THÈSE

présentée par

Yves BAULAC

pour obtenir le titre de

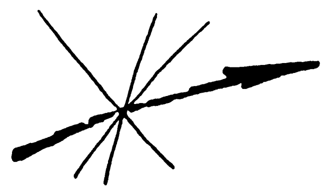
DOCTEUR DE L'UNIVERSITÉ JOSEPH FOURIER - GRENOBLE I

(arrêté ministériel du 5 Juillet 1984)

(spécialité: **INFORMATIQUE**)

Un micromonde de géométrie,

Cabri-géomètre



Thèse soutenue le 7 Février 1990

Composition du jury:

Jacques **MOSSIÈRE** (président)
Jean-François **DUFOURD**
Jean-Marie **LABORDE**
Janine **ROGALSKI**
Laurent **TRILLING**
Martial **VIVET**

Thèse préparée au ***Laboratoire de Structures Discrètes et de Didactique*** **(IMAG)**

Merci, Jean-Marie, pour ton accueil chaleureux et tes idées généreuses qui ont mené à la rédaction de cette thèse.

Je remercie vivement Jean-François Dufourd et Martial Vivet qui ont bien voulu rapporter sur ce travail.

Je remercie également Jacques Mossière, Janine Rogalski et Laurent Trilling pour leur confiance donnée en participant au jury.

Last but not least, merci à tous ceux qui ont participé à la réalisation de Cabri-géomètre et de ce texte.

Plan

Introduction

Mon travail de recherche s'est déroulé au sein du Laboratoire de Structures Discrètes et de Didactique à l'IMAG avec pour thème la *conception et la réalisation d'applications interactives pour l'enseignement de la géométrie.*

Recherche *appliquée* parce que les résultats ont été mis en œuvre dans un logiciel d'aide à l'apprentissage de la géométrie, un cahier de brouillon interactif pour la géométrie, Cabri-géomètre.
Recherche *d'équipe* parce que pour mener à bien un tel projet, des informaticiens, des pédagogues, des didacticiens, des cogniticiens, des experts de la matière en jeu - ici donc des mathématiciens - doivent mettre, ont mis ou mettront leurs efforts en commun. Les idées originales de ce travail sont le fruit d'une réflexion collective de l'équipe de Cabri-géomètre et je transmets ici en quelque sorte l'état d'avancement de ces recherches.
Notamment Frank Bellemain, qui a été depuis le début un des principaux artisans de la réalisation informatique. La thèse de didactique qu'il soutiendra dans quelques mois recoupera donc dans une certaine mesure le présent document.
Recherche *inachevée* parce qu'il est toujours difficile de mettre un terme à un travail dont le support est un projet en constante évolution. Comme cela n'est jamais le bon moment d'acheter un ordinateur parce qu'il faut attendre le nouveau modèle qui va révolutionner l'ancien. A l'heure où Cabri-géomètre cesse d'être la "chose" d'une petite équipe pour se structurer en un projet plus vaste à caractère pluridisciplinaire et dépassant le cadre de notre seul laboratoire, il me paraît intéressant de faire le point sur les réalisations et les ambitions qu'il a suscitées.

Les établissements d'enseignement sont aujourd'hui dotés de machines dont la puissance de calcul et la qualité des périphériques ne sont souvent pas pleinement utilisées par les applications destinées à l'éducation.
Dans le monde de la micro-informatique, la communication graphique a pris le pas sur la communication alphanumérique, l'interactivité des logiciels est une règle d'or, de nouveaux traitements de l'information sont mis en œuvre. Le monde de l'éducation, ou tout du moins de la formation initiale, à cause sans doute de l'étroitesse du marché, n'a pas encore été touché par ces développements.

La géométrie est, à notre avis, un terrain d'expérience privilégié pour essayer de nouvelles idées dans une optique d'apprentissage:
Il existe de nombreux travaux et réalisations proposant des dispositifs informatiques pour l'enseignement de la géométrie. Citons ici GEOMETRY TUTOR ([Anderson & al. 85] et les dérivés du langage de la tortue LOGO ([Papert 80]), deux archétypes pour l'aide à la démonstration et l'exploration de figures respectivement.
Le passage de la feuille de dessin à l'écran d'un ordinateur ne change pas la nature de la représentation d'une figure, ici et là plane et limitée en taille et en résolution.
Les théories géométriques sont axiomatiques (et fondées sur la logique des prédicats) ou algébriques (et associées au corps des réels) et l'on peut raisonnablement espérer mettre au point des outils partiels de démonstration automatique.
Il s'agit là d'une formation initiale, et les options pédagogiques sont donc susceptibles d'être reprises dans d'autres domaines.

Les figures géométriques sont des dessins que l'on élabore, gomme, rature, surcharge, modifie pour chercher des solutions, essayer des idées, ...

Cabri-géomètre transpose ces activités de la feuille de papier à l'écran d'un ordinateur. Le confort d'utilisation et la convivialité des interfaces ont été une des préoccupations majeures dans la définition de cette transposition. La puissance de calcul de la machine permet non seulement d'améliorer la rapidité et la justesse des constructions mais aussi de proposer de nouveaux moyens comme l'animation des figures, la simulation de transformations géométriques, les ruptures d'hypothèse, grâce auxquels les invariants et autres phénomènes géométriques sont facilement observables.

Une figure de géométrie est composée d'objets (points, droites, cercles,...) dont la position relative est précisée par un ensemble de relations géométriques (appartenance, équidistance, parallélisme,...). La caractéristique essentielle de Cabri-géomètre est la possibilité de modifier dynamiquement la position des objets dits *quelconques* dans une figure. Celle-ci est alors réactualisée en temps réel, les relations définies entre les objets à la construction étant conservées. On peut ainsi visualiser rapidement un grand nombre de ***cas de figures*** et accéder aux propriétés de la ***classe de figures*** correspondante à un énoncé. Par exemple en déplaçant sur l'écran un des sommets d'un triangle (en le "tirant" à l'aide d'une souris), on *voit* que ses médiatrices restent concourantes quelle que soit la forme du triangle.

Cabri-géomètre est un ***Cahier de Brouillon Interactif***.
Cette métaphore du cahier de brouillon est au centre du projet CABRI ([Habib & Laborde 83]) dont Cabri-géomètre est issu. D'autres cahiers de brouillon ont été réalisés, notamment pour la théorie des graphes ([Tallot 85], [Benzaken 86], [Laborde 87], [Baudon 89]).
Le logiciel Cabri-géomètre est actuellement diffusé commercialement. Il a reçu le Trophée APPLE 88 de la meilleure réalisation pour l'éducation et a été récemment agréé par le Ministère de l'Education Nationale sous forme d'une licence mixte facilitant son acquisition par les établissements d'enseignement.

Dans la première partie de ce document, intitulée "***Dans le domaine de l'E(I)AO,...***" nous abordons quelques aspects de la problématique de l'enseignement assisté par ordinateur en présentant notamment les tuteurs et les micromondes comme étant aux deux extrémités sur l'éventail des environnements d'apprentissage. Le tuteur pilote l'apprenant pour lui transmettre des connaissances sur le domaine, le micromonde propose à l'apprenant de découvrir, d'explorer les lois du domaine en conduisant ses propres expériences.
Le concept de micromonde est très représentatif des activités que l'on propose lors de l'enseignement de la géométrie de nos jours (en France tout du moins). En effet, les activités de construction de figures, d'exploration de ces figures pour y mettre en évidence des invariants géométriques, sont privilégiées par rapport à la seule démarche de la démonstration d'une propriété.
Nous exposons ensuite le cahier des charges du Cabri-géomètre réalisé, en soulignant l'originalité des outils et des interfaces proposés ainsi que leur justification sur le plan pédagogique. Un historique du projet rappelle les principales étapes, réalisations et personnes qui ont marqué son évolution.

Une 2ème partie est consacrée aux ***spécifications fonctionnelles*** des différents éléments de ce micromonde. Nous détaillons la représentation dans Cabri des concepts de la géométrie (objets, relations, transformations,...). L'exposé des fonctions essentielles (construction, déformation, redéfinition des figures) et des principes de l'interface nous permettra de mettre en évidence les problèmes posés par cette modélisation ainsi que les limites des structures associées dans Cabri aux figures de géométrie.
La thèse sous-jacente à ce document est que la définition d'un micromonde dans un contexte d'apprentissage nécessite la mise en œuvre de moyens logiciels importants et de processus que l'on peut qualifier d'intelligents, tant pour assurer la cohérence interne du système que pour aller vers un véritable environnement tutoriel.

Quelle sont les conséquences de l'introduction de nouveaux outils, inédits dans l'environnement papier-crayon-règle-compas? Peut-on mettre en évidence une appropriation plus rapide ou plus profonde des concepts géométriques avec ce nouvel environnement? L'apprenant y trouve-t-il une nouvelle motivation pour la géométrie? Un renouvellement des pratiques pédagogiques est-il nécessaire? La réponse à ces questions, essentielle puisqu'elle constitue la seule *évaluation et preuve du programme*, n'entre pas dans le cadre de ce travail. D'autres chercheurs se sont engagés dans cette recherche ([Bellemain 88], Keskessa, Tahri, Strässer) et leurs résultats serviront de base à la définition des futurs Cabri-géomètre.

I. Dans le domaine de l'E(I)AO...

E (pour enseignement, exploration, échanges, environnement,...) A (pour assisté, accessible, apprentissage, amélioré, administré, ...) O (pour ordinateur): Martial Vivet [Vivet 89] en présentant de nombreuses interprétations de l'acronyme "EAO" met l'accent sur la multiplicité des approches possibles.
Nous ferons tout d'abord le constat d'un **manque de maturité** de cette branche de l'informatique. Les systèmes de **communication de connaissances** que sont les logiciels éducatifs modernes requièrent la mise en œuvre de processus intelligents dont nous préciserons le rôle.
Nous évoquerons ensuite les deux principales approches pédagogiques que sont les **tuteurs intelligents** et les **micromondes**.

Cela nous amènera à définir les objectifs de Cabri-géomètre puis à donner les principaux éléments du **cahier des charges** et de son originalité.
Enfin nous donnerons un aperçu des diverses réalisations et projets en matière de logiciels pour l'enseignement de la géométrie en décrivant les divers travaux dans ce domaine.

1.1 quelques généralités sur les logiciels éducatifs

1.1.1 un constat de manque de maturité

• La prolifération de logiciels éducatifs, sans doute plus de 10.000, reflète la vigueur de cette branche de l'informatique que l'on a considérée dès les années 1970 comme un support révolutionnaire pour l'enseignement. Les réalisations ne sont en général pas à la hauteur des ambitions initiales ([Lawler 86]) et l'intérêt pédagogique de la plupart de ces logiciels est réduit. Cet état de fait nuit certainement à la crédibilité des recherches dans ce domaine et à l'enthousiasme partagé des utilisateurs finaux, à

savoir les professeurs et leurs élèves. Il est aussi le reflet de la complexité des objectifs à atteindre et de l'ampleur du travail restant à accomplir.
On peut également constater une certaine "dispersion" dans la dénomination des logiciels. C'est ainsi que sous le terme générique de logiciels éducatifs on trouve des didacticiels, des tutoriels, des micromondes, des tuteurs intelligents, des systèmes auteurs, des systèmes d'EAO traditionnels, des imagiciels, des ludiciels, des environnements d'apprentissage, des systèmes d'EAOI ou des systèmes d'EIAO. Les auteurs anglo-saxons ont aussi une terminologie riche: ITS pour Intelligent Training System ou Intelligent Tutoring System, (I)CAI pour (Intelligent) Computer-Aided Instruction, CAL pour Computer Assisted Learning, knowledge communication systems, ...
Cette volonté de distinction est en partie une conséquence de l'inadéquation des réalisations actuelles.

• L'engouement pour ce nouveau moyen de communication qu'est l'ordinateur a conduit, dans un premier temps, de nombreux enseignants à écrire eux-mêmes les programmes informatiques, mais avec le plus souvent un manque de professionnalisme tant au point de vue de la conception que de la réalisation. L'élaboration d'un logiciel éducatif requiert la mise en commun des compétences d'enseignants, de didacticiens, d'experts du domaine et d'informaticiens.
Les expérimentations et évaluations de tels logiciels sont longues et difficiles à mettre en œuvre mais sont pourtant indispensables à la mise au point définitive. Trop souvent cette phase est réduite à quelques expérimentations "in vitro" hors du contexte final d'utilisation.

• Le logiciel éducatif utilise sous une présentation différente les supports classiques d'apprentissage: cours, questionnaire, entraînement, jeu, simulation, etc. Il n'y a souvent qu'une transposition des outils des environnements classiques, transposition qui nuit généralement à la flexibilité des activités. L'attrait de la nouveauté et l'aspect ludique du maniement de l'ordinateur sont alors vite épuisés.

• l'impact de l'enseignement assisté par ordinateur est limité "sur le terrain" par d'autres considérations indépendantes de la qualité des produits, comme

- l'inadéquation du parc de machines disponibles dans les établissements d'enseignement,
- la faible rentabilité économique de ce type de logiciel, qui limite les réalisations "industrielles",
- parfois même l'inertie du personnel éducatif pour s'adapter à une nouvelle technologie.

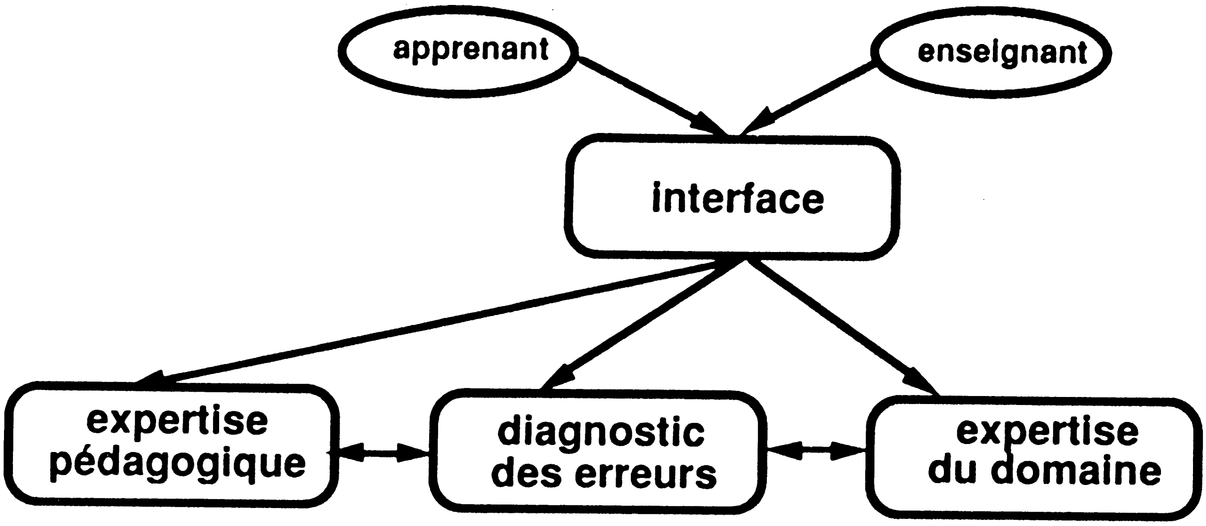
• D'un autre côté depuis quelques années on assiste à l'émergence d'une nouvelle génération de systèmes informatiques pour l'enseignement. Des techniques issues de l'intelligence artificielle y sont mises en œuvre pour codifier les connaissances de la matière enseignée, du professeur et de l'élève et les vrais problèmes semblent être dorénavant posés sinon résolus.
Citons quelques systèmes souvent donnés comme référence : AMALIA, APLUSIX, BUGGY, GUIDON, SCHOLAR, SOPHIE, WEST, GEOMETRY TUTOR (décrits dans [Nicaud & Vivet 87] et [Wenger 88]) et auxquels nous nous référerons dans les paragraphes suivants.
La plupart n'ont cependant pas encore dépassé le stade du développement et ne sont pas diffusés en dehors de leur cadre expérimental: d'une part la complexité des tâches a conduit à focaliser tel ou tel aspect du problème au détriment de la mise au point finale du produit, d'autre part leur fonctionnement nécessite souvent des machines puissantes.
Cabri-géomètre, en tant que système expérimental d'aide à l'enseignement de la géométrie, participe à cet effort.

1.1.2 la communication des connaissances

• Si l'EAO, dit traditionnel, avait pour but la production de logiciels éducatifs par des enseignants, les recherches en EIAO ont pour objectif la conception d'environnements spécialisés prenant en charge, dans une certaine mesure, des éléments jusque là dévolus à l'enseignant, à savoir la généralisation des concepts, l'interaction pédagogique, la présentation des connaissances et l'individualisation de l'enseignement. Les tendances modernes de recherche en enseignement intelligemment assisté par ordinateur ont fait l'objet de nombreux articles et ouvrages de synthèse ([Wenger 88], [Nicaud & Vivet 87a], [Yazdani 88], [Dede 87], [Lelouche 87]).

• Wenger dans "Artificial Intelligence and Tutoring Systems" ([Wenger 88]) présente les logiciels éducatifs comme des ***systèmes de communication de connaissances***.
On peut identifier quatre entités intervenant dans ces dispositifs de communication : l'apprenant, l'enseignant, la matière enseignée et l'ordinateur.
La matière enseignée est l'**objet de la communication**, l'apprenant est le **destinataire de la communication**, l'ordinateur est un support à une nouvelle **forme de la communication** , l'**art de cette communication** c'est la pédagogie de l'enseignant.
Ces quatre aspects de la communication ne peuvent être gérés, pris en compte, simulés que par des processus intelligents. Les connaissances que sous-entendent ces processus peuvent-elles être représentées par des modèles calculables?

L'architecture d'un tel système de communications de connaissance s'articule autour de quatre modules selon un schéma du type suivant, que l'on retrouve chez de nombreux auteurs:



Remarquons que l'apprenant semble souvent avoir l'exclusivité du dialogue avec le système et que l'enseignant est omis. Son intervention avant, pendant ou après le travail de l'apprenant restera pourtant, à notre avis, essentielle.

la connaissance experte

La connaissance des lois du domaine est nécessaire à la fois pour résoudre les problèmes posés à (ou par) l'utilisateur et pour la gestion interne des données.

En géométrie, comme dans la plupart des disciplines, on trouve plusieurs formes de connaissances [Nicaud & Vivet 87] :

- des faits géométriques (ex: les points d'un cercle sont équidistants du centre)
- des règles de base, des théorèmes (ex: la transitivité du parallélisme)
- des méthodes, des stratégies, des heuristiques, des contextes d'application [Chouraqui & Inghilterra 87] guidant l'emploi de ces formes de connaissance (ex: utiliser les cas d'égalités des triangles pour...).

Les *meta-connaissances*, jouent un rôle différent dans un système cognitif classique et dans un système pour l'apprentissage, en ce sens qu'elles doivent également être transmises à l'apprenant, notamment dans les domaines de base comme la géométrie dont une des finalités est l'apprentissage du raisonnement.

Les systèmes experts classiques développent une base de faits en cherchant à appliquer les éléments d'une base de règles, la sélection et l'application de ces règles étant du ressort d'un moteur d'inférence.
La base de règles contient donc les éléments d'une théorie modélisant le domaine. Le moteur d'inférence résoud le cas échéant les conflits entre les différentes règles applicables. Par exemple le moteur PROLOG lit les clauses dans l'ordre où elles lui ont été fournies, le moteur OPS5 cherche la règle la plus récemment utilisée ou celle présentant les préconditions les plus simples. Ces stratégies privilégient en général l'efficacité sur "l'intelligence" des solutions trouvées.
Une codification explicite des meta-connaissances (les plans, les heuristiques), abordée par exemple dans AMALIA ([Vivet 87]) ou ARCHIMEDE ([Chouraqui & Inghilterra 87]) reste encore difficile et incomplète.

Une première étape, que nous abordons dans Cabri-géomètre (voir le §2.3.5), est de conférer au système la capacité de résoudre les problèmes de validation de faits géométriques par des mécanismes internes, éloignés du raisonnement humain. L'explicitation et l'explication des étapes de la résolution de ces problèmes ne seront pas accessibles. Nous verrons que la connaissance des propriétés géométriques d'une figure est nécessaire, non seulement pour donner à l'apprenant un instrument de validation, mais aussi pour assurer la cohérence des interfaces et des figures représentées à l'écran.

la connaissance pédagogique et la connaissance de l'apprenant

Dans la plupart des domaines enseignés et plus encore pour les enseignements de base comme la géométrie, les lois du domaine sont bien arrêtées et le problème est un problème de représentation plus que de modélisation de la connaissance.

Par contre l'interaction didactique et la prise en compte des connaissances fragmentaires d'un apprenant suppose une modélisation qui n'est qu'abordée par certains systèmes comme DEBUGGY [Burton 82] ou GEOMETRY TUTOR [Anderson & al. 85].
L'interaction didactique peut s'appuyer sur des scénarii, des règles de guidage, des stratégies pédagogiques, des générateurs d'exercices qui doivent être paramétrés par le niveau de connaissances de l'apprenant.

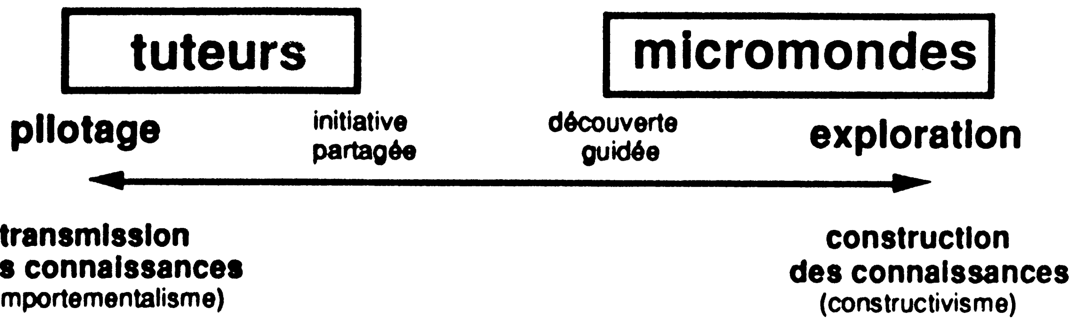
Les fonctions mêmes du modèle de l'apprenant ne sont pas totalement définies. On peut distinguer des fonctions correctives, des fonctions élaboratives, des fonctions stratégiques, des fonctions diagnostiques, des fonctions prédictives, des fonctions évaluatives ([Self 87]).

La prise en compte ***dynamique*** des connaissances correctes ou erronées de l'apprenant, de sa démarche, et l'intervention résultante sont des tâches encore peu maitrisées, que nous n'avons pas abordées dans le Cabri-géomètre actuel.

Par contre la prise en compte ***statique*** des caractéristiques de l'élève et de l'interaction souhaitée peut être envisagée. Une définition adaptée des interfaces, le paramétrage de la complexité des outils d'apprentissage proposés, le choix du mode d'interaction, entrent dans ce cadre et nous verrons dans les paragraphes suivants leur application dans Cabri-géomètre.

1.1.3 tuteurs et micromondes

L'enseignement traditionnel propose des cours magistraux, des travaux dirigés, des travaux pratiques ou des exercices d'entraînement.
L'initiative dans la conduite de l'activité y est répartie entre l'enseignant et l'apprenant.
Dans un dispositif informatique, le système et l'utilisateur interagissent de la même manière. Le choix du ***mode d'interaction*** permet de caractériser les logiciels éducatifs. On peut représenter linéairement ces modes d'interaction, du plus au moins directif, et leur associer un type de logiciels et une théorie de l'apprentissage:



Il n'y a pas de discontinuité entre ces deux modes d'acquisition que sont la transmission et la construction de connaissances. C'est cependant par ce biais que l'on trouve souvent dans la littérature une dichotomie entre deux types de logiciels éducatifs, les tuteurs intelligents et les micromondes.

La transmission des connaissances

L'enseignant, qui transmet des connaissances, impose une certaine démarche à l'apprenant. L'enseignement est directif et piloté. Le déroulement d'une session est une suite de stimulus-réponses, la suite des étapes est déterminée par les connaissances préalables et les erreurs de l'apprenant.
Cette approche comportementaliste (behaviourism) se traduit par un enseignement programmé qui, par sa nature même, se prête parfaitement au support informatique. Historiquement les premiers logiciels éducatifs ont suivi ces concepts sous la forme rudimentaire (questionnaire à choix multiples sans rétroaction) qu'autorisait le développement des technologies informatiques de l'époque.
Les récents ***tuteurs intelligents*** sont issus de cette approche directive, le plus souvent tempérée par une *initiative partagée* qui permet à l'apprenant d'influer dans une certaine mesure sur le déroulement de la

session et la nature des connaissances transmises.
SCHOLAR [Carbonell 70] est un des premiers logiciels de cette génération des tuteurs intelligents. L'extrait suivant donne une idée du dialogue avec SCHOLAR:

Scholar>	*What is the language of Argentina?*
Student >	Probably spanish
Scholar>	*Very good. Approx what is the area of Argentina?*
Student >	Tell me something about Peru
Scholar>	*Peru is a country*
	It is located in South America
	The capital of Peru is Lima
	Now answer the question you didn't answer before.

La construction des connaissances

L'apprenant est ici au centre du processus d'apprentissage, c'est lui qui a l'initiative. Le travail de l'enseignant consiste à proposer des activités permettant l'exploration, la découverte et la construction de nouvelles connaissances. Libre de sa démarche, placé en situation de découverte, exploitant son potentiel cognitif, procédant par expérimentations et tâtonnements, l'apprenant élabore ainsi ses propres structures mentales [Picard &Braun 88]. Cette approche est directement issue de l'approche constructiviste de l'apprentissage mise en évidence par Piaget [Piaget 47]. L'archétype des réalisations informatiques fondées sur cette approche est le langage LOGO développé par Seymour Papert au MIT dès 1971: «ce n'est plus l'ordinateur qui programme l'enfant, mais l'enfant qui programme l'ordinateur» [Papert 80].
L'acquisition de connaissances, produit d'une recherche personnelle, ne se limite pas ainsi à des connaissances factuelles, mais englobe également les connaissances liées à des savoir-faire, des raisonnements, des heuristiques, des stratégies.
Le dispositif informatique fixe le cadre de travail et fournit un ensemble d'objets du domaine et un ensemble d'outils pour conduire des expériences. L'ensemble des objets et des procédures que l'on peut leur appliquer constitue un petit sous-ensemble du monde physique, un ***micromonde***. C'est sous ce nom (introduit par Papert à propos de la

tortue LOGO mais Winograd désignait déjà le monde des blocs comme un "miniworld" [Lawler 87]) que l'on désigne ces environnements d'apprentissage dans lesquels l'apprenant construit lui-même ses connaissances.

L'avenir des logiciels éducatifs est sans doute dans la coexistence de ces deux modes d'interaction dans un même système, le contexte déterminant la part d'initiative de chacun des intervenants
«Les phases de guidage, de découvertes dans un micro-monde adapté, de résolution libre ou guidée d'exercices, de transmission directe de connaissances, d'approche de nouveaux concepts, de création par des situations problèmes par l'apprenant sous des contraintes données... doivent pouvoir être gérées harmonieusement [Nicaud & Vivet 87]».

Les développements futurs de Cabri-géomètre iront dans ce sens.

1.1.4 le cas de la géométrie

Il semble raisonnable, quand on pose le problème de l'enseignement assisté par ordinateur, de fixer un domaine expérimental précis. Une démarche de travail, des interfaces, des types d'interaction pourront y être dégagés et s'appliquer à d'autres domaines.
Le choix de la géométrie élémentaire comme matière d'enseignement n'a pas été dans notre cas le résultat d'une analyse, mais celui d'un concours de circonstances: l'apport des mathématiciens et de l'équipe de didactique du laboratoire, l'expérience acquise avec d'autres logiciels graphiques, l'existence de nombreux travaux sur le domaine (décrits au §1.3) ,...

• La géométrie élémentaire est une formation de base. L'objectif d'un enseignement de la géométrie n'est pas seulement la connaissance des formes du monde physique. L'apprentissage de la rigueur et du raisonnement déductif, de la créativité et de l'imagination, de la dextérité

sont également essentiels.
La représentation des concepts géométriques est souvent une figure, un dessin sur une feuille de papier. La traduction de ce dessin à l'écran d'un ordinateur ne pose guère de problèmes quand à la nature de la représentation. En effet, le résultat d'une construction géométrique (les traits de crayon ou les pixels noircis sur un écran) garde la même nature, plane, limitée en dimension et limitée en résolution. On évite ainsi les problèmes de modélisation liés aux domaines non traditionnellement reproduits sur une surface plane.

• Par ailleurs, les théories liées à la géométrie élémentaire sont des théories de la logique du premier ordre ou des théories algébriques pour lesquelles il est raisonnable d'espérer mettre au point des outils de démonstration automatique.

• Le développement des environnements interactifs graphiques ([Foley & Van Dam 84]), notamment pour la CAO, a mis en évidence certains critères de traitement et d'affichage de l'information graphique, qui trouvent une application dans le traitement des figures de la géométrie scolaire. Et les interfaces graphique-souris que l'on retrouve désormais autour des principaux systèmes d'exploitation des microordinateurs ou des stations de travail donnent un cadre préalable et simplificateur au développement.

• La tendance actuelle semble privilégier une géométrie construite par rapport à une géométrie des structures ([Pluvinage & Rauscher 86]) dans laquelle la manipulation des images est essentielle.
Donner à l'apprenant la possibilité de construire lui-même les figures de la géométrie, puis de les explorer semble un préalable à tout système d'apprentissage de la géométrie. Et le raisonnement, la démonstration de faits géométriques, s'appuie sur les phénomènes mis en évidence par cette activité d'exploration.

• Le premier objectif que nous nous sommes fixés avec Cabri-géomètre est donc de réaliser un éditeur de figures géométriques, un éditeur capable non seulement d'afficher une figure, mais aussi muni de

fonctionnalités comme par exemple de permettre un accès immédiat à *toutes* les configurations représentatives d'une construction donnée ou de modifier la définition des objets de la figure.
Cabri est l'acronyme de **Ca**hier de **br**ouillon **I**nteractif (ou **I**nformatique ou **I**ntelligent?). Sur ce cahier de brouillon, l'apprenant pourra mettre en œuvre ses idées, expérimenter des phénomènes, identifier des propriétés, comme il le fait sur son cahier de brouillon classique, mais avec des potentialités à la hauteur de la puissance de calcul d'un ordinateur.

1.2 Le cahier des charges

Le développement de Cabri-géomètre n'a pas suivi linéairement la trilogie classique "cahier des charges, spécifications, implémentation". Les idées originales de Cabri-géomètre tant au point de vue des fonctionnalités que de l'interface ou de l'implémentation se sont dégagées d'une suite d'essais-erreurs, de maquettes et de prototypes, la version actuelle n'étant que le dernier élément de cette suite.
Le cahier des charges que nous allons détailler n'est pas le résultat final d'une analyse didactique et informatique mais une expression de l'état d'avancement de nos recherches.

Nous allons faire d'abord un **rappel chronologique** des différentes étapes du projet et des réalisations. Nous donnerons ensuite les **objectifs** de l'utilisation de Cabri-géomètre au cours d'activités en géométrie , les **critères de qualité** que doit présenter un tel logiciel, et les **fonctionnalités** nécessaires à la réalisation de ces objectifs.

