HAL
open science

Vers une approche déclarative pour les logiciels de dessins géométriques

Denis Bouhineau

To cite this version:

Denis Bouhineau. Vers une approche déclarative pour les logiciels de dessins géométriques. 4e journée EIAO de l'ENS Cachan, 1995, Cachan, France. pp.55-66. hal-00190299

HAL Id: hal-00190299

https://telearn.hal.science/hal-00190299v1

Submitted on 23 Nov 2007

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

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

Vers une approche déclarative pour les logiciels de dessins géométriques

Denis BOUHINEAU
IMAG-LGI Bureau D 300I
BP 53x
38041 Grenoble Cedex

RÉSUME : Cet article est consacré à la conception d'un environnement d'apprentissage de la géométrie orienté vers la manipulation directe de figures géométriques. À travers deux exemples d'activités, proposées en classe, on distingue les notions de construction géométriques, de spécification, et de manipulation directe. Dans le cadre ainsi défini, nous présentons notre approche. Elle consiste à intégrer des mécanismes et la méthodologie de l'intelligence artificielle dans un logiciel de spécification de figures géométriques au niveau de l'utilisateur et du programmeur Cette approche repose sur une démarche déclarative pour l'acquisition de la spécification de la figure et se distingue donc des environnements procéduraux et constructifs. Nous montrons son intérêt dans divers domaines : la recherche de constructions, la découverte et la vérification de propriétés.

1. Introduction

Depuis quelques années avec l'apparition des logiciels Cabri [Baulac et al 92] et Geometer Sketchpad destinés à la construction et la manipulation directe en géométrie plane, l'enseignement de la géométrie évolue. Il semble même qu'une nouvelle branche de la géométrie apparaisse : la géométrie des animations ("dynamic geometry" en anglais). Elle a pour origine l'animation des figures rigides que l'on peut dessiner sur le papier. Au-delà de la capacité d'animation introduite, ce qui la différencie de la géométrie des problèmes de construction, de la géométrie des démonstrations, et de la géométrie des transformations du plan, c'est la manière de considérer la notion de propriétés pour une figure, et l'importance accordée aux lieux de points.

Plus précisément, la géométrie des animations s'intéresse à l'observation des invariants et des changements au niveau d'une figure géométrique quand on modifie progressivement la position d'un objet intervenant dans celle-ci, tout en conservant les propriétés de définition de la figure. On parle d'animation de la figure ou de manipulation directe. Par exemple, Cabri propose à chaque instant de la construction

Environnements Interactifs d'Apprentissage avec Ordinateur, Eyrolles, Paris, 1995.

d'une figure, la manipulation directe des objets de base de cette figure, et réaffiche en temps réel un dessin renouvelant la construction de l'utilisateur ayant défini la figure, à partir des nouvelles positions des objets initiaux (ou objets de base). On observe ainsi comment la figure se déforme tout en respectant les propriétés qui l'ont définie.

Les notions de propriétés et de lieu de points sont définies en fonction des animations. Une figure possède telle propriété, si cette propriété reste vérifiée à chaque instant, dans le cadre de la géométrie statique, lors de toutes déformations de la figure. Par exemple dans Cabri, les constructions définissent des propriétés d'une figure, puisque la reconstruction de la figure en temps réel s'effectue en respectant ces constructions. On retrouve la définition suivante: "une propriété est un invariant d'une classe de figures", où le mot "invariant" prend une connotation expérimentale, et où le terme "classe de figure" désigne l'ensemble des figures obtenues par animation. Le lieu d'un point d'une figure, est l'ensemble des positions prises par ce point au cours d'une déformation de la figure. Par exemple, le lieu du centre de gravité **G**, d'un triangle **(ABC)**, lorsque **A** décrit une droite **d**, **B** et **C** restant fixe, est une droite **d'** parallèle à **d**. C'est une des conséquences du théorème de Thalès qui s'applique, car **G** est au *2/3* de la médiane passant par **A**.

