Analyse et prise en compte des contraintes didactiques et informatiques dans la conception et le développement du micromonde de preuve Cabri-Euclide

Vanda Luengo

Analyse et prise en compte des contraintes didactiques et informatiques dans la conception et le développement du micromonde de preuve Cabri-Euclide

V. Luengo

Laboratoire Leibniz - IMAG
46 avenue Félix Viallet
38031 Grenoble Cedex
Vanda.Luengo@imag.fr

RÉSUMÉ. Nous présentons dans cet article les contraintes didactiques et informatiques dans le développement d'un logiciel de preuve en géométrie : Cabri-Euclide. La contrainte de départ est de produire un logiciel de construction de la preuve sans avoir des solutions, à propos du problème, préalablement construites. Cabri-Euclide est conçu comme un micromonde, l'essentiel est ainsi dans le fait que la tâche dévolue au système est de vérifier une cohérence locale, plutôt que d'affecter des déductions automatiques globales.

ABSTRACT. This paper shows the didactical and computational constraints in the design of software for mathematical proof in geometry: Cabri-Euclide. The earliest constraint is to produce software for mathematical proof without has the previous solutions concerning the problem. Cabri-Euclide is a microworld, Thus, the essential is that the system have to verify a local coherence, and not to designate a global automatic deduction.

MOTS CLES : preuve, réfutation, résolution de problèmes, contraintes didactiques, contraintes informatiques, Cabri Géomètre, géométrie.
KEY WORDS: mathematical proof, refutation, problem resolution, didactical constraints, computational constraints, Cabri Géomètre, geometry.

1 Introduction : pourquoi un logiciel de preuve centré sur la négociation plutôt que sur un ensemble de solutions produites automatiquement.

Nos travaux précédents ([LUE 93], [LUE 98]) ont montré les contraintes liées au principe que la machine détermine a priori les solutions satisfaisantes vis-à-vis du travail de l'utilisateur dans une situation de résolution de problèmes. Ces contraintes font référence à des problèmes de compréhension, par la machine, de la tâche de l'élève et par conséquent des difficultés sur la communication entre l'élève et la machine. Ainsi, par exemple, dans les cas des logiciels d'aide à la preuve que nous avions étudiés ([LUE 98]), nous avions remarqué les problèmes d'incompréhension de la part de la machine quand l'élève introduit un objet qui ne fait pas partie de l'énoncé du problème et des solutions envisagées par la machine.

Nous nous sommes donc posé la question de la possibilité d'un logiciel qui ne soit pas centré sur des solutions construites a priori, mais dont la tâche soit centrée sur le suivi du travail de l'élève et que, dans le cas de contradictions au cours de la résolution, le logiciel puisse prendre en charge une négociation de façon à faire évoluer la connaissance.

A partir de cette idée, nous avons décidé de concevoir et développer un logiciel né d'un ensemble des contraintes basées sur une analyse didactique et informatique. Notre domaine est celui de la preuve en géométrie. Dans cet article, nous analyserons certaines de ces contraintes en montrant les conséquences de ces choix tout au long de notre travail.

2 Les contraintes

La proposition fondamentale, source de notre travail, est que l'interaction entre la machine et l'élève soit centrée sur la cohérence du travail de l'élève et non sur des choix de la machine par rapport à l'objet d'apprentissage (modèle du domaine) et/ou des choix relatifs à l'élève (modèle de l'apprenant). Cette proposition vient de l'idée d'enlever les contraintes relatives à des choix préalables, qui sont faits automatiquement par la machine, par rapport à la connaissance qu'elle manipule et qui, comme nous avons montré ([LUE 98]), vont perturber l'interaction entre l'élève et la machine. Mais ce choix introduit d'autres contraintes que nous avons prises en compte lors de la conception et le développement. Les contraintes que nous présentons et analysons dans la suite sont les suivantes :

- Permettre la représentation de l'analyse didactique.
- Permettre à l'élève de construire sa propre représentation.
- Les modes de représentation doivent êtres conformes à l'analyse épistémologique.

2.1 Permettre la représentation de l'analyse didactique

L'analyse didactique nous a permis de produire le type de représentation à prendre en compte. Cette analyse nous permet de dégager les aspects concernant le savoir en jeu, c'est-à-dire, les connaissances qui interviendront dans l'interaction entre la machine et élève et qui sont source d'apprentissage.

Une analyse des travaux en didactique nous a permis de déterminer la représentation de la preuve ainsi que son organisation dans une activité de résolution de problèmes. Cette représentation concerne d'une part le domaine de phénoménologie

([BAL 94], p. 33), c'est-à-dire la façon dont l'élève construira et manipulera les objets pour les différentes phases de production d'une preuve, et d'autre part la représentation interne à la machine, ou système formel (ibid.). Nous présentons dans cet article l'aspect lié au domaine de phénoménologie.

Ainsi, Balacheff ([BAL 91]) présente l'argumentation comme un obstacle dans l'apprentissage de la preuve. Cela est lié au fait que l'argumentation est un type de discours utilisé dans des situations de la vie courante. L'auteur montre le besoin de négocier explicitement de nouvelles règles qui vont permettre le passage de l'argumentation, à la preuve sans pour autant rejeter l'argumentation mais en permettant une adaptation au nouveau type de processus (ibid. p. 189).

> "...une argumentation n'est pas une démonstration. Ce qui les sépare tient à leurs contraintes respectives d'organisation. Pour qu'un raisonnement puisse être une démonstration, il est nécessaire qu'il soit un raisonnement valide. L'argumentation, au contraire, est un raisonnement qui n'obéit pas à des contraintes de validité mais à des contraintes de pertinence" (Duval 1992, p.42).

"Pour passer d'un mode de fonctionnement à l'autre (de l'argumentation à la démonstration), une décentration à l'égard du contenu d'une part, et une prise de conscience de l'existence d'une autre valeur épistémique d'autre part, sont donc nécessaires" (ibid., p. 47). L'auteur souligne ainsi, que les valeurs épistémiques de l'argumentation sont centrées sur la compréhension du contenu des propositions, tandis que dans le cas de la démonstration, ces valeurs épistémiques sont centrées sur le statut théorique des propositions.

Dans ce sens, nous retenons la différenciation, proposée par Duval ([DUV 91]), qui donne des éléments explicites à prendre en compte dans des situations de construction de preuves. L'organisation ternaire du raisonnement déductif et le rôle des valeurs épistémiques des énoncés sont des éléments qui vont différencier explicitement la preuve en mathématique de l'argumentation. L'organisation ternaire est représentée de la manière suivante :

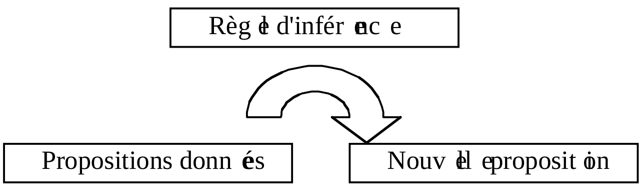


Figure 1, Diagramme proposé par Duval ([DUV 91]) et représentant le fonctionnement ternaire d'un pas d'inférence.

Cette organisation est caractérisée par deux composantes : l'enchaînement et l'inférence.

L'enchaînement est le passage d'un pas de l'organisation déductive à un autre et l'inférence est le pas du raisonnement lui-même. Cette représentation ternaire met en évidence le caractère opératoire des énoncés participant à un raisonnement déductif qui différencie ce dernier du raisonnement argumentatif :

“ Cela introduit une première différence importante entre le raisonnement déductif et le raisonnement argumentatif. Celui-ci recourt à des règles implicites qui relèvent en partie de la structure de la langue, et en partie des représentations des interlocuteurs : le contenu sémantique des propositions y est donc primordial. Au contraire, dans un pas de déduction, les propositions n’interviennent pas directement en fonction de leur contenu mais en fonction de leur statut opératoire, c’est-à-dire de la place qui leur est préalablement assignée dans le fonctionnement du pas ” ([DUV 91], p.235). Le statut opératoire est donc la place qui est assignée à l'énoncé dans le modèle ternaire de la déduction. Il y a donc trois types de valeurs opératoires : les propositions d’entrée, la règle de déduction et la proposition de sortie.

Nous avons fait le choix de permettre à l'utilisateur de produire sa preuve pas à pas, en indiquant quel est la proposition à déduire, quels sont les énoncés donnés et par quelle règle de déduction cet énoncé sera déduit (Figure 2). Le logiciel réagit par rapport à cette organisation déductive et contraint l'élève à organiser sa connaissance sur cette forme. Par exemple, si l'élève construit un pas de preuve où la règle de déduction est manquante le logiciel demandera à l'élève de rajouter un théorème ou une définition.

