cylindre à une carte colorée des départements français. [3] décrit un exemple de glyphe 3D utilisé afin de visualiser le trafic d'un réseau.

#### 3.2. Gestionnaires de placement

Les **gestionnaires de placement (GP)** regroupent des composants graphiques, c'est à dire soit des glyphes 3D ou soit d'autres GPs. Ils ont pour fonction de placer géométriquement et d'orienter leurs éléments à partir de politiques de placement prédéfinis. A titre d'exemple, les GPs peuvent organiser leurs éléments en orbite autour d'un centre (modèle héliocentrique), ou en file le long d'un axe, ou encore en ligne/colonne sur un plan, voire sous la forme d'un "cone tree" [4] (cf. figure 4). Un GP peut également ajouter des éléments visuels comme une boîte semi-transparente [5] qui englobe ses enfants afin d'améliorer la visualisation. Placer des composants graphiques proches les uns des autres donne à l'utilisateur l'impression qu'ils partagent des propriétés [6] [7], c'est pourquoi les GPs sont en général associés aux relations, mais la position et l'orientation des composants graphiques peut aussi permettre de visualiser des informations autres que les relations (la distance entre 2 entités peut, par exemple, permettre de visualiser une valeur).

La conception des GPs doit prendre en compte l'aspect dynamique du système. En effet, chaque modification du service peut avoir un effet non négligeable en particulier lorsque le nombre ou la taille des composants gérés par un GP sont modifiés, car dans ce cas le GP doit alors recalculer la position de ces composants et propager les modifications à ses parents et à ses fils. De ce fait, une modification mineur peut parfois avoir des répercussions importantes sur le monde dans sa globalité et déstabiliser l'utilisateur [4]

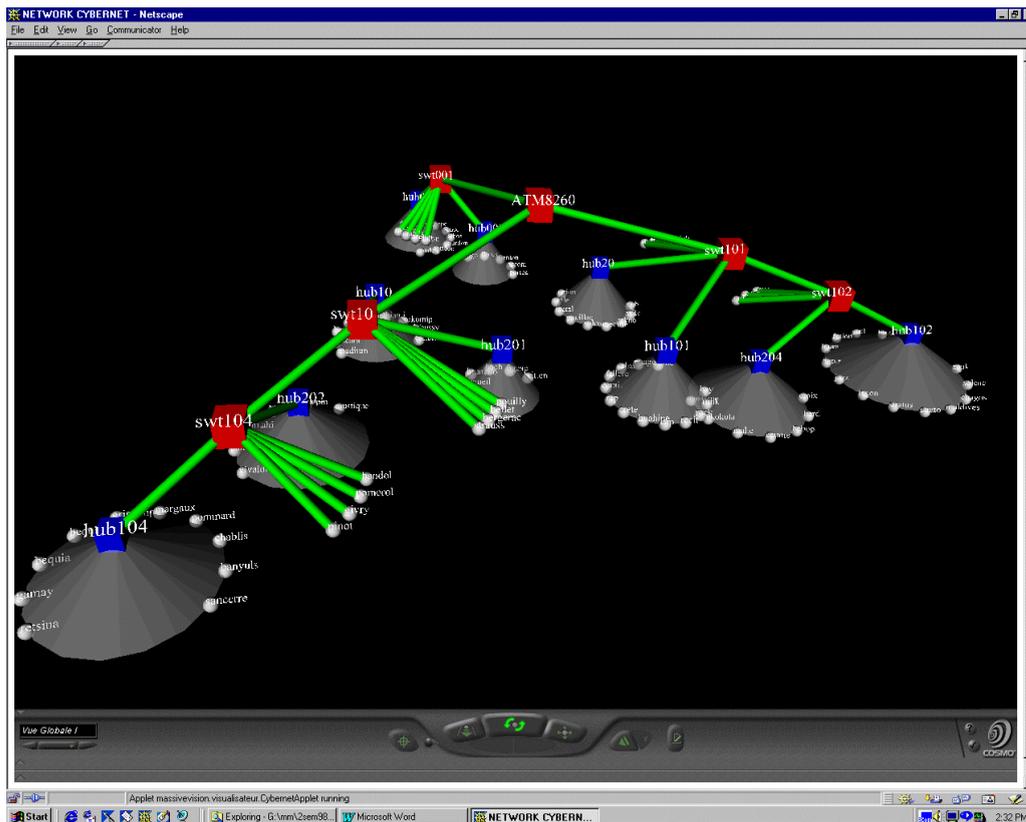


Figure 4 : visualisation du réseau de l'institut Eurecom en utilisant un gestionnaire de placement du type « cone tree »