La géométrie des animations est tributaire des logiciels qui ont participé à sa naissance. Deux approches sont possibles pour l'acquisition de la figure et son animation. La première approche est constructive, Cabri en est un exemple. Elle demande à l'utilisateur une construction de la figure à partir d'objet donnés que l'on nomme objet de base. Elle s'apparente aux langages de programmation procéduraux. La classe des animations possibles correspond aux animations des points de base de la figure. La seconde approche est déclarative. Elle demande à l'utilisateur une spécification logique de la figure et indépendamment, un ensemble d'objets de base. Elle s'appuie sur des algorithmes de construction automatique pour transformer la spécification en figure. La classe des animations envisagée est beaucoup plus large, puisqu'elle s'appuie sur un ensemble d'objets de base que l'on peut changer sans modifier la spécification logique de la figure. Elle dépend explicitement des capacités de construction automatique du logiciel. C'est le problème qui nous intéresse.

Cet article se découpe en deux parties. La première, constituée de deux exemples, met en évidence les différentes notions introduites précédemment. Les deux approches, constructive et déclarative, sont comparées. Ces deux exemples ont été choisis parmi des activités effectuées en classe, au collège et au lycée, avec ou sans ordinateur. Dans une seconde partie, les problèmes rencontrés dans l'élaboration d'un système favorisant la géométrie des animations sont mis à jour. Cette partie commence par donner une réponse à la problématique suivante : "quel cadre choisir pour obtenir des figures sur lesquelles on puisse effectuer un nombre maximum d'animations ?". Notre choix s'est porté sur une approche déclarative de la géométrie et sur une implantation avec un langage de haut niveau intégrant les paradigmes de la programmation déclarative et de la programmation par contraintes (Prolog III [Colmerauer 90]). L'algorithme de construction automatique, nécessaire pour notre approche, est explicité ensuite. Les apports engendrés par ce nouvel outil sont discutés à la fin de cette partie, ainsi que la recherche d'une notion de complétude pour notre algorithme de construction automatique.

2. Exemples d'activités de géométrie

Deux exemples d'activité de géométrie sont décrits ici. Le premier exemple concerne un problème de construction, le second la recherche d'une démonstration

pour un théorème donné. L'apport des animations est clairement exprimé, les deux approches constructive et déclarative sont mises en concurrence.

2.1. Autours d'un problème de construction.

Commençons par observer une activité de résolution pour un problème de construction. Considérons l'énoncé suivant tiré de [Capponi 94] et proposé comme activité de construction de figures contraintes géométriquement sur Cabri, à des élèves d'une classe de troisième :

> **Étant donnés deux droites d1, d2 et un point m, construire les points a et b, tels que a ∈ d1, b ∈ d et tel que m soit le milieu de [a,b]**

Une méthode classique pour résoudre les problèmes de construction consiste à utiliser la méthode des lieux, i.e. chercher le lieu de **b** quand **a** bouge dans l'énoncé moins contraint :

> **Étant donnés deux droites d1, d2 et un point m, construire a et b, tels que a ∈ d1 et m milieu de [a,b]**

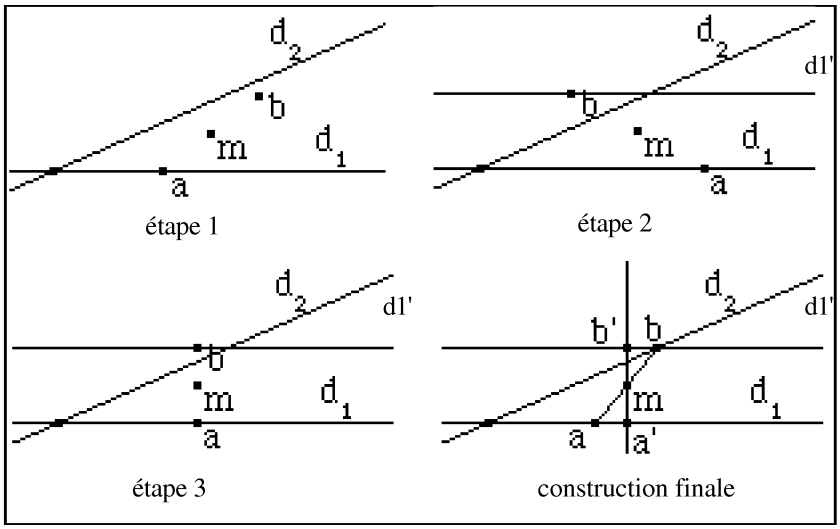

Figure 1 *dessin d'un segment s'appuyant sur deux droites*

Première étape : On résout ce problème facilement (cf figure 1), en prenant pour **a** un point quelconque de **(d_1)** et en construisant **b** le symétrique de **a** par rapport à **m**.