1.2.1 historique du projet

L'histoire du projet a commencé en 1983 avec la formation du groupe CABRI [Habib & Laborde 83]) dont voici un extrait du document de constitution :

« *Dans le domaine de la théorie des graphes ou des ensembles ordonnés comme dans celles qui en font usage, il arrive très fréquemment que l'on ait recours à la réalisation de "petits dessins", formés essentiellement de sommets et d'arêtes; cela dans le but de les manipuler, de leur appliquer concrètement certaines transformations, ne serait-ce que pour vérifier telle ou telle propriété, conforter ou infirmer une idée, une conjecture.*

Malheureusement, cette pratique menée sur un cahier de brouillon ordinaire souffre de limitations, dues par exemple à la taille relativement limitée des exemples que l'on peut traiter à la main ou encore au fait que la vérification d'une propriété peut-être fort longue (comme c'est le cas, par exemple, lors de l'examen d'une "criticalité"); l'idée est donc naturelle de vouloir élargir les possibilités d'exploration par la mise à la disposition des chercheurs du domaine (et d'étudiants) d'un outil graphique où le traditionnel cahier de brouillon serait remplacé par l'écran interactif d'un terminal associé à toute la puissance de calcul et de mémorisation de l'ordinateur.
C'est la réalisation d'un tel outil, sensé être d'accès aussi facile que possible pour être utilisé, pour partie, par des non informaticiens, qui constitue l'essence du projet CABRI. (...) » .
A cette époque CABRI etait l'acronyme de cahier de brouillon informatique.

Après la réalisation des premières maquettes et versions de Cabri-graphes ([Benzaken 86], [Tallot 85]), l'idée a surgi de transposer ces concepts dans un autre domaine qui présente des similarités quant à la représentation graphique de concepts mathématiques, la géométrie.
Il a été ainsi formulé par Jean-Marie Laborde en 1986 ([Laborde 86]) :
«Développement d'un système interactif de traitement de figures géométriques, telles celles de l'enseignement secondaire.
Les primitives usuelles de dessin de figures géométriques seraient disponibles à partir de menus déroulants.
Il serait possible de modifier l'ensemble d'une figure en ne changeant que les caractéristiques des éléments de base qui l'ont déterminée.
Ce logiciel permettrait aux élèves de l'enseignement secondaire de tracer rapidement les figures proposées lors de problèmes, de multiplier les exemples, voire les contre-exemples à leurs conjectures et serait ainsi un outil d'aide à la résolution de problèmes.
Il est à développer sur Macintosh et s'inspirera des concepts déjà développés pour Cabri, logiciel de traitement de graphes. (...) »

C'est à partir de ces spécifications qu'une première maquette a été réalisée (programmée en langage C sur Macintosh) en 1987 ([Baulac & Cayet 87]). Des expérimentations, menées par Franck Bellemain et Bernard Capponi (Bellemain 88]), dans des classes de collège ont permis de mettre en évidence et de remédier à certaines incohérences et difficultés d'appropriation par les élèves de ce prototype.
La faisabilité du cahier des charges initial ayant été démontrée, l'étape suivante a consisté à ne plus considérer Cabri-géomètre comme une simple alternative commode aux outils classiques pour faire de la géométrie (papier, crayon, règle,...), mais comme un véritable outil d'aide à l'apprentissage et à l'enseignement de la géométrie scolaire. Par véritable outil il faut entendre outil utilisable c'est à dire ayant à la fois un degré satisfaisant de fiabilité, de cohérence et présentant un éventail de fonctionnalités justifiant son emploi dans des classes, non pas seulement pour des expérimentations dans le cadre de recherche didactique, mais aussi "en vraie grandeur" par des professeurs et leurs élèves.

La possibilité de construire la plupart des figures de la géométrie scolaire, l'accès à la notion de classe de figures par une modification continue des objets de base, les outils de visualisation de lieux, de mesure, la paramétrisation possible par l'enseignant, l'extensibilité des primitives de construction, les facilités d'édition, la normalisation de l'interface datent de cette mise à niveau.
La mise au point définitive de ces premières versions du logiciel a été effectuée en vue de sa diffusion extérieure au laboratoire et je tiens à souligner le côté quelque peu rébarbatif de cette phase ainsi que le temps important que Franck Bellemain, Jean-Marie Laborde et moi-même y avons consacré.
Cabri-géomètre est actuellement dans sa version 2.0 pour le Macintosh et 1.0 pour les IBM/PC.

Si nos travaux se sont soldés par la réalisation de logiciels commercialisables, ils sont aussi à l'origine de nouvelles ambitions.
Un groupe de travail s'est constitué en 1988 sur le thème "Vers un Cabri intelligent". L'objectif est à terme de fournir un système intelligent d'apprentissage de la géométrie, centré sur l'utilisation d'un micromonde, mais intégrant la conduite par le système du déroulement d'une session ([Laborde & Trilling 89]).
Ceci nécessite au préalable une réécriture complète des programmes pour permettre entre autres une configurabilité des interfaces et l'intervention d'utilisateurs semi-finaux (didacticiens) au niveau du fonctionnement du logiciel.

1.2.2 objectifs

Qu'attend-on de l'enseignement de la géométrie?
La géométrie fait partie des enseignements de base de l'enseignement secondaire. Elle est même un de ceux dont la pratique est considérée actuellement comme très importante dans l'apprentissage des mathématiques: « *Les mathématiciens reconnaissant, pour beaucoup de leur réflexions, le bénéfice d'images mentales issues le plus souvent de la géométrie euclidienne plane, les utilisations ont largement évolué et font appel à bien d'autres connaissances que le seul repérage, le ou plutôt les traitements d'images se sont introduits avec force dans de multiples secteurs d'activité (très souvent aujourd'hui dans l'industrie, on travaille sur image avant de se lancer sur la matière* » [Pluvinage & Rauscher 88].
L'intérêt de la matérialisation graphique d'objets mathématiques et l'importance des raisonnements déductifs que les figures suscitent en sont les principaux attraits.
Les bénéfices qu'un élève est sensé retirer de l'apprentissage de la géométrie vont bien au-delà de la simple connaissance géométrique.

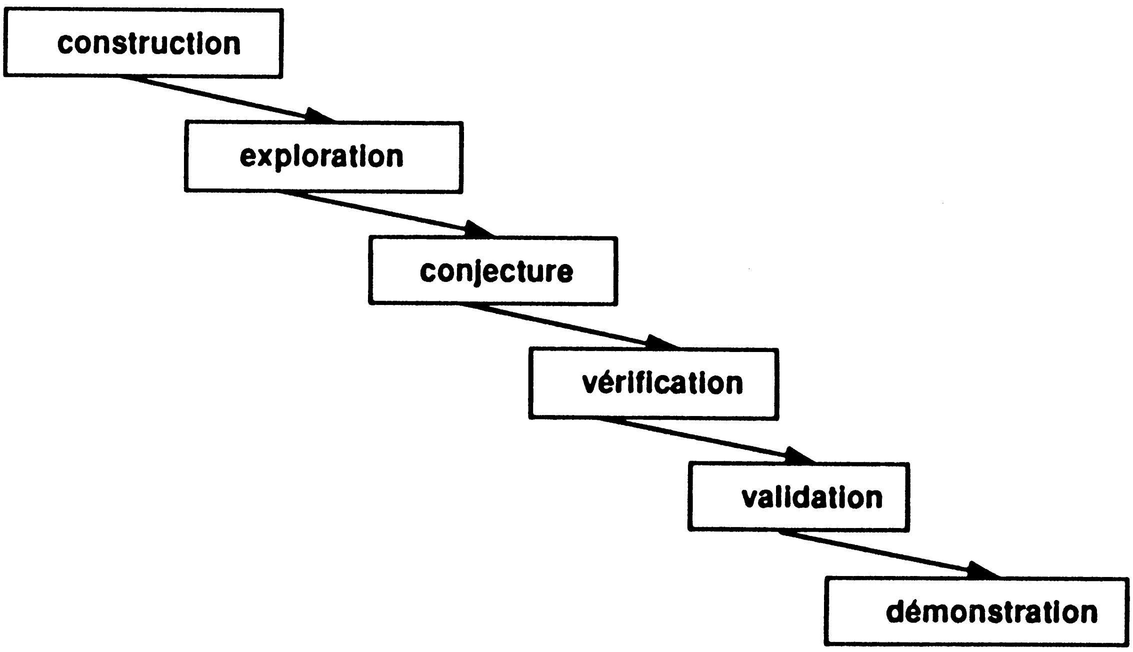
D'une manière un peu simpliste on peut dire que la géométrie développe chez l'apprenant:

• La **dextérité** avec une maitrise du maniement des outils classiques (règle, compas, rapporteur, ...) lors des constructions.

• La **créativité** et l'imagination (constructions complexes, élaboration de conjectures, découverte de propriétés dans une classe de figures ou d'une méthode de construction)

• Le **raisonnement** (déduction, démonstration de théorèmes, généralisation, ...)

Les activités de géométrie proposées aux élèves sont extrêmement variées, mais on peut dans la plupart discerner les différentes phases décrites dans le schéma suivant:



Le Cabri-géomètre que nous présentons s'attache plus aux phases de **construction** et d'**exploration** de la figure. Des instruments de mesure permettent d'opérer certaines **vérifications** sur une figure, et une **validation** de certaines propriétés déductibles de la figure peut être effectuée par le système. L'explicitation et l'explication de démonstrations n'est pas réellement abordée.

La métaphore du **cahier de brouillon interactif** est une image exacte du fonctionnement de ce micromonde. Il n'y a pas d'interaction didactique et l'élève est le maitre du déroulement de sa session de travail.
Le système n'intervient que pour signaler une incohérence ou une impossibilité lors de l'élaboration ou de la manipulation de la figure.
L'intervention de l'enseignant pour proposer des activités ou évaluer les résultats reste essentielle: le système lui propose des outils permettant de configurer le logiciel et de reproduire le déroulement d'une session.

1.2.3 quelques critères qualitatifs

On peut très pragmatiquement donner quelques critères qui conditionnent la bonne acceptation d'un logiciel éducatif par ses utilisateurs finaux, à savoir les enseignants et leurs élèves. Le souci de respecter le mieux possible ces critères qualitatifs a été présent tout au long de la réalisation de Cabri-géomètre.

• Le logiciel doit être un produit fini:
Pour être utilisable "in vivo" dans le monde de l'éducation, nous avons développé Cabri-géomètre dans une optique de diffusion extérieure au laboratoire, ce qui nécessite une phase de mise au point et de finition comme par exemple la gestion complète des entrées sorties (enregistrement , impression soignée des figures, ...). Cette phase au contenu peu scientifique est cependant nécessaire si l'on veut effectivement s'assurer de l'adhésion d'un public large dont les retours d'informations sont indispensables aux développements futurs.

• Dans le même ordre d'idées, le choix de la machine cible conditionne la bonne acceptation du logiciel. Pour une application telle que Cabri-géomètre, le choix d'un micro-ordinateur disponible dans les établissements scolaires du premier et du second cycle s'impose. Nous avons développé les applications sur les micro-ordinateurs de la gamme

Macintosh et nous avons retenu les normes d'interface de ces machines. La portabilité de l'application est un élément indispensable au vu de l'hétérogénéité du parc des machines disponibles pour l'enseignement.

• La convivialité de l'interface et donc le confort d'utilisation sont un élément essentiel dans la définition de Cabri-géomètre. La phase d'apprentissage du fonctionnement du logiciel doit être réduite à un minimum et le scénario d'une session de travail ne doit impliquer qu'un minimum de mémorisation du déroulement de la session.

• Pour être effectivement adopté par les utilisateurs finaux le logiciel doit couvrir une partie importante du domaine et pouvoir être utilisé comme support tout au long de l'enseignement d'un programme scolaire. Cabri-géomètre permet de construire toutes les figures de la géométrie euclidienne plane enseignée dans les lycées et collèges.

• La transposition des outils des environnements classiques, la feuille de papier, le crayon ,la règle et le compas pour Cabri-géomètre, doit s'accompagner d'améliorations des performances: la rapidité de construction d'une figure et la justesse des représentations obtenues sont de celles-là.

• Une simple transposition ne présente guère d'intérêt si elle ne s'accompagne pas de nouveaux outils qui vont au delà des possibilités de l'environnement papier-crayon. La puissance de calcul des micro-ordinateurs modernes et la qualité de leurs écrans graphiques ont permis de proposer de nouveaux outils permettant d'appréhender différemment les figures de la géométrie.

1.2.4 Interfaces et fonctionnalités

Les fonctionnalités de Cabri-géomètre peuvent se regrouper en deux principales catégories :

- construction de figures géométriques,
- exploration des figures,

Nous allons décrire ces différents éléments en essayant de justifier brièvement leur bien-fondé pédagogique.
Les spécifications fonctionnelles et informatiques précises font l'objet du chapitre suivant et une description de toutes les fonctionnalités de Cabri-géomètre est donnée dans l'annexe C. Nous avons été amenés à choisir des expressions pour paraphraser un nouvel outil ou à donner une signification non usuelle à certains termes: nous profitons de cet exposé des fonctionnalités de Cabri-géomètre pour donner la définition d'expressions (en gras et en italiques) que nous utiliserons tout au long de ce document.

Interfaces

L'***interface*** est la partie de l'application qui gère le dialogue homme-machine. Du point de vue de l'utilisateur, l'interface se présente comme un ensemble de manipulations possibles et d'objets assurant la présentation des résultats ou lui permettant de lancer des procédures, d'en fixer les paramètres et d'en contrôler l'exécution. L'interface de Cabri-géomètre est dérivée de celle du Macintosh et nous en donnons ici les principaux éléments, dont les principes sont détaillés au paragraphe 2.4:

• Les manipulations se font à l'aide d'une souris et opèrent directement sur les objets de l'application. Le déplacement de la souris est associé au déplacement d'un curseur sur l'écran et le système reçoit des informations indiquant une impulsion sur le bouton de la souris et la position du curseur au moment de l'impulsion. Cette ***manipulation directe*** permet de ne pas avoir recours à un langage spécifique pour indiquer les procédures et leurs paramètres.

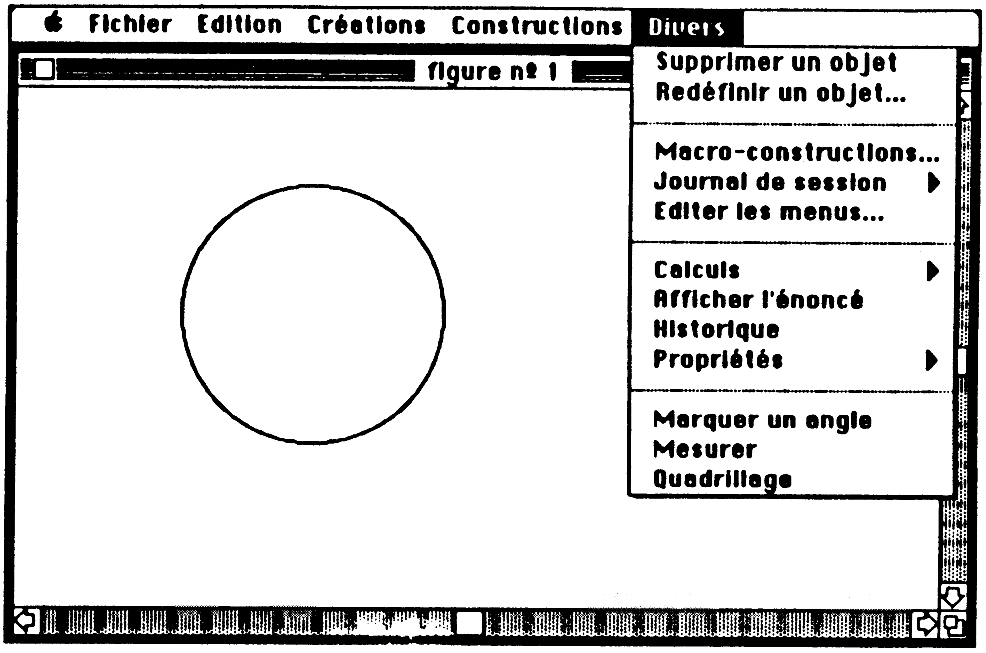
• Les figures de Cabri-géomètre sont visualisées dans des ***fenêtres*** qui délimitent une portion de l'écran. Ces fenêtres sont superposables et mobiles dans l'écran. Si les informations dépassent le cadre de la fenêtre, des ***ascenseurs*** horizontaux et verticaux permettent de faire apparaître une autre partie du dessin. Le cadre de la fenêtre indique le titre éventuellement donné à la figure et possède des cases de contrôle permettant de fermer, d'agrandir et de déplacer la fenêtre.

• Les fonctionnalités sont réunies par thème sous forme d'***articles*** dans une barre de ***menus déroulants***. Certaines touches du clavier permettent de modifier l'exécution de la procédure. Cabri-géomètre permet d'éditer l'ensemble des menus disponibles. L'enseignant peut ainsi ne laisser disponibles que les articles nécessaires à l'activité qu'il propose.

• Au cours d'une opération l'utilisateur doit indiquer des objets comme paramètres d'une procédure. Cette sélection ou ***désignation*** des objets se fait à l'aide de la souris. Les objets sélectionnés sont soulignés par un clignotement jusqu'à la fin de l'opération.

• Les choix que l'utilisateur est invité à faire ou les messages d'erreur sont affichés dans des fenêtres de ***dialogue,*** munies de ***boutons de contrôle*** correspondant aux différents choix.
Des icônes et des curseurs de formes différentes sont associés aux diverses opérations pour aider l'utilisateur à maitriser le déroulement de sa session de travail.

Le schéma suivant montre les principaux éléments d'interface de Cabri-géomètre:



construction de figures

Les figures de la géométrie sont composées d'***objets élémentaires*** (principalement des points, des droites, des cercles, des segments), ces objets présentant éventuellement des relations qui lient leurs positions. Ces relations sont celles qui sont définies dans une géométrie axiomatique (comme la géométrie de Hilbert [Hilbert/Rossier 71]); on trouve ainsi les relations d'équidistance, d'appartenance, de parallélisme, d'orthogonalité, ... Nous préciserons, au §2.1, la nature exacte des objets et des relations manipulées par Cabri-géomètre.
Une figure de géométrie peut être décrite par une suite de constructions de base (point quelconque, milieu, intersection de deux droites, droite parallèle, bissectrices,...). Cabri-géomètre propose de construire une figure en appliquant des ***primitives de construction***, pour lesquelles l'utilisateur spécifiera la position relative dans le plan ou bien les objets qui permettent d'appliquer cette primitive.
Le système ne se contente pas de donner la représentation graphique du résultat de la construction; les hypothèses sont également mémorisées ce qui permettra de modifier la figure tout en conservant certaines ou toutes ces hypothèses.

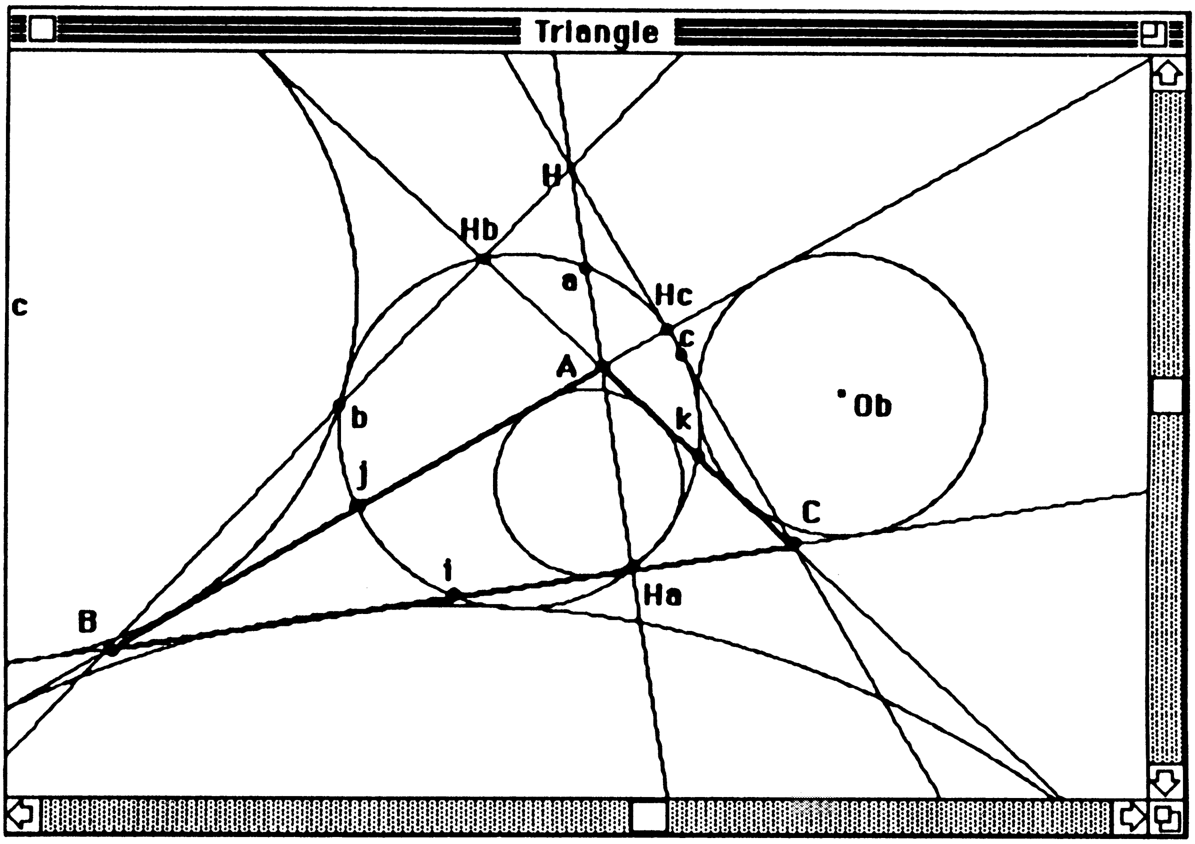
Nous avons choisi d'obliger l'utilisateur à expliciter par une primitive tous les objets auxquels il se réfèrera, même ceux qui sont implicitement identifiés dans l'environnement papier-crayon comme les points d'intersection. De même seules les hypothèses explicitées à travers ces primitives seront retenues (un point quelconque n'aura la propriété d'appartenir à une droite que s'il a été défini comme tel, pas s'il apparaît sur la droite lors de son positionnement à l'écran).

Une suite de constructions simples peut être définie comme une unique construction, disponible et exécutable identiquement aux constructions de base. Les ***macro-constructions*** ainsi obtenues permettront d'effectuer beaucoup plus rapidement une construction complexe (cercle circonscrit à un triangle par exemple), voire de donner la possibilité à un élève de manier des objets sans avoir besoin de connaitre leur construction (pentagone régulier par exemple). L'enseignant peut ainsi construire une bibliothèque de macro-constructions adaptée à tel ou tel type d'activités.
N'oublions pas de souligner l'apparition de macro-constructions dans un micromonde!!

La possibilité de construire une figure s'accompagne de facilités d'***édition de figures*** et de ***mesures*** dont voici les principales:
• Une figure de Cabri-géomètre est dessinée sur une ***feuille de dessin*** de dimension fixe, supérieure aux dimensions de l'écran. L'écran permet d'afficher telle ou telle partie de cette feuille.
• Les points, les droites, les cercles peuvent être nommés et les noms déplacés pour être mis à l'endroit le plus approprié.
• les objets peuvent être cachés ou épaissis pour faire ressortir les principaux éléments de la figure.
• La mesure de segments peut être affichée, les angles définis par trois points peuvent être marqués par un arc de cercle et mesurés.

• Des calculs d'aire, de longueur, de pente, ... peuvent être affichés dans une fenêtre de calculs liés à la figure.
• L'énoncé d'une figure sous forme de chaines de caractères associées à chaque objet peut être affiché également dans une autre fenêtre.

La copie d'écran suivante montre le résultat obtenu après la construction d'un triangle ABC et de certains de ses éléments caractéristiques:



exploration de figures

• Sur un tableau noir ou sur un cahier de brouillon, l'élève ou le professeur qui dessine une figure de géométrie est souvent amené à modifier le dessin de sa figure, pour faire ressortir tel ou tel objet de la figure, pour essayer de mettre en évidence un invariant géométrique, pour envisager ou éliminer les cas particuliers, pour étudier une figure sensiblement différente en rajoutant ou en supprimant une hypothèse,

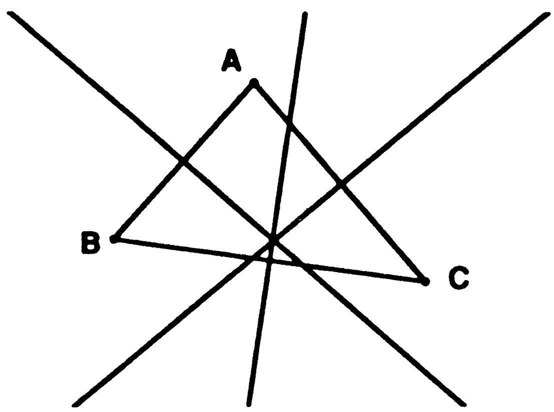
pour étudier l'évolution (et donc le lieu géométrique) d'un des points de la figure ou tout simplement pour rectifier une erreur ou pour pallier les limitations physiques de son dessin (qui n'a jamais soupiré en s'apercevant que deux droites, résultat d'une suite compliquée de constructions, ne se coupent pas sur la feuille?).
Redessiner la figure à l'écran est certes possible, mais n'est plus nécessaire: Cabri-géomètre propose des outils pour entreprendre ces explorations de la figure, explorations qui consistent notamment à modifier certaines caractéristiques d'un objet.

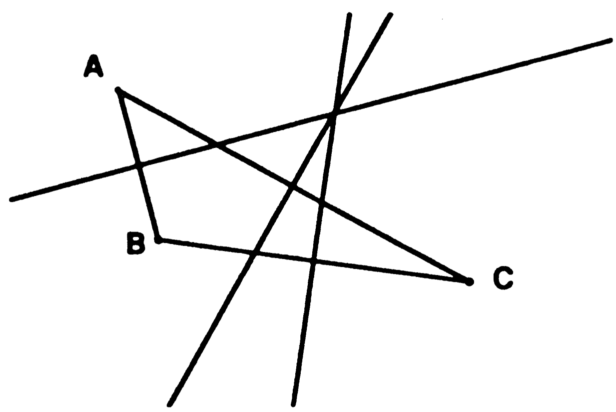
• La ***redéfinition d'un objet*** est la modification des hypothèses sur un objet. Un objet peut être redéfini comme le résultat d'une autre primitive de construction à condition que la nouvelle figure soit constructible par une suite d'applications des primitives. On peut ainsi redéfinir un point quelconque comme un point appartenant à une droite et voir évoluer la figure avec cette nouvelle contrainte.

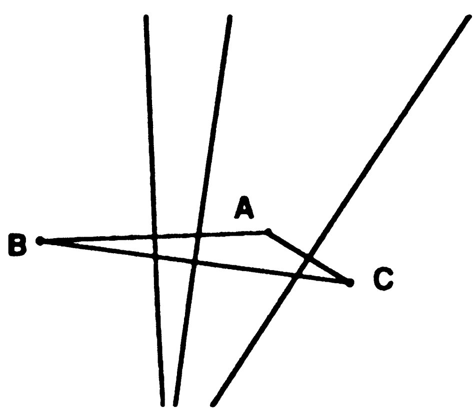
• Parmi les divers éléments d'une classe de figures certains objets sont entièrement déterminés par des relations avec d'autres objets (le milieu de deux points par exemple). D'autres objets possèdent une certaine "liberté"(une droite quelconque ou un point quelconque sur un cercle donné par exemple).
La ***déformation de la figure*** est l'opération qui consiste à modifier un des éléments libres tout en conservant les relations géométriques entre les différents objets. Cette opération est à la fois une des principales originalités de Cabri-géomètre et celle qui offre le plus de potentialités didactiques.
Modifier un des éléments de la figure tout en conservant le même ensemble d'hypothèses modifie l'aspect de la représentation graphique et c'est ce changement d'aspect que nous avons appelé déformation. Cette déformation est obtenue par une manipulation directe de l'objet. L'utilisateur "saisit" l'objet avec la souris (le curseur prend alors la forme

d'une main saisissant un objet) puis le déplace sur l'écran.
L'ordinateur calcule alors la nouvelle position de cet objet ainsi que des autres objets de la figure de façon que les relations explicitées à la construction soient conservées.
La rapidité de calcul et de l'affichage des machines permet d'avoir une **actualisation en temps réel** de la figure et une **continuité apparente** dans la déformation.
Cette opération est la plus souvent utilisée au cours d'une session de travail et elle est disponible à tout moment sans avoir à la sélectionner dans la barre des menus.

• Sans pouvoir traduire dans ce document l'idée de mouvement associée à cette opération nous pouvons illustrer son principe par les trois diagrammes suivants qui montrent l'évolution d'une figure composée du triangle ABC et de ses trois médiatrices quand on fait glisser le point A.

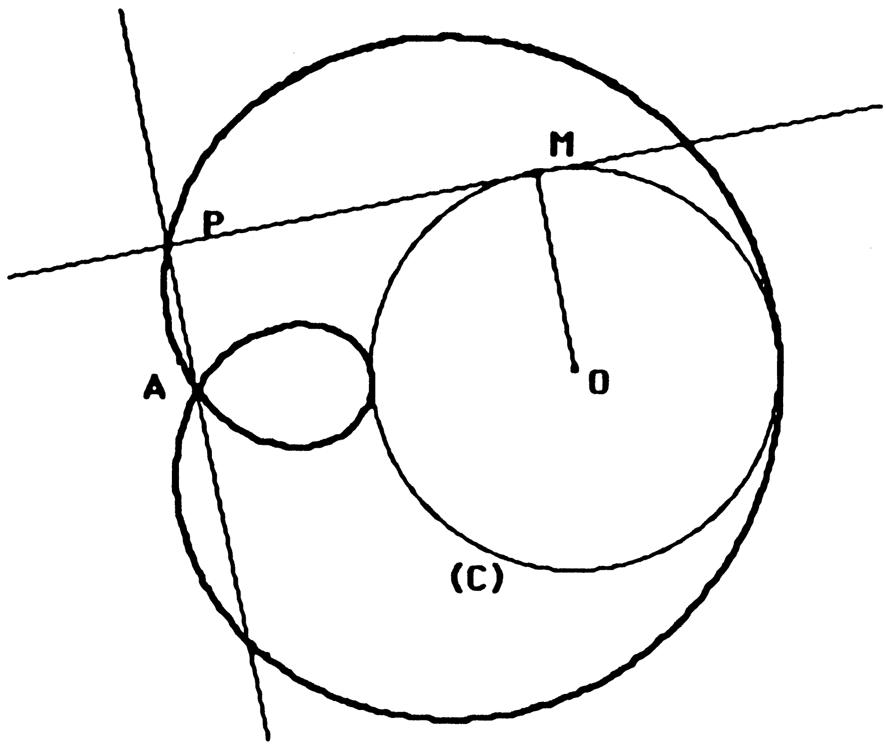






Ce nouvel outil est apparu comme étant un nouvel outil de ***validation empirique*** pour l'utilisateur. Par validation nous entendons une phase intermédiaire entre la simple vérification à l'aide d'un instrument de mesure et la démonstration complète de la propriété par une suite de déductions basées sur un ensemble d'axiomes et de théorèmes.
Une mauvaise construction ou une construction *au jugé* pourra être invalidée facilement.
La formulation d'une conjecture pourra être confortée par l'invariance de tel ou tel élément de la figure au cours de la déformation: la mise en évidence des invariants est visuellement favorisée par la continuité de la déformation.
Plusieurs chercheurs de l'équipe sont engagés dans des expérimentations pédagogiques dont l'objectif est de mesurer l'impact de cet outil.