## 4. Visualisation des informations

Nous avons vu comment nous construisons des services et quels sont les composants graphiques qui permettent de construire un monde virtuel. Chaque service est visualisé avec un monde virtuel qui reflète les informations contenues dans ce service. Nous allons maintenant décrire la manière d'associer un service à un monde virtuel. Cette association est soumise à des contraintes et à des choix que nous allons expliquer.

### 4.1. Les services : des graphes non cycliques

Les langages graphique 3D de type VRML, OpenInventor, Java3D ou OpenGL imposent tous une structuration hiérarchique de la description de scène. Par conséquent pour pouvoir faire correspondre la structuration, donc les relations, d'un service avec une structure de description de scène 3D, il est nécessaire d'obtenir un arbre (sous-graphe non cyclique) lors de la création d'un service à partir du graphe d'entités.

### 4.2. Construction de la scène graphique

Il existe deux grands groupe d'informations à visualiser: les entités et les relations entre celles-ci. Dans sa représentation la plus simple, la construction de la scène graphique se calque sur la structure de l'arbre qui forme le service que l'on veut visualiser. Les feuilles de l'arbre, i.e. les entités, sont représentés par des glyphes 3D et

les noeuds internes, i.e. les relations, par un GP. En fonction de son type et de sa position dans l'arbre chaque entité possède un glyphe 3D spécifique et les informations contenues dans chaque entité sont visualisées par l'intermédiaire des paramètres visuels du glyphe 3D associé. D'autres possibilités de construction sont possibles car il n'existe pas de contraintes en terme d'association entre nœud du service et composants graphiques. Une même entité du service peut en effet être associée à 0 ou N composants graphiques et un même composant graphique peut être associé à 0 ou N entités du service. Cette liberté permet de prendre en compte les points suivants :

- Au niveau de l'interaction de l'utilisateur avec le monde virtuel, il est important d'offrir à l'utilisateur la possibilité de synthétiser un sous arbre du service (et donc une partie du monde virtuel) dont les informations détaillées ne l'intéresse pas. Dans ce cas, les paramètres visuels d'un glyphe 3D sont associés à des informations calculés à partir de certaines données provenant des entités du sous arbre du service. Dans la scène 3D, on substitue donc un ensemble de glyphes 3D et GPs par un glyphe 3D unique résumant les informations contenues dans ces derniers.
- A l'inverse il peut être utile de visualiser en détail certaines informations et donc de projeter des informations d'une même entité sur plusieurs composants graphiques.

Dans tous les cas, il est important de prendre en compte le fait que l'association doit être effectuée de façon dynamique car la structure de l'arbre correspondant au service évolue dans le temps (apparitions et disparitions de noeuds et de branches).

### **4.3. Métaphores**

#### **Un jeu de composants graphiques**

Un des buts du projet CyberNet est d'utiliser des métaphores visuelles, à l'image de la métaphore du bureau que l'on trouve la plupart des ordinateurs, afin d'augmenter la lisibilité de l'information. Une métaphore est constituée d'un jeu de composants graphiques représentant le concept choisi. Par exemple, pour créer une métaphore de ville nous avons besoin de glyphes tels que des maisons, des immeubles, des routes ainsi que des GPs qui placent ces derniers dans différents quartiers [8]. Pour utiliser une métaphore du système solaire, nous allons devoir créer des glyphes 3D de planètes, d'étoiles, de satellites et un GP qui place ses éléments en orbite autour d'un centre.