Figure 2, Un pas de preuve dans Cabri-Euclide

L'autre aspect que nous avons retenu pour la différenciation entre le raisonnement déductif et le raisonnement argumentatif est la valeur épistémique d'un énoncé. La valeur épistémique est le degré de certitude ou de conviction attaché par l'usager à un énoncé ([DUV 92]). " Pour passer d'un mode de fonctionnement à l'autre (de l'argumentation à la démonstration), une décentration à l'égard du contenu d'une part, et une prise de conscience de l'existence d'une autre valeur épistémique d'autre part, sont donc nécessaires " (ibid., p. 47). L'auteur souligne ainsi, que les valeurs épistémiques de l'argumentation sont centrées sur la compréhension du contenu des propositions, tandis que dans le cas de la démonstration, ces valeurs épistémiques sont centrées sur le statut théorique des propositions.

Afin de favoriser le raisonnement déductif, Duval souligne l'importance du passage des différentes valeurs épistémiques en valeur opératoire : “ Les valeurs épistémiques doivent être " traduites " en l'un des trois statuts opératoires pour que les propositions puissent être reliées déductivement ” (ibid., p.46). Ainsi, dans un pas de déduction admis, les propositions d’entrée sont acceptées comme prémisses dont la vérité n’est pas mise en doute (dans ce pas de déduction). La règle d’inférence permet de faire la transition et elle est donc acceptée comme vraie (ou institutionnalisée) par la communauté, enfin la proposition de sortie doit devenir vraie une fois que ce pas est accepté par cette communauté, au sens où la valeur épistémique est impliquée par l'inférence acceptée.

Dans Cabri Euclide, nous acceptons comme énoncés possibles d’entrée des hypothèses, des lemmes, des conjectures ou des propriétés vraies (pas de la preuve qui est déjà prouvée). La règle de déduction peut être une définition (par exemple [AB] et parallèle à [CD] et [AD et parallèle à [BC] par la définition de parallélogramme ABCD est un parallélogramme) ou un théorème. Enfin, la proposition de sortie peut être une conjecture, une propriété vraie ou la conclusion. Nous expliquons dans la suite chaque statut plus en détail.

Conclusion (CN) :

> La conclusion désigne la cible d'une preuve en cours de construction. La notion de conclusion ici est plus restreinte qu’elle ne l’est dans la communauté mathématicienne parce qu'un seul énoncé peut être qualifié de conclusion dans un problème. Ce choix est fait pour bien marquer la différence entre la conclusion cible du problème et les conclusions intermédiaires, qui dans notre cas seront appelées "propriétés vraies". Il est possible d’utiliser les propriétés vraies dans les pas de preuve, mais il est impossible de placer la conclusion du problème ailleurs qu'au terme de la preuve.

Propriété vraie (PV) :

> Une propriété vraie a été construite dans une session de travail, elle est associée à une figure particulière et peut faire partie d’une preuve plus générale. Une "propriété vraie" peut devenir un théorème, mais elle doit alors passer, du point de vue didactique, par un processus d’institutionnalisation et, du point de vue informatique, par un processus de généralisation, ou décontextualisation, qui la détache du problème dans lequel elle a été construite. Au cours de la résolution d’un problème, on peut donner ce statut à une conclusion intermédiaire.

Hypothèse (HY) :

> Une hypothèse est un énoncé qui n'a pas de preuve car l'hypothèse est admise comme donnée du problème.

Théorème (TH) :

> Un théorème est une règle acceptée comme vraie pour tous les problèmes car elle a été démontrée à un moment donné ou bien elle a été

admise. Du point de vue didactique, les théorèmes sont pour nous les énoncés institutionnalisés, c'est-à-dire acceptés par le professeur. La gestion de ces règles doit être prise en charge par le professeur. C'est lui qui aura accès à la modification de ces règles (il pourra les construire, détruire, modifier). Le théorème est donc soit un élément de base du logiciel (le professeur l'a mis dans la liste de théorèmes admis ou licites), soit un élément rajouté par le professeur lors d'une interaction avec Cabri-Euclide suite à la construction de sa preuve.

Définition (DF) :

Une définition fixe les caractéristiques d'un objet avec les objets associés, sa syntaxe et ses propriétés. Un énoncé qui a le statut de "définition" peut avoir une preuve associée, la fonction de cette preuve étant de décrire l'objet avec les propriétés qui lui sont associées. Il existe deux types de définitions : les définitions construites par l'utilisateur dans un chantier particulier, et les définitions généralisées et institutionnalisées qui ne sont associées à aucun chantier en particulier. Une définition créée par un utilisateur peut devenir une définition institutionnalisée sous les mêmes conditions qu'une propriété vraie peut devenir un théorème.

Conjecture (CJ) :

Une conjecture est un énoncé qu'on suppose vrai mais qui n'est pas prouvé. Les conjectures peuvent être utilisées dans la construction d'une preuve, mais le logiciel demandera de prouver les conjectures pour que la preuve soit acceptée comme valide.

Conjecture fausse (CF) :

Une conjecture fausse est un énoncé qui a une réfutation par rapport aux énoncés de sa preuve ou par rapport au dessin, dans le dernier cas, cette réfutation est montrée avec un contre-exemple. On peut vouloir conserver une conjecture fausse soit parce qu'on veut raisonner juste avec une figure que l'on sait fausse, car sa construction exacte est elle-même un problème, soit parce qu'on veut mettre en évidence la réfutation d'un énoncé, soit parce qu'on veut le conserver comme problème à traiter plus tard.

Lemme (LE) :

Un lemme est une proposition qui a été introduite sans preuve parce que l'utilisateur pensait pouvoir le prouver plus tard, ou parce qu'il lui est possible de le prouver par analogie avec une autre partie de preuve. Pour le moment, le logiciel se limite à accepter comme vrai l'énoncé, mais si l'utilisateur demande la vérification de sa preuve, le logiciel lui rappellera qu'il serait prudent de prouver les énoncés qui ont le statut de lemme. Cela dit, la conclusion est acceptée comme vraie dans ce cas-là.

Selon nous le processus de construction de preuve est constitutif de la résolution du problème. La construction d'une preuve peut ne pas aboutir à une démonstration au sens des mathématiciens, mais seulement à une explication (satisfaisant le sujet). Dans le travail de construction d'une preuve, un enjeu sera l'explicitation des faits et des connaissances mis en œuvre, aussi que celle de leur organisation structurée.

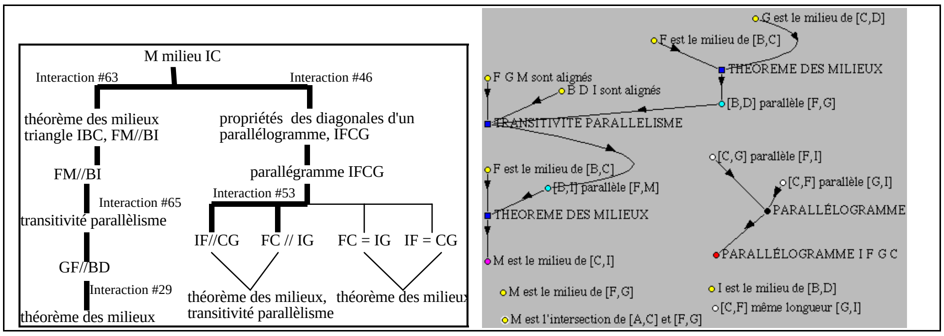
Nous voyons ainsi dans le raisonnement déductif avant tout un processus qui permet d'établir la validité d'une proposition ([BAL 87], p. 148). Il accepte les intuitions, les errances, les implicites. Il n'est pas nécessairement séquentiel : dans une preuve, on peut arriver à la conclusion cible en divisant le problème en sous-problèmes, et réaliser le processus d'argumentation et de raisonnement déductif à l'intérieur de chaque sous-problème. La résolution des sous-problèmes n'est pas nécessairement effectuée de manière hiérarchisée.

Ainsi, nous pouvons voir que les différents choix réalisés par rapport au raisonnement lié au processus de résolution des problèmes peuvent contraindre les interactions entre l'élève et la machine : "Les contraintes imposées par DEFI 1.0 sur l'expression d'une solution par les élèves ont eu un effet amplifié par les capacités limitées du logiciel à prendre en compte leurs intentions" [LUE 98, p.20]. Par exemple, dans l'expérimentation réalisée avec DEFI [LUE 93], il a été montré comment l'impossibilité de produire une preuve en chaînage arrière a perturbé le travail de l'élève : "Lorsque les élèves ne satisfont pas ces contraintes de chaînage, DEFI 1.0 réagit en terme de validité de l'articulation conclusion-théorème-hypothèses. Il ne ‘comprend’ donc pas le problème rencontré par les élèves qui, eux-mêmes, ne parviennent pas à interpréter la réaction du logiciel" (ibid., p.21).

Nous avons donc la contrainte de permettre à l'élève de produire des preuves qui ne soient pas nécessairement monotones[1]. La construction et l’association de ces énoncés se fait dans Cabri Euclide de façon dynamique, c’est-à-dire qu’il est possible de les construire, les détruire, les reconstruire, les associer et les dissocier ; notons que la preuve peut également se construire en chaînage arrière, avant ou mixte.