Deuxième étape : En prenant d'autres positions pour le point **a** sur **(d_1)**, c'est à dire en animant **a**, on observe que le lieu du point **b** est une droite. Comme on peut le constater la géométrie des animations s'est introduite dans le problème naturellement. En dessinant le lieu de **b**, on observe de plus que la droite décrite par **b** est parallèle à **(d_1)** et qu'elle coupe **(d_2)** en un point solution. Pour résoudre notre problème de construction, il faut donc trouver comment construire cette droite, que l'on va nommer **(d_1')**.

Troisième étape : Déplaçons **b** sur **(d_1')**. Comment faire cela ? Dans une optique constructive, cette animation nécessite une nouvelle construction car **b** est 'construit'. Dans une optique déclarative, on libère **a**, et anime **b**. Amenons **b** proche de **m**, on observe **a** se déplacer sur **(d_1)** et venir à l'endroit de **(d_1)** le plus proche de **m**. On en conclut que **(d_1)** et **(d_1')** sont à égale distance de **m**, ce qui permet de construire définitivement la figure demandée. Au passage, on redécouvre le résultat suivant de la géométrie des transformations du plan :"l'image d'une droite **d** par une symétrie de centre **m** est une droite parallèle à **d** et à égale distance de **m**"

Construction finale : En partant donc du point **m**, on construit le point **a'** projeté de **m** sur **(d_1)**. Puis le point **b'** symétrique de **a'** par rapport à **m**. On peut alors construire **(d_1')**, parallèle à **(d_1)** passant par **m**. Enfin on prend **b** intersection de **(d_1')** et de **(d_2)** et on construit **a** symétrique de **b** par rapport à **m**.

La comparaison des deux approches dans le cas d'activités de construction, ou d'énoncé d'exercice non-constructif, est en faveur de l'approche déclarative. Outre le fait que la classe d'animation est plus riche, l'utilisation d'un algorithme de construction automatique dans cette approche permet d'obtenir plus rapidement un dessin solution du problème que l'utilisateur peut explorer. Ici par exemple, la spécification initiale permet d'obtenir un dessin où l'on observe que **b** est fixe pour toute une classe de points **m** décrivant une parallèle à **(d_1)**.

2.2. Autours de la démonstration d'un théorème

Considérons l'exercice portant sur la démonstration de la propriété suivante :

> **Étant donnés un triangle (ABC) isocèle en B, et D l'intersection des parallèles à (BC) et (AB) passant par A et C respectivement, montrer que le triangle (ACD) est isocèle en D.**

Le schéma de résolution de ce type d'exercices a été imaginé par [Allen et al. 90] pour le système Mentoniezh (voir aussi [Py 90]), il peut comporter quatre phases :

• acquisition de la figure : l'élève construit une figure conforme aux hypothèses du théorème à démontrer.

• appropriation de la figure : l'élève fait évoluer la figure pour découvrir des invariants ou des lieux de points intéressants.

• exploration des propriétés : l'élève examine d'éventuelles propriétés suggérées par le système.

• organisation de la démonstration : enfin l'élève construit une démonstration visant à prouver le théorème recherché.

L'animation d'une figure représente donc la deuxième phase du processus imaginé pour Mentoniezh. Imaginons que l'élève confronté au problème décrit au début de ce paragraphe obtienne une figure correspondant aux hypothèses de son énoncé. L'élève se trouve donc en face d'un premier dessin qui peut être selon le triangle de départ, le dessin 1, 2 ou 3 de la figure 2. Ces dessins sont riches d'enseignements; les figures que l'on y voit sont différentes. Déformons la figure. Dans une approche constructive la classe des animations est contrainte par la construction du triangle **(ABC)**, a priori on peut déplacer certains points parmi **A, B** et **C** selon des modalités qui dépendent de la construction et du logiciel de dessin. Dans une approche déclarative

on peut manipuler les quatre points plus librement. La différence entre les deux approches parait minime mais elle est significative car **D** appartient à la partie de la figure sur laquelle porte la conclusion du théorème. La manipulation conduit au second dessin, le plus général, qui représente un losange. En fait, plus précisément, en manipulant la figure, sans forcément atteindre le deuxième dessin, on se rend compte que la figure est un losange et seulement un losange, et non pas un carré, conclusion que l'on aurait pu tirer si l'on avait seulement le dernier dessin.