#### **Une cohérence nécessaire**

Chaque jeu de composants graphiques est conçu pour être utilisé seul à l'intérieur d'un monde virtuel. En effet, il s'agit d'obtenir une cohérence au sein d'un monde sur le plan métaphorique mais aussi sur le plan des paramètres visuels. Imaginez un monde où la couleur permettrait à la fois d'identifier la nature d'un objet et de visualiser une valeur. Placé devant un objet d'une couleur donnée, l'utilisateur ne pourrait déterminer de quelles informations il s'agit. Pour cela, il est nécessaire d'instaurer des lois d'utilisation des paramètres visuels à l'intérieur d'une même métaphore, lois qui s'appliquent à tous les composants graphiques qui la compose.

#### **Indépendance vis-à-vis du service**

Nous voulons également offrir à l'utilisateur la possibilité d'expérimenter plusieurs types de métaphores pour la visualisation d'un même service. Pour cela, une métaphore ne doit pas être liée à un type de service. Pour chaque association

informations/composant graphique, on doit donc définir la façon d'associer les données réelles obtenues sur le réseau aux paramètres visuels du composant. Ceci est réalisé en utilisant un adaptateur [9] spécifique capable de transformer le domaine de valeur du monde réel (ex. le nombre de paquets transmis) vers le domaine de valeur du paramètre visuel utilisé (ex. la taille d'un cylindre). Toute modification des informations est bien sur immédiatement répercutée au niveau du ou des paramètres visuels auxquels cette valeur est associée.

A l'heure actuelle, ces associations sont prédéfinies et le respect de la cohérence est donc laissé à la charge du concepteur. Toutefois un formalisme de caractérisation des données récoltées qui, couplé à un formalisme de caractérisation des paramètres visuels nous permettra d'automatiser ces associations, est à l'étude [9].

