# *UN OUTIL POUR LA CONSTRUCTION GEOMETRIQUE*

**Gabriel BRAUN**

This paper presents Geophile, a soft designed to manipulate geometric concepts, that can be used at two levels : at the teacher level Geophile is an object oriented language specialized in geometric knowledge representation and the student level provides tools to manipulate easily this knowledge. This article deals only with this student use of Geophile.

First we show how the user can create interactively geometric constructions with primitive objects an relations (defined by the teacher) in any order. Then we explain how to operate some modifications that will be automatically propagated in the whole construction. Some more elaborated tools making possible locus or enveloppes visualisation and a few formal manipulations are shown at end.

## INTRODUCTION

La géométrie telle qu'elle est enseignée aujourd'hui dans les lycées et collèges tend à réintroduire la pratique de la construction géométrique, favorisant ainsi une approche visuelle, de cette discipline plutôt que calculatoire ou formelle [PLU 86].

La visualisation de constructions (figures géométriques) constituant une extension du domaine sensible, favorise une approche plus perceptive, privilégiant ainsi l'intuition [INH 47].

Cependant la réalisation concrète de constructions présente d'innombrables difficultés d'ordre matériel. En effet le tracé de figures est long et délicat, en raison des contraintes inhérentes à cette activité : le maintien de la figure dans les limites de l'épure demande souvent de procéder par approximations successives, la complexité d'une figure nécessite la réalisation de constructions intermédiaires qui seront effacées, un minimum de précision est requis, toute modification est coûteuse en temps.


*2(1989) (p. 111-133) IREM de Strasbourg*

La construction géométrique est donc une activité de haut niveau, malheureusement très difficilement praticable.

Lever cette contradiction a été l'élément moteur qui nous à conduit à l'élaboration d'un outil informatique adapté aux besoins de la construction géométrique : Géophile.

Géophile s'adresse tout d'abord aux étudiants pour qui il constitue un outil élaboré, facile d'accès, permettant la réalisation, la modification, l'animation de constructions géométriques définies à partir d'objets et de constructeurs de base.

La connaissance géométrique étant variable selon le niveau des élèves, l'ensemble des constructeurs utiles n'est pas figé. Ainsi, outre les nombreux constructeurs de base définis dans le système, il est possible à l'enseignant de rajouter de nouveaux objets et de nouveaux constructeurs élémentaires.

Le logiciel que nous présentons peut donc être employé à deux niveaux :

- le niveau utilisateur qui permet la manipulation de constructions par des élèves
- le niveau programmeur qui permet à l'enseignant d'adapter les notions géométriques connues par le système aux besoins des étudiants utilisateurs

L'utilisation de Géophile tant au niveau utilisateur (élève) que programmeur (enseignant) utilise un formalisme proche de celui en vigueur en mathématique et ne requiert donc quasiment aucune connaissance informatique.

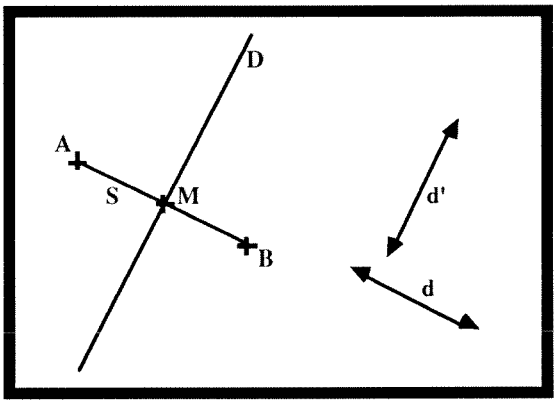
Dans cet article, nous présentons Géophile exclusivement d'un point de vue utilisateur, la description de Géophile en temps que langage étant quelque peu technique. Pour une description complète tant au niveau utilisation que programmation nous reportons le lecteur à [BRA 88].

Quant à la géométrie traitée, nous nous sommes restreints à la géométrie plane euclidienne élémentaire. Elle nous a semblé constituer un cadre suffisemment large pour illustrer les possibilités de Géophile.

## CARACTERISATION D'UNE CONSTRUCTION GEOMETRIQUE

**Exemple** : médiatrice d'un segment

Dans le cadre d'un exercice scolaire, la construction d'une médiatrice peut constituer un problème qui pourrait être ainsi formulé : déterminer la droite D passant par le milieu de A et B, extrémités du segment donné S, ayant pour direction la direction perpendiculaire à la droite passant par A et B.




Remarquons qu'une première lecture linéaire de l'énoncé précédent sans une analyse étape par étape est bien insuffisante pour se faire une idée du résultat de la construction. Par contre un simple coup d'oeil sur la figure illustrant l'énoncé donne instantanément une vision globale et complète de la situation.

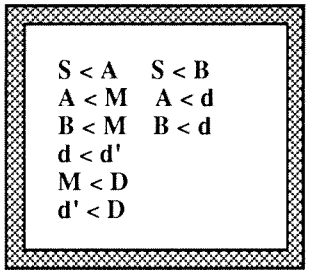
Nous pouvons transcrire l'énoncé précédent en un ensemble de relations entre les divers éléments de la contruction :

**S = segment donné**
**A = point origine de S**
**B = point extrémité de S**
**M = milieu de A et B**
**d = direction de la droite AB**
**d' = direction perpendiculaire à d**
**D = droite passant par M et de direction d'**

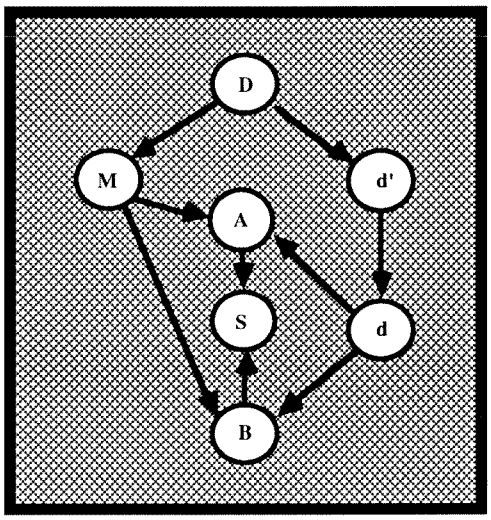
Chaque relation de la construction peut être assimilée à une équation de la forme : **X = f ( Y Z ....)**, f étant une fonction dont on connait une description procédurale (milieu de , direction perpendiculaire à, etc.) et X,Y et Z sont des entités mathématiques.