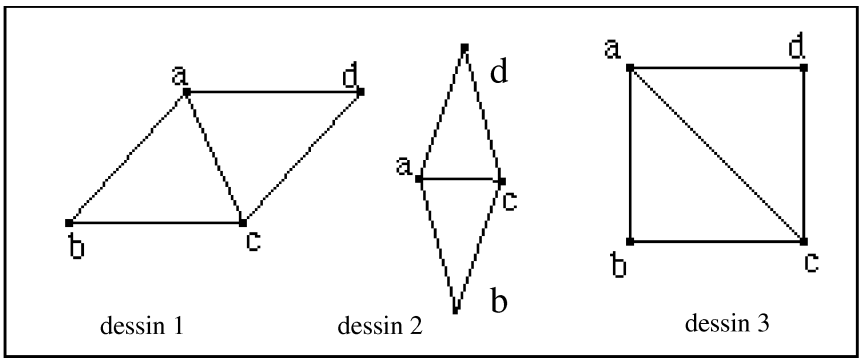

Figure 2 *trois losanges obtenus à partir de triangles isocèles*

En fait, on trouve le résultat suivant : l'animation de figures géométriques permet la compréhension et l'appropriation de la notion de classes de figures géométriques (comme la classe des rectangles ou celle des losanges), et des liens d'inclusion entre les différentes classes de figures (comme l'inclusion de la classe des losanges dans celle des parallélogrammes). Par ailleurs, l'animation d'une figure, obtenue pendant la résolution d'un problème, permet de découvrir plus justement son appartenance à la bonne classe de figures pointées par l'énoncé de l'exercice.

3. Mise en œuvre d'un environnement de dessin et de manipulation pour la géométrie des animations

L'écriture d'un logiciel prototype, GéoSpécif, qui insiste sur la manipulation directe des figures géométriques nous a permis de mieux comprendre les exigences imposées par un tel projet, en particulier l'importance de l'algorithme de construction automatique. Nous présentons ici, le résultat de notre travail. Nous montrons d'abord que la géométrie des animations nécessite une approche déclarative. Nous décrivons ensuite l'algorithme de construction automatique utilisé. Enfin nous terminons par quelques perspectives de travail dans les domaines de la vérification des surcontraintes et dans la difficile recherche d'algorithmes de construction automatique complets pour une classe de figures compréhensible par l'utilisateur.

3.1. Géométrie des animations et géométrie déclarative

Les logiciels comme Cabri ou Geometer Sketchpad ont permis de découvrir toute la puissance des animations. Les activités présentées dans la première partie ne sont que partiellement réalisables avec ces logiciels. En effet, ces environnements restreignent la classe des animations possibles aux animations des points de base d'une figure. Cela vient du mode opératoire choisi pour obtenir une figure. Ces logiciels construisent une figure progressivement à partir des objets de base, et

d'opérations géométriques introduisant de nouveaux objets dit 'construits'. Lors de la manipulation des objets de base la figure se reconstruit en temps réel en suivant le processus de construction donné initialement. Les objets 'construits' ne peuvent pas être manipulés de la même manière, car ils sont les résultats d'opérations de constructions qui ne sont pas inversibles. Cette difficulté est connue depuis le XIXe siècle et les travaux de Galois puis de Wantzel sur la recherche de moyen pour reconnaître si un problème de géométrie peut se résoudre par la règle et le compas. Elle est reprise dans [Lebesgue 1989] qui montre que certains problèmes ne peuvent être résolus à la règle et au compas quand bien même il suffirait d'inverser un modèle comme dans le cas de la trisection de l'angle. Cette limitation théorique s'applique au domaine de la construction automatique : Il n'y a pas de méthode générale pour résoudre tous les problèmes de géométrie à la règle et au compas. Si l'on ajoute que les algorithmes les plus performants dans le domaine sont de complexité exponentielle, on comprendra que notre objectif est d'offrir un algorithme pratique de construction aussi puissant que possible, et qu'il constitue l'un des points cruciaux de notre approche.