L'exemple ci-dessous, montre la démarche non monotone suivie par un binôme lors d'une expérimentation avec Cabri-Euclide [LUE 97]. A gauche, nous montrons le diagramme des solutions avec le parcours suivi par le binôme (indiqué par le numéro d'interaction) et, à droite, le graphe de résolution produit par le binôme.

[1] Une démarche monotone implique que la preuve doit être vraie à pendant toute la construction de la preuve.



Exemple 1, Démarche de résolution suivi par un binôme d'élèves dans une expérimentation avec Cabri-Euclide [LUE 97, p.222]

2.2 Permettre à l'élève de construire sa propre représentation

Le principe de permettre à l'élève de construire sa propre représentation vient de l'idée qu'un système informatique doit permettre à l'élève de raisonner en produisant lui-même des représentations graphiques et symboliques ([DIM 95], p. 189). Entre autres, pour nous, une des conditions centrales de ce principe est que le système ne doit pas substituer à l'action de l'élève des automatismes qui d'une certaine façon préjugeraient de ses intentions. Nous pouvons voir, par exemple, que le logiciel Chypre ([BER 93]) présente la résolution d'un problème comme la construction d'un réseau de faits proposés par l'utilisateur et reliés entre eux, automatiquement, par l'intermédiaire de théorèmes. L'utilisateur ne participe pas au choix des théorèmes, tandis que dans la tâche d'organisation déductive que nous avons montrée (§ 2.1) le choix du théorème joué un rôle important, car il met en évidence la structure ternaire et les statuts opératoires (nous développerons cela dans la suite §2.2.1). Dans la conception de Cabri-Euclide, au contraire, l'idée de ne pas réaliser d'automatismes vis-à-vis de la connaissance est prépondérante.

"L'organisation des connaissances joue un rôle fondamental dans l'activité de résolution de problèmes" [GUI 89]. Par ce principe, nous cherchions également à respecter, autant que possible, les modes de représentation des élèves afin de les faire évoluer, et donc de permettre l'apprentissage.

Nous avons ainsi fait le choix de permettre à l'élève d'organiser et d'articuler sa connaissance dans le Chantier. Le "Chantier" est défini comme le lieu privilégié de la résolution du problème. La cohérence d'un chantier est intrinsèque à lui-même. Nous avons fait en sorte que chaque élément d'un chantier soit analysé par rapport aux éléments qui interviennent dans ce chantier.

Nous avons associé trois types de registres à la construction de la preuve : dessin, texte et graphe. Pour le dessin et le graphe, nous utilisons des outils qui ont été conçus pour un domaine plus large que celui que nous traitons. Aussi, nous avons été amenés à faire des choix relatifs au rôle de ces deux registres et aux objets qui seront utilisés dans notre conception ([LUE 97a], p.78).

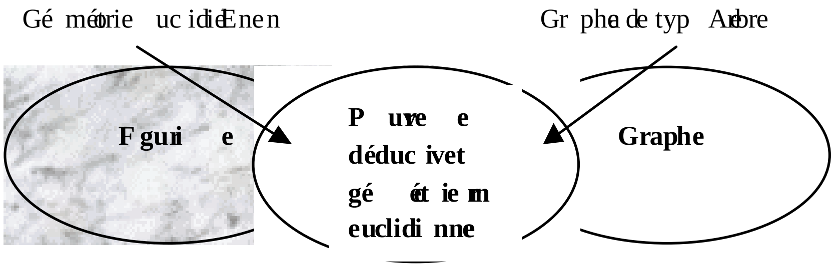


Figure 3, les trois registres

La contrainte de permettre de dérouler une preuve de façon non monotone oblige dans l'articulation de ces trois registres à considérer la possibilité de retour en arrière et celle de s'engager dans des tentatives qui n'aboutissent pas à la solution du problème. Ainsi, dans l'exemple que nous avons montré (Exemple 1), l'élève a suivi une démarche où il a commencé avec une stratégie puis il a essayé de résoudre le problème avec une autre stratégie et finalement il est revenu sur la première stratégie. Cette démarche est faite en étroite relation entre les trois registres, l'élève peut faire, par exemple, des modifications sur la figure qui doivent être prises en compte dans le registre du texte.

De même, si l'élève produit un énoncé qui se révèle faux à la suite d'une réfutation, il peut rester cependant dans le chantier qui représente la trace du travail de l'élève dans l'espace de résolution du problème. De cette façon, l'élève a le choix de garder cet énoncé pour le travailler plus tard comme un problème à part ou de reformuler les propositions (la figure ou le texte) pour que cette proposition devienne valide.

Dans la suite, nous exposerons les choix didactiques et les choix informatiques qui justifient les trois registres et les caractéristiques que nous associons à chacun d'eux.

2.2.1 Le registre du Texte

Dans le registre textuel, il est possible de formuler des objets et des propriétés de la géométrie euclidienne. Il est également possible d'associer ces énoncés afin de produire une organisation déductive. Nous offrons la possibilité de mettre en rapport les énoncés sous la forme d'un raisonnement déductif. Les règles de déduction sont accessibles grâce à la "Boîte à outils".

La "Boîte à outils" peut être considérée comme l'endroit où nous trouvons tous les énoncés institutionnalisés, en d'autres termes, des énoncés que nous pouvons réutiliser dans un autre contexte sans que le caractère de vérité de ces énoncés soit remis en question. La ”Boîte à outils” est donc composée de définitions et de théorèmes. La contrainte d'évolution de la connaissance dans une situation d'apprentissage, oblige à que la "Boîte à outils" soit adaptable à la situation d'apprentissage souhaité, c’est-à-dire qu’il doit être possible d’enlever ou de rajouter des règles (définitions et théorèmes) selon les situations d'apprentissage qui sont en jeu. Dans notre conception, l'idée est que la gestion des théorèmes est accessible seulement par le professeur, car c'est lui qui organise le travail avec Cabri-Euclide et qui est le responsable du processus d'institutionnalisation. Par contre, la construction de définitions est accessible à tout utilisateur, car elle joue un rôle essentiel dans la dialectique des preuves et réfutations que nous expliquerons plus loin (§ 2.3).

La différence entre le théorème et la définition est que la définition possède deux rôles. La définition permet de formuler un objet sous forme d'énoncé qui ne se trouve pas parmi les objets de base du logiciel (c. f. dans Exemple 2, partie gauche) et elle peut, de plus, être utilisée comme règle de déduction (c. f. dans Exemple 2, partie droite). Le théorème est utilisé seulement comme élément de déduction.

Définition en tant qu’outil de formulation : avec la définition parallélogramme, on exprime l'énoncé parallélogramme ABCD. PARALLELOGRAMME A B C D (CJ)	Définition en tant qu’outil de déduction : la définition du parallélogramme permet de déduire la propriété de parallélisme. [G,H] est parallèle à [K,L] (CN) car PARALLÉLOGRAMME (DF) [A,C] est parallèle à [G,H] (PV) [A,C] est parallèle à [K,L] (?)

Exemple 2, Utilisation d'une définition.

En résumé, l'élève doit choisir une règle de déduction pour qu'un pas de preuve soit validé. Cette règle peut provenir de la "Boîte à outil" et elle est donc une règle institutionnalisée ou elle peut provenir d'une nouvelle définition qui a été créée dans le chantier.

Comme l'interaction avec l'élève ne se base pas sur des preuves préalablement construites, le système ne sait pas déterminer quand l'élève a fini sa preuve ou son pas de preuve. Cela implique que dans le registre textuel, quand on produit un pas de preuve, il y a deux types de vérifications : des vérifications automatiques et des vérifications qui sont réalisées à la demande de l'utilisateur.

Les vérifications automatiques sont produites dans l'organisation déductive : si l’utilisateur introduit un énoncé, et non une règle de déduction, Cabri-Euclide vérifie d’abord si l’énoncé n’était pas déjà présent dans le pas de la preuve et puis il vérifie que l’introduction de cet énoncé ne produit pas une preuve circulaire. Si l’utilisateur rajoute un théorème ou une définition dans un pas de preuve, Cabri-Euclide vérifie que l’énoncé n'a pas déjà un autre théorème ou une autre définition. Il vérifie aussi si cette règle peut déduire l’énoncé.

Les vérifications demandées par l'utilisateur se font dans l'interface par une icône appelé "Vérification de la cohérence", laquelle on utilise quand on veut vérifier si une preuve ou un pas de preuve est correcte. Ces vérifications sont liées à la vérification de l'exactitude de l'organisation déductive et aux statuts des énoncés, dans le registre textuel. La première vérification consiste à examiner la présence dans un pas de preuve des propositions données comme prémisses, de la règle de déduction et de la conclusion intermédiaire. Ainsi, le théorème, ou la définition permet de vérifier si les hypothèses sont suffisantes et appropriées dans ce pas de déduction. La deuxième vérification est réalisée en fonction du statut de l’énoncé proposé. Les vérifications sont ici d’ordre épistémique. Par exemple, un message envoyé par Cabri-Euclide peut être du type : "La preuve est correcte, mais il faut démontrer les conjectures" ou "L'énoncé est une conjecture fausse : la preuve n'est pas valide" [LUE 97b p. 94].