La construction d'une médiatrice semble donc intégralement déterminée par le système d'équations ci-dessus. L'ordre de construction correspond à un ordre possible de résolution du système : on ne détermine un élément que lorsque tous les éléments dont il dépend sont déterminés.

Plus généralement si nous notons "<" la relation d'ordre partiel "doit être déterminé avant" nous avons les contraintes suivantes pour notre construction :



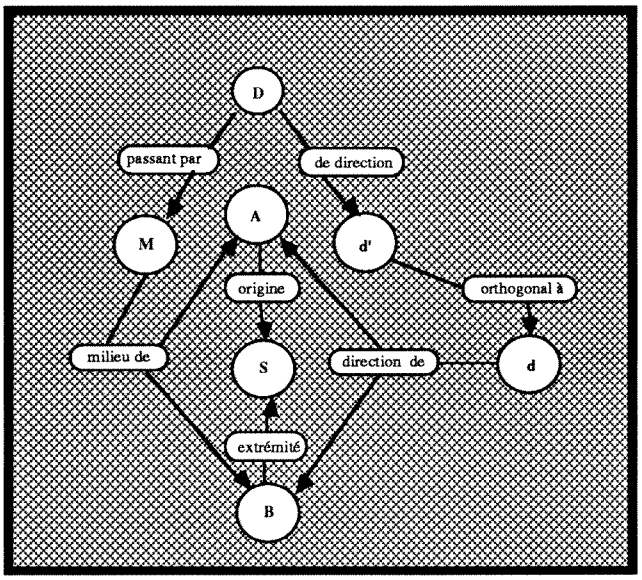

Nous pouvons également représenter l'ordre partiel entre les divers composantes de la figure sous forme d'un graphe :




Ce graphe qui ne représente que les diverses relations entre les objets sans en préciser la nature est appelé : "**graphe d'interdépendances**" .

Un tel graphe ne contient pas la complète information nécessaire à la réalisation de la figure, mais permet seulement de définir un ordre partiel de construction entre les divers objets entrant dans la composition du résultat.

Notons que, pour peu que l'on rajoute le terme fonctionnel au niveau de chaque noeud du graphe afin de préciser la relation entre deux objets reliés par une arête, nous avons une représentation complète de la construction :




Cette représentation sous forme de graphe orienté dont les arêtes sont étiquetées est appelé "**réseau de construction**" associé à la construction.

## Généralisation

De façon plus générale nous pouvons représenter une procédure de construction d'une figure par un système d'équations de la forme suivante :

$$\begin{cases} X_1 \; X_2 \; \ldots \; X_N \text{ donnés} \\ X_{N+1} = f_{N+1}\,(X_1\, X_2 \ldots X_N) \\ \ldots\ldots\ldots\ldots \\ X_{N+K} = f_{N+K}\,(X_1\, X_2 \ldots X_N\; X_{N+1} \ldots X_{N+K-1}) \\ \ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots \\ X_{N+M} = f_{N+M}\,(X_1\, X_2 \ldots X_N\; X_{N+1} \ldots X_K \ldots X_{N+M-1}) \end{cases}$$

$X_1, X_2$ .... $X_N$ sont les objets connus initialement

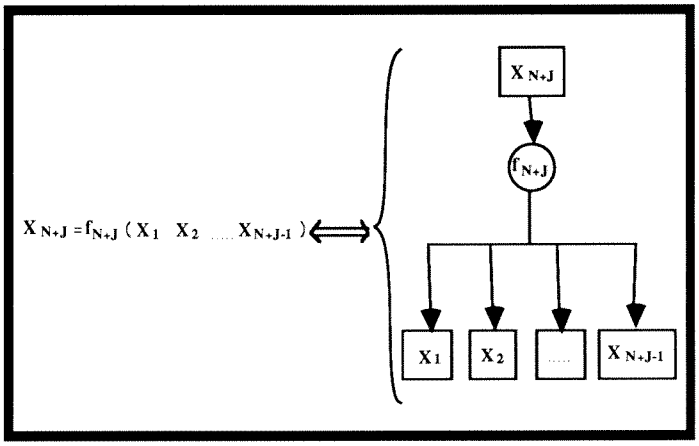
$X_{N+1}, X_{N+2}$ .... $X_{N+K}$ ..... $X_{N+M-1}, X_{N+M}$ sont des objets calculés

$f_{N+J}$ est une fonction qui, dépendant de certains objets pris parmi $(X_1 .. X_{N+J-1})$, détermine $X_{N+J}$

Ce type de système d'équations a été largement étudié dans le cadre de travaux sur les langages à affectation unique et sur la méthode de programmation déductive [HUC 77], [PAI 79]. Nous ne présentons ici que les aspects essentiels dans le cadre de la construction géométrique.