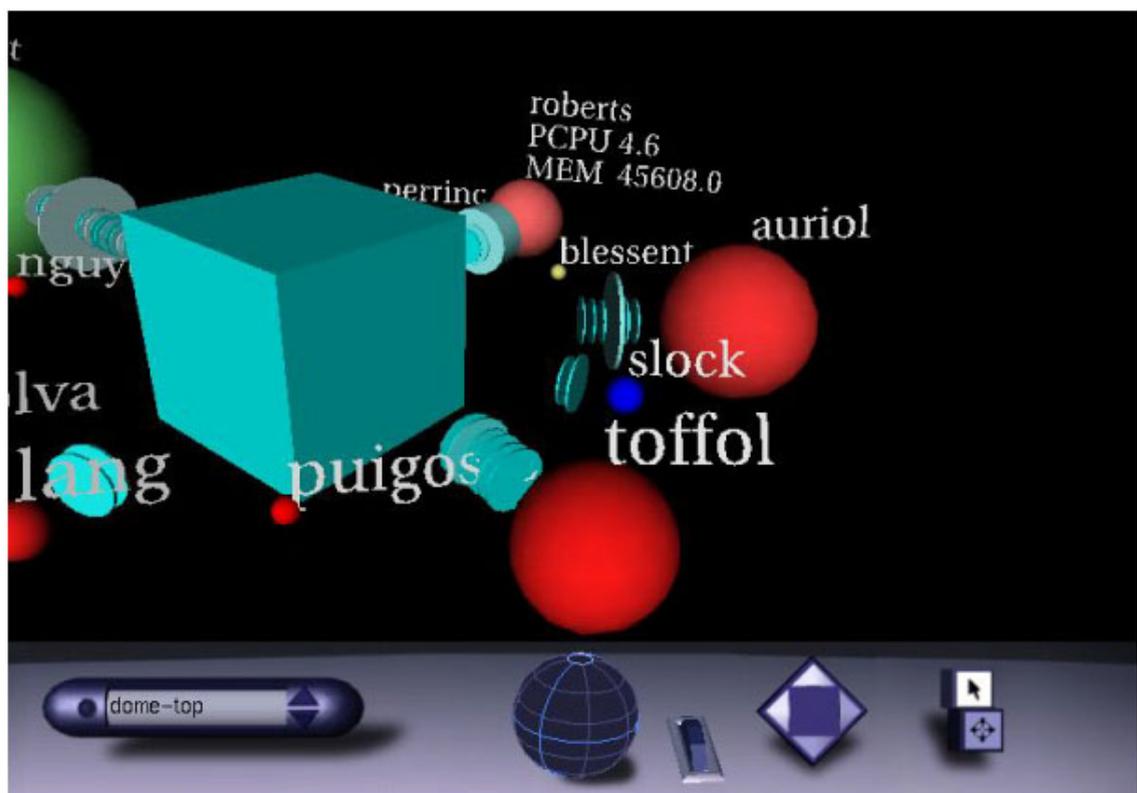


Figure 5 : Visualisation des utilisateurs et des processus d'une machine. Les glyphes sont des cubes pour les machines, des cylindres pour les processus et des sphères pour les utilisateurs. Les GPs rangent ces différents objets selon un modèle héliocentrique : la machine au centre et les utilisateurs en position de satellites plus ou moins éloignés en fonction de leurs CPU. Dans ce modèle, les processus sont placés entre l'utilisateur et la machine. Les paramètres d'un cylindre représentent les attributs du processus associé : la longueur du cylindre est proportionnelle à la consommation de CPU, le rayon à la mémoire utilisée, la couleur au CPUTime. De même, la couleur de la sphère représente le groupe de l'utilisateur, la nuance son CPUTime et le diamètre la mémoire utilisée.

## 5. Conclusion

Nous avons vu tout au long de cet article comment, à partir d'informations collectées sur le réseau, on pouvait construire un monde virtuel permettant la visualisation de services liés à l'administration de réseau. Nous avons proposé un modèle de structuration (entités, relations, graphe d'entités et services) qui nous donne la possibilité d'organiser efficacement les informations dynamiques. Les structures ainsi obtenus sont utilisées afin de construire des visualisations sous la forme de mondes virtuels composés de 3D glyphs et de gestionnaires de placement définissant des modèles métaphoriques. Plusieurs sujets sont encore à étudier. Nous devons maintenant rechercher de nouvelles façon de visualiser les services et donc développer de nouvelles métaphores. La construction automatique passe également par une poursuite du travail sur l'automatisation des associations données- métaphores. Enfin nous pensons que notre système est adaptable à d'autres domaines d'applications où l'on trouve de vastes quantités d'informations dynamiques et nous envisageons donc de le tester dans un de ces domaines.

Le projet CyberNet est supporté par France-Télécom et l'Institut Eurécom.

## 6. Références

- [1] S. Mukherjea, J. Foley and S. Hudson (1995): "Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views", Proceedings of ACM SIGCHI '95, May 1995, Denver, Colorado, pp. 331-337.
- [2] Chernoff H., The use of faces to represent points in k-dimensional space graphically. Journal of American Statistic Association, Vol. 68, 1973, Pages 331-368
- [3] Swing, E., Flodar : Flow visualization of network traffic. IEEE Computer Graphics and Applications. September/October 1998. Pages 6-8.
- [4] Robertson, C.G., and Mackinlay, J.D., and Card, S.K. Cones trees: Animated 3D visualizations of hierarchical information. Proceedings CHI'91 Human Factors in Computing Systems, ACM, Pages 173-179
- [5] Hendley, R.J., and Drew, N.S., and Wood, A.M., Beale, R. , Narcissus: Visualising information. Proceedings IEEE Information Visualization 95. Pages 90-96.
- [6] Benford, S., Snowdon, D., Greenhalgh, C., Ingram, R., Knox, I., and Brown, C. "VR-VIBE: A Virtual Environment for Co-operative Information Retrieval" Proceedings Eurographics '95, pp 349-360
- [7] Chalmers, M. and Chitson, P., Bead: Explorations in Information Visualisation, Proceedings of SIGIR'92, ACM Press, pp. 330-337.
- [8] Ingram, R. and Benford, S., Legibility Enhancement for Information Visualisation, Proceedings of Visualization 1995, Atlanta, Georgia, October 30 - November 3, 1995
- [9] P.Abel, P.Gros, D.Loisel, J.P.Paris : Network management and virtual reality . Proceeding des 7ème journées du GT Réalité Virtuelle. Laval, June 1999