Mais avant de considérer la notion de construction automatique, attardons-nous sur un choix essentiel. Pour favoriser la classe des animations possibles, il faut reconsidérer les priorités attribuées à la construction et à la spécification. On passe ainsi de la programmation géométrique procédurale, à la programmation géométrique déclarative. Dans l'approche constructive, la construction constitue un élément indispensable : on ne peut ajouter une spécification à une figure, si elle ne correspond à une nouvelle construction. De fait, l'importance accordée à la construction divise les objets en deux classes figées, la classe des objets de base et la classe des objets construits, et les animations sont déterminées. Dans l'approche déclarative, la spécification est privilégiée par rapport à la construction. Par exemple dans GéoSpécif les constructions se font automatiquement, lorsque cela est possible, à partir des objets donnés par l'utilisateur et des spécifications qui lient ces objets. À la différence de l'approche constructive, le statut des objets géométriques n'est pas figé. Un point n'a plus un statut figé de point de base ou de point construit, mais un statut que l'utilisateur peut faire évoluer au cours de la session selon la figure 3.

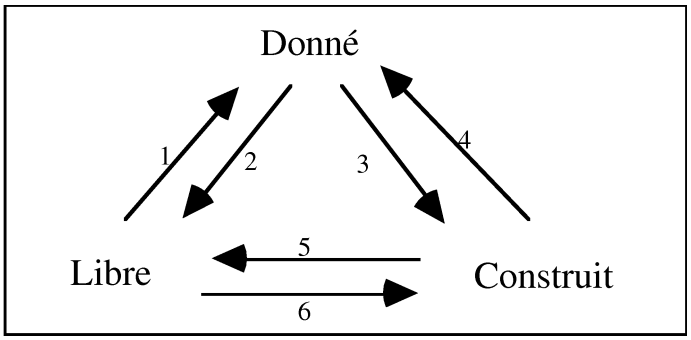

Figure 3 *statut d'un objet dans une approche déclarative*

Les points donnés et construits sont ceux qui apparaissent à l'écran à l'emplacement désigné par l'utilisateur ou calculé par le logiciel pour respecter la spécification. Les points libres apparaissent dans la spécification de la figures mais ne sont pas affichés parce qu'ils sont insuffisamment contraints ou qu'ils restent

inconnus. Les flèches 1 et 4 correspondent à des instanciations, 2 et 3 à des libérations, ces quatre flèches correspondent à des actions volontaires de l'utilisateur relative à l'animation des objets et non à la spécification géométrique de la figure. Les flèches 5 et 6 correspondent aux constructions automatiques ou aux défauts de construction. Elles ne sont pas directement commandées par l'utilisateur, elles le sont soit par un changement dans la spécification de la figure, soit par l'utilisation de l'algorithme de construction automatique.

3.2. Description de l'algorithme de construction automatique

Pour valider notre approche déclarative, nous nous appuyons sur un algorithme de construction automatique qui prend en compte la spécification logique de la figure de manière non déterministe (au sens de Prolog) et incrémentale. L'algorithme utilisé opère en deux temps. Tout d'abord il complète automatiquement la spécification avec des contraintes géométriques vérifiées par la figure. Ensuite il effectue une traduction algébrique du problème et tente de résoudre le système d'équations algébriques obtenu. Pour améliorer les performances de cet algorithme, l'utilisateur est mis à contribution pour enrichir la spécification, avec l'aide de l'ordinateur, au cours d'une exploration de la figure organisée par GéoSpécif.

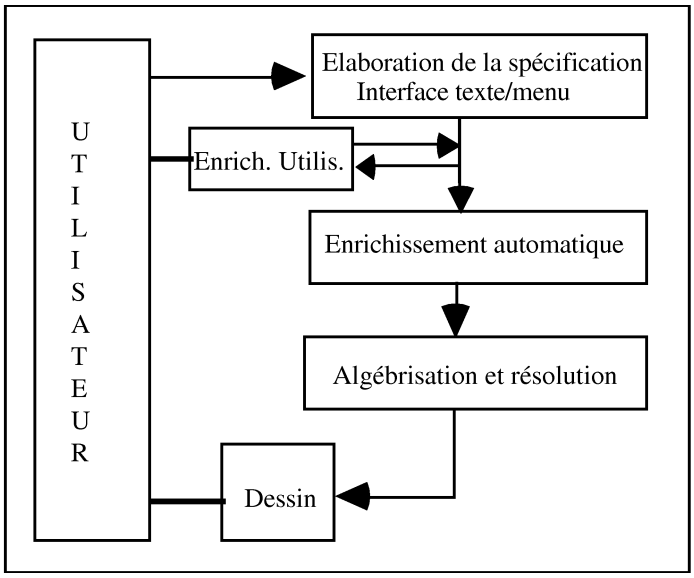

Figure 4 *architecture de GéoSpécif et construction automatique*

3.2.1. Enrichissement automatique

L'idée de départ consiste à ajouter des objets intermédiaires à la spécification qui peuvent servir pour la construction, lorsque celle-ci comporte des configurations prédéfinies. On peut voir l'ajout de ces objets intermédiaires comme si l'on complétait la spécification avec des informations redondantes.