**Réseau de construction associé à un système d'équations**

Pour chaque équation de la forme $X_{N+J} = f_{N+J} (X_1 .....X_{N+J-1})$ on crée un noeud du réseau de nom $X_{N+J}$ pointant vers tous les $X_K$ effectivement arguments de la fonction $f_{N+J}$.
On obtient ainsi le graphe d'interdépendance. Afin d'aboutir au réseau sémantique il faut et il suffit de stocker au niveau de chaque noeud $X_K$ sa définition, en l'occurrence la fonction $f_K$.




La relation d'ordre doit être déterminé avant se traduit en terme de chemin au niveau du graphe d'interdépendance.

$X_I$ doit être déterminé avant $X_J$ est équivalent à dire qu'il existe un chemin de $X_J$ à $X_I$ dans le graphe d'interdépendance.

Le réseau est un graphe sans cycle en vertu de l'antisymétrie de la relation d'ordre. Il ne peut y avoir de définition circulaire dans le système d'équations définissant la construction géométrique.

## ELABORATION D'UN RESEAU DE CONSTRUCTION

Rappelons que l'un des objectifs essentiels du système est de minimiser l'intervention de l'utilisateur. Dans cette optique il est souhaitable que ce dernier ne soit pas contraint de construire une figure selon l'ordre imposé par le système d'équations la représentant.

Mais, ne respectant plus l'ordre de construction, l'utilisateur ne dispose a priori plus d'aucun critère de complétude de son système de définition. Le système est dit **complet** lorsque toute les équations nécessaires à sa résolution sont définies.

Afin de répondre à ces deux impératifs nous avons conçu un protocole de construction n'obligeant pas l'utilisateur à définir un ordre et rendant impossible toute omission d'une définition par l'intervention du système.

### Le protocole de construction

Le protocole de construction d'une figure est le suivant :

Lorsque l'on définit une entité $X_J = f_J$ (certains $X_K$) avec $K < J$

- $X_J$ fait partie des **objets en cours de définition** qui représentent à un instant donné toutes les entités dont la valeur n'a pu être déterminée, faute de connaitre la définition de tous ses arguments

- il faut avant tout qu'aucun des arguments $X_K$ de $f_J$ ne fasse partie de l'ensemble des **objets en cours de définition** car il s'agirait d'une définition circulaire ; si ce n'est le cas le système refuse la définition et en demande une nouvelle

- la définition n'étant pas circulaire, le système cherche parmi les arguments $X_K$ ceux qui ne sont pas encore définis et en demande la définition à l'utilisateur ; le même protocole s'applique récursivement à tous ces arguments indéfinis

- tous les arguments de la définition étant définis et évalués il est à présent possible pour le système de déterminer la valeur (évaluer) du sommet $X_J$ traité, qui est alors extrait de l'ensemble des **objets en cours de définition**, étant dès lors intégralement défini..

Notons que la procédure termine toujours, car tôt ou tard on introduit des définitions constantes qui correspondent aux données connues de la construction. Le protocole inspiré de la méthode déductive [HUC 77] [BEL 78], autorise les trois types de construction suivants :

a) Construction exclusivement **descendante** ou **déductive**

Une construction **descendante** de la médiatrice a l'allure suivante :

définir D = droite passant par M et de direction d'
**? donnez la définition de M argument de D** (interrogation du système)
M = milieu de A B
**? donnez la définition de A argument de M**
A = origine de S
**? donnez la définition de S argument de A**
S = #(segment #(point 0 0) #(point 1 2) ) (notation d'un segment)
**? donnez la définition de B argument de M**
B = extrémité de S
**? donnez la définition de d' argument de D**
d' = direction perpendiculaire à d
**? donnez la définition de d argument de d'**
d = direction de A B
**----> #(droite -6 4 11)** (droite d'équation : -6x + 4y + 11 = 0)
("---->" indique le résultat d'une évaluation)

Notation : les réponses du système sont imprimées en **caractères gras**

Notons que lorsque plusieurs objets dépendent d'un même argument la définition de ce dernier n'a lieu qu'une fois. Par exemple, lors de la définition de B, S est déjà connu ayant été défini comme argument de A.

La définition de D se termine par la donnée de sa valeur. Il est bien entendu possible pour le système d'afficher au fur et à mesure toutes les valeurs des objets intermédiaires évalués entrant dans la définition de D.

b) Construction exclusivement **remontante** ou **incrémentale**

Il s'agit de définir les objets selon l'ordre induit par le réseau de construction :

| |
|---|
| définir S = #(segment #(point 2 3) #(point 5 (cos 0))<br>----> **#(segment #(point 2 3) #(point 5 1))** |
| définir A = origine de S<br>----> **#(point 2 3)** |
| définir B = extrémité de S<br>----> **#(point 5 1)** |
| définir M = milieu de A et B<br>----> **#(point 3.5 2)** |
| définir d = direction de A B<br>----> **#(vecteur 3 -2)** |
| définir d' = direction orthogonale à d<br>----> **#(vecteur 2 3)** |
| définir D = droite passant par M et de direction d'<br>----> **#(droite -6 2 11)** |

Il est bien entendu possible de combiner à volonté constructions remontantes et descendantes. Il est ainsi possible de constituer le réseau par sous-réseaux complets successifs.

On appelle **sous-réseau complet** toute partie de réseau étant un réseau à part entière c'est-à-dire complètement défini et dont les valeurs de tous les noeuds ont pu être calculées.

c) Construction partiellement déductive

définir A origine de S
**? donnez la définition de S argument de A**
S = #(segment #(point 2 3) #(point 5 1) )
**----> #(point 2 3)**

définir M = milieu de A B
**? donnez la définition de B argument de M**
B = extrémité de S
**----> #(point 3.5 2)**

définir D = droite passant par M et de direction d'
**? donnez la définition de d' argument de D**
d' = direction perpendiculaire à d
**? donnez la définition de d argument de d'**
d = direction de A B
**----> #(droite -6 4 11)**

Ainsi l'utilisateur peut librement choisir sa façon de construire étant assuré par le système de ne pas oublier de définition.