2.2.2 Le registre du dessin

De nombreux articles traitent du rôle du dessin dans le processus de preuve ([ALM 92], [BAZ 93a], [BEL 92], [HOY 98] [SCH 85]) et la résolution de problèmes en géométrie ([LAB 92], [LAB 94]) :

“La lecture structurée, le codage graphique qui l'accompagne, permettent de faire jouer à la figure le rôle d'un condensé de sens, d'un "réservoir d'hypothèses", ...recherche d'indices, ...la figure est source des conjectures.” ([BAZ 93b]).

“ La particularité du dessin est de constituer un outil privilégié de mise en évidence des propriétés géométriques. Il possède la faculté d'organiser graphiquement des données formelles. Il peut ainsi faire apparaître des relations entre des éléments graphiques ; relations qui n'apparaissent pas dans une description formelle du fait de son caractère séquentiel. Le dessin favorise la mise en évidence d'assertions en proposant une réorganisation des données formelles ”. ([BEL 93])

“La figure est très souvent utilisée comme un révélateur de sous problèmes, il s'agit d'une structuration de la figure (et sous-figures) en termes de propriétés géométriques et de relations prenant en compte la conclusion du problème” ([ALM 93]).

La mise en évidence des propriétés géométriques qui vont permettre la résolution d'un problème de géométrie se réalise dans une interaction figure-preuve. Nous avons choisi de favoriser cette interaction en permettant l'association des énoncés et des figures, l'association des preuves et des figures et le traitement cohérent de ces interactions.

Comme nous pouvons le voir dans les travaux de Bazin ([BAZ 93a]), à plusieurs étapes de la résolution il y a des moments où se produisent des enrichissements de la figure initiale, des extractions de sous-figures, dans des allers-retours entre la figure et la preuve. Cela impose comme contrainte que la construction de la figure et la construction de la preuve ne soient pas exécutées de manière indépendante (selon un schéma par exemple du type figure-preuve).

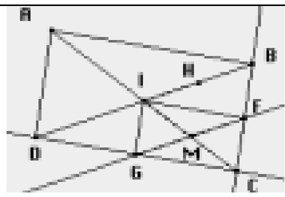
Un énoncé peut avoir des relations privilégiées avec certains objets de la figure, dans ce cas, ces objets composent une sous-figure qui lui sera associée. De la même façon, on peut établir des relations privilégiées entre une figure et une preuve d'un énoncé.

Dans le cas de Cabri-Euclide il y a une exigence de cohérence entre la construction de la figure et la déclaration des énoncés, ainsi à chaque construction d'un nouvel objet ou d'une nouvelle relation (soit dans la figure, soit dans la preuve), des vérifications sont réalisées par rapport aux éléments déjà existants. La même vérification est exécutée au moment de l'élimination d'un objet. Les vérifications portent sur l'existence et sur les propriétés géométriques qui ont été construites dans la figure et dans la preuve.

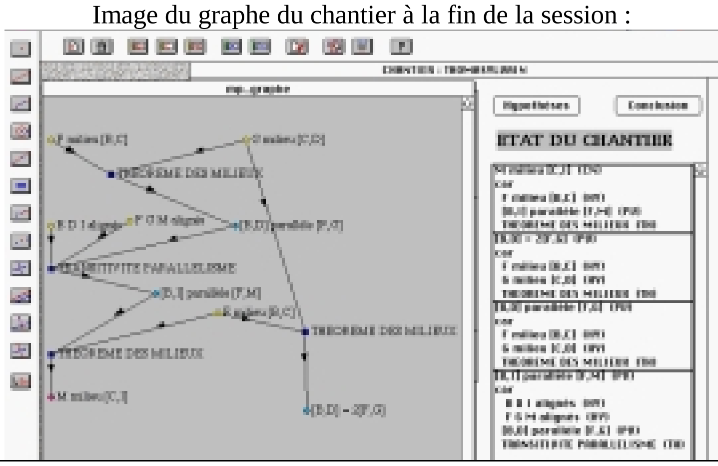
Ainsi, tous les objets cités dans le registre textuel doivent exister dans le registre du dessin, i.e. ils doivent exister en tant qu'objet du dessin (si on produit l'énoncé [AB] parallèle à [CD], les segments [AB] et [CD]). Nous avons imposé cette contrainte pour différentes raisons, entre autres, la construction de la figure est une source assez riche d'information par rapport au travail de l'élève. Pour avoir un suivi cohérent, il est nécessaire que les informations soient complètes dans un des deux sens (figure-texte, ou texte-figure). Nous aurions pu penser faire des constructions automatiques lors des formulations textuelles, mais dans ce cas-là, le travail heuristique aurait été plus éloigné des interactions entre la machine et l'élève. Dans ce cas, nous irions contre le principe de ne pas produire d'automatismes par rapport au raisonnement de l'élève, car si le logiciel produit les constructions c'est la machine et non le sujet qui prendra des décisions à propos des objets mathématiques construits. De même, si nous avions permis de formuler un objet qui n'existait pas dans la figure, nous aurions été obligés de faire des déductions automatiques sur le travail de l'élève en laissant, éventuellement, de côté d'autres propriétés de l'objet produit. Par exemple, nous pouvons formuler qu'un point est milieu d'un segment, mais ce point peut être plus que cela (point d'intersection, milieu des diagonales), et cette information peut être précieuse lors du suivi du travail de l'élève.

La possibilité d'utiliser des solutions préalables joue également un rôle dans l'interaction figure-preuve. Ainsi, l'impossibilité de créer de nouveaux objets qui n'ont pas été pris en compte dans ces solutions préalables [LUE 98 p. 36], est un choix relatif au processus de recherche de solution qui peut contraindre le raisonnement de l'élève. Nous avons pu constater dans l'expérimentation avec DEFI en 1993 que les élèves ont pu rajouter l'objet dans le dessin mais au moment des formulations cela n'a pas été compris par la machine et a perturbé le travail de l'élève [LUE 98].

Dans le cas d'une expérimentation avec Cabri-Euclide, nous avons constaté que, au contraire, le rajout d'un objet a aidé les élèves dans leur processus de recherche d'une solution (Exemple 3).

<table>
<tr><th>Dialogue des élèves</th><th>Production dans l'ordinateur et observations</th></tr>
<tr><td>91. S : je ne sais pas. Attends, on a un triangle là aussi. On as BIC, donc avec le théorème des milieux, peut être que aussi on pourrait aboutir à M, si on fait le milieu de là-bas, peut-être, ha oui, non ça ne marche pas. Si !, regarde, si on fait le milieu de BI,
92. S : bien H par exemple.
93. T : hypothèse H milieu de BI.
94. ...
95. T : tu fais dans hypothèse, dans le triangle qu'on va faire, il faut démontrer que M c'est le milieu de IC.
96. S : oui mais après il faut démontrer que BC//HM. Si tu veux démontrer que M c'est le milieu.
97. …</td><td>

H milieu [B,I] (HY)
Ils discutent pendant un moment sur plusieurs propriétés, mais ils ne les formulent pas dans Cabri-Euclide. Puis ils regardent le théorème des milieux

Ils ont abandonné l'idée avec le point H, mais cette démarche leur a permis de prendre en compte le théorème des milieux.</td></tr>
<tr><td colspan="2">Image du graphe du chantier à la fin de la session :

</td></tr>
</table>

Exemple 3, Extrait d'une expérimentation avec Cabri-Euclide [LUE 97, p. 229]

2.2.3 *Le registre du graphe*

> "Le recours à des organigrammes ou à des réseaux s'est imposé pour représenter la structure profonde de la démonstration (Truong 1972 ; Anderson 1982, 1978 ; Reynes 1981 ; Gaud 1984; ...). Et plus généralement les travaux faits dans le cadre de l'intelligence artificielle ont imposé la représentation en réseau comme outil pour toute représentation des connaissances et par suite pour tous les modèles de compréhension du discours (Quillian 1969, Rumelhart 1972,1975). Les réseaux mettent en relief les relations qui lient entre eux les énoncés ainsi que le sens de ces relations. Celles-ci deviennent accessibles indépendamment de la compréhension des termes désignant l'implication, l'équivalence ou tout autre connecteur logique. En outre, par l'appréhension synoptique qu'ils donnent de l'organisation déductive, les réseaux permettent de détourner l'attention du contenu des énoncés vers leur statut" ([DUV 89], p.30).