À l'heure actuelle, l'algorithme d'enrichissement automatique cherche les cordes communes des cercles où se trouvent des points pour ajouter la spécification

supplémentaire "linéaire" suivante : les points appartenant à deux cercles, sont sur la corde commune de ces deux cercle. Cet enrichissement a été choisi car il favorise la résolution du problème de construction en ajoutant une équation linéaire. En effet l'algorithme de construction automatique repose essentiellement sur l'algorithme du simplexe qui traite les équations linéaires. Cette approche ressemble aux travaux de [Schreck 94] qui assure la progression de la construction automatique dans Progé par l'application de règle de productions.

3.2.2. Résolution algébrique

Prolog III résout les systèmes d'équations et d'inéquations linéaires. Aussi chaque contrainte géométrique est transformée en une équation algébrique pseudo-linéaire (pour les contraintes où interviennent des droites) ou quadratique (pour les contraintes où interviennent des cercles), et lorsqu'une équation devient linéaire, elle est transmise à Prolog. Plus précisément, chaque équation relève d'un processus appartenant à l'une des deux classes suivantes, pseudo-linéaire ou quadratique, qui est activée au cours de la résolution du système d'équations algébrique. L'activation des processus donne le résultat suivant (l'activation des deux classes est décrite plus loin):

- si l'équation est linéaire : elle est introduite dans le résolveur linéaire de Prolog III et le processus s'arrête.
- si l'équation porte sur une seule variable et que c'est une équation du second degré : une procédure annexe la résout. et le processus s'arrête.
- si toutes les inconnues de l'équation ont été déterminées : l'équation n'apporte pas d'information, c'est une surcontrainte que le système essaie de vérifier (voir chapitre ci dessous), et le processus s'arrête.
- Sinon, l'équation est laissée en attente de résolution.

L'activation des deux classes de processus s'effectue de la manière suivante : La première classe de processus pseudo-linéaires est activée tant que des processus atteignent le premier cas. Puis la classe de processus quadratiques est activée. Si un processus atteint le second cas, on reprend l'activation des processus pseudo-linéaires. L'activation s'arrête quand toutes les variables sont connues, ou qu'aucune activation ne mène aux premier ou second cas. On peut vérifier que la priorité est donnée à la résolution des équations linéaires.

3.2.3. Enrichissement guidé par l'utilisateur

Il est proposé à l'utilisateur une exploration automatique de sa figure pour y déceler des propriétés supplémentaires réalisées, voir [Rouchon 94]. On opère de la manière suivante : on demande à l'utilisateur de construire sa figure à partir d'autres ensembles de points de base, et on recherche sur la figure construite des propriétés supplémentaires réalisées (pour l'instant, seulement les incidences). Ces propriétés sont alors proposées à la sagacité de l'utilisateur pour qu'il puisse les intégrer ou non à la spécification de sa figure. Par exemple dans la construction du centre de gravité **G** d'un triangle comme point d'intersection de deux médianes, l'exploration de la figure demande s'il faut ajouter la propriété d'appartenance de **G** à la troisième médiane.

3.2.4. Résultats

La combinaison de ces méthodes de construction automatique permet d'avoir déjà des résultats acceptables. Par exemple les figures 1 et 2 sont obtenues directement à partir de leur spécification, en respectivement 0,1 seconde et moins de 0,1 seconde. Ces performances permettent une animation en temps réel. Pour la figure 1, par exemple, la spécification introduite est: **A ∈ d1, B ∈ d2, P milieu de [A,B]**