Notons que la construction descendante n'est autre qu'une méthode déductive telle que celle employée par des informaticiens pour l'analyse d'un problème [DUC 84], [FIN 85], [HUC 77], [PAI 79].

## MODIFICATION D'UNE CONSTRUCTION

Ayant défini un certain nombre de réseaux nous désirons à présent opérer des modifications. On peut en distinguer deux sortes : les modifications d'objets **terminaux** d'un réseau (ou **feuilles**) et les modifications d'objets non terminaux. Un objet est dit **terminal** si sa définition est indépendante d'arguments, c'est à dire constante. La distinction entre les deux porte sur le but de la modification plutôt que sur la méthode utilisée. En effet le changement d'un objet terminal correspond à un simple changement d'une donnée de base de la construction, celle-ci restant inchangée dans son principe, alors que toute autre modification change la construction en tant que procédure. De façon équivalente nous avons au niveau du système d'équations la possibilité de changer soit une donnée de base soit certaines équations sans pour autant modifier la méthode de résolution.

### Modifications terminales

Il s'agit donc de modifier des objets ne dépendant pas d'autres entités. Ce sont les données du système d'équations ou les feuilles du graphe d'interdépendance.

Ce type de modification correspond à la démarche classique de l'étudiant désirant représenter plusieurs cas de figure d'une même construction afin de mieux en saisir la généralité. D'un point de vue d'informaticien en assimilant la construction à la procédure qui l'exécute, la modification d'un objet terminal revient à un simple changement du jeu d'essai.

Lorsqu'une feuille du réseau de construction est modifiée, tous les objets qui en dépendent de façon directe ou indirecte devront également subir une modification.

Plus précisément, si nous nommons F l'objet terminal modifié, il est nécessaire de réévaluer tout objet X vérifiant la relation d'ordre :

F doit être déterminé avant X <==> il existe un chemin de X à F

Dans l'optique de minimiser l'intervention de l'utilisateur nous avons permis à ce dernier de définir des constructions sans se soucier de la relation d'ordre de construction entre les objets. Cette démarche serait peine perdue si, lors d'une modification, l'utilisateur était obligé de faire réévaluer les divers objets le nécessitant dans le bon ordre qu'il devrait alors tout de même connaître. Aussi dans ce même esprit d'intervention minimale, nous rendons automatique la réévaluation des diverses entités à modifier.

#### Stratégie de réévaluation

A première vue on pourrait être tenté d'appliquer la stratégie suivante : dans un premier temps on corrige tous les objets qui dépendent de la donnée modifiée, puis tous les objets qui dépendent de ces objets et ainsi de suite jusqu'à ne plus trouver d'objets dépendants.

Cette façon de procéder réalise bien la modification, mais présente un défaut. En effet une telle remontée (en largeur) dans un graphe peut conduire à parcourir plusieurs fois un même sommet. Il suffit que plusieurs de ses arguments soient modifiés pour engendrer plusieurs réévaluations du même objet.

Afin de palier ce défaut la modification automatique se fait en deux étapes :

- le système fabrique la liste des sommets rencontrés en effectuant une remontée en largeur à partir de la feuille modifiée en ne retenant que la dernière occurence de chaque sommet
- les sommets sont réévalués dans l'ordre de cette liste

Utilisation : on peut distinguer deux sortes de modifications terminales du réseau.

- La modification d'une feuille en feuille

Ce type de redéfinition d'un objet a pour finalité de représenter une même construction avec différentes valeurs des données de bases.

Dans l'exemple de la médiatrice l'utilisateur pourrait donner différentes valeurs au segment S, unique donnée de base de la construction.

- La modification d'une feuille en un objet non terminal

Ce type de modification remplace une donnée de base par une construction ou une feuille par un réseau.

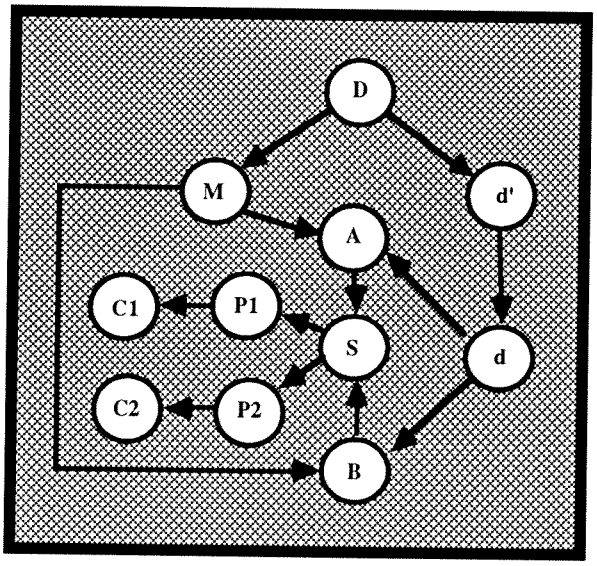
Lorsque l'utilisateur redéfinit une feuille en la faisant dépendre d'arguments, ceux-ci sont à leur tour définis de façon déductive comme dans le cas de la définition d'un réseau. C'est le système qui interroge l'utilisateur pour définir les nouveaux objets.