Le rôle du graphe ici est celui de l'aide à l'organisation du travail de résolution de problèmes et à la prise en compte du rôle opératoire des énoncés. Nous voulons montrer avec le graphe la structure sous-jacente à l'organisation déductive des énoncés. Le graphe est orienté, chaque sommet est associé à un énoncé et chaque arête met en rapport les propositions avec les règles d'inférence (théorèmes ou définitions).

Quant à l'aide pour l'organisation dans la résolution du problème, nous pensons que la possibilité de pouvoir accéder à des sous-graphes de preuve permet à l'utilisateur de se situer par rapport à sa construction ([GUI 95], [BER 95]). Ainsi, il est possible de demander le graphe du chantier, mais aussi le graphe de la preuve d'un énoncé ou d'une partie de la résolution. Dans le graphe du chantier sont représentés tous les énoncés ainsi que leur organisation déductive, tandis que dans le graphe de la preuve d'un énoncé il y a l'énoncé et sa preuve.

Le graphe est un outil d'aide au raisonnement déductif, car il est possible de visualiser les deux passages d'une preuve déductive que nous avons présentés plus haut (§ 2.1), c'est-à-dire l'inférence et l'enchaînement. Les différents types de statuts épistémiques des énoncés sont également visibles à chaque sommet du graphe.

Le graphe peut être également utile pour le professeur, car il permet à ce dernier de comprendre le processus de résolution suivi par l'élève lors de la résolution du problème.

Dans le prototype actuel de Cabri-Euclide, le graphe est une représentation graphique du registre textuel, l'utilisateur n'a pas la possibilité de les créer. Les graphes accessibles sont les graphes correspondant à l'état du texte du chantier ou d'une partie de la preuve qui a été choisie. L'utilisateur peut produire des conjectures au niveau du texte et le voir dans le chantier (chaque statut est représenté par une couleur différente dans le graphe), cela peut aider dans la recherche d'une solution et laisse la possibilité de faire le point sur l'état du travail [GUI 89].

L'illustration suivante (Figure 4) représente un chantier dans Cabri-Euclide. Côté gauche, il y a la figure construite par l'utilisateur avec Cabri-Géomètre ([LAB 85]) et le graphe de la conclusion construit avec Cabri-Graphe ([CAR 95]), et côté droit, il y a la liste des énoncés et enchaînements que l'utilisateur a proposés.

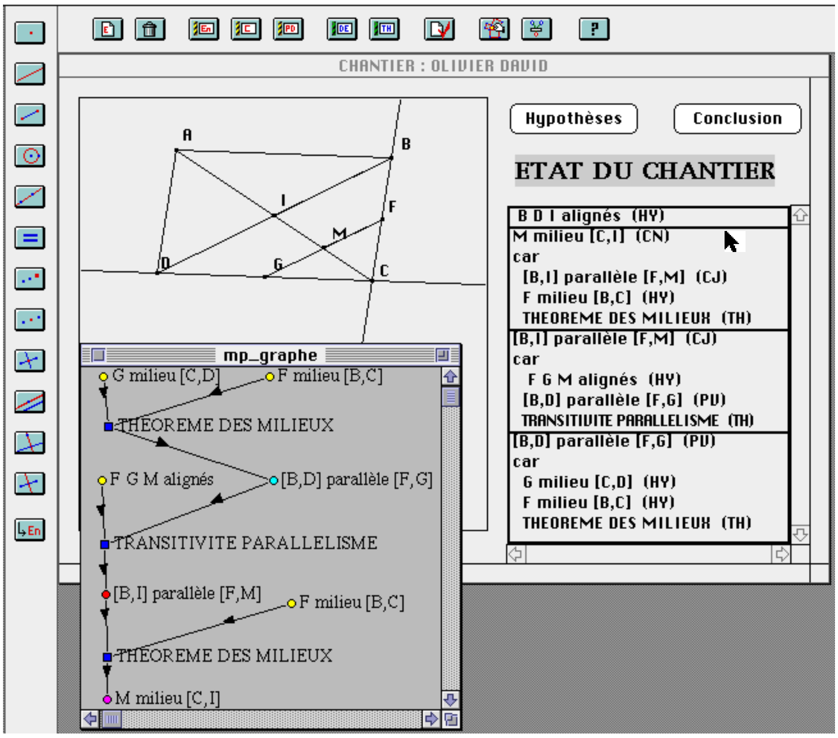


Figure 4, Un chantier de Cabri-Euclide

2.3 *Les modes de représentation doivent être conformes à l'analyse épistémologique*

"Il convient d'examiner à chaque fois si les outils offerts pour construire les représentations et les activités proposées sont conformes à l'analyse épistémologique" ([DIM 95], p. 191). Pour nous, l'analyse épistémologique permet la modélisation des interactions souhaitées entre la machine et l'élève de façon à faire émerger la connaissance qui est en jeu.

Nous avons retenu deux aspects de l'analyse didactique et épistémologique à propos de la preuve : la construction des preuves pragmatiques et des preuves intellectuelles, et la dialectique des preuves et réfutations. Cette analyse concerne le domaine de phénoménologie, car nous définissons en fonction de cette analyse les contraintes d'interactions entre les différents types de registres (figure, texte et graphe) et les élèves.

2.3.1 Preuve pragmatique et preuve intellectuelle

Balacheff ([BAL 88]) a proposé une analyse, épistémologique et expérimentale à propos de la preuve et la démonstration, et une catégorisation des preuves fondée sur les moyens utilisés pour "croire" dans la validité d'une formulation.

L'auteur définit les preuves pragmatiques comme les preuves fondées sur l'action effective mise en œuvre sur des représentations d'objets mathématiques. Les preuves intellectuelles, par contre, sont détachées de l'action, inscrites dans des conduites langagières exprimant les objets et leurs propriétés et calculant leurs relations.
Le passage des preuves pragmatiques aux preuves intellectuelles repose sur une évolution conjointe de trois pôles : action, formulation, validation, en étroite interaction. Le tableau ci-dessous ([BAL 96]) illustre la correspondance entre ces différents pôles :

Niveaux d'action	**Formulation**	**Validation**
	ostension	
pratiques(savoir-faire, théorème-en-acte)		preuves pragmatiques
	langage de la familiarité	
connaissances comme objet (savoir)		preuves intellectuelles
	langage fonctionnel	
connaissance théorique (système axiomatique, théorème)		
	formalisme naïf	démonstration

Tableau 1 : Tableau proposé par Balacheff. Ce tableau suggère des positions relatives et non des correspondances strictes entre les niveaux des différentes colonnes.

L'auteur distingue quatre types de preuves (empirisme naïf, expérience cruciale, exemple générique, expérience mentale) qui vont de la preuve pragmatique à la preuve l'intellectuelle.

> "On peut distinguer parmi les preuves pragmatiques et intellectuelles quatre principaux types qui occupent une place privilégiée dans la genèse cognitive de la démonstration : l'*empirisme naïf,* l'*expérience cruciale,* l'*exemple générique* et l'*expérience mentale* (Balacheff 1987). Au-delà de l'expérience mentale nous situons le *calcul sur des énoncés,* forme dans laquelle s'exprime la démonstration. Les deux premiers types perdront le nom de preuve dans toute communauté ayant acquis un peu de culture scientifique, ils sont les racines de la

preuve au sens commun. Par ailleurs, il existe une rupture fondamentale entre ces deux premiers types et les deux suivants. En effet, pour l'exemple générique et l'expérience mentale il ne s'agit plus de "montrer" qu'un énoncé est vrai parce que "cela marche", mais d'établir le caractère nécessaire de sa validité en en dégageant des raisons." (ibid.)

Permettre le passage de la preuve pragmatique vers la preuve intellectuelle invite à laisser à l'élève un espace pour le travail heuristique et un espace pour la preuve intellectuelle qui doit être décontextualisé de la figure et qui impose des contraintes qui rendent impossible la proposition des preuves pragmatiques.

Ainsi, nous avons choisi de disposer d'un lieu de travail sur la figure qui permette l'analyse heuristique et la production des preuves pragmatiques, et d'un lieu pour le traitement des expressions langagières et l'analyse de leur organisation du point de vue de la logique pour produire des preuves intellectuelles. Autrement dit, nous concevons un environnement qui permet l'expression de preuves pragmatiques, à l'aide du registre graphique (dessin et figure) et des preuves intellectuelles dans le registre langagier. Cabri-Euclide donne accès aux deux et organise l'interaction entre le deux.

Les validations seront un élément décisif par rapport à la production des preuves pragmatiques ou des preuves intellectuelles : " l'accès possible ou non à l'expérience constitue une caractéristique de la situation qui va jouer un rôle central dans le fonctionnement de cette dialectique de validation. La mise à exécution d'une décision, ou la réalisation du contenu d'une affirmation, permet ce que nous appelons des validations pragmatiques de la décision, ou des preuves pragmatiques lorsqu'elles sont effectuées par l'élève lui-même pour établir la validité d'une proposition. Lorsque cet accès à la réalisation n'est pas possible, alors les validations sont nécessairement intellectuelles. La production de ces preuves intellectuelles requiert notamment l'expression langagière des objets sur lesquels elles portent et de leurs relations " ([BAL 87], p. 157).