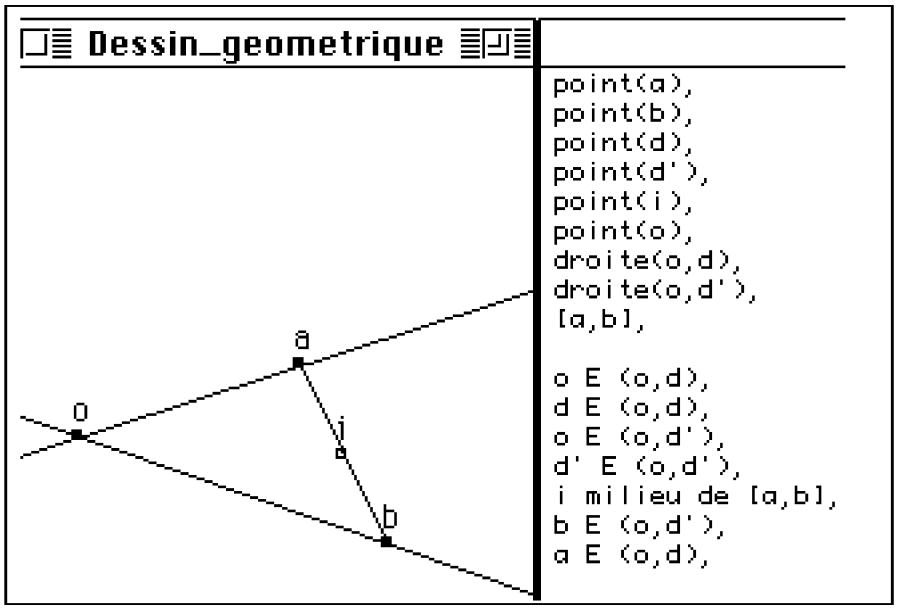

Figure 5 *Copie d'écran de GéoSpécif relative au premier exemple.*

3.3. Prise en considération des surcontraintes géométriques, et classes de problèmes traités

3.3.1. Les surcontraintes géométriques

Les deux processus de complétion d'une figure décrits plus hauts et l'utilisateur peuvent naturellement introduire des contraintes redondantes dans la spécification. Lorsque toutes les inconnues d'une contrainte ont été éliminées, on obtient une surcontrainte. La première solution pour gérer ces surcontraintes consiste à les ignorer. C'est ce qui se passe pendant l'animation des figures géométriques pour accélérer cette animation. Pendant la construction initiale, les surcontraintes linéaires sont vérifiées en utilisant une arithmétique en précision infinie sur les rationnels. En effet le résolveur d'équations linéaires de Prolog III effectue ses calculs en précision infinie sur les rationnels et permet la vérification des surcontraintes linéaires. Notons que ce calcul permet d'avoir une arithmétique correcte pour la géométrie à la règle et à l'équerre. Il reste donc à vérifier les surcontraintes du second degré qui caractérisent la géométrie au compas. Des travaux sont effectués actuellement pour résoudre ce problème difficile. En effet, il présente la même difficulté que la démonstration automatique de théorèmes.

Deux voies sont envisagées. Le calcul sur intervalle, qui permet d'avoir une arithmétique exacte sur les intervalles et permet d'éviter les erreurs numériques. Cependant il faut noter que l'arithmétique sur les intervalles ne permet pas de prouver une surcontrainte, mais seulement sa négation. Une deuxième voie s'inspire des travaux de [Guillot et al 91] en généralisant la méthode proposée à n'importe quel ordre. Il s'agit de calculer non plus dans R, mais dans des extensions successives de Q. En effet les nombres intervenant en géométrie à la règle et au compas appartiennent à des extensions algébriques de Q dont l'ordre est une puissance de 2, ce sont les nombres constructibles. Les nombres constructibles s'introduisent avec les intersections entre cercles et droites. L'idée consiste à partir de K=Q pour décrire les objets de la figure. À chaque fois qu'une racine carré est extraite lors d'une intersection avec un cercle, on vérifie que cette racine n'appartient pas à l'extension courante K de Q, auquel cas on continue les calculs dans K[x], où x représente l'élément de K dont on a extrait la racine. Si la racine de x appartient à K on continue les calculs dans K. Le nombre d'extension de Q dépend du nombre d'intersections de cercles indépendantes. La représentation des nombres constructibles, selon n bases successives des n extensions de Q données par les racines des x introduites, nécessite 2^n rationnels. Les additions s'effectuent en temps linéaire en fonction de la longueur $l=2^n$ des nombres, les multiplication en $l^{5\ln(2)}$. Un premier essai sur la construction du pentagone régulier permet d'espérer que la complexité algorithmique exponentielle, limitée dans la classe des problèmes qui concernent l'enseignement, n'empêche pas l'emploi de ces algorithmes.