• "Trouver le lieu géométrique du point P lorsque le point M décrit la droite D" fait partie des énoncés classiques.
Si le point M a été construit comme un point de la droite D, il peut être saisi et déplacé (il restera cependant assujetti à la droite D). Le point P se déplace et par un effet de rémanence on peut avoir une idée du lieu géométrique du point P.
La ***visualisation de lieux géométriques*** est possible en demandant au système de conserver la trace d'un ou plusieurs points au cours d'une déformation. Cette trace est lissée par interpolation entre les différentes positions calculées par le système.
Ainsi, dans la copie d'écran suivante, le point P a été construit comme la projection d'un point A sur la tangente à un cercle C en un point quelconque M du cercle, le point P décrit un limaçon de Pascal quand le point M décrit le cercle C:



• La ***validation de propriétés*** est également un outil que l'élève n'a pas à sa disposition dans l'environnement papier-crayon. Avoir une information sur la validité d'un fait géométrique semble intéressant pour conforter définitivement une conjecture et se lancer avec optimisme dans une démonstration dont on sait qu'elle a toutes les chances d'aboutir.
Par exemple si l'on décompose un problème, il est préférable d'avoir immédiatement une information sur la validité des sous-buts que l'on s'est fixé.
Nous avons vu plus haut que la déformation de la figure permettait d'effectuer une validation expérimentale. Nous avons également introduit dans Cabri-géomètre un outil de validation théorique (voir ci-dessous en §2.3.5) mais c'est avec prudence que nous le proposons.
Prudence pour son utilisation qui n'a pas été réellement expérimentée jusqu'à présent, prudence surtout parce que cet outil ne repose pas sur une validation formelle, algébrique ou logique, mais sur des calculs

analytiques et probabilistes. Nous chercherons à justifier cette méthode, qui ne recouvre pas le même domaine de validité que les méthodes symboliques, et qui se rapporte à un micromonde fini tant au point de vue du nombre d'objets, que de l'éventail des relations de base et des points d'un écran.

L'utilisateur choisit une propriété (appartenance, parallélisme, alignement,...) dans un menu puis désigne les objets impliqués dans la propriété recherchée. Le système évalue cette propriété et propose un contre-exemple dans le cas où la propriété n'est pas générale mais semble visuellement vraie à l'écran

1.3 D'autres réalisations pour la géométrie

La géométrie est un domaine d'enseignement qui a donc été l'objet de nombreuses recherches et réalisations. Nous allons faire dans ce paragraphe une description de certains de ces systèmes sans cependant entrer dans une analyse critique de leurs particularités, car nous ne les connaissons qu'à travers de courtes séances de démonstration et/ou d'articles descriptifs.
De cet inventaire, on pourra noter la diversité des approches qui revient à privilégier telle ou telle phase(s) lors de l'activité de résolution de problèmes : la phase de construction et d'acquisition d'un énoncé sous la forme de la commande d'une représentation graphique d'une figure, la phase d'exploration par reconfiguration, calculs, comparaisons d'éléments de la figure, expressions de conjectures,..., enfin la phase de démonstration par explicitation de ses différentes étapes, des théorèmes employés,....

L'éventail des systèmes va d'ailleurs bien au delà de ceux cités ici [Schumann 88], [Guin 89]. Nous répertorions dans ce paragraphe uniquement les projets dont l'objectif est effectivement la réalisation de logiciels éducatifs dans le domaine de la géométrie (d'autres travaux, notamment dans le domaine de la démonstration géométrique, seront évoqués au paragraphe 2.3.5).
Nous allons décrire successivement les systèmes Archimède, Eloge, Euclide, Geom, Geometer's sketchpad, Geometric supposer, Geometry Tutor, Geophile, Mentoniezh, Tricon (la date indiquée pour chaque système correspond au dernier programme ou article en notre possession).

On pourra remarquer qu'aucun de ces systèmes, pas plus que ne le fait Cabri-géomètre, ne peut prétendre être à l'heure actuelle, un outil complet pour l'étude de la géométrie, encore moins une alternative à l'enseignement classique.

Archimède :
(1988)

Archimède [Chouraqui & Inghilterra 87] [Chouraqui & al. 88] est un système à l'état de projet qui est en cours d'implémentation à l'Université de Marseille.
Le langage utilisé est le langage OBJ-LOG, un langage orienté objet et développé autour de PROLOG.

L'objectif est de réaliser un système expert pour l'apprentissage de la démonstration géométrique en classe de quatrième des collèges, avec comme fonctionnalités principales :
- créer ou modifier les connaissances géométriques et la banque d'exercices,
- saisir et agencer une progression pédagogique de l'apprentissage
- résoudre tout exercice qu'il propose ou que l'utilisateur propose
- justifier la trace de sa résolution
- suivre le cheminement démonstratif de l'élève et le guider en cas d'erreur, de blocage ou d'appel d'aide
- évaluer et mémoriser les acquisitions de l'apprenant

Il s'agit donc d'un environnement d'apprentissage complet dont l'ambition implique une réalisation très échelonnée.
Les recherches ont porté pour l'instant principalement sur une analyse de la connaissance géométrique et de sa représentation en machine.
Les objets géométriques sont considérés comme des agrégats de relations organisés sur un "réseau hiérarchique" du type de celui gérant l'héritage dans les environnements de programmation orientée objet.
Les règles, les théorèmes sont également exprimés sous formes de schémas prototypes et d'instances.
La résolution des problèmes ne se fera pas par un balayage aléatoire des règles, mais c'est la perception d'indices pertinents dans les données du problème qui indiquera quel groupe de règles est éligible.
Ces indices sont appelés des contextes d'application.

Le moteur du système consiste en une factorisation récursive des prémisses communes dans le groupe et la création d'un réseau expert. La structure même du futur logiciel, l'interface utilisateur n'ont pas encore été entièrement définis. Il sera alors possible de tester le fonctionnement d'une telle base de connaissances et de son adéquation aux objectifs d'enseignement.

Eloge :
(1984)

Développé à Strasbourg en LSE sur Micral 80-22.

Eloge aborde le problème de la démonstration non pas directement mais par l'analyse des étapes proposées par l'élève [Fleck & al. 84].
Il est organisé comme un système expert dans lequel la base de connaissances contient les théorèmes de géométrie utilisables par l'élève. La base de faits contient les hypothèses de l'énoncé et s'enrichit au fur et à mesure de la démonstration.
Le moteur d'inférence met en œuvre deux fonctions principales :
- le suivi pas à pas d'une démonstration faite par un élève,
- une fonction d'aide que l'élève en difficulté peut activer à tout moment.

La démonstration est élaborée pas à pas par l'élève et le système en vérifie la cohérence en demandant des références d'axiomes et des précisions sur les objets impliqués dans ce pas de démonstration.
L'élève peut consulter à tout moment la base de faits, c'est à dire les hypothèses et les propriétés déjà démontrées.
Une fonction "secours" donne une assistance pour la suite de la démonstration mais se borne à proposer à l'élève les règles applicables sans cependant vérifier le bien-fondé de son application pour effectuer la démonstration demandée.
Eloge est un logiciel à l'état de prototype dans lequel les problèmes relatifs à une interface utilisateur n'ont pas encore été véritablement étudiés.

C'est sur la base de ce prototype que se sont développées à l'IREM de Strasbourg des recherches visant à créer un véritable système d'EAO de la géométrie en utilisant des langages mieux adaptés à la représentation et manipulation des connaissances et du langage naturel (le langage PROMESS) [Guin & Rousselot 87]. Ces recherches n'ont pas encore abouti à une réalisation concrète.

Euclide :
(1988)

Euclide [Allard & Pascal 88] est sans doute le logiciel d'aide à l'enseignement de la géométrie le plus utilisé à l'heure actuelle par les enseignants des établissements secondaires français. Des versions existent pour les IBM/PC, les Nanoréseaux et les Macintosh.
Il se présente comme une extension du langage LOGO sous forme d'une bibliothèque de procédures.
Bien que nécessitant l'apprentissage préalable de la manipulation du langage LOGO, il offre l'intérêt de construire la plupart des figures de la géométrie scolaire.
L'exploration de la figure est envisagée par la possibilité de vérification analytique de relations entre objets.
Un récent module (Euclide espace) permet également de construire des représentations perspectives de figures de la géométrie dans l'espace.

Le logiciel comprend environ 200 fonctions regroupées dans divers modules.
Le module construction permet de construire des points, des droites, des cercles, des vecteurs, des lignes brisées, en fixant certains paramètres ou certaines relations avec d'autres objets.
La position des points se fait selon les machines avec une souris, des flèches de déplacement du curseur ou un crayon optique.
Une droite peut être construite comme passant par deux points, comme

médiatrice d'un bipoint, comme parallèle (perpendiculaire) à une droite ou un vecteur et passant par un point, comme bissectrices de l'angle formé par trois points ou comme la droite d'équation ax+by+c=0.
La commande **CLCP pt1 pt2** crée le cercle de centre pt1 et pasant par pt2.

A partir de ces constructeurs de base l'utilisateur peut créer ses propres constructeurs sous la forme d'une fonction LOGO.
Il est possible d'interroger le système sur la valeur numérique d'un rayon de cercle, d'une distance de deux points ou droites,... et de vérifier analytiquement certaines propriétés d'appartenance, de parallélisme,...
Un module transformations permet de construire les images d'un objet dans une des transformations géométriques classiques, rotation, symétrie, translation, homotéthie, projection,.... Des objets peuvent être regroupés dans une liste à laquelle on pourra appliquer globalement la transformation.
En utilisant la syntaxe du langage LOGO on pourra par ailleurs écrire des fonctions qui permettent de visualiser des lieux géométriques

Geom :
(1987)

Geom [Gras 88] est développé à Rennes en PROLOG sur Macintosh. Il s'agit d'un logiciel, à l'état de prototype, qui s'intéresse plus particulièrement à la phase d'exploration de la figure et de recherche de la démonstration.
Le module de construction de la figure par le biais d'une tablette graphique, développé dans le cadre du projet Mentoniezh (voir ci-dessous) pourra y être intégré.
Les problèmes sont proposés par l'enseignant qui en précise d'une part l'énoncé et d'autre part le contenus des divers fichiers PROLOG comprenant les consignes, les questions, les théorèmes qui s'y rapportent.

L'exploration de la figure se fait donc sous forme textuelle. Elle est dirigée par le système qui suggère des conjectures que l'élève doit instancier avec des objets de l'énoncé. L'élève a également un rôle actif par les déclarations de sa capacité à apporter la preuve d'une conjecture. Cette exploration est faite par un chaînage arrière à partir de l'objectif terminal. La démonstration est effectuée pas à pas par l'élève qui annonce ce qu'il veut prouver, le théorème adéquat et ses prémisses.
Un module "Théorèmes" permet d'avoir accès aux différents éléments de la théorie auxquels l'élève peut se référer.
Un module "Bilan" permet d'avoir accès aux différentes assertions (conjectures ou propriétés démontrées) de l'élève.

Cette contrainte de la démonstration à un pas imposée à la fois par le fonctionnement du moteur PROLOG et par une interface liée à des dialogues successifs s'est vérifiée être, lors d'expérimentations, à la fois un obstacle à l'intuition d'un schéma déductif et un investissement bénéfique à la structuration des sous-problèmes.

Geometer's sketchpad
(1989)

Geometer's sketchpad ([Jackiw 89]) est un logiciel développé au College de Swarthmore (USA) et qui est une des réalisations d'un projet plus global "The Visual geometry project".
Développé en Pascal et en assembleur sur Macintosh, Geometer's sketchpad est un micromonde qui présente de notables similarités avec Cabri-géomètre.
La version qui nous a été présentée est une version préliminaire dans laquelle de nombreuses fonctionnalités prévues n'ont pas encore été implémentées.
Geometer's sketchpad permet de construire par manipulation directe les objets de la géométrie élémentaire et de déplacer certains objets en conservant les relations entre les objets.

Sans vouloir comparer deux produits développés indépendamment on peut mettre à l'actif de Geometer's sketchpad certaines originalités comme une meilleure définition des interfaces plus proche des normes définies par le constructeur Apple (palette d'outils, interaction non modale, ...), cependant peut-être plus délicates à appréhender pour des utilisateurs occasionnels.
Cabri-géomètre propose actuellement un éventail de fonctionnalités plus étendu mais il est probable que dans un avenir proche les différences s'estompent entre les deux produits.

Geometric Supposer :
(1985)

Développé au USA, Geometric Supposer [Schwartz 85] est implémenté sur les ordinateurs IBM PC et les Apple II. Geometric Supposer est sans doute le logiciel de géométrie le plus utilisé aux Etats-Unis.
Il se présente sous la forme de plusieurs programmes séparés traitant respectivement des points et des droites, des triangles, des quadrilatères, des cercles.
L'interface utilisateur est un ensemble de menus et de sous-menus dont l'accès est commandé par le clavier.
Chacun de ces modules permet la construction des figures remarquables dans l'étude des objets spécifiés (médiatrices d'un triangle, diagonales d'un quadrilatère,...). Les caractéristiques de l'objet sont indiquées par exemple pour les triangles en explicitant la définition utilisée (trois longueurs, une longueur et deux angles,...) puis des valeurs numériques.
Certaines possibilités d'exploration de la figure sont possibles comme le changement de configuration par redéfinition des paramètres des objets de base, l'explicitation des méthodes de construction des objets remarquables, le calcul de mesure de segment, d'angle, de périmètre, d'aire.

Geometry Tutor :
(1985)

Geometry Tutor [Anderson & al. 85a] [Anderson & al. 85b] est un produit de l'Université Carnegie-Mellon à Pittsburg (USA) et entre dans la catégorie des tuteurs intelligents pour laquelle il est souvent cité comme référence. Il s'agit d'une application à la géométrie des principes d'une théorie cognitive ACT* (pour Architecture of Cognition Theory ou Advanced Computer Tutoring ou Adaptative Control of Thought). Le tuteur suit la conduite de l'apprenant par ses rapports au modèle de l'apprenant idéal et celui de l'apprenant standard [Guin 89].

L'activité de l'élève est ici, non pas de construire et d'explorer la figure, mais d'élaborer la démonstration d'un problème donné.
L'écran se présente de la manière suivante :
• la figure est représentée avec la symbolique usuelle pour dénoter les propriétés faisant partie des hypothèses. Aucune action ne peut être entreprise sur cette figure qui sera cependant actualisée avec les symboles correspondant aux propriétés effectivement démontrées.
• les hypothèses et les conclusions s'inscrivent respectivement en bas et en haut de l'écran dans le langage classique de la géométrie.

L'élève va élaborer sa démonstration en construisant l'arbre qui est associé à cette démonstration.
Chaque pas de démonstration est effectué en indiquant successivement à l'aide de la souris et du clavier les prémisses, les raisons et les conclusions de ce pas s'il désire procéder par un chaînage avant ou ces mêmes éléments dans l'ordre inverse s'il effectue un chaînage arrière.

Le suivi des pas de démonstration de l'élève est confié à un système de production dont nous allons donner quelques éléments du fonctionnement :
• Les lois de la géométrie et le système de connaissances de l'apprenant

sont réunies, sous une même représentation, dans une base des connaissances "idéales et boguées" contenant à la fois les règles permettant de résoudre les problèmes posés et les règles incorrectes auxquelles l'apprenant peut avoir recours. Il y a environ 200 règles.
•Le système analyse le pas de démonstration fourni par l'élève qui peut être ou non l'instanciation d'une règle de la base. En cas d'utilisation d'une règle boguée le système fournit à l'apprenant des éléments pour remédier à son erreur. Si l'apprenant est bloqué ou s'engage trop loin dans une voie qui ne mène pas à la solution, le système lui fournit un pas idéal de démonstration.
• La validité d'une inférence de l'apprenant est également conditionnée par un contexte d'application. Ce contexte est exprimé dans la règle sous la forme d'un but ou sous-but à atteindre.
• Le résolveur par lequel le système peut fournir la démonstration évite l'explosion combinatoire dans la recherche des règles applicables par l'utilisation du diagramme comme dans [Gelertner 63] et la gestion des permutations et symétries des objets de la figure comme dans [Buthion 75].
• Le domaine d'utilisation de Geometry Tutor, version 1985, semble se limiter aux problèmes sur les cas d'égalité des triangles.

Geophile :
(1989)

Développé à l'Université Louis Pasteur de Strasbourg sur SM90, Geophile [Braun 89] permet de construire et de manipuler les figures de la géométrie plane euclidienne élémentaire en exprimant des opérations sur le graphe formé à partir des relations d'interdépendance des objets.

L'utilisateur exprime les étapes de sa construction, le système se charge d'en vérifier la cohérence et de demander, le cas échéant les définitions manquantes. On peut ainsi exprimer une construction selon un mode

descendant (droite D passant par I et perpendiculaire à S, avec I milieu de S et S segment d'extrémités A et B) ou bien ascendant (point A, point B, segment S, I milieu de S, droite D passant par I et perpendiculaire à S). L'utilisateur est ainsi sûr de la cohérence de sa construction.
Le langage de l'interface a de toute évidence une connotation lispienne. L'exemple suivant correspond à la commande de définition d'un segment S ayant pour extrémités le point de coordonnées (2,3) et le point d'abscisse 5 et d'ordonnée le cosinus de l'angle 0 :

définir S= #(segment #(point 2 3) #(point 5 (cos 0)))

Lorsque le réseau est complet, les valeurs des différents nœuds sont calculées et il devient possible d'effectuer des modifications. Ces modifications peuvent opérer sur les objets dits terminaux, c'est à dire qui ne dépendent pas d'autres entités, et les objets non terminaux.
Les modifications d'objets terminaux correspondent à des reconfigurations du même ensemble d'objets liés par les mêmes relations alors que les modifications d'objets non terminaux correspondent à des modifications de la procédure de construction. Ces modifications induisent des réévaluations de certains objets dont l'affichage est alors actualisé.
D'autres manipulations sont possibles comme les consultations, pour chaque objet, de ses valeurs, arguments, prédécesseurs et successeurs sur le graphe ainsi que certaines consultations globales sur le réseau ou sur une classe d'objets.
Les lieux de points, les enveloppes de droites sont modélisés sous la forme de suites d'objets, chaque élément de la suite étant calculé à partir des différentes valeurs d'un autre élément et du calcul d'une expression représentant les relations de ces objets.
De telles expressions permettent également de définir de nouveaux constructeurs à partir des constructeurs de base fournis par le système.

Mentoniezh :
(1989)

Développé à l'Université de Bretagne (Rennes), le système Mentoniezh a pour but l'aide à la résolution de problèmes de géométrie tels ceux de la classe de quatrième des collèges.
Les prototypes ont été écrits en Pascal et en Prolog et utilisent des Bull/Micral 90/50, des IBM PC et des stations SUN.
Une décomposition en modules reprend les trois phases par lesquelles passe l'élève, c'est à dire la compréhension des hypothèses, l'étude des propriétés logiques de la figure puis l'organisation de la démonstration.
Les réalisations actuelles relèvent principalement du premier module en développant des outils pour la construction et la vérification des figures géométriques.
L'enseignant donne les spécifications de l'énoncé dans un langage d'interface (appelé CDL pour Classroom Description Language) proche du langage habituel de la géométrie.
La construction est effectuée par l'élève au moyen d'une "machine à dessin" composée d'une tablette graphique sur laquelle il dessine les objets et d'une interface lui permettant de spécifier dans un langage SCL (pour Student Construction Language) les relations entre ces objets. Le système traduit les tracés de l'élève puis les ajuste pour que les propriétés des objets soient respectées à l'écran.
Ces deux ensembles d'hypothèses sont traduites dans un langage interne LCL (pour Logical Construction Language) par le biais duquel un moteur PROLOG va évaluer la cohérence et l'équivalence des représentations d'un énoncé qu'ont donné le professeur et l'élève.
Les questions sur lesquelles se penche le système sont :
La figure de l'élève est-elle cohérente?
La figure du professeur est-elle cohérente?
Existe-t-il une figure construisible à partir de la spécification du professeur?
La figure de l'élève répond-elle à la spécification du professeur?

comporte-t-elle toutes les propriétés désirées par le professeur? Est-ce un cas particulier?
Cette correction logique se fonde sur une "Théorie de la Géométrie Instrumentale" (TGI). Il s'agit non pas d'une base de théorèmes mais de l'approximation d'une théorie générale de la géométrie telle l'axiomatique de Hilbert. La mise au point de cette théorie et du choix pertinent des axiomes est au centre des recherches actuelles.

Tricon :
(1988)

Tricon est issu de travaux de recherche à l'Université de Giessen (RFA) sous l'égide de G. Holland [Holland 87][Holland 88].
Tricon est opérationnel sur les machines du type IBM/PC et a été développé en Turbo-Prolog.

Le domaine envisagé est la géométrie du triangle. Tricon permet de réaliser ou de visualiser les étapes de la construction d'un triangle dont on a précisé certains éléments comme trois côtés ou deux angles et un côté.
L'apprenant fournit les éléments caractérisant le triangle en choisissant trois valeurs numériques parmi celles que peuvent prendre les angles, les côtés, les hauteurs, les médianes ou les bissectrices.

Le système signale par un message d'erreur si les éléments fournis en hypothèse ne suffsent pas pour caractériser un triangle (les trois angles par exemple).

L'apprenant peut construire lui-même le triangle en utilisant des primitives de construction qu'il exprime par le biais d'un langage (LEKTOR).
Par exemple la commande **f=ortho(AB,C)** dessine l'orthogonale à la droite AB passant par C et cette nouvelle droite est appelée f.

Les primitives de construction (environ 15 primitives) permettent de construire des points, des cercles, des droites, des demi-droites respectant certaines relations d'appartenance, de parallélisme ou d'orthogonalité avec des objets existants.

Le guidage de la session par le système s'effectue à trois niveaux :
- pré-enregistrement d'exercices qui sont soumis à l'apprenant,
- les commandes sont interprétées par le système qui peut les refuser si elles ne mènent pas à la résolution du problème posé.
- Une fonction d'aide permet à l'apprenant de se voir proposer l'étape suivante de la construction.

En mode visualisation, le système donne le détail de toutes les méthodes de construction possibles (par exemple il y a neuf méthodes de construction d'un triangle dont on fixe la longeur des médianes). Les constructions successives sont commentées à l'écran.

Pour élaborer lui-même les constructions ou vérifier la bonne stratégie de l'apprenant, le système expert de Tricon maintient à jour un liste des points remarquables du triangle à construire, une liste des points construits et une liste des constructions élémentaires effectuées.
Le système cherche à faire coincider ces deux listes de points en construisant des lieux géométriques (droites, demi-droites, cercles, segments) dont l'intersection fait partie de la liste des points à obtenir pour telles ou telles données initiales.

II. Spécifications fonctionnelles et opérationnelles

La spécification des objets et des procédures de Cabri-géomètre, tels que nous les avons introduits au §1.2 sous la forme d'un cahier des charges, n'a pas fait l'objet d'une formalisation par le biais d'un véritable langage de spécifications. Nous présentons ces spécifications fonctionnelles et opérationnelles ensuivant le schéma suivant: après avoir défini les **figures de Cabri-géomètre** et leurs différents éléments, nous exposerons les **structures de données** mises en places, puis les **fonctions principales** du logiciel ainsi que les **principes de l'interface. Quelques problèmes** pourront être mis en évidence, et des solutions partielles proposées. Enfin nous dirons quelques mots sur les modalités de **l'implémentation** logicielle.

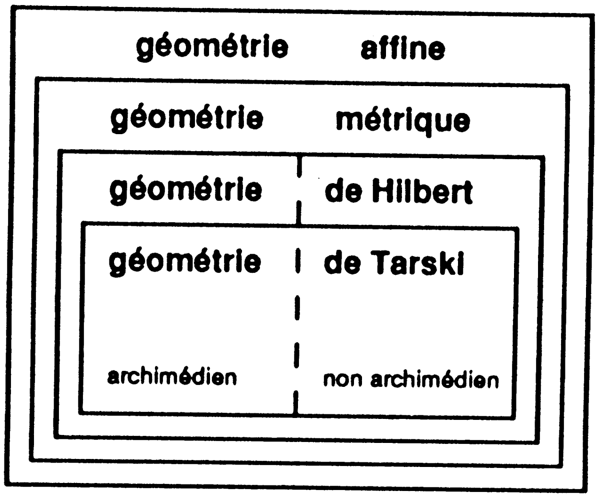
2.1 les figures de Cabri-géomètre

Cabri-géomètre permet de construire et de manipuler des figures géométriques, mais il n'est pas question de prétendre construire toutes les figures de toutes les *Géométries*.
Le modèle que réalise Cabri-géomètre est éminemment restrictif et nous allons dans ce paragraphe définir les différents éléments de ce que l'on peut appeler la "géométrie de Cabri".
Nous verrons dans ce paragraphe et dans les suivants que la consistance et la complétude du modèle actuellement mis en œuvre sont cependant mis en défaut dans certaines situations. Nous proposerons alors des solutions partielles pour restreindre l'impact de ces phénomènes sur l'objectif final de Cabri-géomètre, à savoir d'être un micromonde destiné à l'apprentissage.

2.1.1 La géométrie de Cabri

La géométrie, c'est la science des figures de l'espace, la description de modèles mathématiques pour la représentation de l'espace physique. Il y a donc "la" géométrie et "des" géométries.

Le terme de géométrie est utilisé selon les contextes comme une théorie, un système formel en logique du premier ordre (la géométrie affine par exemple) ou comme un modèle d'une théorie (la géométrie euclidienne est un modèle de la théorie de la géométrie affine) [Chou 85].
nous pouvons évoquer les relations, les inclusions entre les diverses théories illustrées par le schéma ci-dessous que nous reprenons de [Chou & al. 87] :



Le domaine abordé par Cabri-géomètre est la géométrie euclidienne plane, celle qui est enseignée dans les établissements d'enseignement primaire et secondaire français, et qui repose sur une axiomatique comme celle présentée par Hilbert [Hilbert/Rossier 71].
Dans la suite de ce document, le mot géométrie et ses dérivés se référeront uniquement à cette notion.
Cette théorie se fonde sur un ensemble d'axiomes qui précise la nature de certaines relations entre des objets. Nous allons donc préciser quels sont les objets et les relations géométriques qui sont reconnus par Cabri-géomètre.

Par ailleurs il y a une liaison étroite entre géométrie et algèbre, les objets géométriques et les relations pouvant être définis dans le produit cartésien E^n où E est généralement un corps.
La géométrie euclidienne plane s'appuie sur le corps des réels. A chaque élément de la figure on fait correspondre un système d'(in)équations algébriques.
Nous verrons que la représentation graphique des éléments d'une figure dans le sous-ensemble des nombres réels codés en machine est un facteur important d'incohérence.

2.1.2 les objets élémentaires

On peut classer les objets de la géométrie en trois grandes catégories: les points, les courbes et les surfaces. Les courbes et les surfaces sont des ensembles de points. L'ensemble des courbes présente certains sous-ensembles plus particulièrement étudiés: les droites, les cercles et les portions de ces objets (segments de droite et demi-droites, arcs de cercle) ainsi que la réunion de ces objets, notamment les polygones.

Nous appellerons *objets géométriques élémentaires* ou plus simplement *objets* les éléments de ces sous-ensembles.

Les objets qui ont été pris en compte dans la définition du système sont :

- **les points**
- **les droites**
- **les segments de droite**
- **les cercles**
- *les polygones*
- *les arcs de cercle*
- *les demi-droites*
- *les courbes quelconques*

les quatre dernières catégories sont ici en italiques parce que dans la réalisation actuelle, elles ne sont que partiellement représentées:
- les seuls polygones reconnus par Cabri sont les triangles (tout polygone est constructible, simplement il ne sera pas reconnu comme polygone mais comme assemblage de segments de droite)
- les arcs de cercle et les demi-droites ne sont actuellement pas des objets Cabri bien que cela ne présente pas de difficulté particulière de mise en œuvre. Sans que cela soit une justification, on peut expliquer cette absence provisoire par une importance moins grande de ces objets dans les figures de la géométrie scolaire et par une interface de définition plus complexe (il faut quatre informations pour représenter un arc de cercle: le centre du cercle, les extrémités de l'arc et un choix entre les deux arcs ainsi définis).
- les courbes quelconques ne sont pas constructibles en tant que telles mais toutes les courbes qui se décrivent comme le lieu de un ou plusieurs points soumis à certaines contraintes peuvent être visualisées.
Les structures actuelles ne permettent pas de définir ces courbes comme des objets à part entière, pouvant alors subir certaines manipulations.
Par ailleurs si cette courbe se trouve être une droite, un cercle ou tout autre objet de Cabri, elle n'est actuellement pas reconnue comme telle.

Pour finir, précisons que Cabri-géomètre ne considère pas les surfaces si ce n'est pour en fournir éventuellement l'aire (dans le cas des cercles et des polygones).

2.1.3 les relations entre les objets

Les relations géométriques qui lient les objets d'une figure sont des formules qui peuvent s'exprimer sous la forme d'une combinaison logique de certaines relations de base, relations qui sont précisées par les axiomes de la théorie géométrique considérée.
Sans revenir aux fondements de la géométrie ([Hilbert/Rossier 71] [Lelong-Ferrand 85]), nous pouvons répertorier les relations de base

communément utilisées en géométrie euclidienne plane: parallélisme, perpendicularité, incidence, congruence, ordre sur des points d'un objet.

Dans Cabri-géomètre, les relations qu'il est possible de fixer directement entre les objets sont les suivantes:
- **appartenance** d'un point à un objet (droite, cercle, segment)
- **intersection** de deux objets (droites, cercles, segments)
- **paralélisme** de deux droites (ou segments de droites)
- **orthogonalité** de deux droites (ou segments de droites)
- **équidistance** de points

Toutes ces relations se traduisent algébriquement par un système d'équations et elles dénotent en quelque sorte des égalités entre certaines caractéristiques analytiques des objets.
Les relations qui se traduisent par des inégalités ne sont que partiellement prises en compte. Il est possible de préciser l'appartenance d'un point à un segment et donc d'assurer un ordre sur les points d'une droite. Mais il n'est pas possible de mettre en œuvre directement certaines relations comme par exemple l'appartenance d'un point à une portion du plan délimitée par un ensemble d'objets (demi-plan, intérieur d'un cercle,...).

Une autre relation entre les objets coiffe toutes les relations citées. il s'agit d'une **relation de dépendance** $\Re$ sur l'ensemble des objets d'une figure.
x dépend de y ($x \Re y$) si la définition de l'objet x fait intervenir l'objet y. Par exemple si I est le milieu de [AB] on a $I \Re A$ et $I \Re B$.
C'est cette relation de dépendance qui sera utilisée pour relier les structures informatiques associées aux objets.

2.1.4 les figures et leur représentation

Le terme de figure de géométrie est souvent associé à la représentation graphique d'un ensemble d'objets dont certains ne sont pas disposés d'une manière quelconque, mais reliés les uns aux autres.
Outre la représentation graphique, la notion de figure implique une description des objets qui la compose; cette description peut être entre autres:
- (a) un ensemble d'objets et un ensemble de relations géométriques entre ces objets, exprimés sous la forme d'un énoncé ou d'une formule logique.
- (b) une suite (une séquence) de constructions, chacune de ces constructions menant à un des objets de la figure. Une construction est obtenue graphiquement par l'usage d'instruments comme le compas et la règle.