Pour la construction de la médiatrice il est possible de redéfinir le segment S comme un objet dépendant d'arguments. Par exemple S peut être redéfini comme reliant les deux centres P1 et P2 de deux cercles C1 et C2.

exemple : une telle redéfinition est réalisée de la façon suivante :

```
redéfinir  S = #(segment  P1  P2)
? donnez la définition de P1 argument de S
        P1 = centre de C1
        ? donnez la définition de C1 argument de P1
            #(cercle  5  #(point  1 2))
? donnez la définition de P2 argument de S
        P2 = centre de C2
        ? donnez la définition de C2 argument de P2
            C2 = #(cercle  3  #(point  5  0))
-----> #(segment  #(point  1  2)  #(point  5  0))
```

La valeur de la médiatrice a bien entendu été recalculée : D = #(droite -4 1 10.5)




La modification de feuilles est un outil très efficace pour représenter rapidement différentes instanciations du même schéma de construction. Cette possibilité de réévaluer automatiquement permet à l'utilisateur de disposer d'une vision plus générale d'un même réseau grâce à des modifications successives de feuilles réalisées en un minimum d'investissement de sa part.

En fait un utilisateur définit un schéma d'instanciation général en décrivant une construction particulière.

Lorsque l'utilisateur désire modifier partiellement la construction dans son principe, il a recours aux modifications non terminales.

### Modifications non terminales

La redéfinition d'objets non terminaux modifie la construction dans ses principes. De telles modifications permettent de redéfinir des objets qui dépendaient précédemment d'arguments. La nouvelle définition peut être une feuille (sans arguments) ou un réseau (avec de nouveaux arguments).

Le principe de redéfinition d'objets non terminaux est le même que celui appliqué lors de la modification de feuilles. Le nouveau réseau est défini de façon déductive à partir de l'objet modifié.

Cependant il existe une différence essentielle pour la modification non terminale. Les anciens arguments de l'objet modifié n'y sont plus connectés alors que pour la redéfinition de feuilles ces arguments sont inexistants.

Ainsi la modification terminale n'est qu'un cas particulier de la modification non terminale.

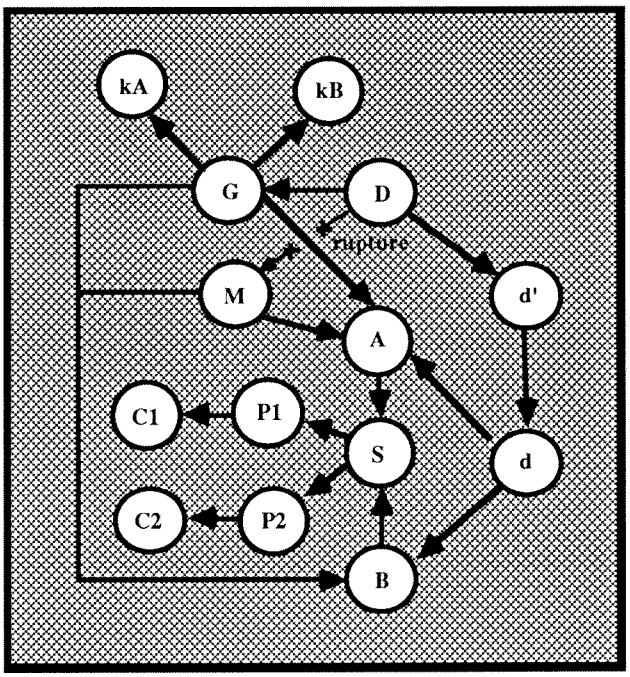
Un seul problème se pose dans le cas général : que deviennent les anciens arguments d'un objet modifié ?

Les arguments d'un objet modifié en sont déconnectés. Cependant certains peuvent également être arguments d'autres objets du réseau. Aussi est-il hors de question de les éliminer du système. Dans le cas particulier où chacun des arguments de l'objet redéfini n'entre que dans la définition de ce dernier ceux-ci vont constituer différents réseaux déconnectés qui pourront être réutilisés ultérieurement pour être rebranchés au réseau principal par exemple.

Nous illustrons ces deux cas pour la construction de la médiatrice dont nous prenons le dernier réseau en cours.

- cas 1 : les arguments de l'objet modifié restent connectés au réseau

Nous décidons de redéfinir la droite D non plus comme passant par le milieu de A et B mais par un barycentre de ces derniers avec des coefficients respectifs kA et kB. Le point M n'est plus argument de D, devient une racine du réseau, mais y demeure connecté par ses arguments A et B. Ainsi dans ce cas de figure nous conservons un réseau connexe.

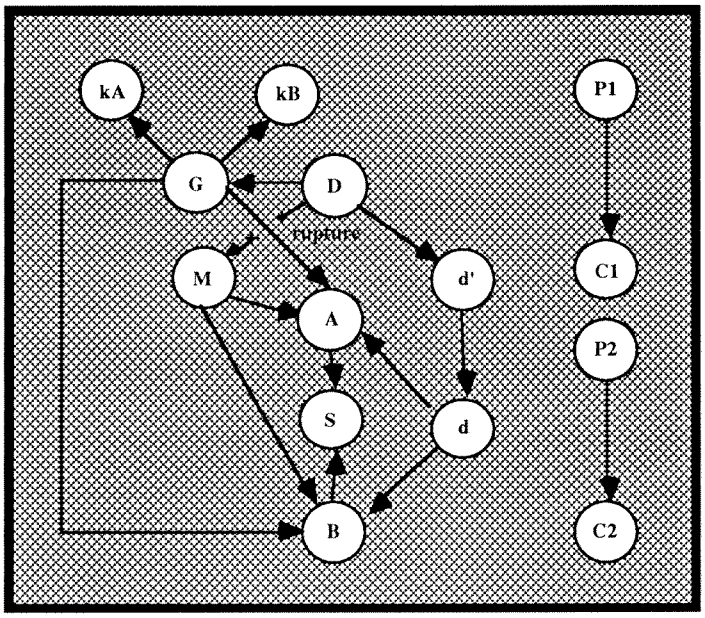


```
redéfinir  D = droite passant par G de direction d'
? donnez la définition de G argument de D
   G = barycentre de A et B avec les coefficient kA et kB
   ? donnez la définition de kA argument de G
      kA = 1
   ? donnez la définition de kB argument de G
      kB = 1
-----> #(droite  -4  1  10.5)
```

Notons que D conserve la même valeur car nous avons utilisé un barycentre particulier qui est le milieu de A B ( kA = kB = 1). Il suffira de modifier les nouvelles feuilles kA et kB pour obtenir D passant par un barycentre quelconque.