La contrainte qui s'impose donc est de concevoir la possibilité de créer des phases de validation dans le registre graphique et dans le registre textuel. Dans Cabri-Euclide nous avons fait le choix que les validations empiriques se réalisent du côté de la figure et que les validations intellectuelles se passent dans le registre textuel. Ainsi dans le registre graphique, les élèves peuvent faire des validations heuristiques en utilisant en particulier le caractère dynamique de Cabri géomètre. Le déplacement constitue un des moyens qui vont permettre à un instant donné de continuer ou non à soutenir une proposition. Mais les validations dans ce registre peuvent rester du côté des preuves pragmatiques et c'est pour cela que nous considérons les phases de validation dans le registre textuel qui vont permettre le passage aux preuves intellectuelles.

Nous avons donc déterminé plusieurs variables pour favoriser ce passage. Nous nous appuyons principalement sur la distinction entre les valeurs opératoires d'un énoncé et sur l'aspect déductif des preuves que nous considérons. Comme nous l'avons présenté plus haut (§2.1), Duval ([DUV 92]) montre comment la capacité de distinguer entre la valeur opératoire des énoncés utilisés dans la résolution et leur valeur informationnelle constitue une évolution essentielle dans le passage de l'argumentation à la démonstration. Quant à l'aspect déductif, la validation fait référence à la règle utilisée dans la déduction, i.e. vérifier si le théorème sert à déduire l'énoncé donné, et elle fait référence également à l'organisation de la déduction, i.e. le logiciel vérifie si les hypothèses sont suffisantes et appropriées à la déduction. Ce type de vérification est détaché de la figure et spécifique aux preuves langagières. L'intérêt de ces moyens de validation tient à leur capacité de réfutation et aux moyens donnés pour le dépassement de cette réfutation.

Nous pouvons distinguer dans le tableau suivant les capacités des différents registres par rapport aux possibles interactions :

	Objet du Savoir	**Formulation**	**Action**	**Validation**
Cabri-géomètre1	Figure géométrique	Objets géométrie euclidienne : points, droites, ..	• Construction, • Macro-construction • Sauver • Ouvrir • Quitter • Changement des propriétés des objets • Déplacement	• Déplacement, • Vérification des propriétés de la figure **preuves pragmatiques**
Cabri-Euclide	Preuves déductives en géométrie euclidienne	Propriétés, définitions, théorèmes, règle d'inférence Hypothèses => Conclusion	• Construction • Macro-construction • Association • Sauver • Ouvrir • Quitter • Changement des propriétés des objets	• Vérification de la cohérence • contre-exemple • Organisation déductive • statuts **preuves intellectuelles**
Cabri-graph	Graphes de preuve	Couple : (sommet ou énoncé, arête ou lien inférentiel)	• Construction • Déplacement • Sauver • Ouvrir • Quitter	**modalité de présentation des preuves intellectuelles**

Tableau 2, types d'interaction.

Il faut souligner que toutes les capacités de Cabri-géomètre 1 et de Cabri-graph ne sont pas accessibles dans la version effectivement réalisée de Cabri-Euclide. Nous avons uniquement choisi celles qui, selon nous, sont valides dans un contexte de construction de preuves et une production de réfutations (§ 2.3.2). En particulier pour le graphe, étant donné que nous l'utilisons seulement comme un élément d'aide à la visualisation de la structure et de l'organisation des énoncés, nous ne nous intéresserons pas à l'analyse de ses propriétés.

2.3.2 La dialectique de preuves et réfutations

Le processus des preuves et réfutations est proposé par Lakatos ([LAK 76]) pour modéliser la logique de la découverte et de l'élaboration des concepts en mathématique. Le moteur de ce processus est une dialectique articulant des phases de proposition de conjectures pour répondre à un problème ou à une question ouverte, des phases de critique des conjectures à l'aide de contre-exemples, et des phases de dépassement de ces critiques grâce à divers moyens tels que l'augmentation du contenu de la conjecture ou la délimitation de son domaine de validité.

Les protagonistes de cette dialectique sont le sujet formulateur (qui propose les conjectures) et le sujet antagoniste (qui accepte ou refuse ces propositions).

Le processus de preuves et réfutations, dans une optique de résolution de problèmes, permet à chaque sujet d'être à la fois le sujet formulant les conjectures et le sujet antagoniste. En effet, celui qui est le sujet antagoniste devient le sujet formulateur d'une conjecture qui est la réfutation de la conjecture initiale. Le sujet antagoniste doit aussi convaincre de la vérité de sa réfutation.

La dialectique des preuves et réfutations est un modèle qui nous permet de représenter les interactions qui sont du côté de la formulation des conjectures et l'évolution de ces conjectures vers une preuve.

De cette orientation découle la contrainte de ne pas suivre le travail de l'élève avec des solutions pré-calculées. Cela implique que s'il y a un outil de déduction automatique, il ne soit pas utilisé pour produire des solutions préalables qui contraindront la négociation entre l'élève et la machine [LUE 98]. Une démonstration automatique est, pour nous, un outil d'aide dans l'interaction entre l'élève et la machine (pour négocier ou réfuter, par exemple), mais pas un outil qui doit contraindre le processus de recherche de solution. Par conséquent, le choix des interactions de construction, explication et réfutation visent cet objectif. Il découle également de l'analyse épistémologique la nécessité de mettre en place des moyens de négociations relatives à la dialectique des preuves et réfutations.

Cela se traduit par la contrainte de disposer d'un environnement qui soit capable d'argumenter et de communiquer et où le dialogue puisse permettre un examen critique des propositions de l'utilisateur, autrement dit qui permette un processus de négociation. Nous cherchons à aboutir à ce type d'interactions grâce à des dialogues où l'utilisateur puisse interroger la machine lors d'une rétroaction exprimant qu'une action est jugée non cohérente ([LUE 97a]).

Les explications du logiciel prennent la forme de messages. Elles peuvent être vues comme des moyens de surmonter ou de comprendre les contraintes du logiciel.

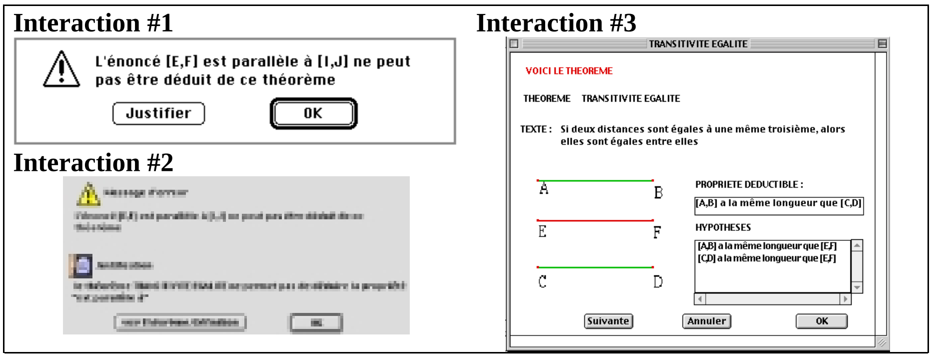
Ainsi, il y a des explications qui font référence à la bonne utilisation du logiciel de façon à faire comprendre les contraintes d'interface. Néanmoins, celles qui nous intéressent sont les explications faisant référence au contenu qui interviennent dans le processus de négociation et celles relatives à la dialectique des preuves et réfutations.

Les messages portant sur le contenu doivent, selon nous, êtres relatifs à l'objet de savoir en jeu. Ce choix est fait en fonction de notre objectif initial qui est d'accompagner l'élève dans son processus de construction et de résolution, et d'être vigilant par rapport à la cohérence de son travail. Cette interaction est réalisée par ailleurs sans lui montrer quelle doit être la façon de résoudre le problème, car dans la dialectique des preuves et réfutations, *aucun des sujets qui interviennent dans le processus ne connaît a priori la solution du problème*.

Dans cette problématique de négociation, nous avons conçu des moyens d'explication entre la machine et l'utilisateur pour permettre d'avancer dans l'interaction. Ainsi, les moyens d'explication existant dans Cabri-Euclide pour envoyer une explication à l'utilisateur sont : les contre-exemples et les justifications des réfutations produites par le logiciel. Les moyens donnés à l'utilisateur sont : la déclaration des statuts et la possibilité de créer de nouvelles définitions.