A un énoncé donné il correspond généralement plusieurs constructions possibles. Certains énoncés ne peuvent pas être traduits par des constructions avec les instruments classiques (la trisection de l'angle par exemple). La correction d'un énoncé est donné par la validité de la formule logique qui lui est associé. La correction d'une figure est donnée par la correction de chaque pas de construction.
L'une et l'autre de ces représentations sont à terme indispensables pour pouvoir répondre aux sollicitations des utilisateurs dans des activités aussi variées que la construction d'une figure donnée, la découverte de certaines propriétés ou la démonstration d'une propriété.

Dans Cabri-géomètre nous nous sommes plus attachés, et cela apparaît dans le cahier des charges, aux activités de construction et d'exploration. C'est la représentation (b) qui est présente dans les structures du système.

On peut donc donner maintenant une suite de définitions qui permettent de caractériser une figure de Cabri-géomètre:

(1) ***Une figure est un ensemble d'objets (et sa représentation graphique) obtenu par l'application d'une suite ordonnée de constructions élémentaires.***

(2) Une primitive de construction est la représentation, sous forme d'une fonction, d'une construction élémentaire qui fait correspondre, à un n-uplet N ($n \geq 0$) d'objets, un nouvel objet présentant des relations avec les objets de N.

Les primitives de construction sont de deux types :
- Celles qui permettent d'appliquer ces fonctions à un ensemble d'objets de la figure en cours d'élaboration.
- Celles qui permettent de se donner des objets a priori, c' est à dire n'entretenant pas de relations avec des objets obtenus par une construction antérieure.

(3) La position d'un objet dans le plan peut ne pas être entièrement déterminée par les relations ayant été déclarées lors de sa construction. Ces objets seront appelés ***objets de base****.*

(4) Une ***classe de figures*** *sera définie comme l'ensemble des figures correspondant à un ensemble de constructions.*

Nous faisons maintenant une série de remarques concernant ces définitions:
- Le terme de figure sera le plus souvent utilisé au lieu de classe de figures et nous parlerons de configurations quand il s'agira de différencier des éléments d'une même classe.

- Les primitives de construction nécessaires à la prise en compte des relations mentionnées au §2.1.3 sont celles qui permettent de réaliser les constructions possibles à la règle et au compas, à savoir:
  - point, droite et cercle quelconques
  - droite (segment) passant par (ayant pour extrémités) deux points donnés
  - cercle de centre donné et passant par un point donné
  - intersection de deux objets donnés
  - spécification d'un point sur un objet donné

- Toutes les autres constructions (de la géométrie de la règle et du compas) peuvent être obtenues à partir d'une composition de ces constructions.

Par exemple le milieu d'un bipoint (A,B) peut être obtenu comme l'intersection du segment [AB] et du segment [EF], E et F étant les points d'intersection des cercles de centre A (respectivement B) et passant par B (respectivement A).

Ces deux dernières affirmations ne sont pas ici démontrées mais peuvent être raisonnablement conjecturées dans le cadre de la géométrie scolaire qui nous intéresse.

- Pour améliorer l'efficacité de Cabri-géomètre, l'utilisateur a la possibilité de définir une composition de constructions et de la déclarer comme une primitive. Ce sont les macro-constructions qui réalisent l'extensibilité des possibilités de construction.

- Pour les mêmes raisons, certaines compositions très utilisées de constructions sont prédéfinies et font partie intégrante des primitives de base. C'est le cas du milieu, des droites passant par un point et perpendiculaire à une direction ,...

- Les objets de base dans Cabri-géomètre sont d'une part les objets *quelconques* (c'est à dire dont la position n'est pas déterminée par celle d'autres objets), d'autre part ceux entretenant la *seule* relation d'appartenance à un objet. C'est la modification de ces objets de base qui permet d'accéder à différents éléments d'une classe de figures.

- Si une figure est une suite de constructions élémentaires, elle ne correspond pas forcément à la suite chronologique des constructions parce que les objets peuvent être redéfinis postérieurement à leur construction.

Les remarques précédentes permettent de mettre en évidence les principales fonctionalités de Cabri-géomètre à savoir :

- la construction d'objets selon une primitive donnée
- la déformation de la figure par modification des objets de base
- la composition automatique des primitives de construction
- la redéfinition d'objets postérieurement à leur construction

Nous allons dans les deux paragraphes suivants, tout d'abord présenter les structures de données qui ont du être mises en place puis ensuite les algorithmes permettant de réaliser ces fonctionnalités.

2.2 les structures de données

Un programme informatique manipule des éléments propres à une application donnée et des éléments de contrôle qui permettent de mettre en œuvre les fonctionalités du programme. Ces éléments de contrôle sont en majeure partie sont des éléments d'interface dont les structures sont définies dans le système d'exploitation de la machine ou dans une sur-couche indépendante de l'application. Nous étudierons ces objets de contrôle au §2.4.

Nous allons donner ici les principaux éléments de la structure informatique associée à un objet intrinsèque de Cabri-géomètre puis aux liens entre ces objets.

2.2.1 l'objet géométrique

Les figures de Cabri-géomètre apparaissent dans des fenêtres délimitant une portion de l'écran à laquelle est donc associé une liste d'objets géométriques et un système de coordonnées permettant de donner une position à ces objets.

La représentation en machine d'un objet géométrique est réalisée dans Cabri-géomètre en utilisant les concepts de la programmation orientée objet même si, comme nous le verrons au §2.6.2, l'environnement de programmation utilisé pour l'implémentation n'intègre pas ces concepts.

Nous précisons d'abord la signification des termes employés pour caractériser un objet:

type: le type de l'objet est sa catégorie, telle que nous les avons définies au paragraphe 2.1.2, à savoir point, droite, cercle, segment, polygone (uniquement les triangles dans l'implémentation actuelle).

classe: la classe d'un objet détermine les relations qu'il entretient directement avec les objets ayant servi à sa construction. Ces classes sont au nombre d'une quarantaine dont voici quelques exemples :

- milieu d'un bipoint,
- intersection de deux droites,
- cercle défini par son centre et un point périphérique,
- symétrique d'un point par rapport à une droite.

La liste complète des classes d'objets est donnée dans l'annexe A.

attributs: ce sont les attributs graphiques d'un objet comprenant notamment son nom éventuel et la position de ce nom, le graphisme de l'objet (VISIBLE ou INVISIBLE, FIN ou GRAS,...).

constituants: nous dirons que les objets Y_i ayant servi à la définition de X sont les constituants de X (si I est le milieu de A et B, alors A et B sont les constituants de I).

dépendants: les dépendants de X sont tous les objets de la figure dont la définition fait intervenir directement ou indirectement X.

caractéristiques analytiques: ces caractéristiques sont les nombres réels qui seront donnés en paramètres aux algorithmes de tracé. Ces caractéristiques analytiques sont, pour chaque type d'objet:

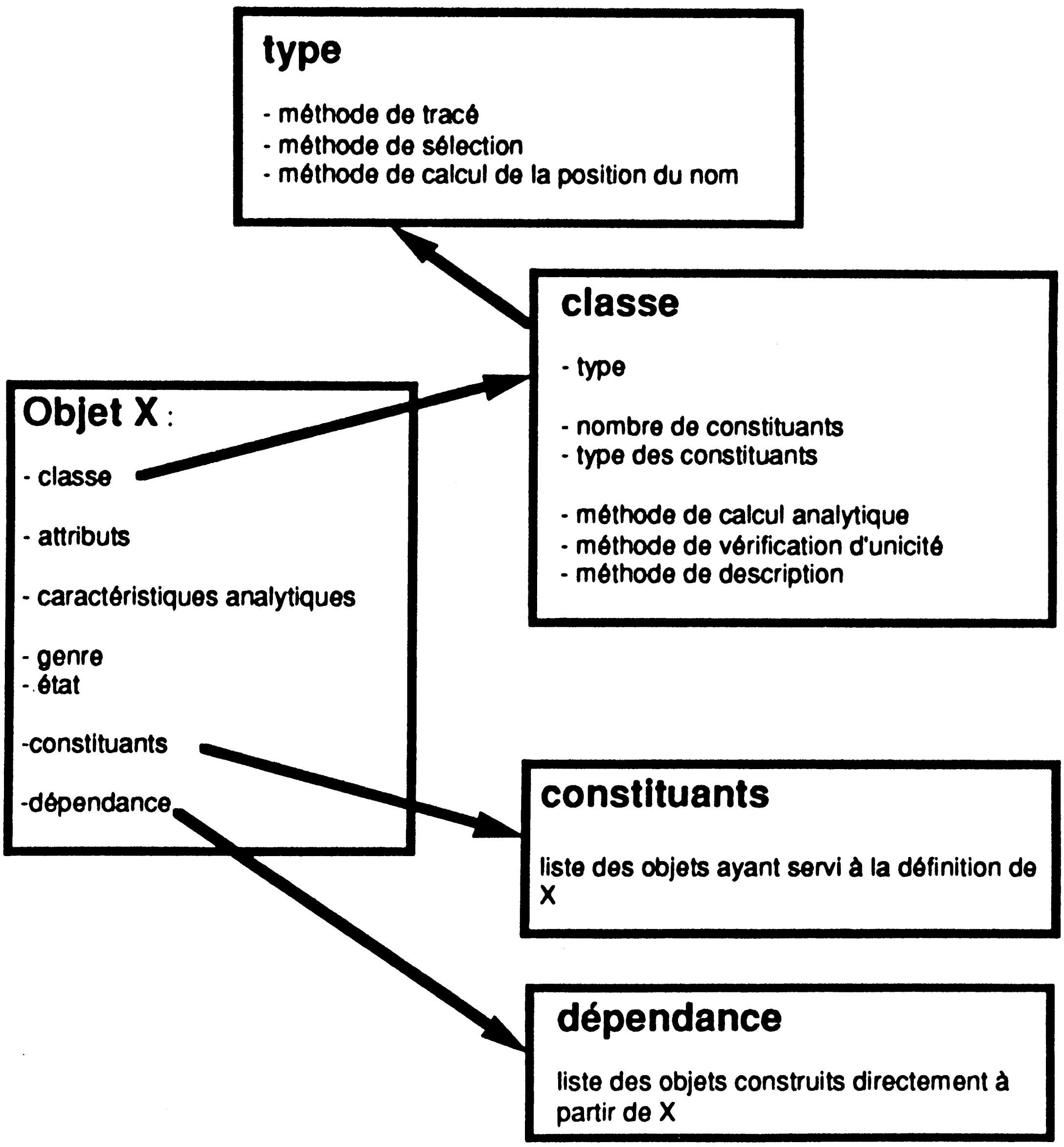
- point: ses coodonnées dans le repère lié à la figure
- droite: un point et un vecteur directeur
- cercle: un point et un rayon
- segment: les extrémités du segment

état: ce champ spécifie si un objet a effectivement une représentation graphique dans le plan. Il trouve sa raison d'être dans la possibilité de déformation de la figure. En effet un objet peut ne pas exister dans certaines configurations accessibles par modification des objets de base de la figure.

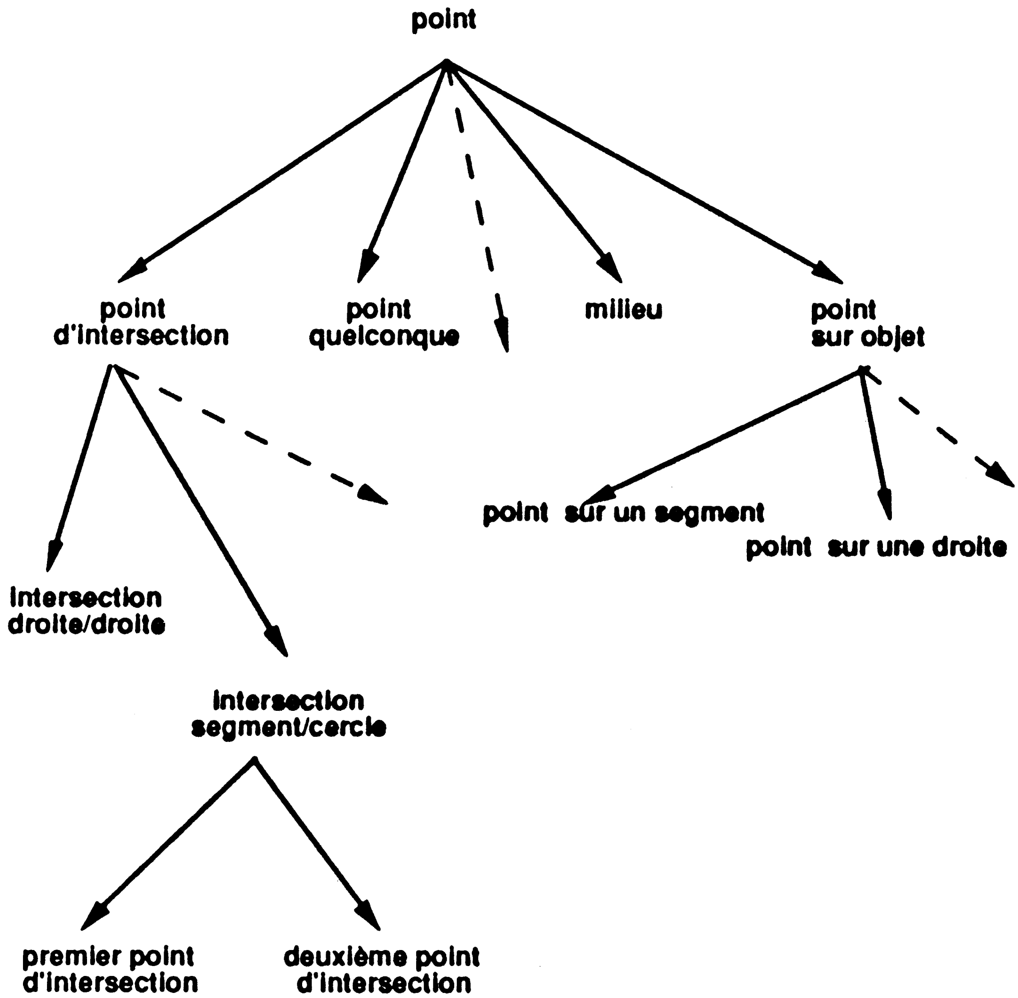
genre: les objets ne jouent pas tous le même rôle pour l'utilisateur. Un objet peut être une construction intermédiaire nécessaire à une macro-construction, il peut aussi avoir été implicitement créé comme par exemple les côtés d'un triangle. Ce genre peut prendre les valeurs NORMAL, IMPLICITE, INTERMEDIAIRE.

La réponse d'un objet aux différents messages seront conditionnées par son genre et son état.

Le schéma de la page suivante présente les différents éléments de la structure (les structures C sont détaillées dans l'annexe B):



Cette structure de données permet de mettre en évidence deux hiérarchies entre les objets. D'une part une hiérarchie de classes qui permet à un objet d'hériter des méthodes d'une classe supérieure et d'autre part une hiérarchie induite par la relation de dépendance entre les objets d'une même figure. Cette dernière sera précisée au paragraphe suivant.
La hiérarchie des classes (au sens de la programmation orientée objet) se présente sous la forme d'un arbre dont le shéma ci-dessous donne une vue partielle:



Les structures associées aux classes et aux types comprennent les éléments (données et procédures) communes à toutes les instances.
Les principales méthodes communes à tous les objets d'un même type sont les différentes méthodes de tracé, de sélection (calcul de la distance

du curseur à l'objet)et de calcul de la position du nom.
Tous les objets d'une même classe héritent entre autres des procédures de calcul des caractéristiques analytiques, de vérification d'unicité (pour ne pas avoir d'objets identiques). Nous indiquerons le contenu de ces méthodes lors de la description des algorithmes au §2.3.

Cette structure s'est avérée suffisamment flexible au cours du développement pour permettre l'incorporation de nouvelles classes d'objets, voire de méthodes non prévues dans le cahier des charges initial.

Des primitives de construction comme les bissectrices de deux droites ont été ainsi rajoutées à faible coût quand il s'est avéré que leur accessibilité directe (et non par l'intermédiaire de macro-constructions plus lentes) était nécessaire.
Par ailleurs cette structure a permis d'implémenter d'autres éléments dans la figure :
- Marquer un angle par un arc de cercle est une opération courante dans l'environnement papier-crayon. ces marques sont considérées par le système comme des objets à part entière d'un nouveau type "angle".
- Marquer la mesure d'un segment ou d'un angle a consisté à considérer cette mesure comme le nom du segment ou de l'angle correspondant.
- Décrire la figure sous forme d'une suite de phrases détaillant chacun des objets a pu être réalisé en rajoutant une nouvelle méthode à la structure de classe, cette méthode ayant pour paramètres les constituants de l'instance considérée.

Bien que cela n'ait pas été réalisé, la structure permet d'envisager la création dynamique de classes (voir §2.5.4), ces nouvelles classes étant soit une particularisation des classes existantes (cercle circonscrit à un triangle), soit l'union de classes existantes (polygones réguliers ou non).

2.3.2 le graphe de dépendance

Nous avons vu qu'une figure est caractérisée par un ensemble d'objets reliés par des relations géométriques ou par une suite logique de constructions élémentaires.
Une construction est une fonction qui prend ses arguments X_i dans l'ensemble des objets d'une figure et qui produit un (ou plusieurs) objet Y ayant certaines relations géométriques avec les X_i.
Il est tentant de représenter la figure par un réseau sémantique dans lequel les nœuds sont les objets de la figure et les arêtes sont les relations géométriques entre les objets.
Un tel réseau serait extrêmement complexe de par la nature même des relations géométriques qui ne sont pas forcément binaires et qui sont elles-mêmes interconnectées (par exemple $D \parallel D'$ & $D \perp D'') \Rightarrow D' \perp D''$).

Nous avons choisi de représenter non pas directement ces relations géométriques, mais la relation de dépendance entre les objets.
Cette relation est un ordre strict sur l'ensemble des objets d'une figure.
Elle est effectivement antisymétrique:

$$(x \neq y \;\&\; x \,\Re\, y) \Rightarrow \neg\, y \,\Re\, x$$

Si y est obtenu à partir de x, y ne peut pas exister sans x et a fortiori ne peut pas permettre de construire x.
Elle est aussi transitive:

$$(x \,\Re\, y \;\&\; y \,\Re\, z) \Rightarrow x \,\Re\, z$$

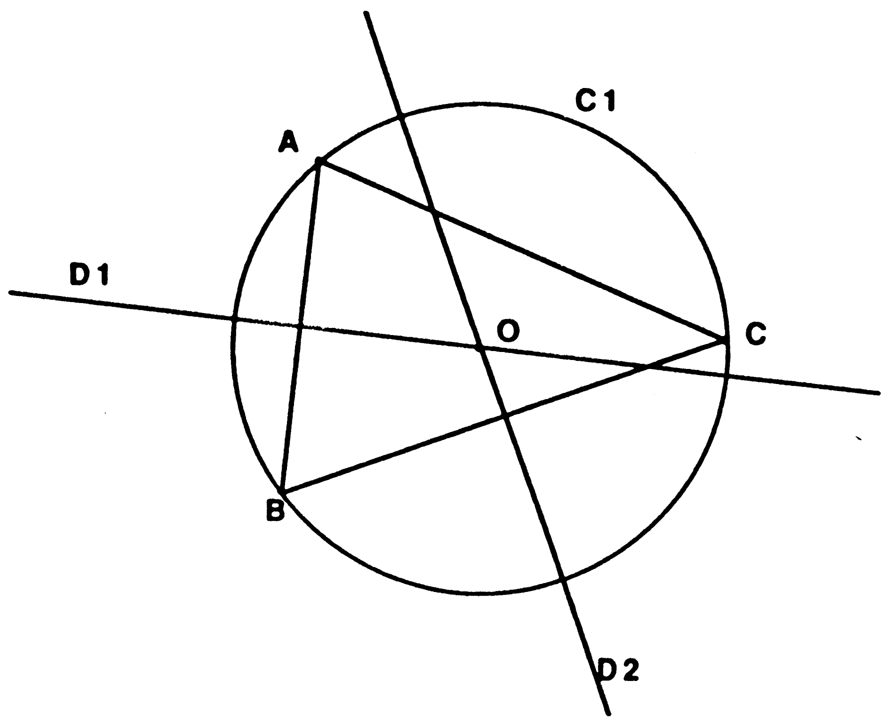
Il ne s'agit pas d'une véritable relation d'ordre partiel, la réflexivité n'ayant pas de signification ici.
On peut donc associer à cet ordre strict un diagramme de Hasse.

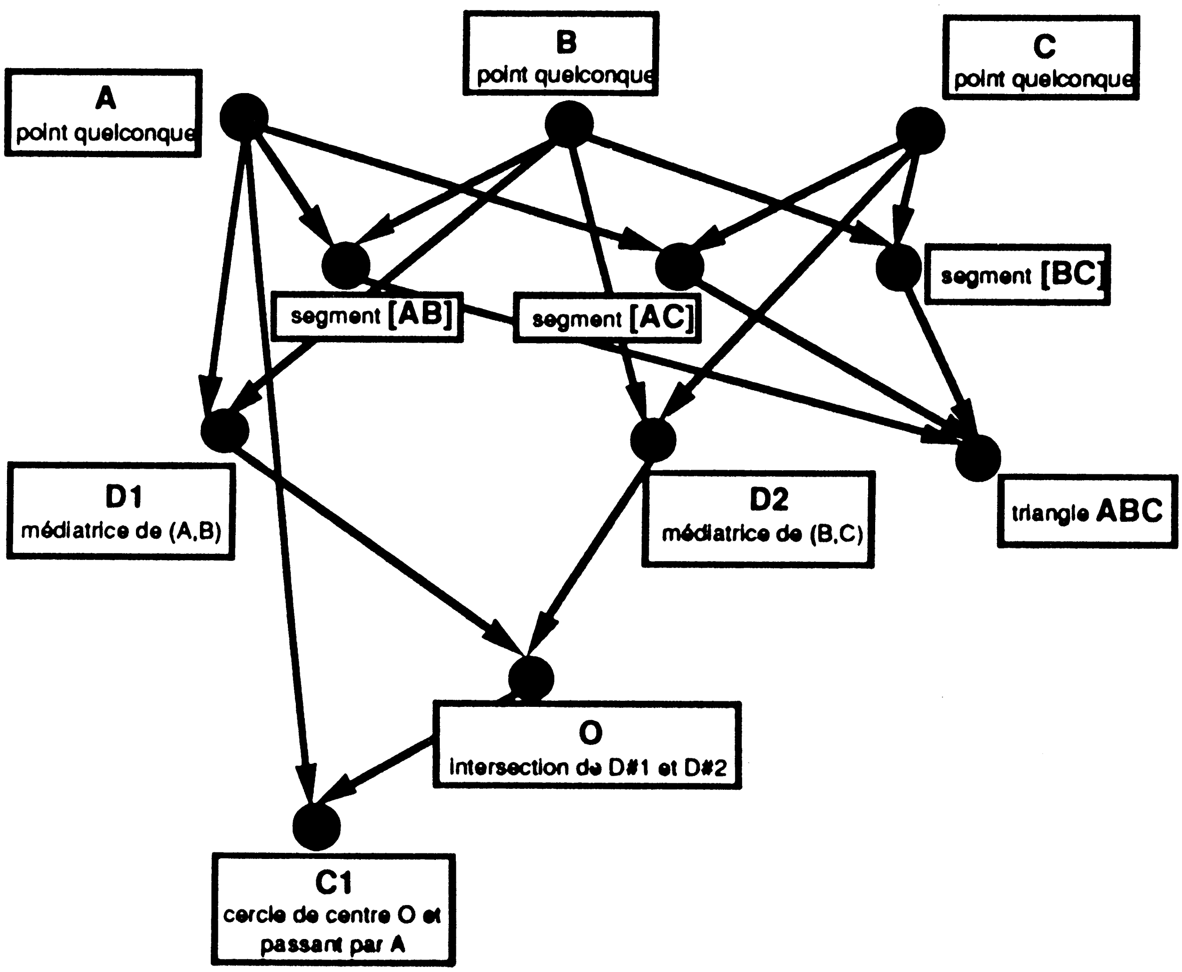
Cabri-géomètre construit donc un graphe, le **graphe de dépendance** de la figure; les sommets sont les différents objets de la figure et il y a une arête entre X et Y si Y a été construit à partir de X.
Les informations sur les relations géométriques liant un objet à ses prédécesseurs sont contenues dans les structures de X sous forme de

fonctions décrivant la construction.

Ce graphe est **orienté**, la relation de dépendance est un ordre, et **sans circuit**, la relation de dépendance est antisymétrique.
Les sources du graphe (les sommets sans prédecesseurs) sont des objets de base de la figure.
Les structures des objets permettent de conserver en permanence à la fois le graphe des prédécesseurs (constituants) et des successeurs (dépendants).
La déformation de la figure par déplacement d'un objet de base consistera à modifier les informations d'une source du graphe et à propager les effets de cette modification. La redéfinition d'un objet a posteriori consistera à modifier les arcs du graphe.

Voici un exemple de figure et du graphe de dépendance associé:




A
point quelconque
B
point quelconque
C
point quelconque
segment [AB]
segment [AC]
segment [BC]
D1
médiatrice de (A,B)
D2
médiatrice de (B,C)
triangle ABC
O
intersection de D#1 et D#2
C1
cercle de centre O et
passant par A

2.3 les fonctions principales

Les algorithmes qui permettent de manipuler les figures de Cabri-géomètre sont en grande partie des parcours et des modifications du graphe de dépendance associé à la figure.
La mise au point a consisté à se ramener aux algorithmes de parcours DFS, de mise en niveaux, de tri topologique, de suppression de sommets, d'extraction ou de fusion d'un sous-graphe, etc.
Nous ne décrirons pas ces algorithmes classiques ([Reingold, Nievergelt, Deo 77], [Xuong 88]).
Pour chacune des fonctionnalités principales de Cabri-géomètre, nous évoquerons les particularités des étapes de l'exécution et préciserons comment elle se ramène à tel ou tel algorithme.
Le problème de la consistance et de la complétude des constructions et des transformations de figure ne sera qu'évoqué dans ce paragraphe et nous l'examinerons plus en détail au §2.5.
Les opérations principales que nous allons étudier sont la construction d'un nouvel objet, la déformation de la figure par déplacement d'un objet de base, la création de macro-constructions, la redéfinition d'un objet et la validation de propriétés (voir le §2.1.4 pour les définitions de ces procédures).

2.3.1 construction d'un objet

Lors de l'opération de construction d'un nouvel objet les différentes tâches à exécuter sont successivement:

- détermination de la classe de l'objet à construire.
- sélection des constituants et/ou, dans le cas d'un objet de base, détermination de la position de l'objet dans la figure.
- vérification de la non-existence préalable d'un objet identique
- vérification de l'existence de l'objet dans le plan euclidien sous une forme non dégénérée et calcul des caractéristiques analytiques de l'objet

- mise à jour des structures et du dessin

Sélection d'objets

• Un objet est sélectionnable s'il est du type attendu (ces types sont des constantes de la classe de l'objet à construire) et si la distance de la souris à l'objet est inférieure à un seuil.
• la sélection des objets se fait à l'aide de la souris, le curseur indiquant le type de l'objet actuellement sous le curseur. La détection du ou des objets présents sous le curseur est donc préalable à l'activation du bouton de la souris et le résultat de cette détection est actualisé à chaque déplacement du curseur.
• L'optimisation de ces procédures (propre à chaque type d'objets) nous a conduit à utiliser la distance euclidienne (x^2+y^2), la distance Manhattan ($|x| +|y|$) ou la distance du maximum ($\sup(|x|,|y|)$) selon les cas.
• Il arrive que plusieurs objets soient simultanément sélectionnables. L'utilisateur est alors invité à lever l'ambiguïté dans une liste, ordonnée chronologiquement, de ces objets.

Unicité des objets

• Avant d'accepter la création d'un nouvel objet le système vérifie qu'un objet identique n'existe pas déjà sur la figure. Laisser l'utilisateur construire plusieurs exemplaires du même objet ne nuirait pas à l'aspect graphique de la figure mais ultérieurement conduirait à des ambiguïtés.
• Il faut remarquer qu'être capable de détecter systématiquement l'identité de deux objets (au sens de la logique géométrique) revient à résoudre la plupart des problèmes de géométrie. Par exemple détecter que les intersections deux à deux des médiatrices d'un triangle sont identiques, c'est démontrer que les médiatrices sont concourantes.
• A chacune des classes d'objets est associée une méthode de vérification d'unicité. Cette méthode, outre la vérification de l'identité au sens strict (même classe, mêmes constituants), utilise certaines

propriétés géométriques et règles d'inférence qui lui permettent d'opérer des déductions simples pour mettre en évidence la présence d'un objet identique.
Donnons comme exemple les composantes de la méthode associée à la classe intersection de deux droites:

- L'intersection existe déjà s' il existe un point commun à ces deux droites
- Un point A appartient à une droite D si:
    - la droite D a été construite comme passant par A
    - A est de la classe point sur droite et D est son constituant
    - A est milieu de deux points appartenant à la droite
    - A est le symétrique d'un point de D par rapport à un autre point de D
    - A est déjà l'intersection de D avec un autre objet

• Il est évident que de telles méthodes ne sont pas complètes. Elles permettent cependant d'éviter les duplications d'objets relevant non pas d'une méconnaissance de telle ou telle propriété géométrique mais d'une erreur de la part de l'utilisateur.

Existence d'un objet dans le plan

Un objet n'a pas toujours une représentation dans le plan euclidien. Ainsi par exemple l'intersection de deux droites parallèles est rejetée à l'infini, l'intersection d'une droite et d'un cercle peut être imaginaire.
Le système n'acceptera pas de créer un objet dont on ne peut pas avoir de représentation graphique.
Les relations géométriques que l'on donne à un objet sont représentées par des équations permettant de calculer les caractéristiques analytiques d'un objet en fonction de celles de ses constituants.
La résolution de ces équations n'est pas une résolution formelle mais une résolution numérique. Nous ne discuterons pas ici les problèmes d'approximation inhérents à une telle résolution numérique (voir le §2.5).
La détermination de l'existence de l'objet est une conséquence de l'application de la méthode de calcul associée à chaque classe d'objets.

Dégénérescence

Nous dirons qu'un objet est dégénéré dans les cas où sa représentation graphique n'est pas conforme au cas général. Par exemple un cercle dont le centre est à l'infini ou la médiatrice de deux points confondus. Bien que de tels objets aient une représentation en géométrie classique (un cercle de rayon infini est une droite) nous avons choisi dans Cabri-géomètre de ne pas accepter la construction de ces objets.

Si toutes ces conditions préalables sont remplies, l'objet est alors mémorisé et dessiné. Au graphe de dépendance de la figure est adjoint un nouveau sommet relié par un arc à chaque constituant de l'objet.

2.3.2 Déformation de la figure

La déformation de la figure est une des principales fonctionnalités de Cabri-géomètre et notamment une de celles qui n'ont pas d'équivalent dans l'univers papier-crayon.
Rappelons qu'elle permet de saisir un des objets de base de la figure (point, droite, cercle quelconque ou point simplement lié à un objet) et de le modifier tout en conservant les relations géométriques définies entre les autres objets à leur construction.