- cas 2 : les arguments de l'objet modifié sont déconnectés

Nous considérons la construction de la droite D (qui n'est plus une médiatrice) en l'état. Nous redéfinissons le segment S non plus comme dépendant de P1 et P2, mais comme une donnée de base. Nous obtenons ainsi trois réseaux déconnectés qui demeurent accessibles dans le système.



## AUTRES MANIPULATIONS DE RESEAU

### Consultations

A tout moment l'utilisateur peut souhaiter consulter l'information contenue dans le système, en particulier lors d'une construction déductive. En effet, les diverses définitions des objets, la liste des objets déjà créés, la liste de ceux en cours de définition, l'ensemble des fonctionnalités existantes, etc. sont des informations indispensables pour une utilisation efficace de Géophile.

En effet le système permet toutes sortes de consultations et en particulier lors d'une construction déductive durant laquelle toute opération est autorisée. Lorsque le programme demande une définition d'objet pour construire le réseau il suffit à l'utilisateur de taper un point d'interrogation "?" signifiant qu'il désire interroger le système avant de lui fournir une réponse. Il peut alors donner toute commande au système et en particulier des commandes de consultation.

Nous inventorions les différentes informations accessibles à l'utilisateur que nous classifions en trois catégories :

Consultation des sommets du réseau

| | |
|---|---|
| - Valeur d'un sommet | valeur de D = #(droite -4 1 10.5) |
| - Définition d'un sommet | déf G : barycentre de A B avec kA et kB |
| - Arguments d'un sommet | arguments de G : ( A B kA kB) |
| - Sommets dont dépend un sommet | G dépend de : A B S P1 P2 C1 C2 |
| - Sommets dépendants d'un sommet | sommets dépendant de S : A B M G d d' D |
| - Expression d'un sommet en fonction d'autres sommets | expression de S en fonction de C1 et C2 : segment d'origine le centre de C1 et d'extrémité le centre de C2 |

Consultation globale au réseau

- Racines du réseau
- Feuilles du réseau
- Objets existants définis dans le réseau

Consultation par classe d'objet

Il est possible de consulter l'information relative à une classe donnée d'objets. Par exemple :

- quels sont tous les points définis dans le système ?

- quelles sont toutes les primitives de bases applicables à un cercle ?

- comment est représenté une droite ?

## Utilisation avancée des réseaux

Terme fonctionnel en argument

Dans tous les réseaux que nous avons décrits jusqu'à présent une fonction permettait de définir des relations entre les objets, mais jamais une fonction n'était elle-même considéré comme un objet. Il y a une distinction formelle entre les $X_I$ et les $f_I$.

Cette distinction convient fort bien pour des constructions où les relations entre objets sont relativement figées et où l'on fait varier certains de ces objets non fonctionnels.

Cependant dans certains cas, ce sont plutôt les fonctions que l'on désirerait faire varier. Ceci est rendu possible grâce à l'implantation du langage Lisp, sur qui repose Géophile, qui ne distingue pas les données des programmes. Géophile permet de construire un réseau avec un terme fonctionnel en argument.

```
définir y = appliquer  f  à  x
? donnez la définition de f argument de y
        f = 'cosinus
? donnez la définition de x argument de y
        0
----> 1
```

Géophile utilise donc la fonction "appliquer" qui permet de calculer le résultat de l'application d'une fonction qui est argument à certains arguments.

On pourrait ainsi appliquer différentes transformations ponctuelles à un même objet géométrique en donnant différentes valeurs à la fonction représentant la transformation.

Cette façon de procéder nous permet certes d'utiliser la fonction f en argument, mais elle suppose la prédéfinition des fonctions que l'on désire appliquer. Géophile permet la définition d'une fonction de façon dynamique au cours de la création d'un réseau.

Un pont entre utilisation et programmation

Dans certains cas de figure l'utilisateur peut être conduit à effectuer plusieurs fois la même construction dans le temps ou dans l'espace durant une même session ou entre plusieurs sessions. Géophile lui offre différentes possibilités de réutilisation automatique d'un réseau afin de lui éviter de réitérer souvent une même suite d'opérations.

- Durant une même session

L'utilisateur souhaite effectuer plusieurs fois la même construction non dans le temps, auquel cas il lui suffit d'utiliser la redéfinition, mais dans l'espace, c'est-à-dire qu'il désire représenter simultanément plusieurs réseaux identiques aux feuilles près.

Ainsi ayant construit le réseau de la médiatrice au segment S, pour obtenir la médiatrice du segment S1 sans redéfinir un réseau identique on procède ainsi :

définir D1 = appliquer expression sur S1
**? donnez la définition de expression argument de D1**
expression = expression de D en fonction de S
**? donnez la définition de S1 argument de D1**
#(segment #(point 1 0) #(point 0 1))
**----> #(droite 1 -1 0)**

Bien entendu toute modification du réseau D provoque la réévaluation de l'expression de D en fonction de S et modifie donc le réseau D1.

Plus généralement on peut extraire un sous-réseau sans partir d'une racine et sans arriver aux feuilles d'un réseau global.

Cette façon d'opérer s'avère donc efficace pour définir parallèlement plusieurs réseaux semblables durant une même session, mais ne permet pas d'automatiser une construction d'une session à l'autre.

- Entre plusieurs sessions

L'utilisateur peut être conduit à effectuer fréquemment une même construction au cours de différentes sessions.