Les possibilités offertes par un vérificateur de surcontraintes sont nombreuses. Elle ne permettent pas seulement d'éliminer les spécifications incohérentes :

• L'adéquation entre une spécification et une construction peut être établie dans une démarche semblable à [Desmoulins 94]. La spécification serait donnée par un professeur et constituerait l'énoncé à modéliser, la construction serait donnée par l'élève à la règle et au compas.

• On peut aussi penser à un autre mode d'acquisition de spécification, par rigidifications successives. Le système prendrait alors en compte les nouvelles contraintes de l'utilisateur, différemment selon qu'il s'agirait de surcontraintes, ou de nouvelles contraintes à ajouter à la figure.

3.3.2. Classe de problème traitée

Notre travail se développe également en vue de déterminer un algorithme de construction complet pour une classe de figures compréhensible par l'utilisateur. C'est un sujet difficile et semé d'embûches. Par exemple, il est incorrect de dire que la résolution de systèmes linéaires correspond aux constructions à la règle seulement. Le problème de Steiner, qui suit, constitue un exemple de problème soluble à la règle seule mais non par la résolution d'un système d'équations linéaires :

> **Étant données trois droites concourantes et trois points du plan, dessiner un triangle s'appuyant sur ces trois droites et passant par les trois points**

Pour l'instant nous nous orientons vers la classe suivante : celle des constructions de figures dont l'ensemble des objets nécessaires appartient à la spécification.

4. Conclusion

Notre démarche se situe au croisement de deux mouvements appartenant à deux domaines relativement éloignés. Chronologiquement, le premier mouvement s'est développé avec Prolog, il s'agit de la programmation déclarative et non déterministe, il se prolonge aujourd'hui avec l'apport de la programmation par contraintes. Le second mouvement, beaucoup plus récent, a été engendré par Cabri, il s'agit de la géométrie des animations et il se développe dans le domaine de l'éducation avec un grand succès. La réalisation de Géospécif, notre logiciel de géométrie déclarative tourné vers l'animation, et son utilisation montrent le pouvoir d'expression et la facilité d'utilisation des paradigmes mis en jeu. En quelque sorte il participe à la validation de ces deux mouvements. Les problèmes scientifiques importants à résoudre ressortent de la vérification des surcontraintes et de l'obtention d'un algorithme complet pour une classe de problèmes définissable géométriquement. Sur le plan des performances, des optimisations basées sur une compilation de la construction sont en cours. La réalisation prochaine d'une maquette robuste et conviviale devrait permettre de valider ces résultats auprès d'utilisateurs finals. Et on peut imaginer, dans une perspective plus lointaine, l'extension du langage de spécification pour en faire un réel langage de programmation géométrique et logique.

Bibliographie

Allen, R., Nicolas, P., Trilling, L., [1990] *Figure Correctness in an Expert System for Teaching Geometry*, Proceedings of the eight biennial conference of the Canadian society for computer studies of intelligence, Ottawa, pp 154-160

Baulac, Y., Bellemain, F., Laborde, J.M., [1992] *Cabri, the interactive geometry notebook*, Brooks/Cole publishing company, Pacific grove, CA.

Capponi, B., [1994] *Constructions, contraintes*, communication personnelle.

Colmerauer, A., [1990] *An introduction to Prolog III*, Communication of the ACM 33(7):69--90.

Desmoulin, C., [1994] *Etude et réalisation d'un système tuteur pour la construction de figures géométriques*, thèse de l'Université Joseph Fourier, Grenoble.

Geometer's Sketchpad, Key Curriculum Press, Berkeley CA.

Guillot, P., Jolivet, L., Leperd, J.L., Moulin C., [1991] *Démonstration automatique en géométrie*, Actes de l'Université d'été Informatique et Enseignement de la géométrie, Irem de Toulouse editeur.

Lebesgue, H., [1989] Leçons sur les constructions géométriques, réédition, Gauthier-Villard, Paris.

Py, D., [1990] *Mentoniezh, a geometry I.T.S. for drawing and prof setting* Journal of Artificial Intelligence in Education, Vol 1, No 3.

Rouchon, P., [1994] *Etude et réalisation d'un vérificateur de contre-propriétés en programmation logique avec contraintes sur les intervalles*. Rapport de stage de maîtrise, Université Joseph Fourier.

Schreck, P., [1992] *Automatisation des construction géométriques à la règle et au compas*, Thèse de l'Université Louis Pasteur, Strasbourg.