Les modifications que l'on peut faire sur un objet de base dépendent de la classe de cet objet:
point de base: translation de vecteur quelconque
droite de base: translation ou rotation autour d'un point de la droite
cercle de base: translation ou homothétie par rapport au centre
point sur objet: translation-projection sur l'objet

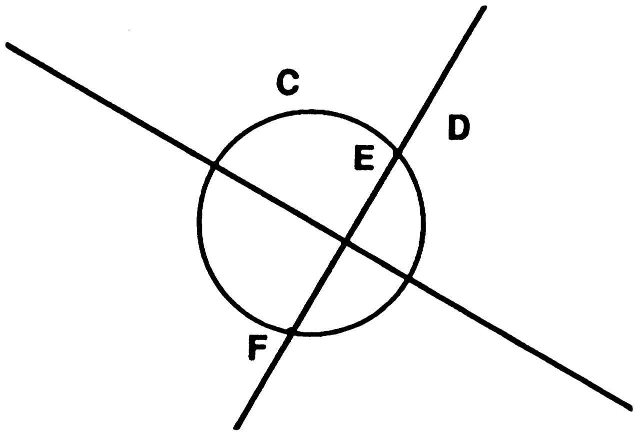
configurations d'une même classe.

Parmi les propriétés de la classe de figures il y a celles qui ont été explicitées à la construction et celles qui sont déductibles de la figure par les lois de la géométrie.
Les propriétés sont représentées dans les structures d'une figure de Cabri-géomètre par le graphe de dépendance et par la classe de chaque objet. Le graphe et la classe des objets ne sont pas modifiés par la déformation et les propriétés seront donc conservées. Les mêmes axiomes et théorèmes pourront être utilisés dans la nouvelle configuration pour effectuer les mêmes déductions.

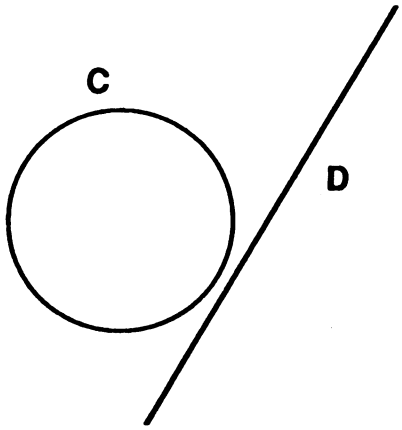
Les propriétés géométriques explicitées lors de la construction et les propriétés déductibles des précédentes par les lois de la géométrie sont conservées lors du déplacement d'un élément de base.

Disparition d'objet

Cabri-géomètre permet de construire des objets qui peuvent ne pas exister dans certaines configurations.
Construisons
- le cercle de base C
- la droite de base D
- les intersections E et F de D et C
- la médiatrice de (E,F)



Les intersections E et F ont été construites parce qu'elles existaient dans la configuration présente à l'écran. Mais si l'on déplace la droite D par exemple, on peut arriver à la configuration suivante dans laquelle les points E et F n'existent plus dans le plan euclidien.



La médiatrice de (E,F) disparaît également. Ces objets ne sont bien entendu pas effacés des structures puisqu'ils pourront apparaître à nouveau à un autre stade de la déformation.
Chaque objet possède un champ (état) qui reflète l'existence de l'objet à un instant donné. Si au cours de la déformation un objet change d'état, il propage cette information à tous ses descendants directs et indirects. Ces objets ne répondront plus aux messages de dessin par exemple. L'ordre dans lequel on signifie aux objets leur changement d'état n'a pas d'importance et c'est un parcours DFS du graphe de dépendance qui réalise cette propagation.
On peut discuter le bien-fondé d'autoriser un déplacement menant à la disparition d'un objet. On pourrait ainsi imaginer un scénario différent: si deux cercles sont sécants dans une configuration et que l'on construit

l'intersection, la propriété pour les deux cercles d'être sécants deviendrait alors une propriété de cette classe de figures. Le déplacement du centre d'un cercle serait alors limité à une couronne de manière à assurer l'existence de l'intersection.
Nous pensons que l'empilement de telles limitations deviendrait rapidement inextricable pour l'utilisateur. Par ailleurs il n'est pédagogiquement pas mauvais de visualiser la limite de validité d'une construction.

Actualisation des points liés à un objet

Un point dont la seule propriété déclarée est d'appartenir à un objet (droite, cercle ou segment) est un objet de base de Cabri-géomètre qui possède un degré de liberté. Comment doit être réactualisé un tel point lorsque son support est modifié?
Il n'y a pas de solution immédiate à cette question et nous avons essayé de limiter au maximum les perturbations dans la figure. Il apparaît vite que la projection du point sur la nouvelle position de l'objet ne donne pas de bons résultats (phénomènes de convergence et d'irréversibilité par exemple dans le cas d'une droite "tournant" autour d'un point).
Deux critères ont prévalu:
- que le point en question revienne à sa position initiale si son support fait de même,
- que le point ne soit pas modifié si le support, bien que descendant de l'objet déplacé, n'est pas lui aussi modifié.

Ces critères semblent triviaux, mais il ne semble pas qu'il soit possible de les satisfaire systématiquement.
Ce n'est que par tâtonnement que nous avons adopté le processus suivant:
un point sur une droite est modifié de manière à garder une distance constante avec un point origine de la droite,
un point sur un cercle conservera l'angle formé par ce point le centre du cercle et l'horizontale,

un point sur un segment conservera le rapport dans lequel ce point divise le segment.

génération de la déformation

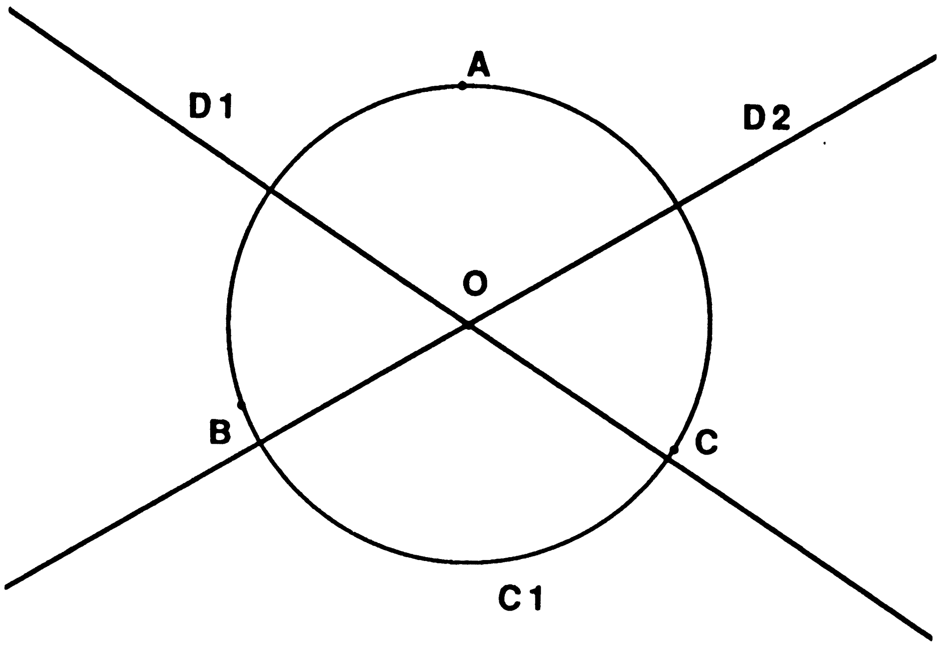
Dans le système actuel la déformation est la conséquence du déplacement non *programmé* d'un *unique* objet de *base*. On peut mentionner ici quelques extensions possibles que les structures de la figure et des interfaces permettent d'envisager:
• être en mesure de déplacer plusieurs objets de base simultanément
• pouvoir déplacer les objets de base selon une transformation géométrique (rotation, translation, homothétie, symétrie,...)
• pouvoir déplacer un objet qui n'est pas un objet de base en spécifiant le nombre nécessaire de "points fixes".

2.3.3 macro-constructions

Une macro-construction est une suite de constructions élémentaires qui peut être reproduite par un unique appel de procédure.
Une construction élémentaire est caractérisée par un nombre de constituants, le type de chaque constituant et la classe de l'objet résultant.
La structure d'une macro-construction en est une extension caractérisée par:
- le type des **objets initiaux** nécessaires à la réalisation de la macro-construction,
- les **objets finaux** qui sont le résultat de la macro-construction,
- les **objets intermédiaires** éventuellement nécessaires pour construire les objets finaux à partir des objets initiaux.

L'utilisateur définit une macro-construction sur une figure déjà élaborée en désignant successivement les objets initiaux et les objets finaux.
L'exécution de la macro-construction est réalisée après désignation d'objets ayant le même type que les objets initiaux.

La figure suivante correspond à la construction des objets suivants:
- points A, B, C quelconques
- droites D1 et D2, médiatrices de (A,B) et (A,C)
-point O, intersection de D1 et D2
- cercle C1, de centre O et passant par A



Dans une macro-construction permettant d'obtenir le cercle passant par trois points donnés:
il y a trois objets initiaux, chacun étant de type point,
il y a un seul objet final, le cercle C1,
les deux médiatrices D1 et D2 ainsi que leur intersection O sont des objets intermédiaires.

L'ensemble de ces objets initiaux, intermédiaires, finaux détermine un sous-graphe du graphe de dépendance de la figure.

Une macro-construction est représentable par un graphe de dépendance.

Les sommets de ce graphe sont des objets formels dont les champs sont:
- pour les sources, le type de chaque objet initial correspondant
- pour les autres sommets, la classe et le genre (intermédiaire ou final) de l'objet plus éventuellement certains attributs graphiques pour les objets

finaux.

La création d'une macro-construction consiste à extraire un sous-graphe G_m du graphe de dépendance G de la figure.
L'exécution de la macro-construction consiste à fusionner la graphe de dépendance G' de la figure et le sous-graphe G_m en identifiant les sources de G_m avec des sommets de G' en respectant les types.

Lors de l'exécution d'une macro-construction, les conditions d'unicité et d'existence sont imposées comme pour les constructions élémentaires (voir le §2.3.1).

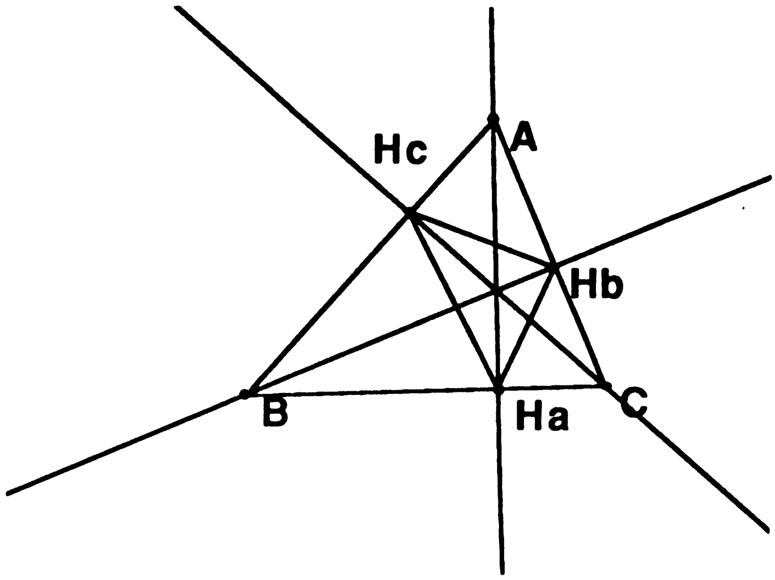
La création d'une macro-construction peut être entreprise dans une figure complexe et les objets initiaux ne sont pas obligatoirement des objets de base de la figure.
Le système teste alors la cohérence de la macro-construction en cours de définition en vérifiant notamment que les objets initiaux déterminent entièrement les objets finaux (dans le graphe des prédécesseurs, tout chemin issu d'un objet final passe par un objet initial)

Les contraintes sur les objets initiaux ne peuvent être que des constantes de type, il n'est pas possible d'imposer des relations entre les objets initiaux. Les relations qui pourraient exister dans la figure ayant servi à la définition de la macro-construction seront ignorées.

Il est par contre possible d'intégrer dans une macro-construction des objets intermédiaires non entièrement déterminés, comme des points appartenant à un objet. A l'exécution un tel point sera positionné aléatoirement sur le support, à charge pour l'utilisateur d'avoir prévu que cette détermination aléatoire n'a pas d'influence sur les objets finaux.

La création de macro-constructions n'est pas une opération très simple dans la mesure où il faut s'assurer de la généralité de la construction

pour qu'elle soit exécutable (et qu'elle donne le résultat attendu!) quelle que soit la configuration des objets initiaux.
Prenons l'exemple d'une macro-construction créant à partir d'un triangle, le triangle dont les sommets sont les pieds des hauteurs du triangle initial.



Dans cette figure les points Ha, Hb, Hc ont été construits comme les intersections de chaque hauteur avec le côté opposé.
Si on exécute la macro-construction sur un triangle dont un angle est obtus, le système refusera la construction demandée parce que deux des hauteurs ne coupent pas le côté opposé.
Pour être générale la macro-construction aurait dû être créée sur une figure dans laquelle les pieds des hauteurs sont construits comme intersection de la hauteur et du *support* du côté opposé.

Cette extensibilité des primitives de construction s'est avérée très utile et a donné lieu à des propositions supplémentaires d'extensions comme:
- Donner à une macro-construction une capacité d'itérer n fois une suite de constructions, les objets finaux devenant les objets initiaux à l'itération suivante (->fractals,...) ou bien un objet initial évoluant sur un support (-> enveloppes,...).
- Prendre en compte des contraintes entre les objets initiaux (par exemple appartenance d'un objet initial à une droite initiale).

- Définir des objets initiaux "globaux" qu'il n'est pas nécessaire d'indiquer à chaque application de la macro-construction.
- Opérer des modifications graphiques sur les objets initiaux après exécution de la macro-construction.

2.3.4 redéfinition d'objets

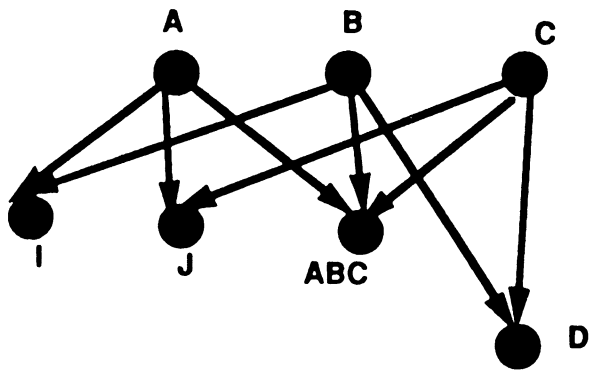
Un objet est défini par sa classe et par un ensemble de constituants. Redéfinir l'objet consiste à modifier certains de ces éléments.
On peut ainsi modifier les propriétés géométriques d'un objet, pour étudier par exemple une classe de figures présentant un ensemble restreint ou étendu de propriétés.

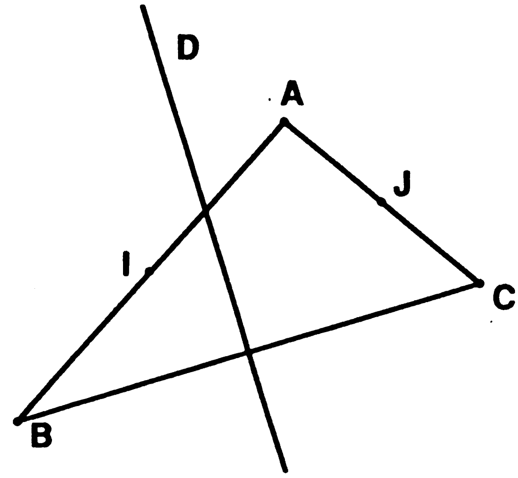
Cette redéfinition peut prendre différentes formes:
• identification de l'objet avec un autre objet de même type
• suppression de toutes les relations avec ses prédécesseurs (l'objet devient un objet de base)
• conservation de la classe de l'objet mais modification des prédécesseurs
• modification de la classe de l'objet

Les points, droites et cercles peuvent ainsi être redéfinis comme le résultat de n'importe laquelle des primitives de construction de Cabri-géomètre, sous la condition de respecter la cohérence des relations géométriques de la figure.

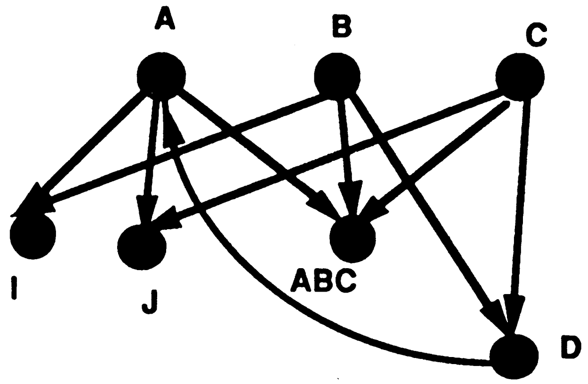
En redéfinissant un objet X, on n'affecte pas l'ensemble des dépendants de X, et les relations géométriques que X entretient avec ces objets seront conservées.

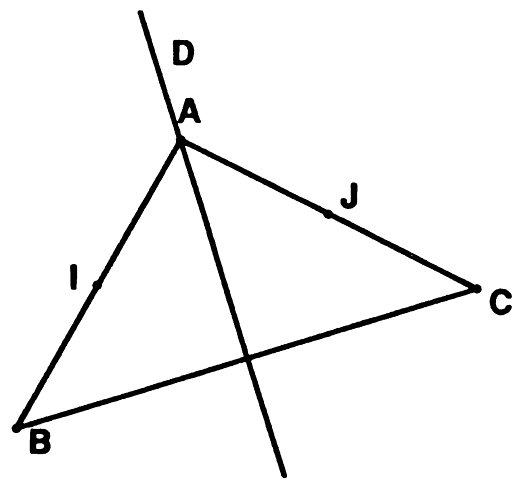
Soit la figure obtenue par les constructions suivantes:
- les points quelconques A,B et C
- le triangle ABC
- la médiatrice D de (B,C)
- les milieux I et J de (A,B) et (A,C) respectivement





Afin de donner au triangle la propriété d'être isocèle, on redéfinit le point A comme étant un point de la droite D. La figure et le graphe deviennent:





I et J restent milieu de (A,B) et (A,C). Le point A est désormais assujetti à la droite D et le restera dans toute déformation ultérieure de la figure.

Les différentes phases de la redéfinition sont:
- sélection de l'objet à redéfinir
- choix de sa nouvelle classe
- désignation des nouveaux constituants
- **vérification de la cohérence de la redéfinition**
- **mise à jour du graphe de dépendance**
- actualisation de la figure

• la nouvelle définition de l'objet se fait selon la même procédure que la construction initiale, à savoir, choix de la primitive correspondante dans un menu (seules certaines primitives, pour lesquelles il existe un exemplaire au moins de chaque constituant, sont activables), désignation à l'aide de la souris des constituants (et d'une position sur la figure si nécessaire).

• Si la redéfinition de X mène à un objet dont un exemplaire Y existe déjà dans la figure, il y a identification de X et Y.

cohérence de la redéfinition

Nous avons vu qu'il y a un ordre strict entre les objets et que la relation de dépendance est antisymétrique. Cela se traduit par le fait que le graphe de dépendance est et doit rester sans circuit.
Si X est construit à partir de Y, Y ne peut pas utiliser X pour sa définition.

Une redéfinition d'un objet qui entraînerait un circuit dans le graphe est rejetée par le système.

Mise à jour du graphe de dépendance

Sur le graphe de dépendance, la redéfinition revient à couper les arcs rentrants d'un sommet et à créer ceux correspondants aux nouveaux constituants. La classe de l'objet change éventuellement ainsi que certains attributs comme les caractéristiques analytiques de l'objet.

Après la mise à jour du graphe l'ensemble de la figure est réactualisé en effectuant comme dans l'opération de déformation un tri topologique sur le graphe engendré par les descendants d'un objet source. Dans le cas présent, la source est un objet fictif ayant pour descendants directs tous les objets de base de la figure.

Certains sommets ne doivent pas être conservés dans la nouvelle figure.

Il faut après une redéfinition épurer le graphe et les structures de tous ces sommets parasites:
- Dans le cas de l'identification de deux objets, le sommet correspondant à l'un d'entre eux doit disparaître.
- Une redéfinition peut entraîner l'apparition d'***objets jumeaux*** (objets ayant même classe et mêmes constituants) qu'il faut éliminer pour éviter des ambiguïtés ultérieurement.
- Les ***objets intermédiaires sans descendance*** ne mènent plus à des objets apparaissant sur la figure et il n'est pas nécessaire de les conserver.
- La redéfinition peut entraîner la dégénérescence complète de certains objets qui ont alors des constituants identiques. Soit par exemple deux points A et B et leur milieu I. On identifie A et B. I devient le milieu de A et de A et doit être supprimé des structures. Ces objets correspondent à des ***arcs doubles*** dans le graphe.

La détection et la suppression des sommets présentant les caractéristiques précédentes nécessitent des parcours successifs du graphe, certains n'étant pas linéaires comme la recherche et la suppression des jumeaux.

2.3.5 Validation de propriétés

Le terme de validation que nous employons ici se distingue de démonstration (dans l'acception scolaire du terme). Ce dernier sous-entend une explicitation du chemin menant au résultat. Nous avons présenté l'outil de validation de Cabri-géomètre comme fournissant un résultat sans prétendre expliciter ou expliquer une démarche (voir le §1.2.4).

La méthode de validation que nous proposons, dans le contexte du micromonde de Cabri-géomètre, se démarque nettement des autres travaux dans le domaine. Nous sommes conscients que son efficacité observable n'est pas, pour l'instant, étayée par une formalisation suffisamment rigoureuse.

Cet outil de validation est effectivement implémenté dans Cabri-géomètre et les résultats que nous avons obtenu sont très positifs.

Les approches logique et algébrique

Il existe de nombreux travaux dans le domaine de la validation de faits géométriques, certains intégrés dans des projets de logiciel pour l'enseignement, d'autres dont le but est de montrer la faisabilité de la validation automatique de faits géométriques, voire de démontrer de nouveaux théorèmes. La Geometry Machine [Gelertner 63a], une des réalisations pionnières, est dans ce dernier cas.

• De nombreux systèmes sont à base de règles de production dans lesquels les axiomes et les théorèmes sont modélisés sous la forme de règles d'inférence, inférence guidée par les techniques classiques de chaînage avant, de chaînage arrière, de parcours d'arbre et/ou, ...
La limitation de ces systèmes est d'ordre fonctionnel, en se restreignant par exemple aux cas d'égalités des triangles comme dans Geometry

Tutor ([Anderson & al. 85]). Elle peut être aussi opérationnelle, par exemple sans introduction de nouveaux objets non spécifiés dans l'énoncé ([Gelertner 65a], [Nevins 85]).

• Le système Mentoniezh ([Trilling & al. 87], [Nicolas 89]) propose la représentation des objets et des propriétés dans le langage de la logique du premier ordre sous forme de prédicats de typage et de prédicats associés aux propriété de base.
Cette *Théorie de Géométrie Instrumentale* est utilisée non pas pour effectuer une démonstration mais pour vérifier l'équivalence logique de deux formules correspondant à une figure construite et aux spécifications d'un énoncé.

• Wu a proposé une méthode de validation algébrique ([Wu 84], [Chou & al. 87], [Kapur 88]). Les faits géométriques considérés sont ceux qui peuvent s'exprimer sous la forme d'une équation polynomiale (les relations liées à l'ordre et au partage ne sont pas dans ce cas).
On résout formellement le système d'équations représentant la propriété à valider. Des paramètres représentent les coordonnées des points qui peuvent être choisis arbitrairement. Si la propriété est validée dans le cas général, les conditions de non-dégénérescence sont exprimées sous la forme d'inéquations sur ces paramètres.

Les méthodes évoquées présentent des obstacles majeurs à leur implémentation dans un micromonde comme Cabri-géomètre.
Le temps de réponse de ces systèmes est de l'ordre de la dizaine de secondes sur des machines de pointe (cf les résultats obtenus par Kapur dans [Kapur 89]). Dans un micromonde où l'interactivité joue un rôle prépondérant, une attente de cet ordre est prohibitive.
Le champ d'investigation de la méthode doit alors être réduit pour limiter l'explosion combinatoire des algorithmes et l'incomplétude qui en découle ne serait pas acceptable pour l'utilisateur final d'un logiciel éducatif.

Principes d'une validation sémantique

La notion de mesure, pourtant liée étymologiquement à la géométrie, est souvent ignorée par les systèmes précédents qui s'en tiennent à une expression formelle et logique des concepts de la géométrie et réalisent une ***approche syntaxique*** de la validation d'une propriété. La Geometry Machine utilise pourtant un *diagramme* sur lequel la vérification d'un sous-but est effectuée avant d'aller plus loin dans l'arborescence.
Dans cette approche une propriété est considérée comme valide sur une classe de figures, déterminée par un ensemble d'objets et de relations entre ces objets, si elle est un théorème, c'est à dire qu'elle peut se déduire des axiomes de la théorie et des hypothèses par une suite de transformations syntaxiques (modus ponens, généralisation).

Cabri-géomètre réalise une ***approche sémantique*** et une propriété est valide sur une classe de figures si elle est valide sur l'ensemble des configurations de la classe, c'est à dire sur l'ensemble des positions possibles des objets de base.
Dans l'espace euclidien muni d'un repère approprié, chaque point est univoquement associé à un couple de nombre réels, ses coordonnées x et y, un objet (une courbe) est un ensemble de points dont chacun vérifie une équation. Une relation entre deux objets se ramène à des égalités ou inégalités métriques sur les diverses coordonnées et paramètres de ces objets.
La validité sur une configuration donnée peut être donnée en utilisant la sémantique métrique de la propriété.

On peut donc séparer deux problèmes :

(1) Comment valider par le calcul la propriété sur une configuration donnée?

(2) Comment et dans quel cas généraliser à toutes les configurations possibles?

configurations possibles?

(1)
Les propriétés d'**appartenance** (d'un point à une droite ou à un cercle), de **parallélisme**, d'**orthogonalité**, de **congruence**, d'**alignement** (de trois points) sont les propriétés le plus souvent recherchées dans la géométrie scolaire. Toutes ces propriétés peuvent se traduire par un système d'équations.

Les coordonnées des points de la figure ne sont connues qu'à la précision près de la machine. Les égalités numériques du type Ei=0 peuvent-elles être déclarées vraies si Ei<epsilon? Cabri-géomètre est un micromonde et l'univers géométrique associé est limité par des paramètres comme les primitives de construction disponibles, le nombre d'objets constructibles dans une figure et la résolution de l'écran qui donne une borne inférieure à la distance de deux objets de base dans une figure.

Etant donnés
- un ensemble de primitives de construction
- une distance minimale entre deux points de base
- un nombre maximum d'objets

existe-t-il un nombre réel tel que pour tout élément d'une classe de figures C, la distance de tout couple de points est soit nulle, soit supérieure à ce nombre?

Nous pensons que si l'on peut donner une réponse affirmative à cette question (mais nous n'en avons encore que l'intuition), il nous semble que la véracité d'une propriété sur une configuration d'une classe de figures peut être acquise numériquement.

(2)
Dans une classe de figures C, soit R une relation ne faisant pas partie des hypothèses. La prise en compte de R partage C en deux sous-ensembles, celui des configurations pour lesquelles R n'est pas vérifiée et son complémentaire. Quelle est la mesure relative de ces deux sous-ensembles?

Prenons un exemple :

Considérons une classe C de figures composées d'un triangle ABC et de ses trois hauteurs. R1 est la relation "les trois hauteurs sont concourantes", les configurations ne vérifiant pas R1 sont celles dans lesquelles les points sont alignés ou non disjoints (cas de dégénérescence du triangle).
R2 est la relation "l'orthocentre du triangle est à l'intérieur du triangle", toute configuration dans laquelle un des angles du triangle est obtus ne vérifie pas R2.
R3 est la relation "l'orthocentre est le sommet A du triangle",seules les configurations dans lesquelles l'angle du triangle en A est droit vérifient R3.
Si l'on choisit aléatoirement une configuration de la classe C, la probabilité pour que R1 soit fausse (ou que R3 soit vraie) est nulle, la probabilité pour que R2 soit vraie ou fausse n'est pas nulle.
R1et R3 sont des propriétés qui se réfèrent aux axiomes d'incidence, de parallélisme, de congruence. R2 se réfère en plus aux axiomes d'ordre et de partage.

Les propriétés que l'on considère en géométrie scolaire sont pour la plupart des propriétés du type de R1 ou R3, quantifiées universellement avec certaines restrictions de non dégénérescence (conditions qui sont le plus souvent implicites). La probabilité d'obtenir par tirage aléatoire une configuration dégénérée est nulle. La véracité d'une telle propriété sur un nombre fini de configurations (voire une seule) permet d'assurer *raisonnablement* la généralité de la propriété.

Il se peut que la propriété implique des objets qui peuvent ne pas avoir d'existence dans une configuration aléatoire comme les intersections avec un cercle. Il est clair que l'utilisateur ne cherche à valider la propriété que dans le cas de l'existence des divers objets impliqués. Le choix pseudo-aléatoire des objets de base est alors soumis à certaines contraintes et éventuellement réitéré pour rester dans ce cas de figure.

choix pseudo-aléatoire des objets de base est alors soumis à certaines contraintes et éventuellement réitéré pour rester dans ce cas de figure.

Les différentes étapes de la validation d'une propriété par Cabri-géomètre sont:
- E1: évaluation du réalisme visuel (A_v) de la propriété sur la figure de l'écran (par exemple un point apparaît sur une droite s'il y a un ou deux pixels entre le point et la droite).
- si A_v alors E2: évaluation de la propriété sur une configuration dans laquelle les objets de base sont aléatoirement déterminés (A_g).
- si A_v et non A_g: détermination et affichage d'un contre-exemple.

contre-exemples

Si la propriété est réputée vraie sur la figure mais fausse dans le cas général, Cabri-géomètre fournit alors un contre-exemple dans lequel la propriété n'est pas vraie.
Ce contre-exemple ne doit pas être quelconque dans la mesure où il doit illustrer significativement la négation de la propriété.
A cet effet des règles en guident la détermination:
- déplacement si possible d'un seul des objets de base de la figure initiale
- maintien des objets dans la fenêtre
- choix d'un objet dont un faible déplacement entraîne un grand écart par rapport à la propriété considérée
- choix de la meilleure direction de déplacement de l'objet
- minimisation du nombre d'objets qui seront modifiés

Le choix final du contre-exemple est le résultat d'un compromis entre ces règles mais ne peut pas tenir compte de caractéristiques qualitatives comme l'aspect d'un triangle, la non proximité des objets principaux, ...

2.4 Spécifications de l'interface

L'interface assure la communication entre une application informatique et son utilisateur. Jusqu'à ces dernières années l'effort d'adaptation, inhérent à tout dialogue, était considéré comme principalement dévolu à l'utilisateur.
Mais dans le cadre de l'enseignement, qui nous intéresse ici, tout l'édifice de la communication de connaissances, que l'agent en soit d'ailleurs un enseignant ou un logiciel, repose sur la forme de cette communication. Que l'interface d'un logiciel éducatif soit compliquée ou peu conviviale, celui-ci sera rapidement abandonné, quelques en soient les qualités intrinsèques. Le manque de confiance et la dissipation de l'attention que provoquent par exemple des erreurs de syntaxe est à prendre en considération.
La définition des interfaces a donc été un point très important dans les spécifications de Cabri-géomètre. Les principaux critères qui ont guidés les choix de ces interfaces sont:

(1) L'apprentissage et l'assimilation du fonctionnement du logiciel ne doit demander que peu d'effort et de mémorisation.

(2) Il doit être possible de travailler sur des objets non nommés, ce qui est très souvent le cas dans l'environnement papier-crayon.

(3) Les manipulations doivent être simples et rapides pour ne pas interférer, par des interruptions trop longues, sur le déroulement d'une session.