Une première solution consiste à stocker la construction (liste des définitions) dans un fichier selon l'ordre incrémental que l'on rechargera à volonté pour définir la construction.

Cette première solution stocke toute l'information contenue dans le réseau que l'on mémorise. Or, bien souvent, lorsqu'une construction est fréquente ce n'est plus la procédure de construction qui est intéressante, mais directement le résultat. En d'autres termes on souhaite qu'une construction beaucoup utilisée devienne une construction de base.

Géophile crée un pont entre utilisation et programmation en permettant à partir d'un réseau de définir un constructeur de base qui restera accessible de sessions en sessions.

L'utilisateur ayant construit le réseau de la médiatrice, peut décider qu'une telle construction sera un constructeur de base permettant de définir une droite à partir d'un segment (médiatrice du segment)

Il suffit de spécifier : le nom du nouveau constructeur de base :"médiatrice_de"
le nom de l'objet résultat dans le réseau : "D"
le noms des objets arguments : "S"

Ce nouveau constructeur peut dorénavant être utilisé comme suit :

**définir D1 = médiatrice_de S1**

Suites définies à partir d'un réseau

En géométrie on est fréquement conduit à étudier certaines **suites d'objets** variant selon un paramètre telles que les courbes qui ne sont autres que des suites de points, les enveloppes des suites de droites, les mouvements des suites de positions, etc.

Le calcul de l'expression du sommet d'un réseau en fonction d'autres sommets trouve ainsi d'autres applications que la fabrication de constructeurs de base comme le montre le paragraphe précédent. En effet l'une de ces applications en géométrie est la définition de suites d'objets.

Une suite d'objets représente l'ensemble des valeurs prises par un objet du réseau pour un ensemble de valeurs prises par un autre objet du réseau dont il dépend. Plutôt que de représenter une suite par la liste des valeurs prises par un objet, nous la représentons par l'expression de l'objet en fonction de l'argument du réseau à faire varier. Cette façon de procéder s'impose car l'expression de l'objet, plus l'intervalle et le pas de variation, sont une représentation minimale de la suite. Ainsi une suite est représentée par :

- un objet1 dont la variation figure la suite
- un objet2 argument de objet1 induisant la variation de objet1
- un corps qui est l'expression de objet1 en fonction de objet2 dans le réseau
- la valeur initiale de la variation de objet2
- la valeur finale de la variation de objet2
- le pas de variation de objet2

A titre d'exemple pour définir la suite des droite D pour la variation du coefficient barycentrique kA (voir dernier exemple de réseau) l'utilisateur procède ainsi :

définir Suite_D = suite des D pour kA allant de 1 à 10 par pas de h
**? donnez la définition de h**
h = 0.5
-----> **#(suite D Ka** *(expression de D en fonction de kA)* **1 10 .5)**
(représentation d'une suite)

Suite_D est à présent connecté au réseau en pointant sur D kA et h et est automatiquement modifié dès que D est modifié.
Bien entendu il est possible de construire des suites de suites et ainsi de suite.

## GRAPHISME

### Principes

- Affichage indépendant de l'existence des entités du réseau

Il n'est pas nécéssairement intéressant de visualiser systématiquement tous les objets graphiques présents dans un réseau, cela peut même constituer une gêne nuisant considérablement à la clarté de la figure.Aussi l'affichage des entités du réseau est totalement indépendant de leur existence. C'est l'utilisateur qui doit demander explicitement l'affichage et l'effacement d'un objet graphique.

- Représentation graphique correlée avec le réseau

Une entité affichée le demeure jusqu'à ce que l'utilisateur donne un ordre d'effacement. Lors d'une modification tous les objets réévalués sont réaffichés automatiquement par le système de sorte à maintenir les contraintes de la figure imposées par le réseau. Cette corrélation permanente entre un réseau et sa représentation graphique est pédagogiquement l'un des interêts essentiels de Géophile.

- Réseau de construction indépendant du repère écran

### Utilitaires d'affichage

- Rafraîchissement de l'écran - Couleur du fond - Repère par défaut

- Trace d'un objet graphique

Il est possible de tracer une entité graphique au fur et à mesure que de modifications lui sont appliquées. Lorsqu'une variable est tracée, lors d'une modification de réseau, son ancienne représentation à l'écran n'est pas effacée et une nouvelle liste d'affichage lui est attribuée. Cette fonctionnalité est largement dépassée par la possibilité de définir des suites d'objets.

# CONCLUSION

## Etat du prototype

A ce jour le prototype que nous avons décrit fonctionne sur SM90 sous Unix. Les connaissances géométriques implantées ont principalement servi de jeux d'essais pour éprouver les principes de Géophile. L'étude du développement systématique de la géométrie 2D est en cours de réalisation en collaboration avec des géomètres.

L'objectif de simplicité d'utilisation et de minimisation de l'intervention de l'utilisateur lors de la création, la modification et la manipulation de construction a été atteint. Cependant l'étude d'une bonne communication homme-machine, dépassant le cadre de ce travail, n'a pas été entreprise. Elle permettrait l'amélioration de la syntaxe des commandes et l'utilisation de dispositifs de sélection, désignation et localisation.

Les délais de réalisation ou modification d'une figure sont très décents pour le géomètre. La complexité des figures traitées a été considérablement accrue par rapport aux constructions réalisables manuellement (lieux, enveloppes, suites, par exemple). Une nouvelle approche de l'enseignement de la géométrie pourrait en découler. De nouveaux problèmes pourront être abordés car le support sensible que constitue une figure pourra être présent à tous les niveaux et stimuler l'intuition.