La figure peut jouer différents rôles dans l'interaction figure-preuve. Le plus manifeste est qu'elle permet de "rejeter" des conjectures. Cela justifie la contrainte que la formulation de la preuve dépende de la figure car il nous paraît essentiel de pouvoir produire des réfutations dans le dessin, étant donné que notre objectif principal est de produire une dialectique de preuves et réfutation. Cela implique, comme nous l'avons signalé plus haut, qu'il est nécessaire de connaître les objets de la figure lors des formulations textuelles pour pouvoir créer ces réfutations. Une conséquence de ce choix est que l'élève ne peut pas formuler un raisonnement correct sur une figure incohérente, dans ce cas le logiciel le reprend. Par exemple, si l'élève a construit deux droites visuellement parallèles mais qui n'ont pas été construites comme telles, et s'il formule le parallélisme de ces deux droites, alors, Cabri-Euclide lui renvoie un contre-exemple dans le registre graphique. Ainsi, à chaque formulation dans le registre textuel d'une propriété, Cabri-Euclide demande à l'oracle de Cabri-géomètre si cette propriété est vraie. Si la réponse est négative (la propriété est fausse), et si Cabri-géomètre a les moyens de produire un contre-exemple, Cabri-Euclide proposera à l'utilisateur ce contre-exemple.

Les explications sont accessibles à la demande de l'utilisateur, qui peut décider du niveau de détail de ces explications (relativement à une granularité préétablie). Ainsi, nous rajoutons dans les dialogues proposés par Cabri-Euclide une fonctionnalité permettant l'accès à l'analyse de la machine. Il consiste en un bouton "justifier" qui permet d'interroger le logiciel à propos du dernier message qu'il a produit :



Exemple 4, une interaction avec l'option "justifier"

Quant aux explications données par l’utilisateur à la machine, elles sont des constructions graphiques ou des constructions textuelles. La machine peut également demander à l’utilisateur de s’expliquer. Si, par exemple, l’utilisateur produit une formulation incluant un objet n’existant pas dans la figure, le logiciel demandera à l’utilisateur de construire cet objet pour saisir la relation de cet objet vis-à-vis des autres, et par conséquent l’utilisateur, en dessinant l’objet, montrera à la machine ce que celui-ci représente.

La déclaration du statut épistémique des énoncés (§ 2) modifie également le contexte commun aux deux interlocuteurs (apprenant et machine) et permet la négociation. En fonction du statut, le logiciel analysera de façon différente la validité ou la cohérence d'un énoncé. D'où la négociation : en fonction du statut, le logiciel interprète les intentions de l'apprenant par rapport à un énoncé et, éventuellement, lui propose un autre statut ou lui demande des conditions pour accepter ce statut.

Comme nous l'avions indiqué, Cabri-Euclide n'a pas la capacité de déterminer quand le travail de l'élève a été fini, il en est du même pour la dialectique de preuves et réfutations, il n'y a pas de validation "finale" sur la production de l'élève. Les interactions des preuves et réfutations sont locales au pas de preuve et la machine est capable d'interagir par rapport au pas qui est analysé. Par contre, l'élève à des moyens au niveau du chantier pour dépasser la réfutation, comme c'est le cas de la création de définitions.

2.3.2.1 La création de définitions : des Macro-définitions

Quand nous choisissons de permettre la dialectique des preuves et réfutations, nous retrouvons la contrainte de devoir proposer des outils qui servent cette dialectique.

Ainsi, la création des définitions est un outil qui peut être fondamental dans cette dialectique. Dans le cas de la réfutation d’une conjecture, la création d’une définition peut servir à l'augmentation du contenu de la conjecture ou la délimitation de son domaine de validité.

Nous mettons en place, donc, la possibilité de créer des définitions qui puissent être vues comme des sortes de macro-constructions[2]. S'il y a un objet que l'utilisateur veut utiliser et que le logiciel ne met pas à sa disposition, alors l'utilisateur a l'option d'exprimer ce nouvel objet avec la construction d'une définition. Comme pour toute explication dans un processus de négociation, elle peut ne pas être acceptée : dans notre cas, la machine n'acceptera pas la nouvelle définition si celle-ci n'est pas cohérente par rapport au chantier et aux énoncés qui lui ont été associés. La cohérence fait référence à l'existence des objets proposés dans la nouvelle définition et à la non-contradiction avec les énoncés du chantier où la définition est proposée. Une fois acceptées les macro-définitions, elles ont le même caractéristiques des définitions qui sont dans la boîte à outil, en particulier elles peuvent êtres utilisés pour créer de nouvelles macro-définitions. Par exemple, on peut créer la définition du parallélogramme, pour ensuite l'utiliser pour créer la définition du carré.

Nous voulons souligner qu'il est possible de construire des définitions qui ne soient pas reconnues par la communauté mathématicienne comme telles, mais qui soient valides dans le chantier sur lequel travaille l'utilisateur. Si par exemple l'utilisateur décide d'introduire une nouvelle définition 'carré', qui n'existe pas dans le logiciel, cette définition pourra avoir des propriétés qui ne sont pas celles reconnues par la communauté mathématique.

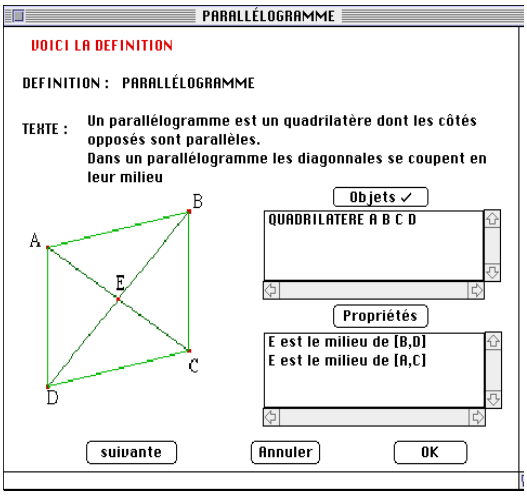
Du point de vue du domaine de phénoménologie, nous cherchons à respecter la contrainte que la création de définition doit être la plus proche possible du système de représentation manipulé par l'utilisateur dans l'environnement. Cela impose un ensemble de contraintes du point de vue informatique afin que le processus de création d'une définition n'exige pas de l'utilisateur d'autres connaissances que celles relatives à la construction d'une preuve en géométrie. Ainsi, par exemple, même si la création d'une définition implique du point de vue informatique un rajout de nouvelles règles, ce rajout doit se faire, aux yeux de l'utilisateur, dans le domaine de la preuve en géométrie (rajout de propriétés, objets) et non dans un domaine qui exige un autre niveau d'abstraction comme l'est la création de règles en calcul de prédicats.

---

[2] C'est-à-dire qu'à partir des objets de base on construit des objets composés.

Résultat de la construction d’une définition :

Cette définition a été construite par manipulation directe par le déplacement des énoncés existants vers la fenêtre de la nouvelle définition.



Nouvelle règle rajoutée dans Cabri-Euclide :
Parallélogramme(A,B,C,D) :
Quadrilatère(Point(A), Point(B),Point(C),Point(D))
$\wedge$ Milieu(Point(E),Segment(Point(A),Point(C)))
$\wedge$ Milieu(Point(E), Segment(Point(B),Point(D)))

Exemple 5, Création d'une macro-définition

Cette construction de macro-définition ([LUE 97a], [LUE 97b], [DES 98] et [MAC 98]) donne la possibilité de créer de nouveaux énoncés qui seront de nouveaux outils de formulation et éventuellement de déduction (c. f. Exemple 2).

3 Conclusion

C’est l’analyse didactique qui nous a permis de concevoir les grandes lignes de l’architecture du logiciel. Il est clair que des contraintes de développement ont pu nous faire revenir sur des choix de conception, mais notre démarche a toujours cherché à produire un outil qui découlait d'une analyse didactique.

Toute conception est basée sur des choix. Nos premiers choix ont été faits pendant l’analyse didactique (situation de preuves et réfutations, résolution de problèmes, raisonnement déductif). Les choix informatiques ont ensuite été réalisés en fonction des choix didactiques préalablement effectués. Néanmoins, les choix informatiques et la programmation du logiciel nous ont montré la complexité et les problèmes de mise en œuvre d'objets abstraits devant êtres manipulés par des utilisateurs. Ces choix ont produit des contraintes de conception et de développement.

Ainsi, permettre le raisonnement déductif implique du point de vue informatique de construire un modèle formel où il soit possible de formuler des objets abstraits représentatifs de cette démarche. Ces objets sont dans notre cas des énoncés qui peuvent être associés au sein d'une structure déductive.

Comme nous avons fait le choix de permettre ce raisonnement dans un cadre de résolution de problèmes, cela exige que la construction et l'association de ces énoncés se fassent de façon dynamique, c'est-à-dire qu'il doit être possible de construire, détruire, reconstruire, associer et dissocier sans les contraintes d'une démarche monotone.

Permettre la dialectique des preuves et réfutations a impliqué pour nous d'une part la production de réfutations par le logiciel, et d'autre part la nécessité de travailler avec des connaissances révisables, susceptibles d'évoluer au cours de l'interaction. Cela impose du point de vue informatique que, pour chaque action de l'utilisateur, le système remet en question la validité des objets qui interviennent dans cette action et que par conséquent il est nécessaire d'évaluer ou réévaluer ces objets.