Nous allons d'abord justifier le choix d'une interface du type graphique-souris pour satisfaire ces critères puis nous exposerons les principes guidant la définition de telles interfaces.

2.4.1 Une interface graphique-souris

De nouveaux concepts d'interface ont été développés au début des années 80 et ont donné lieu à ces interfaces du type graphique-souris, *à la Macintosh.*
Matériellement ce type d'interface se compose:
-pour la sortie des informations d'un ***écran graphique*** dans lequel chaque point est individuellement représenté par une information en mémoire.
- le principal périphérique d'entrée est une ***souris*** dont le déplacement est associé à celui d'un curseur à l'écran. Les informations transmises à la machine se composent d'impulsions sur le (un des) bouton(s) de la souris et de la situation du curseur dans l'écran au moment de l'impulsion.

Si l'écran graphique, associé éventuellement à une imprimante et à un haut-parleur générateur de bips sonores, fait plus ou moins l'unanimité, il n'en n'est pas de même pour la souris.

Nous exposons ci-après quelques avantages et inconvénients des principaux périphériques d'entrée d'information dans le cadre d'un micromonde pour la géométrie.

• Le ***clavier*** était jusqu'à ces dernières années le seul périphérique d'entrée de données. Il offre l'avantage, en enseignement assisté par ordinateur, d'obliger l'apprenant à formaliser dans un langage toutes les étapes de ses expériences et donc dans une certaine mesure d'en avoir assimilé le fonctionnement. Le principal inconvénient est que ce langage, par lequel on va programmer le logiciel, nécessite à lui seul un apprentissage, une mémorisation qui ne sont pas en rapport avec le sujet.
• La ***tablette graphique*** est le périphérique qui se rapproche le plus des instruments traditionnellement utilisés. Un problème qui surgit est celui

de la reconnaissance des formes dans un dessin où l'imprécision des tracés peut être importante. Il est en outre souvent difficile de faire la différence entre les propriétés désirées par l'élève et celles qui sont dues au hasard de la configuration ou à une conception implicite de l'élève. Les systèmes utilisant cette méthode, comme le système Mentoniezh [Trilling & al. 85], se servent d'un langage par lequel l'apprenant détaille les propriétés des objets qu'il trace sur l'écran. Et pour toutes les manipulations ultérieures de la figure, ce langage est bien sûr indispensable.
Un autre inconvénient de cette tablette graphique est qu'elle demande pour chaque machine l'acquisition d'un nouveau périphérique, assez onéreux et peu utilisé dans d'autres logiciels.

• La ***souris*** comme la tablette graphique se manipule facilement dès que l'on a mécanisé le réflexe du passage d'un mouvement de la souris sur un plan horizontal au mouvement du curseur sur le plan vertical de l'écran. Les informations nécessaires sont fournies à la machine par la simple activation du bouton après avoir positionné le curseur à l'endroit ad hoc.
Un inconvénient de la souris est justement cette facilité de manipulation. Un simple clic peut lancer une opération alors que ce mouvement n'est pas la conclusion d'une réflexion et d'un choix de l'élève, mais plutôt le fait du hasard, voire tout simplement de la curiosité.
La souris permet de fournir deux sortes d'informations à la machine, les procédures à effectuer et leurs paramètres. Pour ces deux actions (a) et (b), l'apport de la souris nous paraît primordial.

(a) Indiquer l' opération que l'on désire effectuer
Ce choix est le plus souvent fait en déroulant des menus dans lesquels sont répertoriées les diverses possibilités d'action à un moment donné. Il n'est donc pas nécessaire de mémoriser les commandes si ce n'est leur position - facilement retrouvée - dans l'ensemble des articles de menus.

(b)indiquer les paramètres nécessaires à l'exécution d'une procédure
Si on veut parler d'un objet dans un langage écrit, il faut presque obligatoirement que l'objet soit nommé pour que l'on puisse y faire référence. Or sur une feuille de papier l'élève va par exemple tracer un segment sans qu'il ait besoin de donner un nom à ce segment et/ou aux extrémités du segment pour pouvoir en tracer le milieu.
La souris permet de désigner un objet sans le référencer par un nom mais simplement en activant le bouton de la souris après avoir déplacé le curseur sur cet objet.

• les ***écrans tactiles***, sensibles à la pression du doigt, pourront avantageusement remplacer la souris. La précision et la rapidité de l'entrée d'information est la même qu'avec la souris et on évite ainsi à l'apprenant des activités parasites.

• On peut également mentionner pour mémoire que l'état des recherches et réalisations en compréhension de la langue naturelle ne permettent pas pour l'instant d'envisager une telle communication, même compte tenu de la faible dimension du vocabulaire et des constructions de phrases qui constituent le langage de la géométrie tel qu'on le manipule dans des énoncés ou des démonstrations.

2.4.2 les principes

Les premières réalisations de ces interfaces graphique-souris sont issues de recherches menées au laboratoire de la Rank Xerox et ont été implémentés dès 1981 sur les stations de travail Star.
Les concepteurs du Macintosh ont repris cette architecture de l'interface pour le "Finder", la partie visible du système d'exploitation du Macintosh qui assure la gestion des fichiers. Une des originalités a été alors de généraliser ces concepts et cette architecture à tous les logiciels et de

donner aux développeurs, par le biais d'une boite à outils puissante et rigide, un cadre dans lequel ils mouleront l'interface propre à leur application.
Il est remarquable de noter que cette architecture, longtemps demeurée en marge, se retrouve aujourd'hui dans la plupart des micro-ordinateurs et des stations de travail.

Les chercheurs du projet Star [Smith & al. 82] avaient dégagé de leur analyse que l'interface devait se conformer à quelques principes de base. Les caractéristiques d'une interface "conviviale" s'énoncent ainsi (nous conservons ici les expressions anglaises, simples à comprendre mais difficiles à traduire exactement):

- familiar user's conceptual model,
- seeing and pointing versus remembering and typing,
- what you see is what you get,
- universal commands,
- simplicity,
- modeless interaction,
- user tailorability.

La trivialité apparente de ces principes cache en fait de réelles difficultés de mise en œuvre. Nous allons analyser successivement ces principes en essayant de focaliser d'une part leur importance dans le contexte particulier d'un logiciel éducatif et d'autre part les difficultés et les obstacles auxquels nous nous sommes heurtés pour les respecter lors de l'élaboration de l'interface de Cabri-géomètre.

2.4.3 consistance de l'interface

Ce terme de consistance recouvre deux des principes précédents:
a) Consistance *externe* pour donner au système des analogies avec un ***modèle familier à l'utilisateur***,
b) consistance *interne* pour respecter d'une part un fonctionnement cohérent du système et d'autre part pour assurer une ***universalité des***

commandes tant au niveau de l'application elle-même que d'un ensemble de logiciels.

• Les interfaces du Star et du "Finder" du Macintosh sont profondément liées à la métaphore du bureau électronique sur lequel on écrit, on étale, on classe, on jette, on envoie des documents.
Cabri-géomètre fournit des outils qui sont une transposition des outils de l'environnement classique. Ce sont la feuille de dessin, le crayon, la gomme et nous avons utilisé les divers éléments de l'interface (icônes, articles de menu, curseurs, ...) pour maintenir cette correspondance entre environnement informatique et environnement papier-crayon.

• Les autres fonctionnalités de Cabri-géomètre, suggérées par la capacité de traitement des micro-ordinateurs actuels, sont forcément un peu plus éloignées du modèle familier de l'élève et donc, comme les expérimentations nous l'ont montré, un peu plus difficiles à assimiler.
L'élève doit ici faire lui-même le lien entre trois aspects: la représentation graphique de la figure, la structure logique de sa construction et les propriétés géométriques de sa figure.

• Il est difficile également de respecter tous les artifices utilisés en géométrie pour faciliter la compréhension d'une figure. La mise en place de certaines marques (d'angles droits ou d'angles égaux par exemple) est difficile notamment par le fait des éléments ***implicites*** contenus dans leur positionnement (voir le §2.5.2).

• La notion de consistance interne recouvre la notion de **conformité**: le système doit toujours répondre comme s'y attend l'utilisateur.
L'utilisateur attend à la fois des réponses identiques pour des manipulations identiques, un respect des propriétés et caractéristiques du monde réél modélisé dans un micromonde, la prise en compte implicite d'actions non-dites dans l'environnement classique,...
Donnons deux exemples:

- Dans une figure l'utilisateur trace une droite quelconque puis un point quelconque en faisant en sorte que l'endroit qu'il désigne pour construire le point est très proche de la droite. Il s'attendra à ce que le système donne de lui-même au point la propriété d'appartenir à la droite.
- Ce principe de conformité ne peut pas être systématiquement respecté quand la représentation des objets du domaine est approximative. Soit par exemple une figure composée d'une droite et d'un cercle quelconques (dans une configuration où ils sont sécants) et de leurs intersections. En déplaçant la droite, les points d'intersections se déplacent mais il est rare de pouvoir faire tangenter la droite au cercle et d'avoir les deux points confondus.

Nous examinerons en détail ce problème de la **cohérence des figures affichées** dans le §2.5.4.

• Cette dernière rermarque montre également les limitations du principe *WYSIWYG* (What You See Is What You Get) dans une application telle que Cabri-géomètre. Dans un traitement de texte, la qualité des objets (des caractères) visibles à l'écran doit être une bonne approximation du résultat final désiré, à savoir un document imprimé. Un micromonde comme Cabri-géomètre n'est pas un simple éditeur de figures géométriques, que l'on imprimera d'ailleurs avec une résolution souvent bien supérieure à celle de l'écran. L'apprenant construit ses expériences et ses raisonnements sur la figure telle qu'elle lui apparaît à l'écran et c'est celle-ci qui doit respecter au mieux les hypothèses.

• Nous allons, en donnant ici aussi deux exemples encore présents dans les versions actuelles de Cabri-géomètre, montrer qu'il faut rester très vigilant pour assurer l'universalité des commandes au niveau de l'application.

- Un objet de contrôle, sous la forme d'un bouton à activer, est souvent visible dans la barre de titre d'une fenêtre de Cabri-géomètre. Mais son activation signifie, selon les cas, l'*abandon* de l'opération en cours, la *fin* d'une sous-opération (macro-constructions) ou l'*attente* (lieu de points). Bien sûr le texte, le mot, inscrit dans ce bouton renseigne sur la nature de son usage mais demande un effort de lecture et d'interprétation supplémentaire.
- Une opération classique lors de l'élaboration d'une figure se solde par la disparition à l'écran de certains objets; il peut s'agir de l'*annulation* de la dernière construction, de la

suppression de ces objets ou de leur *masquage* pour éclaicir la figure et le mot *effacer* convient parfaitement pour ces trois opérations. Le choix des termes ainsi que du graphisme des curseurs et des icônes relatifs à ces trois opérations dont les conséquences sont très différentes a fait - et fait encore- l'objet de nombreuses discussions.

• L'universalité des commandes au niveau d'une famille de logiciels donne une facilité d'apprentissage d'une autre application de la même famille. Cette unification des commandes dans Cabri-géomètre, un micromonde uniquement destiné à l'enseignement, reflète peut-être un certain optimisme quant au développement de ce type d'applications et d'environnements.

2.4.4 manipulation directe

La *manipulation directe* des objets de l'application est le fait de pouvoir appeler directement un objet sans le recours à la description de cet objet dans un langage quelconque.
Les objets de l'application sont soit des objets de contrôle (fenêtres, procédures, boutons,...) soit des objets propres au domaine de l'application (des objets géométriques dans le cas de Cabri-géomètre).
Pour pouvoir manipuler directement les objets il faut que ceux-ci soient *visibles*. Cette visibilité passe par la transcription des objets sous une forme qui peut être affichable à l'écran.
Une conséquence de la visibilité de tous les objets est que l'utilisateur n'a pas à encombrer sa mémoire avec le catalogue et la description de ces objets.
Une autre, tout aussi importante, est que les objets n'ont pas besoin d'être nommés pour être passés en paramètres à une procédure par exemple.
Ce concept de manipulation directe, qui est à la base des interfaces graphique-souris tel celui du Macintosh, est une des données de base des interfaces de Cabri-géomètre.
L'application de ce concept n'est pas exempt de problèmes: la faible taille

de l'écran oblige ainsi à recourir à des artifices pour rendre visibles tous les choix possibles. L'utilisateur s'habitue à une rapidité de fonctionnement et il sera difficile de lui imposer la lecture et la compréhension simultanée d'un grand nombre d'informations.

• les fonctionnalités sont regroupées dans une hiérarchie de menus que l'on déroule à l'aide de la souris et la difficulté du déplacement dans cette hiérarchie est fonction directe de la complexité du logiciel. La transcription des outils disponibles sous forme d'une palette permet de repousser la limite de cette complexité.
• Certains dialogues peuvent être ainsi trop denses si les choix possibles sont trop nombreux. L'utilisation d'un arbre de choix n'est pas valable si l'on ne peut pas passer facilement d'une branche de cet arbre à une autre.
• La transcription de la signification d'un objet sous forme d'un mot ou d'une icône pose souvent un problème de précision ou d'ambiguïté.

Un problème posé par la manipulation directe des objets est celui de l'ordre dans lequel les informations sont données au système. L'utilisateur désire exécuter une procédure portant sur tels ou tels objets, doit-il indiquer l'action à effectuer puis les éléments sur lesquels portera l'action ou bien sélectionner un ensemble d'objets avant de choisir l'action à exécuter?
Dans le premier cas le fonctionnement sera dit modal: dès que l'on a choisi une opération, le système entre dans le mode correspondant et ne répondra qu'aux manipulations propres à ce mode:

C'est le cas par exemple de l'éditeur de texte "vi" d'Unix qui ne prendra en compte la frappe d'un caractère que si l'utilisateur s'est placé dans le mode écriture.

Une interaction non modale est un des principes émis par les pionniers des interfaces à manipulation directe. Elle permet de mieux assurer l'universalité des commandes à l'intérieur de l'application.

Cabri-géomètre a un fonctionnement extrêmement modal: on choisit par exemple l'article *Cercle* puis on entre un mode dans lequel le système ne répondra qu'à la désignation successive du centre et d'un point du cercle ou à la sortie du mode .
Ce choix d'une interaction modale dans Cabri-géomètre n'est pas le résultat d'une analyse des tâches, mais provient d'une simplification de l'implémentation logicielle à ses débuts. La modification de cette interaction demande une refonte complète de la structure de l'application et c'est là une des raisons pour laquelle la poursuite du projet Cabri-géomètre passe par une réécriture des programmes.
On peut cependant reconnaître à cette interaction modale de limiter les possibilités de manipulations erronnées.

2.4.5 simplicité et adaptabilité

La consistance de l'interface et la manipulation directe des objets de l'application sont deux éléments, nous venons de le voir, importants pour assurer la simplicité du système. Nous allons ici en mentionner quelques autres.

• La réalisation d'une opération peut se décomposer en actions simples (choix d'une procédure dans un menu, désignation des paramètres à l'aide de la souris, réponses à un dialogue,...) mais l'agencement spatial et temporel de ces diverses actions élémentaires peut, si l'on n'y prend pas garde, aboutir à une opération globale complexe pour l'utilisateur. Complexe parce que l'utilisateur a besoin de réfléchir pour trouver sur l'écran les informations dont il a besoin et/ou complexe parce que l'utilisateur doit mémoriser une suite d'opérations à effectuer et savoir à tout moment à quel stade de réalisation il en est.
Un exemple dans Cabri-géomètre illustre ce problème. Dans les premières versions du logiciel, la création d'une macro-construction et son enregistrement sur le disque faisait apparaître successivement quatre dialogues pour donner un nom à la macro-construction, en

donner une description (phrase d'aide), confirmer l'enregistrement et enfin donner un nom au fichier résultat. Un meilleur agencement de ces dialogues a permis d'en supprimer deux sans nuire, au contraire, à l'efficacité.

• Dans un micromonde tel que Cabri-géomètre, l'utilisateur, surtout s'il est novice en géométrie et/ou inexpérimenté dans le maniement de l'ordinateur, n'est pas à l'abri de mauvaise manipulations. Un certain nombre de *garde-fous* ont donc été mis en place pour éviter des aberrations géométriques ou la perte du contrôle de l'opération en cours. Citons par exemple:

- la possibilité d'abandonner une opération en cours ou bien d'annuler l'opération précédente et de revenir à l'état antérieur.
- seules les opérations ayant une signification dans la figure représentée sont disponibles (par exemple si la figure ne comporte qu'une seule droite, l'article Bissectrices est désactivé).
- la construction d'un nouvel objet n'est possible que si un objet identique n'a pas été préalablement construit.
- le système intervient en affichant des messages pour signifier telle ou telle impossibilité ou pour demander la confirmation d'une opération ayant des conséquences importantes.
- ...

• Dans le même ordre d'idée, des effets visuels sont utilisés pour donner des indications sur l'opération en cours, par exemple:

- les objets désignés clignotent jusqu'à la fin de l'opération
- le curseur change de forme selon la nature de l'opération:

- le type d'un objet sélectionnable est indiqué à côté du curseur:

- le nom de l'opération en cours est indiqué pendant toute l'opération
- une aide *en ligne* est disponible à tout moment
- ...

• La simplicité d'utilisation d'un logiciel passe souvent par une rapidité d'exécution des différentes opérations.

-Le choix d'une procédure dans des menus déroulants peut être fastidieux quand il se repète trop souvent. La plupart des systèmes utilisant des menus offrent la possibilité d'activer tel ou tel article du menu par une combinaison de touches, un *raccourci-clavier* correspondant au choix de l'article. Cabri-géomètre propose également un raccourci permettant de relancer immédiatement la dernière opération effectuée.
- Les macro-constructions, qui réalisent une extensibilité des fonctionnalités de base du logiciel, vont également dans le sens de la simplicité d'utilisation.

• Un logiciel éducatif doit pouvoir s'adapter à différents contextes d'application et donc permettre à l'enseignant d'adapter l'interface afin de proposer à l'élève les éléments dont il aura besoin et uniquement ceux-ci. La simplicité et l'adaptation de l'interface à l'activité demandée sont ici assurées par l'enseignant. Quelques outils sont actuellement disponibles pour permettre cette configuration de l'interface:

- possibilité de restreindre l'ensemble des fonctionnalités disponibles
- affichage d'un repère et des caractéristiques analytiques des différents objets
- ...

Cette adaptabilité de l'interface à des contextes d'enseignement est un élément très important dans un micromonde pour lequel il n'y a pas de solution actuellement reconnue comme étant la meilleure. Abordée très superficiellement pour l'instant, la définition d'une interface configurable fait partie des thèmes principaux de recherche en vue du développement de Cabri-géomètre. Deux questions [Laborde et Trilling 89] prédominent:
- Est-il techniquement envisageable de multiplier les facettes d'interface et de passer facilement de l'une à l'autre?
- Peut-on (doit-on?) envisager à terme d'appréhender les préférences d'un utilisateur en cours de session de travail et de configurer dynamiquement l'interface en conséquence?

2.5 Quelques problèmes

Cabri-géomètre met en jeu différents modèles de la géométrie: un modèle de géométrie axiomatique permettant de déclarer et de maintenir des relations entre des objets, et un modèle de géométrie analytique pour dessiner les figures à l'écran. Les limitations viennent respectivement de la profondeur limitée des déductions dont est capable un système interactif et de la précision finie avec laquelle sont menés les calculs (précision des nombres réels en machine) et les dessins (résolution de l'écran). Nous verrons que ***la consistance et la complétude des constructions*** sont souvent mises en défaut et que ceci se répercute sur la ***cohérence des interfaces*** proposées.

Un deuxième problème provient de la reconnaissance, en géométrie, de nombreux éléments implicites traditionnellement reconnus. Un "implicite" du tracé est la différence de représentation d'un point quelconque (par une croix ou un pâté) et d'un point d'intersection (les objets définissant l'intersection suffisent). Un "implicite" d'énoncé est d'omettre la spécification des cas particuliers à éviter: "Construire un triangle ABC et ses médiatrices" contient le fait que les trois points sont disjoints et non alignés. Un "implicite" de configuration est que la base d'un triangle isocèle est horizontale. Une propriété géométrique "implicite" est qu'un point dessiné sur une droite possède la propriété d'être sur cette droite, une marque "implicite"est le petit carré à l'intersection de deux droites reconnues perpendiculaires. Un objet implicite est par exemple le segment joignant deux points d'une droite, le triangle formé par trois segments ayant une extrémité commune deux à deux.
. Une telle ***gestion des implicites*** supposerait que ceux-ci soient inventoriés, que leur utilisation systématique soit reconnue, problème qui à lui seul fait l'objet d'un travail de thèse en cours de réalisation.

Un troisième point est la grande variété des objets et des propriétés manipulées en géométrie. Si les macro-constructions permettent de définir des primitives de construction pour des objets qui sont un assemblage d'objets élémentaires, ceux-ci ne sont pas reconnus en tant que tels par le système. C'est également le cas des lieux de points que Cabri-géomètre trace sur l'écran mais qu'il n'est pas possible de conserver ni de reconnaître. Nous verrons les problèmes liés à une telle ***extensibilité des catégories d'objets*** et la ***flexibilité de la structure d'une figure***.

Une règle générale et simplificatrice dans la structure de Cabri-géomètre est la systématisation du traitement. Ainsi tous les objets dégénérés sont considérés comme inexistants, tous les calculs sont effectués avec la même précision, toutes les relations doivent être explicitées, tous les points ont la même représentation, ...
Il apparaît que cette règle, qui a simplifié les structures de données et l'implémentation logicielle, peut être un obstacle à la prise en compte de certains phénomènes de géométrie. Une redéfinition des structures dans les versions ultérieures de Cabri-géomètre sera sans doute nécessaire.

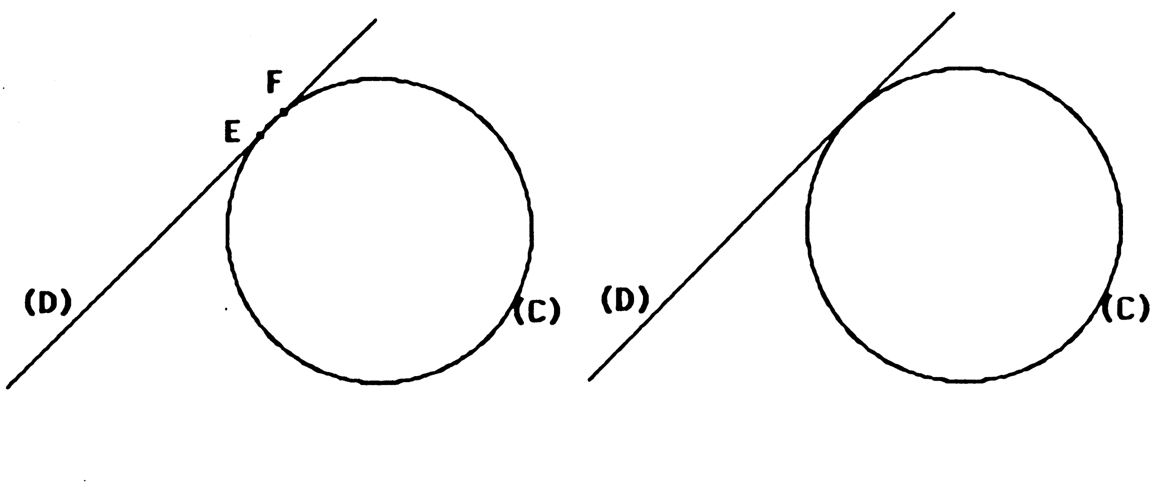
2.5.1 consistance et complétude des constructions-répercussion sur les interfaces

complétude

Peut-on obtenir, avec Cabri-géomètre, toutes les figures de la géométrie euclidienne plane? Que le mot figure signifie ici une représentation graphique ou une structure logique, la réponse est non, bien que les conséquences ne semblent pas trop pénaliser l'utilisateur.

• Les coordonnées des éléments de base de la figure sont déterminées par la souris et prennent des valeurs discrètes dans le maillage de l'écran. Cette limitation ne semble cependant pas poser de problèmes et l'on imagine bien que les propriétés d'une figure ne dépendent pas de la distance initiale des éléments de base.

• Les coordonnées des points calculés prennent leur valeur dans un autre sous-ensemble de R qui est l'ensemble des nombres réels accessibles par la machine. La limitation n'est pas celle des structures de ces nombres dans un compilateur. La précision des calculs peut être très grande mais elle est en tout cas fixée et limitée par des contraintes matérielles de taille de la mémoire et de rapidité d'exécution.
Le dessin d'un point se fera après troncature de ses coordonnées réelles pour le placer dans le maillage de l'écran.
Nous allons donner un exemple dans lequel cette troncature mène à l'invisibilité de certaines configurations particulières que l'on aimerait pouvoir mettre en évidence.
Construisons un cercle C et une droite D puis les intersections E et F.
Si l'on déplace la droite D les points E et F finissent par converger puis à se confondre avant de disparaître du plan. Il n'est que rarement possible d'obtenir à l'écran une configuration dans lequel la droite et le cercle soient tangents et que cette tangence se matérialise par l'identité des points E et F. Voici deux configurations obtenues en essayant de réaliser cette tangence:



Nous n'envisageons pas de solution immédiate à ce problème qui ne pourra être repoussé que par une résolution plus fine des écrans.

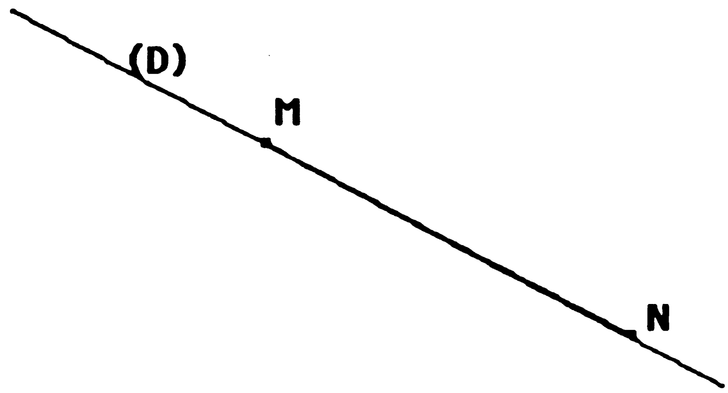
• Les valeurs admissibles pour les algorithmes de tracé donnent une autre limite aux figures de Cabri-géomètre. Un objet n'existe que si les valeurs le caractérisant sont en deçà de ces limites.
Prenons le cas de deux droites D et D', sécantes en O, un point A et le segment [OA]. On rend, par déformation, ces droites suffisamment parallèles pour que les coordonnées de O prennent des valeurs supérieures aux valeurs admissibles pour l'algorithme LineTo. Cabri-géomètre considère actuellement que le point O et donc le segment [OA] n'existent pas dans le plan, ce qui paraît un peu brutal. Une solution, qui n'est somme toute qu'une tricherie, consisterait à placer O "à l'infini" dans la direction de D et D', c'est à dire au maximum des valeurs admissibles dans cette direction. Nous nous sommes interdits ce genre de tricheries, qui semblent à première vue minimiser l'incomplétude mais dont nous ne pouvons pas maîtriser les effets de bord.

• Une autre limitation de Cabri-géomètre est l'ensemble des primitives de construction qui ne permettent pas de mettre en œuvre toutes les relations géométriques. Il est par exemple impossible de spécifier qu'un point appartient à un demi-plan par exemple.

consistance

Est-ce que la représentation graphique et la structure informatique associées à une figure de Cabri-géomètre respectent et traduisent les ordres de constructions donnés? là encore les imprécisions de tracé et de logique géométrique peuvent en faire douter.

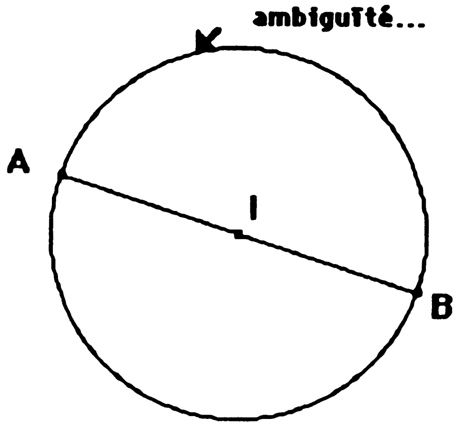
• Construisons par exemple une droite D quelconque puis deux points M et N de cette droite et enfin le segment MN.
La figure obtenue est bien souvent ainsi:



les segments [MN] et la droite D n'apparaissent pas colinéaires (ceci est plus net à sur un écran que sur la copie ci-dessus). Cette mauvaise superposition correspond à un noircissement de pixels différents provoqués par des bornes différentes paramétrant l'algorithme LineTo. Peut-être une meilleure définition de ces paramètres réduira l'apparition de ce phénomène, mais celui-ci ne pourra être traité que par le biais d'une connaissance des propriétés de la figure permettant de repérer la colinéarité de deux segments ou droites et de ne tracer qu'une fois la partie nécessaire de la droite support.

• La figure peut également comporter des objets logiquement identiques bien qu'obtenus par des méthodes de construction différentes.
Par exemple construisons le segment [AB], le milieu I de (A,B), le cercle C1 de centre I passant par A et le cercle C2, de même centre I et passant par B. Ces deux cercles sont confondus mais Cabri-géomètre ne peut actuellement pas le détecter.
Graphiquement la figure est bonne, mais la structure informatique s'est encombrée d'un objet double non reconnaissable parce que ces deux objets n'apparaissent pas sous la forme de jumeaux dans le graphe de dépendance.

Si ultérieurement l'utilisateur a besoin de se référer à ce cercle par exemple pour une nouvelle construction, le système constatera une ambiguïté:



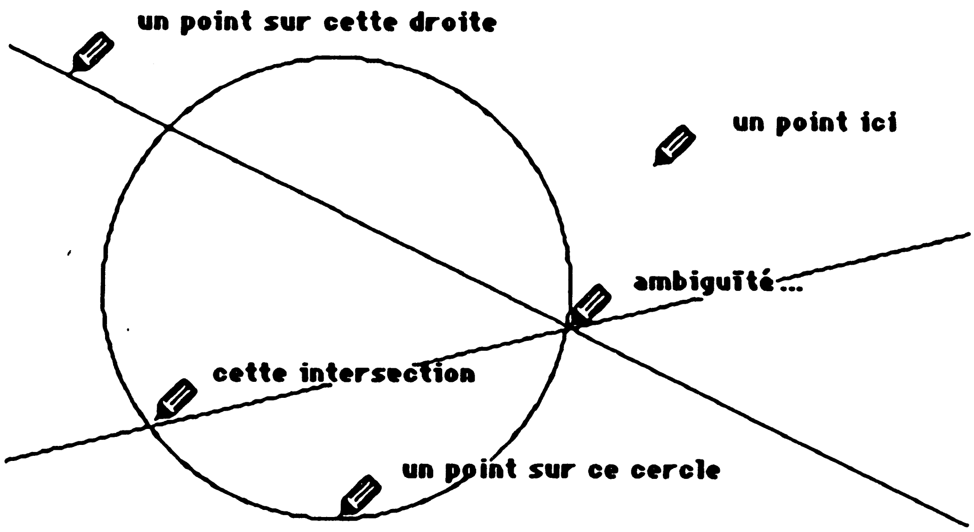
Certains des problèmes mentionnés dans ce paragraphe ne se résolvent qu'à travers une gestion des propriétés de la figure. Le système doit être en mesure de détecter les propriétés de la figure et d'en tirer les conséquences tant pour le dessin à l'écran que pour l'actualisation de la structure informatique. La vitesse d'exécution d'une telle détection est ici encore plus critique puisqu'elle doit pouvoir s'effectuer en temps réel au cours d'une déformation par exemple. La méthode de validation proposée pourra-t-elle être utilisée d'une manière fiable pour ces processus?