## Comparaison avec des logiciels existants

Le logiciel le plus proche de Géophile, en temps qu'outil de construction et de dessin géométrique, est Euclide [ALL 86], développé à l'I.R.E.M de Grenoble. Euclide est en fait un ensemble de constructeurs de base en géométrie agissant sur des objets de base. On n'y trouve pas les notions de définition déductive et de réévaluation automatique d'une figure modifiée. L'extension du système ne pouvant se faire qu'en Logo, langage d'implantation d'Euclide, celle-ci relève bien plus d'une activité de programmation que d'une expression de connaissances en langage quasi mathématique comme le permet Géophile.

Cabri-Géomètre est un logiciel également développé à Grenoble. C'est outil possède un mécanisme de redéfinition automatique. Cependant les primitives de bases étant figés, la construction déductive est impossible. Cet outil quoiqu'ayant des idées communes avec Géophile n'en est qu'un cas particulier.

## Qualités essentielles de Géophile

Le développement de la géométrie est facilité par la représentation "orientée objets" des connaissances, représentation modulaire donc mieux contrôlable et relativement proche de la description mathématique des concepts [BRA 88]. Géophile est un langage géométrique essentiellement évolutif ce qui en permet des utilisations particulières selon les niveaux d'enseignement.

La gestion de réseaux de construction est un point essentiel. Elle permet d'une part la définition d'une construction de façon déductive ou incrémentale comme pour la méthode déductive de programmation. Elle permet d'autre part toute modification d'une construction en minimisant l'intervention de l'utilisateur grâce au processus de redéfinition automatique de réseaux.

De plus la représentation sous forme de réseaux autorise du calcul formel sur les constructions (définition de méthodes ou fonctions, expression formelle d'un lieu ou d'une enveloppe). Cette propriété lance un pont entre utilisation et programmation, un réseau pouvant être transformé automatiquement en un constructeur de base.

L'utilisation des réseaux peut donc constituer un outil général de conception déductive de programmes.

L'extensibilité du système à toute géométrie et à d'autres applications voisines en font également un outil très puissant.

## Développements futurs

Géophile ayant été conçu dans le but de couvrir les problèmes de constructions en général, il semble désormais possible de développer de nouvelles applications gérant des objets quelconques dépendant de façon quelconque les uns des autres.

Le développement systématique de la géométrie 2D étant en cours, nous envisageons une immédiate extension en 3D, toujours sur une base de géométrie analytique. Il serait intéressant de s'assurer qu'il est facile d'y intégrer des applications voisines telles que la cinématique, l'animation ou la mécanique en introduisant des notions physiques. L'intégration de fonctionnalités des modeleurs 3D de la C.A.O. en mécanique ou de la synthèse d'image semble également possible.

Le développement de Géophile en un progiciel destiné à l'enseignement de la géométrie est à l'étude.

Outre l'améloration sensible de la communication homme-machine mentionnée précédement, un tel développement passe par une exploration et expérimentation systématiques du domaine, actuellement réalisées par des enseignants en géométrie.

Des outils destinés à l'E.A.O , tels que la reconnaissance automatique d'une configuration donnée dans une construction , proposée par un enseignant, pourraient être étudiés.

Une autre direction de recherche concerne l'introduction de géométrie formelle permettant la démonstration automatique et réduisant le rôle de la géométrie analytique à la gestion des dessins affichés.

Enfin, étant donnée la similitude entre programmation déductive et la définition de constructions géométriques il pourrait être intéressant d'étudier les conditions d'utilisation de Géophile dans le cadre d'un enseignement de la programmation.

## REFERENCES

[ALL 86] Jean-Claude Allard, Claude Pascal, Logedif, IREM de Grenoble : "Euclide,un langage pour la géométrie", Cedic/Nathan 1986.

[BEL 78] F. Bellegarde, J.P.Finance, B. Huc, J. Jaray, P. Lescanne, J. Maroldt, C Pair, A Quere, J.L. Rémy : "MEDEE, a type of language for deductive programming method" , Conference on reliable software, ACM, Bonn, 1978.

[BRA 88] Gabriel Braun : "Sur la programmation de constructions géométriques", Thèse d'informatique à l'Université Louis Pasteur de Strasbourg.

[CHA 86] Jérome Chailloux : "LE_LISP 15.2", INRIA, 1986.

[DUC 84] Amédée Ducrin : "Programmation 1 : du problème à l'algorithme", Dunod Informatique 1984.

[FIN 85] J.P. Finance, J. Souquières: "A method and a language for constructing iterative programs", Science of Computer Programming n° 5, p. 201 à 218, North Holland, 1985.

[GOL 83] Adele Goldberg et David Robson : "Smalltalk-80" the language and its implémentation, Addison Wesley, 1983.

[HUC 77] B. Huc : "Mise en oeuvre de la méthode de programmation déductive", Thèse de docteur ingénieur ,Université de Nancy-I, 1977.

[HUL 84] Jean-Marie Hulot : "programmer en Ceyx version 15", rapport INRIA 1984.

[INH 47] B. Inhelder et J. Piaget : "La représentation de l'espace chez l'enfant", PUF.

[PAI 79] C.Pair : "Construction des programmes", R.A.I.R.O Informatique Computer Science (Vol.13, n°2, 1979 , p. 113 à 137).

[PLU 86] François Pluvinage, Jean-Claude Rauscher : "La géométrie constructive mise à l'essai", Petit X 1986 n°11, IREM Grenoble.

[SOU 80] Jeanine Souquières : "Méthode pour la construction d'algorithme dans le cas de conflit de structure", actes du Congrès de l'AFCET, p. 203 à 215, 1980.

[WIS 84] Patrick Henry Wiston et Bergdold Klaus Paul Horn : "Lisp", Addison Wesley, 1984.

[WER 85] H. Wertz : "Lisp, une introduction à la programmation", Masson, Paris,1985