Une des fonctionnalités que nous avons ajoutées au système est la possibilité de créer de nouvelles définitions, ce qui implique en informatique de permettre la création de nouvelles règles sur lesquelles le logiciel doit se baser pour faire de nouvelles évaluations. Cela est traduit par la création de nouvelles règles sans pour autant obliger l'utilisateur à les formuler dans un langage formel inconnu et sans rapport avec son travail géométrique. Nous avons résolu ce problème en permettant la production de définitions sous la forme d'un pas de preuve qui sera traduit par un arbre de déduction [LUE 97a].

Nous avons également réalisé le choix fondamental de suivre le travail de l'élève sans avoir comme référence des solutions préalables, cela nous a amenés à définir des moyens qui soient centrés sur l'activité propre à l'élève, comme par exemple interroger la figure, vérifier la preuve par la validation des pas de preuve et de leur organisation et par la valeur épistémique des énoncés.

Avec ce choix, les interactions ne seront pas perturbées par des problèmes relatifs à une solution qui n'est pas connue par l'utilisateur. Nous avons ainsi enlevé les contraintes liées aux solutions préalablement construites par la machine, contraintes que nous avons montrées lors des travaux précédents ([LUE 93], [LUE 98]).

Le processus des preuves et des réfutations est à la base de notre conception, et c'est dans ce cadre qui sont placées les perspectives de ce travail. Elles sont d'une part du côté de la réfutation, en trouvant plus de moyens pour produire des réfutations, par exemple la production d'un contre-exemple qui ne soit pas dans le registre de la figure, mais dans la preuve elle-même (le registre du texte). Et d'autre part, les perspectives sont placées du côté de la négociation, en produisant des explications plus précises du côté de la machine, par exemple quand il y a plusieurs erreurs ; et en ajoutant d'autres moyens de compréhension du travail de l'élève, comme peut-être la vérification des preuves par analogie (l'utilisateur dit qu'un énoncé est prouvé de la même façon qu'un autre énoncé, et Cabri Euclide vérifie la validité de la proposition).

4 Bibliographie

[ALM 92] Ag-Almouloud S., 1992. L'ordinateur, outil d'aide à l'apprentissage de la démonstration et de traitement de données didactiques. Thèse de troisième cycle. Université de Rennes 1.

[BAL 87] Balacheff N., 1987. Processus de preuve et situations de validation. Educational Studies in Mathematics 18, 1987, pp. 147-176.

[BAL 88] Balacheff N., 1988. Une étude des processus de preuve en mathématiques chez les élèves de Collège. Thèse d'état Université Joseph Fourier, Grenoble.

[BAL 91] Balacheff N., 1991. The benefits and limits of the social interaction : the case of mathematical proof. In Mathematical Knowledge : Its Growth Through Teaching, J. Bishop et al (eds). Kluwer Academic Publishers, pp. 175-192.

[BAL 94] Balacheff N., 1994. Didactique et Intelligence Artificielle. Recherches en didactique des mathématiques, Volume 14/1.2. La Pensée Sauvage.

[BAL 96] Balacheff N., 1996. Apprendre la preuve. In : Sallantin J., Szczeciniarz J.J. (eds.). La preuve à la lumière de l'intelligence artificielle, à paraître. Paris : PUF.

[BAZ 93a] Bazin J.M., 1993. Un modèle d'expert en résolution de problème de géométrie, Journées EIAO Cachan 1993.

[BAZ 93b] Bazin J.M. 1993. GEOMUS : un résolveur de problèmes de géométrie qui mobilise ses connaissances en fonction du problème posé. Thèse Université Paris VI.

[BEL 92] Bellemain F., 1992. Conception, réalisation et expérimentation d'un logiciel d'aide à l'enseignement de la géométrie: Cabri-géomètre. Thèse LSD2-IMAG, Université Joseph Fourier, Grenoble.

[BER 93] Bernat P., 1993. CHYPRE : Un logiciel d'aide au raisonnement, Repères - IREM n° 10.

[BER 95] Bernat P., 1995. Calques 2, manuel de l'utilisateur. Topiques Editions, Pont-à-Mousson.

[CAR 95] Carbonneaux Y., Laborde J.-M. & Madani M. R., 1995. Cabri-graph: A Tool for Research and Teaching in Graph. Graph Drawing'95, Frank J. Brandenburg (Eds.), Springer Verlag, Lecture Notes in Computer Science n°1027, pp. 123-126.

[DES 98] Desmoulins C., Macrelle M., 1998. Interopérer via des macro-définitions pour partager des connaissances, application au EIAH de géométrie, actes de Ingénierie des Connaissances, Pont-à-Mousson.

[DIM 95] Dimitracopoulou A., 1995. Le tutorat dans les systèmes informatisés d'apprentissage : étude de la conception et réalisation d'un tutoriel d'aide à la représentation physique des situations étudiées par la mécanique. Thèse Université Paris VII.

[DUV 89] Duval R., 1989. L'organisation déductive du discours. In Annales de Didactique et Sciences Cognitives 2, IREM de Strasbourg. pp. 25-40.

[DUV 91] Duval R., 1991. Structure du raisonnement déductif et apprentissage de la démonstration. Educational Studies in mathematics 22, pp. 233-261.

[DUV 92] Duval R., 1992-1993. Argumenter, démontrer, expliquer : continuité ou rupture cognitive ? In Petit x, n° 31, pp. 37-61.

[GUI 89] Guin D. Réflexions sur les logiciels d'aide à la démonstration en géométrie. *Annales de Didactique de Sciences Cognitives,* (1989), pp. 89-109.

[GUI 95] Guin D., 1995. A Cognitive Analysis of Geometry Proof Focused on Intelligent Tutoring Systems. Nato Asi Series. Berlin : Springer Verlag pp. 82-93.

[HOY98] Hoyles C., Jones K. (1998), Proof in Dynamic Geometry Contexts. In: Mammana C., V. Villani V. (eds) Perspectives on the Teaching of Geometry for the 21st Century (pp.121-128) Dordrecht: Kluwer.

[LAB 92] Laborde C., Laborde J.-M., 1992. Problem solving in geometry: from microworlds to intelligent computer environments. In : Ponte J. et al. (eds.) Mathematical Problem Solving and New Information Technology (NATO ASI Series vol. 89, pp.177-192). Berlin : Springer Verlag.

[LAB 94] Laborde C., Capponi B., 1994. Cabri-géomètre constituant d'un milieu pour l'apprentissage de la notion de figure géométrique. Recherches en Didactique des Mathématiques, Volume 14/1.2. La Pensée Sauvage, pp. 165-210.

[LAB 85] Laborde J., 1985. Projet d'un Cahier Brouillon Informatique de Géométrie. Rapport interne LSD (IMAG). Grenoble.

[LAB 95] Laborde J., 1995. Intelligent Microworlds and Learning Environments. Nato Asi Series. Berlin : Springer Verlag. pp. 113-132.

[LAK 76] Lakatos, I, 1976. Proof and Refutations. Cambridge : Cambridge University Press (traduction française : Preuves et réfutations. Paris : Hermann, 1985).

[LUE 93] Luengo V., 1993. Contraintes Informatiques et apprentissage de la démonstration à propos de trois logiciels. Mémoire du DEA de Didactique des disciplines Scientifiques. Grenoble.

[LUE 97a] Luengo V., 1997. Un micromonde de preuve intégrant la réfutation : Cabri-Euclide, Actes de Journées EIAO, Cachan mai 1997, pp. 85-97.

[LUE 97b] Luengo V., 1997. Cabri-Euclide : un micromonde de preuve intégrant la réfutation. Principes didactiques et informatiques. Réalisation. Thèse. Grenoble : Université Joseph Fourier, 1997.

[LUE 98] Luengo V., Balacheff N. (1998) Contraintes informatiques et environnements d'apprentissage de la démonstration en mathématiques. Sciences et Technologies Educatives 5(1) 15-45.

[MAC 98] Macrelle M., Desmoulins C., 1998. Macro-definitions, a Basic Component for Interoperability between ILEs at Knowledge level: Application to Geometry, in proceeding of 4th International Conference ITS' 98, San Antonio, Texas, pp. 46-55.

[SCH 85] Schoenfeld A. (1985) Mathematical Problem solving. Orlando : Academic Press.

Vanda Luengo a soutenu sa thèse (1997) dans l'equipe EIAH (Environnement Informatique d'Apprentissage Humain) du laboratoire Leibniz à l'IMAG. Son travail de thèse porte sur la conception et développement un logiciel de preuve en géométrie à partir du logiciel Cabri-géomètre. Elle est ATER en informatique à l'université Joseph Fourier et poursuit des recherches en informatique et didactique centrées sur l'apprentissage de la preuve, la représentation de la connaissance, l'interface homme-machine et les agents rationnels.