2.5.2 gestion des implicites

En début de paragraphe nous avons évoqué la variété et l'ambiguïté des implicites en géométrie. Certains points ont cependant été étudiés et implémentés dans des prototypes.

les propriétés implicites

Quand l'utilisateur choisit avec la souris la position d'un élément de base, il donne à ce point des propriétés implicites.
Peut-on, faut-il prendre en considération ces propriétés? Ce n'est pas le cas actuellement et toutes les relations inter-objets doivent être précisées par un appel de primitives et la désignation des objets impliqués.
Dans une figure un tant soit peu complexe, l'utilisateur sera malgré lui amené à donner à un nouvel objet une position lui conférant apparemment certaines propriétés non désirées.
La relation d'appartenance d'un point à un objet joue un rôle un peu différent en ce sens que sa perception visuelle est facile et précise. Il est douteux que l'utilisateur, construisant un point en cliquant sur un des objets de la figure, ne souhaite pas implicitement conférer cette propriété d'appartenance au point construit.
Nous avons construit un prototype dans lequel une primitive permet de construire les points quelconques ou les points quelconques sur un objet ou les points d'intersection. Le curseur donne constamment à l'utilisateur la nature du point qu'il va construire:



On pourrait envisager d'aller plus loin. Par exemple donner la classe milieu à un point lorsque celui-ci est créé au voisinage du milieu d'un bipoint. Outre les ambiguïtés que celà impliquerait, il ne semble pas raisonnable d'admettre toutes les relations qui sont presque réalisées sur une configuration comme étant des propriétés implicites. cela irait même à l'encontre de l'opération de déformation qui, nous l'avons dit, invalide expérimentalement les constructions au jugé.

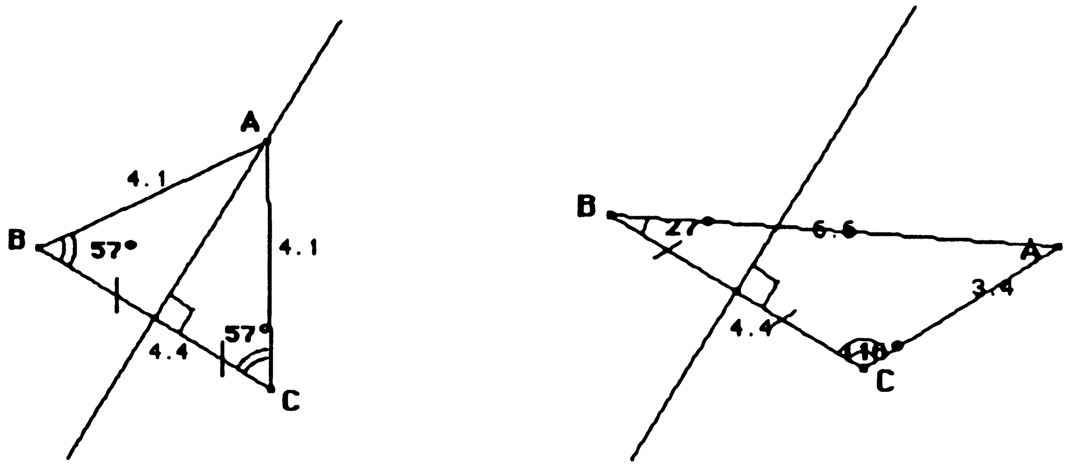
les points d'intersection

Le cas des points d'intersection de deux objets présente un problème supplémentaire qui est celui de leur représentation. Puisqu'un tel point est entièrement déterminé par des objets présents à l'écran, est-il nécessaire d'obliger sa construction explicite et doit-il avoir la même représentation que les points de base? Sur la feuille de papier, ces points ne sont pas soulignés par une croix ou un rond comme les autres points. On peut envisager différentes solutions qui devront peut-être coexister:
• le (ou les) point d'intersection de deux objets est créé et marqué d'un petit rond noir après désignation des objets (méthode actuelle lourde et peu comprise des utilisateurs).
• le point d'intersection est créé et marqué en cliquant à l'endroit de cette intersection (conforme à l'idée première de l'utilisateur débutant mais la désignation peut être difficile dans une figure complexe).
• le point n'est pas créé ni marqué explicitement mais le deviendra dès que l'utilisateur désignera l'intersection (une solution élégante mais qui alourdit beaucoup les algorithmes de recherche des objets présents sous le curseur)

les noms et les marques

Pour améliorer la lisibilité d'une figure, on a souvent recours à nommer les objets et à leur mettre des marques permettant de faire ressortir des propriétés peu visibles au premier coup d'œil (une ou plusieurs petites barres transversales pour indiquer les segments égaux, un petit carré à l'intersection de deux droites perpendiculaires,..).

• Le positionnement de ces marques conditionne la lisibilité de la figure: par exemple si la première des deux configurations suivantes est éclaircie par les différentes marques, ces dernières perturbent la lisibilité dans la deuxième configuration.



Le déplacement de ces noms et marques doit suivre un processus qui tient compte de la place disponible pour apposer une marque sur un objet mais aussi d'un choix d'objets privilégiés et d'implicites traditionnels (le nom des sommets est à l'extérieur du triangle par exemple). Une solution intermédiaire consiste à permettre le déplacement de ces marques par l'utilisateur, mais cette manipulation ne devrait être que le perfectionnement d'un placement automatique.

• Les marques doivent-elles être automatiquement tracées, par exemple dans le cas de droites perpendiculaires?. Et si le théorème : "Si D est parallèle à D' et D est perpendiculaire à D" alors D" et D' sont perpendiculaires" est connu, faut -il apposer toutes les marques?

les segments d'une droite

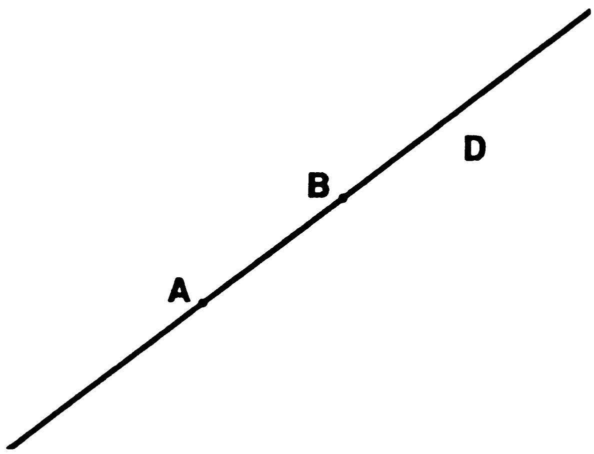
Soit une droite et plusieurs points appartenant à cette droite par construction (point sur objet, milieu, intersection,...). Ces points déterminent implicitement des segments de droite que l'utilisateur peut vouloir mesurer par exemple.
Or Cabri-géomètre ne reconnaît que les objets dont la construction a été explicitée.
La création systématique de tous les segments colinéaires à une droite conduirait vite à une saturation des structures. En effet si n points sont sur une droite, il y a $n(n-1)/2$ segments de droite identifiables.
Comme nous l'envisageons pour les points d'intersection, les algorithmes de recherche pourraient faire la liste de tous les segments identifiables sous le curseur à un endroit donné et le segment choisi dans cette liste serait alors créé. Les ambiguïtés seraient alors souvent nombreuses et par ailleurs le système devrait être en mesure de détecter l'appartenance d'un point à une droite, même si le point n'a pas été construit comme tel mais qu'il s'agit cependant d'une propriété déductible des autres constructions.

2.5.3 flexibilité des structures

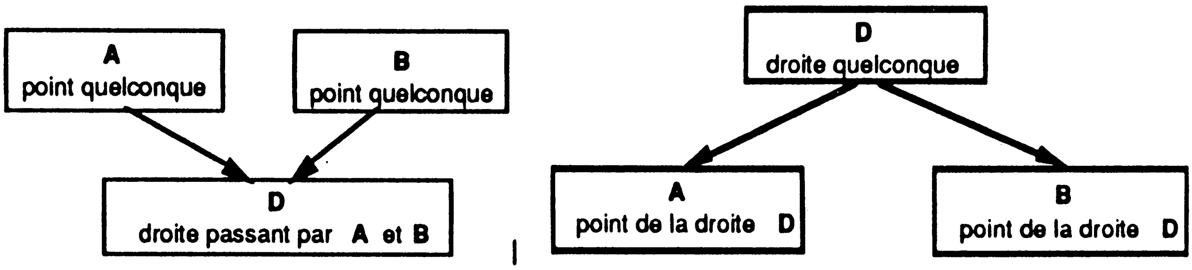
• La représentation d'une figure de Cabri-géomètre est une suite de constructions élémentaires qui se modélise par le graphe de dépendance des objets de la figure.
Cette représentation ne tient pas compte du fait qu'une figure (au sens d'un ensemble d'objets présentant des propriétés identiques) peut être obtenue par des méthodes différentes. Ces méthodes peuvent être triviales et l'utilisateur passe mentalement très vite de l'une à l'autre dans ses explorations. Il sera perturbé par le fait que le système, lui, associe une structure rigide à la figure, modifiable explicitement seulement.

Un exemple simple illustre bien ce problème:

La figure suivante est composée d'une droite D et de deux points A et B de la droite.



Dans Cabri-géomètre voici deux graphes de dépendance qui mènent à cette figure:



Dans le premier cas les points A et B sont des objets de base dont la position pourra être modifiée d'une manière quelconque mais la droite est déterminée par les points. Dans le deuxième cas, la droite est quelconque et pourra être modifiée. Les points A et B sont également des objets de base; ils peuvent être déplacés mais resteront assujettis à la droite.
La redéfinition des objets permet dans une certaine mesure de passer d'une représentation à l'autre mais nécessite des opérations intermédiaires.

Si cette rigidité de la structure pourrait être contournée dans certains cas précis, ***les relations géométriques entre les objets possèdent certaines propriétés que la structure actuelle ignore:***
- Les relations *appartient à* et *passant par* sont des relations duales
- La relation *parallèle à* est une relation d'équivalence donc symétrique
- La relation *perpendiculaire à* s'étend à tous les éléments d'une même classe d'équivalence de la relation *parallèle à*

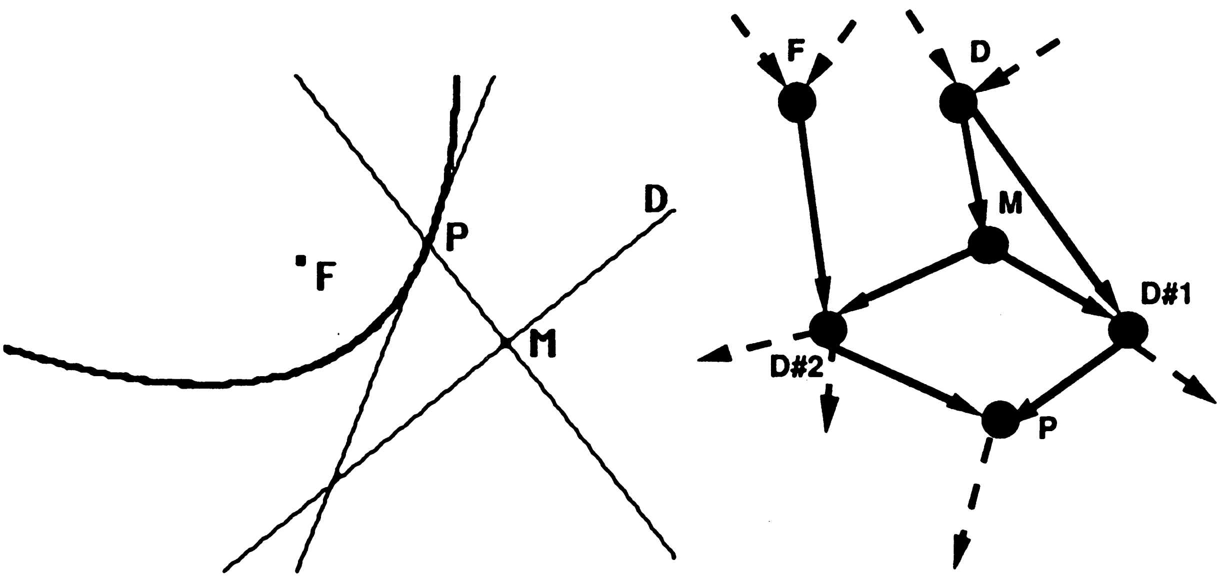
Le graphe de dépendance ne peut pas refléter ces propriétés des relations géométriques et les manipulations des figures ne peuvent pas en tenir compte.
Lors de certains parcours du graphe, des règles permettent de simuler ces propriétés (dans la vérification de l'unicité d'un objet à sa construction par exemple) mais il semble que la structure actuelle de la figure (objets élémentaires de classe donnée et graphe de dépendance)a atteint sa limite de validité pour les manipulations souhaitées.

2.5.4 extensibilité des catégories d'objets manipulés

Nous avons caractérisé les objets de Cabri-géomètre par leur ***type*** et leur ***classe***. Les objets sont du type point, droite, cercle, triangle, segment de droite. Un point est de la classe milieu, intersection, ...
Le type est une caractéristique d'ordre supérieur qui peut donner naissance à plusieurs classes. La classe d'un objet précise des éventuelles relations de cet objet avec d'autre objets, ces derniers ayant un type donné. Par exemple une instance de la classe milieu est un objet de type point qui entretient une relation d'équidistance avec deux objets, également du type point. A chaque classe d'objet on peut associer une primitive de construction.

Parmi les catégories d'objets qui font défaut dans Cabri-géomètre on peut citer:
• les courbes obtenues comme lieu géométrique de points
• les triangles remarquables et les polygones quelconques ou réguliers
Ces catégories d'objets ne peuvent pas être regroupées dans quelques types et classe statiquement créées.
Si l'ensemble des types et des classes d'objets est figé dans le Cabri-géomètre actuel, la structure interne autoriserait facilement l'implémentation de la création dynamique de nouvelles catégories d'objets.

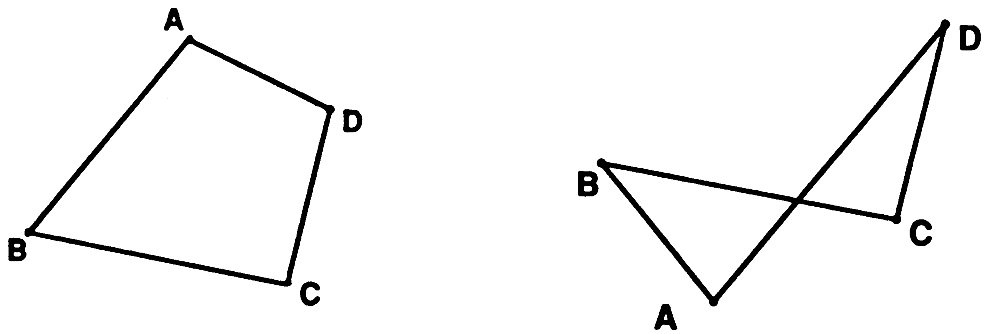
• Un lieu géométrique de points est une classe caractérisé par le(s) **point**(s) générateur du lieu quand un autre **point** se déplace sur un **support**. La donnée de ces deux points et de ce support permet d'isoler un sous-graphe dans le graphe de dépendance d'une figure. La classe parabole aurait comme constituants un point (le foyer F) et une droite (la directrice D), le point générateur de la parabole est l'intersection de la perpendiculaire à D en M, point quelconque de D et de la médiatrice de M et F.



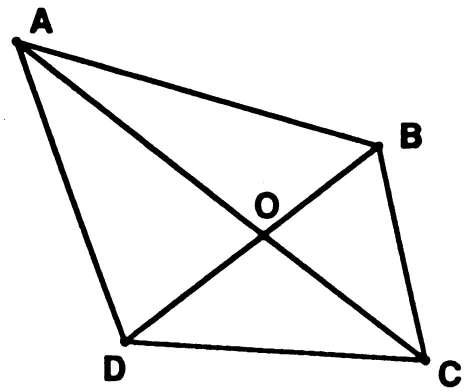
Un *type lieu* aurait comme méthode de tracé le tracé d'une suite de points, qui constitueraient les caractéristiques analytiques du lieu et comme méthode de calcul de distance, le minimum des distances à ces points.
La méthode de calcul de la *classe parabole* serait la répétition du calcul de P pour un ensemble de positions du point M. Le problème réside alors dans le choix des positions de M pour obtenir un ensemble de points P qui donne un lissage suffisant à la courbe. C'est possible mais relativement lent compte tenu du nombre de points à calculer. Cette lenteur, peu problématique lors de la création du lieu, deviendra un problème crucial si ultérieurement l'utilisateur modifie, dans notre

exemple, le point F. La parabole doit être recalculée et la continuité/rapidité du déplacement ne peut plus être garantie.

• Les polygones sont un assemblage de segments de droites. Dans Cabri-géomètre les extrémités d'un segment ou les sommets d'un triangle sont explicites et caractérisent le segment ou le triangle (il y a un type triangle et une classe triangle, dont les constituants sont trois points).
On peut envisager de rajouter statiquement un type polygone mais l'utilisateur qui construit un triangle ou un pentagone sera surpris si le système s'y réfère sous le nom générique de polygone. Il faut donc créer dynamiquement des types comme quadrilatère ou pentagone. Comme pour le triangle actuellement la création de ces objets entraîne la création implicite des structures correspondant aux côtés du polygone. Les différentes méthodes associées au type et classe pentagone sont héritées des classes composantes.
Il n'est pas possible d'imposer un ordre sur les points d'un tel polygone pour que le polygone reste convexe. Le déplacement du point A déforme le quadrilatère ABCD qui, de convexe, passe croisé.



• Il faut également envisager une détection automatique d'objets composés par le système. Actuellement le système ne reconnaît un triangle non pas s'il détecte trois segments ayant une extrémité commune deux à deux, mais si un triangle a été effectivement construit. L'introduction des polygones peut rapidement mener à une complexification de l'interface. En effet dans la figure suivante, il y aurait ambiguïté entre 5 objets, le segment AB, les triangles ABC, ABD, ABO, et le quadrilatère ABCD:



• La notion de triangle remarquable ou de polygone régulier s'intègre moins dans la structure actuelle parce que la construction d'un polygone régulier fait jouer un rôle privilégié à certains points du polygone.
Il est par exemple possible de définir un carré comme une construction déterminée par deux points opposés du carré. On peut ici aussi associer à la classe carré un graphe dont les sources sont deux points et les puits sont les quatre côtés du carré.
Dans le cas des triangles remarquables, il y a le problème de la détermination du troisième sommet du triangle lorsque l'on a fixé la base. Ce troisième sommet possède un certain degré de liberté (point d'un cercle pour le triangle rectangle, point d'une droite pour le triangle isocèle, élément d'un couple de points pour le triangle équilatéral).

Rappelons que cette création dynamique de nouvelles classes n'est pas réalisée dans le cabri-géomètre actuel.

2.6 l'implémentation

Nous donnons ici un bref aperçu du contexte et de certaines particularités de l'implémentation logicielle de Cabri-géomètre.

En quelques chiffres, le logiciel Cabri-géomètre représente
- l'écriture effective de **15000** lignes de programmes source en C,
- un code exécutable de **220** kilooctets,
- **4** hommes-années environ pour la réalisation du logiciel.

Cette première implémentation a été réalisée sur les microordinateurs de la gamme Macintosh, mais reste portable dans tout autre environnement matériel muni d'un écran graphique. Un portage sur les machines IBM/PC (qui sont les machines les plus fréquentes dans les établissements d'enseignement) a d'ailleurs été récemment effectué.

Précisons également que la constatation du "bon fonctionnement" n'est pas pour l'instant étayée par des études spécifiques de robustesse, de fiabilité ou de validité du programme.

Nous allons faire quelques remarques sur la boite à outils du Macintosh, le langage de programmation utilisé et les performances du logiciel.

2.6.1 la boite à outils du Macintosh

• L'architecture du programme suit le modèle préconisé par le constructeur. Le programme est une boucle d'événements, à chaque type d'événement est associé une fonction déterminée.
Un événement est soit occasionné par l'utilisateur (manipulation du bouton de la souris, entrée au clavier, insertion de disquette, ...) soit généré par le système (activation et/ou demande de mise à jour d'une fenêtre,...). Les événements sont stockés dans une pile avec des

niveaux de priorité puis traités séquentiellement. Le programmeur peut intervenir sur la gestion de cette pile en insérant ou en supprimant certains événements.

• Le fichier d'une application est un ensemble d'éléments distincts et de différents types, les ressources. Certaines ressources contiennent le code exécutable, d'autres des paramètres de l'application comme par exemple l'icône sous laquelle elle sera représentée, les articles de menu, les chaînes de caractères, les différents curseurs, etc...
C'est ainsi que le programme fait appel à telle ou telle ressource avant d'exécuter telle ou telle série d'instructions.
La modification des ressources, rendue possible grâce à des éditeurs spécifiques, n'implique en aucun cas une modification du programme. Par exemple la traduction d'un logiciel se fait sans nécessiter de recompilation. Cabri-géomètre existe d'ailleurs en version française, anglaise et allemande.

• Les normes d'interface préconisées par le constructeur (et qui assurent cette consistance externe dont nous avons parlé au paragraphe 2.4.3) sont extrêmement précises et leur respect conditionne en grande partie l'acceptation du logiciel par des utilisateurs courants de ce type de machine. Signalons, pour donner un exemple caricatural, que le nom d'un article de menu, s'il mène à un dialogue dans lequel l'utilisateur indiquera son choix, doit être suivi de trois points de suspension!

• La boite à outils du Macintosh réunit plusieurs centaines de fonctions (environ 800), pour la majorité inscrites dans des ROM, et que la plupart des environnements de programmation permettent d'appeler. Ces fonctions sont décrites dans Inside Macintosh [Apple 84-89] et sont regroupées dans des gestionnaires assurant tel ou tel aspect de l'interface ou accès aux fonctions du système d'exploitation.
Il y a entre autres Quickdraw qui regroupe toutes les fonctions graphiques, le gestionnaire de mémoire qui permet les allocations et le

remaniement de la mémoire, les gestionnaires de fenêtres, de dialogues, de menus qui donnent des fonctions de haut niveau pour la gestion de ces éléments d'interface, le gestionnaire d'événements, le gestionnaire des ressources, ...
Citons par exemple, la fonction NewWindow, qui permet de créer une nouvelle fenêtre, et qui va avec ses huit paramètres, réserver la place nécessaire pour la structure associée, rajouter cette fenêtre à la liste des fenêtres de l'application, appeler la procédure de dessin du cadre, inscrire le titre de cette fenêtre, ...

• Le gestionnaire de la mémoire du Macintosh présente certaines particularités qui permet une optimisation des allocations mais fait du maniement des blocs une des principales sources de bogues dans un programme [Knaster 86]. Le programmeur travaille sur la mémoire réelle et la moindre erreur se répercutera tôt ou tard sous la forme d'une erreur système menant dans le meilleur des cas à la classique "bombe" qui peut nécessiter un redémarrage de la machine. Pour permettre au système de faire des compactages de mémoire les blocs sont le plus souvent alloués par double indirection (les "handles") et le programmeur doit veiller constamment à leur cohérence.

2.6.2 langage et environnement de programmation

• Cabri-géomètre est écrit en langage C dans un des environnements les plus performants sur Macintosh, Think C [Think tech. 88] dont les principales qualités concernent
- la taille réduite et la rapidité d'exécution du code compilé,
- la convivialité dans la gestion de projets importants avec des facilités d'édition, de précompilation des différents modules,
- la puissance du debugger symbolique intégré.

• Les structures de Cabri-géomètre font largement appel aux concepts de la programmation orientée objet mais les langages correspondants ne sont disponibles sur Macintosh que depuis peu. La réécriture de programme, rendue nécessaire par les profondes modifications de structures et d'interface que nous envisageons, utilisera sans doute le langage C++ [Stroustrup 86].
Nous avons pu simuler certaines fonctionnalités de ces langages orientés objets [Cox 86][Schmucker 86] en utilisant les possibilités de C, notamment celle qui permet d'avoir accès à l'adresse du point d'entrée du code d'une fonction et à l'appel d'une fonction par son adresse.
Les adresses des méthodes communes aux objets d'un même type ou d'une même classe font partie de la structure associée.
L'utilisation des "macros"du langage C permet décrire dans le programme des instructions sous forme de message tel que:

Dessiner (unobjet)

qui sera traduit par le précompilateur par

(*(Ttype[(quelobjet)->type].procdessin))((quelobjet)->val);

puis qui sera compilé sous la forme de l'appel de la procédure de dessin associée au type de *quelobjet.*

2.6.3 performances

La plupart des opérations dans une application interactive font intervenir une manipulation de l'utilisateur et ne sont donc pas soumises à des contraintes de performance, le temps de réaction de l'utilisateur avant le passage à une autre opération est suffisamment important.
D'autre part la taille de la mémoire centrale des machines est bien suffisante pour les besoins de l'application et il n'a pas été réellement nécessaire d'optimiser la taille des structures.

Cependant certaines procédures doivent être effectuées "en temps réel" pour donner une impression de continuité.

C'est par exemple le cas notamment pour l'opération de déformation de la figure et de la sélection des objets lorsque le curseur (ou la chaîne de caractères qui lui est éventuellement attachée) change selon le type de l'objet le plus proche. On peut citer quelques points qui ont fait l'objet d'une programmation plus soignée ou pour lesquelles nous envisageons d'autres améliorations.
- Les méthodes de calcul des coordonnées des points caractéristiques des objets font appel à de nombreux calculs en virgule flottante (nombres réels codés sur 80 bits). L'optimisation de ces procédures requiert à la fois un bon choix du système d'équations à résoudre, le passage à des calculs en nombre entiers lorsque c'est possible et l'utilisation de bibliothèques de fonctions mathématiques plus performantes que celles du Macintosh.

- les méthodes de dessin font appel aux fonctions de tracé de lignes et de cercles de la Toolbox mais celles-ci ne sont pas toujours adaptées. Par exemple les cercles sont dessinés en utilisant l'algorithme de Michener, qui devient trop lent dès que le rayon des cercles est important et que seul un arc de faible rayon est visible. D'autre part, l'algorithme LineTo du Macintosh, surtout destiné à tracer des droites verticales ou horizontales, est moins performant pour les autres directions.
L'écriture de fonctions propres à Cabri permettrait par ailleurs d'obtenir des tracés en pointillés réguliers, ce qui n'est pas le cas actuellement.

- Une autre perte de performance est due à la gestion de la mémoire par le système d'exploitation du Macintosh. La réservation dynamique de mémoire est relativement lente puisqu'elle peut entraîner une compactation des blocs relogeables et reste en tout cas peu adaptée à la création de petits blocs. Or Cabri-géomètre manipule énormément de ces petits blocs par exemple dans les listes chainées d'objets.
Nous pensons que, tout en conservant le gestionnaire de mémoire du système pour les blocs importants ou liés aux fonctions de la Toolbox, il serait appréciable de faire une gestion personnalisée des petits blocs dans un espace mémoire réservé à cet effet.

Conclusion

Le Cabri-géomètre que nous avons décrit dans ce document représente à la fois la *fin* d'une recherche et un *moyen* de recherche. D'un côté, nous avons mené à terme la ***réalisation*** effective d'un logiciel pour l'enseignement de la géométrie, utilisable par les enseignants dans leurs classes. D'un autre côté, Cabri-géomètre est devenu un ***projet*** plus ambitieux de système intelligent d'aide à l'apprentissage de la géométrie dans lequel le micromonde actuel ne serait qu'un des éléments du tutoriel [Laborde et Trilling 89]. Et c'est sur la base de ce micromonde que pourront être imaginés et expérimentés d'autres processus d'apprentissage.

Nous allons, en conclusion, faire le point sur l'accueil qu'a reçu le logiciel et les directions de développement qui nous ont été suggérées par ces premiers utilisateurs.
Nous ne parlons que de développements "mineurs", ne remettant pas en cause la nature des objets manipulés. Nous pensons qu'il reste encore beaucoup de travail pour améliorer le logiciel existant et qu'il est prématuré d'envisager à court terme un Cabri-3D, un Cabri-algèbre ou un Cabri-CAO, même si nos démarches pédagogique et informatique semblent adaptables à ces autres domaines.

Sans tomber dans une trop grande auto-satisfaction, on peut noter que Cabri-géomètre a été très bien accueilli par les enseignants, les étudiants et les autres professionnels concernés (mathématiciens, didacticiens, ...) et nous avons déjà mentionné en introduction les distinctions dont le logiciel a fait l'objet.

• Cabri-géomètre n'a pas l'exclusivité des outils qu'il propose et nous avons vu, au §1.3, que la plupart des opérations de construction et d'exploration de figures géométriques se retrouvent dans l'un ou l'autre des travaux cités. Mais il reste l'un des rares et le premier à avoir introduit le concept de manipulation directe.
Son originalité réside principalement dans le souci constant de se conformer à l'attente des utilisateurs, tant au point de vue des fonctions que de leur mise en œuvre.
On peut citer, à ce dernier titre, la définition moderne de l'interface, le confort d'utilisation qu'apporte une rapidité d'exécution, l'étendue du domaine abordé, la configurabilité des menus disponibles, ...
La *boite à outils géométrique* est bien fournie avec la possibilité de construction de la quasi-totalité des figures de la géométrie scolaire, la déformation de la figure permettant de mettre en évidence les invariants géométriques, la redéfinition des objets, la création de bibliothèque de macro-constructions, la visualisation de lieux géométriques, la validation de propriétés dans le cas général, ...
D'autres sont en cours d'implémentation comme par exemple une calculette géométrique permettant de faire (et de conserver au cours des manipulations) des calculs algébriques ou logiques sur les objets et les propriétés de la figure.

• Les expérimentations menées dans des classes par des chercheurs en didactique de l'équipe semblent montrer également une bonne acceptation du logiciel par les étudiants. L'impact sur leur apprentissage de la géométrie n'est bien entendu pas encore mesurable, mais la large diffusion prochaine du logiciel dans les établissements d'enseignement et les nombreuses recherches menées par des didacticiens nous donneront les retours d'information nécessaires.

Il semble déjà que si l'interface est bien accepté dans ses principes, des contraintes comme par exemple les accès répétés à la barre des menus ou l'explicitation nécessaire de tous les objets et de leurs propriétés soient perturbantes. Nous avons vu au §2.5 les problèmes liés à cette simplicité et cohérence de l'interface pour laquelle une plus grande configurabilité est certainement nécessaire pour tenir compte de l'hétérogénéité des activités et des utilisateurs.

• Pourquoi se limiter à la géométrie de la règle et du compas?
Une activité, classique lors du premier cycle, comme *"construire un triangle dont la base fait 4cm"* ne peut pas être menée directement avec Cabri-géomètre. On peut cependant construire un triangle, mesurer la base puis déplacer une des extrémités de la base jusqu'à obtenir la valeur désirée.
Ou bien, pour faire rouler un cercle sur une droite et observer une cycloïde, on désirerait "*construire un segment ayant pour longueur la longueur d'un arc de cercle donné*". On peut ici aussi réaliser une construction approchée dont la précision est suffisante compte tenu de la résolution de l'écran. Mais il reste que ces constructions et d'autres sans doute devraient être accessibles plus simplement.

• Le titre de cette thèse mentionne un *micromonde de géométrie* plutôt qu'un *micromonde pour la géométrie.* En effet, Cabri-géomètre peut être utilisé pour simuler d'autres phénomènes que ceux de la géométrie scolaire. Citons deux exemples de ces nouveaux micromondes imaginés par des membres de l'équipe:

- On place sur une même droite des points représentant les quantités algébriques 0, 1, a et b puis on construit, en utilisant par exemple le théorème de Thalès, les points représentatifs de a+b, a*b, 1/a, √a, ... Les objets de construction ayant été cachés, le déplacement de a ou de b permet de voit évoluer le produit de deux nombres quand ceux-ci varient et de mettre ainsi en évidence des relations algébriques comme si *a<1 alors a*b<b*.

- les phénomènes d'optique, par exemple la taille et la position d'une image quand on fait varier les paramètres d'une lentille, peuvent être simulés par des constructions géométriques simples.

Si le nombre d'objets d'une figure de la géométrie scolaire n'excède généralement pas quelques dizaines, ce n'est plus le cas de ces nouveaux micromondes dans lesquels évoluent plusieurs centaines d'objets. Une amélioration des performances et des outils d'édition graphique s'avère nécessaire.

La géométrie est une science qui a attiré, à travers les siècles, les savants les plus éminents et il est encore utopique de penser à des processus informatiques mettant en œuvre toutes ces connaissances. Nous espérons cependant, avec Cabri-géomètre, proposer un nouvel outil d'apprentissage de ce domaine.

Bibliographie

Note : une bibliographie "Intelligence Artificielle et éducation" a été publiée dans TSI Vol 7 nº 1 1988 pages 167-172.

[Allard & Pascal 86] J.C. Allard , C.Pascal, *Euclide, un langage pour la géométrie plane,* logiciel et manuel, 1986, Cedic-Nathan.

[Anderson & al. 85a] J.R. Anderson, C.F. Boyle, G. Yost, The geometry tutor, 1985, *Proceedings of the ninth International Joint Conference on Artificial Intelligence.* Los Angeles CA.

[Anderson & al. 85b] J.R. Anderson, C.F. Boyle, B.J. Reiser, 1985, Intelligent Tutoring Systems, *Science*, 228, 456-462.

[Apple 84-88] , *Inside Macintosh*, vol 1-5, 1984-1988, Apple Computers Inc.

[Baudon 89], O. Baudon, 1989, *Cabri-graphes, un cahier de Brouillon Interactif pour la théorie des graphes*, Thèse de l'Université Joseph Fourier.

[Baulac & Cayet 87] Y. Baulac & Philippe Cayet, 1987, *Cabri-géomètre*, Mémoire d'ingénieur ENSIMAG, Grenoble.

[Baulac & Laborde 88] Y. Baulac, J.-M. Laborde, 1989, Sur l'interface d'un cahier de brouillon pour la géométrie, *Rapport de recherche LSD-IMAG* n° 740 I.

[Baulac 88] Y. Baulac 1988, Un système d'EIAO pour la géométrie élémentaire, *1er congrès Intelligence Artificielle et Formation*, APPLICA 88 Lille.

[Baulac, Bellemain, Laborde 88] Y. Baulac, F. Bellemain, J.-M. Laborde, 1988, *Cabri-géomètre, un logiciel d'aide à l'enseignement de la géométrie*, logiciel et manuel d'utilisation , Cedic-Nathan.

[Baulac & Laborde 89] Y. Baulac, J.-M. Laborde, 1989,Cabri-géomètre, pour un nouvel apprentissage de la géométrie. *TSI*, Volume 8, n°4.

[Baulac & Laborde 89] Y. Baulac, J.-M. Laborde, 1989,Cabri-géomètre, pour un nouvel apprentissage de la géométrie. *Congrès Education et Informatique UNESCO* Avril 1989

[Bellemain 88] F. Bellemain, 1988, Séquences d'enseignement sur la symétrie à l'aide de Cabri-géomètre, Document interne LSD.

Benzaken 1986] C.Benzaken, 1986, CABRI, un éditeur de graphe simple, d'aide à l'enseignement et à la recherche, *Rapport technique n° 1, Grenoble LSD-IMAG*.

[Bonnet 84] A. Bonnet , 1984, *L'intelligence artificielle, Promesses et réalités*, InterEditions, Paris.

[Braun & al. 88], G. Braun, J.-F. Dufourd, J. Françon,1988, About geometric construction programming, *Rapport de L'Université Louis Pasteur*, Strasbourg.

[Braun 89], G. Braun, 1989, Un outil pour la construction géométrique, *Annales de Didactique et de Sciences Cognitives*, IREM de Strasbourg, 2, 111-133.

[Burton 82] R. Burton, 1982, *Diagnosis bugs in a simple procedural skill*, Intelligent Tutoring Systems by D.Sleeman & J. S. Brown, London Academic Press

[Buthion 79] M. Buthion, 1979, Un programme qui résout formellement des problèmes de constructions géométriques, *R.A.I.R.O. Informatique* ,13, nº 1.

[Carbonell 70] J. R. Carbonell, Mixed-initiative man-computer instructional dialogues, *BBN report 1971*, Bolt Beranek & Newman Inc., Cambridge, Mass.

[Chou 85] Shang-ching Chou, 1985, *Proving and discovering geometry theorems using Wu's,method*, Technical report 49, Institute for computer Science, Austin, USA.

[Chou & al. 87] Shang-ching Chou, W.F. Schelter, Jin-Gen Yang, Characteristics sets and Gröbner bases in geometry theorem proving, *Computer-aided geometry reasoning, INRIA Worskhop*, June 87.

[Chouraqui & al. 88] E. Chouraqui, C. Inghilterra, J. Veronis J, 1988, *Archimède :un système expert d'enseignement de la géométrie*, GRTC/261/Janvier 88.

[Chouraqui & Inghilterra. 87] E. Chouraqui, C. Inghilterra, 1988, Conception d'une base de connaissances orientée objet pour l'EAO de la géométrie, *Actes de l'Université d'été Intelligence Artificielle et enseignement des mathématiques*, IREM de Toulouse.

[Cordier & al. 84] M. O. Cordier, B. Faller, D. Kayser, J.-F. Nicaud, EIAO : une application des systèmes experts à l'E.A.O., *Bulletin de l'E.P.I.*, 12/84.

[Cox 86] B. J. Cox, 1986, *Object Oriented Programming, an evolutionary approach*, Addison-Wesley

[Dede 86], C. Dede, 1986, A review and synthesis of recent research in intelligent computer-assisted instruction, *International Journal of Man-Machine Studies*, 24, 329-353.

[Fleck & al. 84], C. Fleck, J. Fleck, G. Kuntz, D. Pennerath, ELOGE, un logiciel pour l'enseignement de la géométrie, *Bulletin de l'E.P.I.*, Octobre 84.

[Foley & Vandam 84] J. D. Foley, A. Van Dam, 1984, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley

[Gelertner 63a], H. Gelertner, 1963, realization of a geometry-theorem proving machine, *Computers and Thoughts*, Ed. E.A. Feigenbaum & J. Feldman, McGraw-Hill, New York,p134 152.

[Gelertner 63b], H. Gelertner, J.R. Hansen, D.W. Loveland, 1963, empirical explorations of the geometry-theorem proving machine, *Computers and Thoughts*, Ed. E.A. Feigenbaum & J. Feldman, McGraw-Hill, Newyork,p 153-163.

[Gras 88] R. Gras, 1988, aide logicielle aux problèmes de démonstration géométrique dans l'enseignement secondaire, *petit X*,17,65-83, Grenoble.

[Guin & Rousselot 87] D. Guin, F. Rousselot , 1987, Recherche en vue de la réalisation d'un programme E.A.O. d'aide à la démonstration en géométrie, *Actes du Congrès MARI COGNITIVA 87*, Paris.

[Guin 89] D. Guin, Réflexions sur les logiciles d'aide à la démonstration en géométrie, *Annales de didactique et de sciences cognitives*, IREM de Strasbourg, 2, p 89-109, 1989.

[Habib & Laborde 83] M. Habib et J.-M. Laborde, 1983, Document de constitution du groupe CABRI, *PRC Mathématiques et Informatique*.

[Hilbert/Rossier 71] D. Hilbert, 1971, *Les fondements de la géométrie*, édition critique préparée par Paul Rossier,DUNOD, traduction de Grundlagen der Geometrie, B.G. Teubner Verlag, Stuttgart.

[Holland 88] G. Holland, 1988, Un logiciel de résolution de problèmes de preuve en géométrie utilisé en tant qu'expert d'un système tutoriel, *Actes du premier colloque franco-allemand de didactique*, 275-282

[Holland 87] G. Holland ,*TRICON, ein tutorielles System für Dreieckskonstruktionen*, Université de Giessen (RFA).

[Jackiw 89] Nicholas Jackiw, *Geometer's sketchpad*, Collège de Swarthmore, USA.

[Kaltenbach &Frasson 87] M. Kaltenbach, C. Frasson, 1987, DYNABOARD : User animated display of deductive proffs in mathematics, *International Journal of Man-Machine Studies*, 30, 149-170.

[Kapur 88] D. Kapur, A refutational approach to geometry theorem proving, *Artificial Intelligence*, Vol. 37, December 1988

[Knaster 86] S. Knaster, 1986,*How to write Macintosh Software*,Hayden Book Company, U.S.A.

[Laborde 86] J.M. Laborde , 1986, *Proposition d'un Cabri-Géomètre, incluant la notion de figures déformables*, Sujet d'année spéciale ENSIMAG.

[Laborde 87] J.M. Laborde , 1987, CABRI,un outil d'aide à la recherche en théorie des graphes, *Colloque "La Combinatoire et l'Informatique"*, mai 87, Montreal, conférence invitée.

[Laborde 88] J.M. Laborde , 1988, *A propos de la validation de faits géométriques dans un système d'enseignement assisté par ordinateur dans le domaine de la géométrie élémenatire,* Rap. Recherche n°... LSD IMAG dec 88.

[Laborde & Trilling 89] J.-M. Laborde, L. Trilling, 1989, Conception et réalisation d'un système intelligent d'apprentissage de la géométrie, Présentation de projet LSD-IMAG.

[Laborde 89] J.M. Laborde , 1989, Designing Intelligent Tutorial Systems: the case of geometry and Cabri-géomètre. *IFIP WG 3.1 Working Conference on Educational Software at the Secondary Education Level.*

[Lawler 87] R.W. Lawler, Learning environments : now, then and someday, *Artificial Intelligence and Education*, Ed. R.W. Lawler & M. Yazdani, Ablex publishing NJ.

[Lelong-Ferrand 85] J. Lelong-Ferrand, 1985, *Les fondements de la géométrie*, P.U.F.

[Lelouche 87], R. Lelouche, 1987, Apports de l'E.I.A.O. à l'E.A.O., *Actes du congrès francophone sur l'enseignement assisté par ordinateur*, Cap d'Agde.

[Nevins 75], A.J. Nevins, 1975, Plane Geometry Theorem Proving Using Forward Chaining, *Artificial intelligence*, 6, p. 1-23.

[Nicaud &Vivet 87], J.-F. Nicaud, M. Vivet, 1987, Les tuteurs intelligents, réalisations et tendances de recherche, *Rapport de recherche nº 359 du L.R.I.* .

[Nicolas 89], P. Nicolas, 1989, *Construction et vérification de figures géométriques dans le système MENTONIEZH*, Thèse de l'Université de Rennes I.

[Papert 80], S. Papert, 1980, Jaillissement de l'esprit, ordinateurs et apprentissage, Flammarion.

[Papert 87], S. Papert, 1987, Microworlds : transforming education, *Artificial Intelligence and Education*, Ed. R.W. Lawler & M. Yazdani, Ablex publishing NJ.

[Picard & Braun 87] M. Picard, G. Braun, *Les logiciels éducatifs*, collection Que sais-je?, P.U.F.

[Pluvinage & Rauscher 86], F. Pluvinage, J.-C. Rauscher, 1986, La géométrie construite à l'essai, *petit X*, 11,5-36, Grenoble.

[Py 88], D. Py, P. Nicolas, E.D.A.H.O. : un système d'enseignement assisté par ordinateur appliqué au domaine de la géométrie, *Convention I.A. 88-89*, à paraître.

[Reingold, Nievergelt, Deo 77] E.M. Reingold, J. Nievergelt, N. Deo, Combinatorial Algorithms, Prentice-Hall, 1977.

[Schumann 88] H. Schumann , 1988, 2D-Grafiksysteme für den Geometrieunterricht in der Sekundarstufe, *Beiträge zum Computereinsatz in der Schule*, Weingarten Hochchule, R.F.A.

[Self 87] J. Self, 1987, Student Models : What Use Are They, *proceedings of the IFIP TC3 working conference on artificial intelligence tools in education*, Ed. P. Ercoli & R. Lewis, North-Holland.

[Smith & al. 82] D.C. Smith, C. Irby, R. Kimball, B. Verplank, E. Harslem, *Designing the Star User Interface*, Byte 7 n° 4 1982

[Stroustrup 86] Stroustrup B., 1986, *The C^{++} programing language*, Addison-Wesley, U.S.A.

[Schwartz & Yerushalmy 85] J. Schwartz, M. Yerushalmy, 1985, *The Geometric Supposer*, Sunburst communications, U.S.A.

[Tallot 85] G. Tallot, *Quelques propositions pour la mise en œuvre d'algorithmes combibatoires*, Thèse de troisième cycle, Université des Sciences et Techniques du Languedoc, Montpellier

[Think C 88], THINK Technologies, *LightspeedC, C programming environment,* Symantec Corporation.

[Trilling & al. 87] L. Trilling, R. Allen, P.Nicolas, 1987, Spécification logique de figures pour l'aide à l'enseignement de la géométrie, *Actes Cognitiva 87*, Paris.

[Vivet 87], M. Vivet, 1987, Systèmes experts pour enseigner: meta-connaissances et explications, *Congrès international MARI/COGNITIVA*, Paris

[Vivet 89] M. Vivet, *EAO et IA, quelles nouvelles tendances pour l'EAO, que permettent les "tuteurs intelligents"?*, séminaire IMAG-LGI, séance du 2 février 1989

[Wenger 88], E. Wenger, 1988, *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmann Publishers Inc., Los Altos CA.

[Yazdani 86] M. Yazdani, 1986, Intelligent tutoring systems survey, *Artificial Intelligence Review*, 1, p. 43-52.

[Yazdani 87] M. Yazdani, 1987, Intelligent tutoring systems : an overview, *Artificial Intelligence and Education*, Ed. R.W. Lawler & M. Yazdani, Ablex publishing NJ.

[Wu 84] W. Wu, On the decision problem and the mechanization of theorem proving in elementary geometry, *Theorem Proving : after 25 years*, Eds W.W. Bledsoe & D.W. Loveland, Contemporary Mathematics vol. 29, 1984.

[Xuong 88] Xuong Ngyuen Huy, *Eléments de mathématiques discrètes pour l'informatique*, T 3,UJF/INPG Grenoble

Annexes

Annexe A
les classes d'objets de Cabri-géomètre

Les différents types d'objets sont:

point
droite
segment
cercle
triange
angle

Les différentes classes d'objet sont:

point quelconque
droite quelconque
cercle quelconque
segment
triangle

droite passant par deux points
cercle defini par un centre et un point radial

milieu d'un bipoint
mediatrice d'un bipoint

droite passant par un point et parallèle à une droite
droite passant par un point et parallèle à un segment
droite passant par un point et orthogonale à une droite
droite passant par un point et orthogonale à un segment

centre d'un cercle

intersection de deux droites
intersection de deux segments
intersection d'un segment et d'une droite
1ère intersection d'une droite et d'un cercle
2ème intersection d'une droite et d'un cercle
1ère intersection d'un segment et d'un cercle
2ème intersection d'un segment et d'un cercle
1ère intersection de deux cercles
2ème intersection de deux cercles

point sur une droite
point sur un cercle
point sur un segment

symétrique d'un point par rapport à un point
symétrique d'un point par rapport à une droite

1ère bissectrice de deux droites
2ème bissectrice de deux droites

angle formé par trois points

Annexe B
les structures informatiques de Cabri-géomètre

Nous donnons, in extenso, les structures informatiques mises en place. Le langage de programmation utilisé est le langage C.

Une figure est la structure dans laquelle on trouve notamment les adresses en mémoire des différents objets de la figure:

```
typedef struct
{
	WindowPtr fen;              /* fenêtre associée à la figure */
	int n;                      /* nombre d'objets de la figure */
	objetptr t[NBOBJETS];       /* tableau des pointeurs sur les objets de la figure */
	Boolean petitscarreaux;     /* présence du quadrillage */
	Point origfen;              /* situation de la fenêtre dans la feuille de dessin */
	WindowPtr fentexte;         /* fenetre de la description des objets */
	ListHandle listetexte;      /* liste des descriptions des objets */
	WindowPtr fencalcul;        /* fenêtre des calculs sur la figure */
	ListHandle listecalcul;     /* liste des calculs sur la figure */
	Boolean redef;        /* vrai si la logique de construction n'est plus la logique chronologique */
}FIGURE,*figureptr;
```

Le contenu des structures objet, classe, type a été décrit au §2.2.1:

```
typedef struct objet
{
	int classe;           /* classe de l'objet */
	Boolean masque;       /* variable de travail */
	int coche;            /*        ""        */
	int auxiliaire;       /*        ""        */
	STRUCTNOM lenom;      /* nom et place du nom */
	int trait;            /* épaisseur du trait */
	int genre;            /* NORMAL,IMPLICITE,INTERMEDIAIRE */
	int etat;             /* REEL,IRREEL,DEPENDANT_DIRREL */
	Boolean visible;      /* un objet est visible ou caché */
	VALEUR val[4];        /* valeurs analytiques caracterisant l'objet */
	struct objet *constits[4];        /* pointeurs vers les constituants de l'objet */
	listobjetptr dependance;          /* liste des pointeurs vers les objets dépendants directement
*/
	long ismacronum;      /* référence éventuelle à une macro-consruction */
	int indice;           /* indice dans le tableau des objets */
}OBJET,*objetptr;

typedef struct
{
	ProcPtr procdessin;               /* méthode de tracé de l'objet */
	ProcPtr procdessinpointille;      /* idem en pointillés */
	ProcPtr proccaldist;              /* méthode de calcul de la distance de l'objet au curseur */
	ProcPtr proccalpos;               /* méthode de calcul de la position du nom */
	int nbdecetype;                   /* nombre d'objets de ce type dans la figure */
}TYPE;
```

```
typedef struct
{
      int indnomclasse;        /* référence à une chaine de caractères décrivant la classe */
      int typecl;              /* point,segment,droite,demidroite,triangle ou cercle */
      int Nbconstits;          /* nombre de constituants des objets de cette classe */
      int Tyconstits[4];       /* type de ces constituants */
      ProcPtr proccalval;      /* méthode de calcul des éléments caractéristiques */
      ProcPtr proctexte;       /* méthode de description */
      ProcPtr procaccept;      /* méthode de vérification d'unicité */
      int nbdecetteclasse;     /* nombre d'objets de cette classe */
}CLASSE;
```

Cabri-géomètre manipule de nombreuses listes chainées d'objets, par exemple pour mémoriser tous les objets dépendants d'un même objet:

```
typedef struct enregobjptr
{
      struct objet *obj;
      struct enregobjptr *svt;
}LISTOBJPTR,*listobjetptr;   /* élément des listes chainées d'objets */
```

Les calculs des coordonnées des points se font sur des nombres réels qui sont arrondis avant d'être communiqués aux algorithmes de tracé:

```
typedef struct
{
      double v;
      double h;
}POINTPRECIS;      /* coordonnées en nombre réels d'un point */
```

Les valeurs caractéristiques des différents objets sont des points ou des nombres (rayon d'un cercle par exemple):

```
typedef union
{
      POINTPRECIS p_vpoint;
      double p_vnombre; /* rayon d'un cercle, par exemple */
}VALEUR;
```

Les points, les droites, les cercles peuvent être nommés et la place de ce nom peut être modifiée par l'utilisateur:

```
typedef struct
{
      char nom[7];         /* nom de l'objet :6 caractères au maximum */
      Point laplace;       /* place du nom par rapport à l'objet */
      Point vecteurnom;    /*          ""             */
}STRUCTNOM;
```

Une macro-construction est le sous-graphe correspondant à une suite de constructions élémentaires. La structure contient les données sur le nombre et le type des objets initiaux, et un tableau des objets à construire:

```
typedef struct
{
	Str255 m_nom;			/* nom de l'article de menu correspondant */
	int m_nbobjinitiaux;		/* nombre et type des objets initiaux */
	int m_tyinitiaux[NBMAXINIT];
	int m_nbobj;			/* nombre d'objets finaux ou intermédiaires */
	macroobjptr m_objets[NBMAXFIN];/* tableau des objets de la macro-construction */
	Str255 m_help;			/* phrase d'aide associée à la macro */
	long m_heure;			/* entierlong identifiant la macro */
}MACRO, *macroptr;
```

La structure d'un objet d'une macro-construction contient notamment la classe, le genre de l'objet ainsi que les références à ses constituants dans le tableau de la macro:

```
typedef struct
{
	int m_classe;		/* classe d'un objet d'une macro-construction */
	int m_genre;		/* FINAL ou INTERMEDIAIRE */
	int m_visible;		/* VISIBLE ou CACHE */
	int m_trait;		/* épaisseur du trait */
	int m_constits[4];/* référence aux constituants dans le tableau d'objets de la macro-construction */
}MACROOBJET, *macroobjptr;
```

On peut effectuer des calculs sur une figure, la liste de ces calculs est associée à la figure, la nature du calcul est identifié par l'indice de la fonction à exécuter, les objets impliques à la fois par leur adresse et leur indice dans le tableau d'objets de la figure:

```
typedef struct calcul
{
	int fonc;			/* nature du calcul: aire, longueur, angle,... */
	int indarg[3];			/* références aux objets concernés */
	objetptr objarg[3];		/*		"		*/
	struct calcul *calsvt;
}CALCUL;	/* élément de la liste des calculs associés à une figure */
```

Les fonctions permettent de calculer la distance de deux points, l'aire d'un triangle, la pente d'une droite,...et sont caractérisées par un nombre d'arguments et le type de chacun:

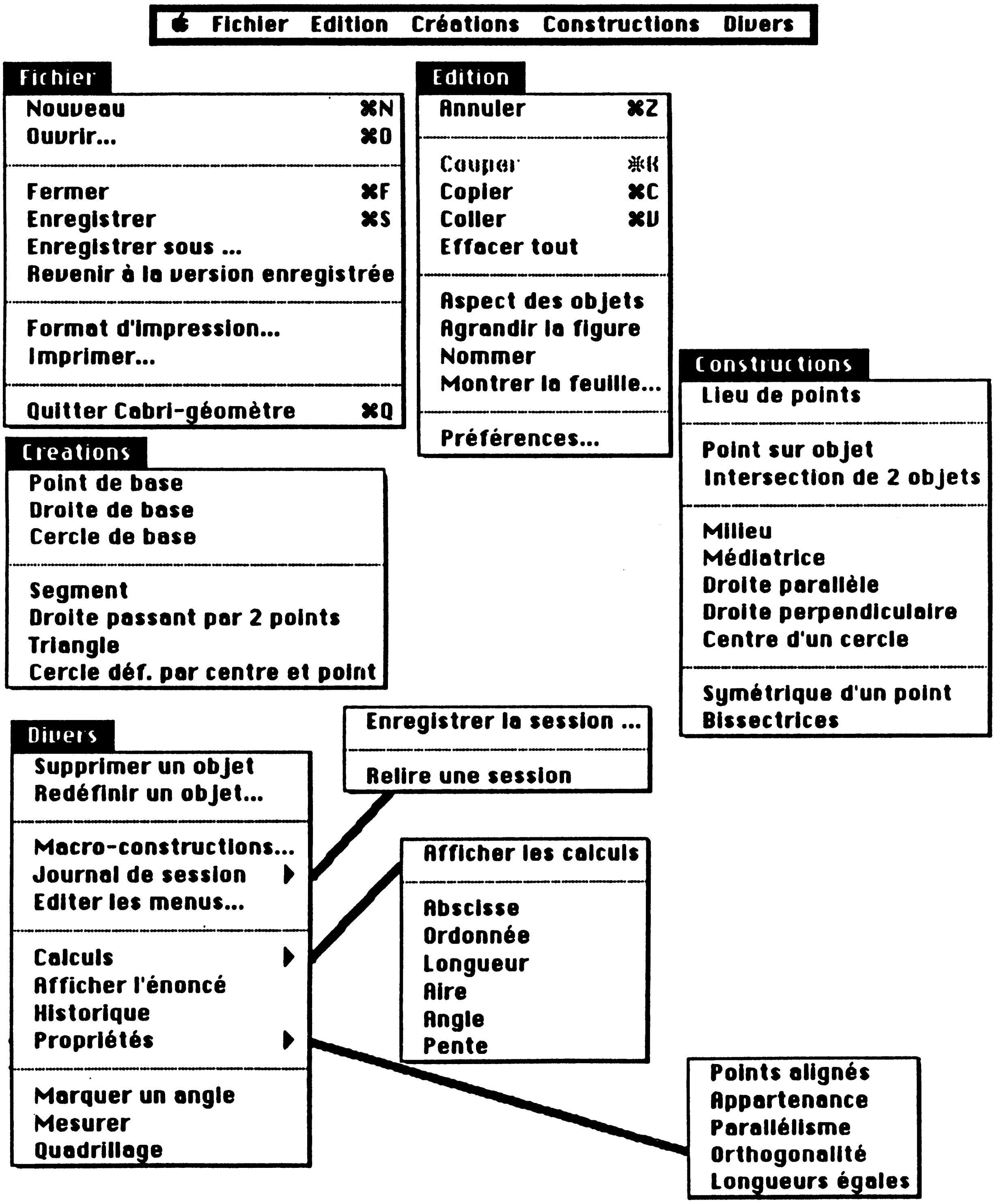
```
typedef struct
{
	int nbarg;			/* nombre et type des objets nécessaires à un calcul */
	int tyarg[4];
	ProcPtr procfonction;		/* méthode de calcul du résultat de la fonction */
}FONCTION;
```

Les propriétés sont également caractérisées par une procédure qui permet de confirmer ou d'infirmer une propriété ainsi que par le nombre et le type des objets impliqués:

```
typedef struct
{
	int Nbconcernes;		/* type et nombre des objets concernés par une propriété */
	int Tyconcernes[4];
	ProcPtr procverifprop;		/* méthode de validation de la propriété */
}PROPRIETE;
```

Annexe C
Les fonctionnalités de Cabri-géomètre

1. menus



2. opérations

Les principales fonctionnalités de Cabri sont ici répertoriées thématiquement et très brièvement commentées.

Constructions de figures

Point, Droite, Cercle

Ces objets, dits quelconques, sont déterminés par la position du ou des points caractéristiques, positions correspondant à l'enfoncement et au relâchement du bouton de la souris.

Segment
Droite passant par 2 points
Triangle
Cercle def. par centre et point

les objets sont obtenus en désignant leurs deux (trois pour le triangle) points caractéristiques. Ces points peuvent être créés dynamiquement au moment de la construction de l'objet.

point sur objet
Intersection de 2 objets

On isole un point sur un objet en désignant cet objet, un point est construit comme la projection de l'endroit indiqué sur l'objet.
L'intersection de deux objets (droites, cercles, segments) doit être explicitée en désignant successivement les deux objets. On obtient deux points si un de ces objets est un cercle.
Ces articles sont amenés à disparaître dans les prochaines versions. La construction d'un point prendra en compte implicitement son appartenance à un ou deux objets.

Milieu
Médiatrice
Droite parallèle
Droite perpendiculaire
Symétrique
Centre d'un cercle
Bissectrices

Ces articles permettent de construire les objets principalement utilisés dans la géométrie scolaire. Ils pourraient tous être le résultat de macro-constructions. Leur présence comme articles propres permet d'améliorer les performances notamment lors de la déformation de la figure.

Après le choix de l'article le système attend la désignation des objets

2. opérations

Les principales fonctionnalités de Cabri sont ici répertoriées thématiquement et très brièvement commentées.

Constructions de figures

Point, Droite, Cercle

Ces objets, dits quelconques, sont déterminés par la position du ou des points caractéristiques, positions correspondant à l'enfoncement et au relâchement du bouton de la souris.

Segment
Droite passant par 2 points
Triangle
Cercle def. par centre et point

les objets sont obtenus en désignant leurs deux (trois pour le triangle) points caractéristiques. Ces points peuvent être créés dynamiquement au moment de la construction de l'objet.

point sur objet
Intersection de 2 objets

On isole un point sur un objet en désignant cet objet, un point est construit comme la projection de l'endroit indiqué sur l'objet.
L'intersection de deux objets (droites, cercles, segments) doit être explicitée en désignant successivement les deux objets. On obtient deux points si un de ces objets est un cercle.
Ces articles sont amenés à disparaître dans les prochaines versions. La construction d'un point prendra en compte implicitement son appartenance à un ou deux objets.

Milieu
Médiatrice
Droite parallèle
Droite perpendiculaire
Symétrique
Centre d'un cercle
Bissectrices

Ces articles permettent de construire les objets principalement utilisés dans la géométrie scolaire. Ils pourraient tous être le résultat de macro-constructions. Leur présence comme articles propres permet d'améliorer les performances notamment lors de la déformation de la figure.

Après le choix de l'article le système attend la désignation des objets

Entrées-sorties

Nouveau/Ouvrir/Fermer
Il est possible d'avoir simultanément six figures sur le bureau, des nouvelles figures ou des figures préalablement enregistrées.

Imprimer/Format d'impression
permet de configurer l'imprimante et de lancer l'impression en choisissant la portion de la figure à imprimer.

Journal de session
permet d'enregistrer puis de relire toutes les étapes d'une session de travail sous la forme de fichiers contenant les figures successivement obtenues.

Exploration

Déplacement des objets de base
les points, droites et cercles quelconques peuvent être déplacés en les tirant avec la souris, les autres objets sont redessinés en conservant les relations géométriques. Cette fonctionnalité est disponible à tout moment sans demander de choix dans la barre des menus.

Mesurer
permet de mesurer des segments ou des angles (un angle doit avoir été préalablement marqué -article **Marquer un angle**). La mesure apparaît à coté de l'objet et peut être déplacée pour éclaicir la figure.

Propriétés
permet de valider dans le cas général l'alignement de trois points, l'appartenance d'un point à une droite (ou un cercle ou un segment), le parallélisme ou l'orthogonalité de deux directions, la congruence de deux segments.

Calculs
permet d'effectuer certains calculs simples, aires, longueurs, et d'afficher les coordonnées des points de la figure. Les résultats apparaissent dans une autre fenêtre.

Historique
permet de refaire pas à pas (clic à clic!) la construction en visualisant en pointillés les objets intermédiaires des macro-constructions.

Afficher l'énoncé

Affiche dans une fenêtre l'énoncé de la figure sous la forme d'une liste des objets construits.

Lieux de points

on peut tracer le lieu géométrique d'un ou de plusieurs points lorsque un objet de base de la figure se déplace. Ce lieu n'est pas un objet de Cabri-géomètre mais l'ensemble lissé des traces des points désignés pour les valeurs successives de l'objet déplacé. Cette trace devra être effacée avant toute autre opération.

Configuration

Editer les menus

permet de créer, d'enregistrer ou de charger une barre des menus personnalisée ne contenant que les articles nécessaires.

Préférences

permet de choisir un certain nombre d'options de fonctionnement, la langue de travail, la police de caractères des noms, la précision des mesures, le dessin du repère associé à la figure, ...