Département de formation doctorale en informatique **École doctorale IAE + M**
UFR STMIA

Prise en compte de l'usager enseignant dans la conception des EIAO. Illustration dans *Calques 3D*

THÈSE

présentée et soutenue publiquement le 17 décembre 1999

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1

(spécialité informatique)

par

Nicolas VAN LABEKE

Composition du jury

Président :	Mme Jeanine SOUQUIERES	Professeur à l'Université Nancy 2
Rapporteurs :	M. Jean-François DUFOURD	Professeur à l'ULP - Strasbourg I
	M. Jean-Pierre PEYRIN	Professeur à l'UJF - Grenoble I
	Mme Jeanine SOUQUIERES	Professeur à l'Université Nancy 2
Examinateurs :	Mme Monique GRANDBASTIEN	Professeur à l'UHP - Nancy I
	Mme Josette MORINET-LAMBERT	Maître de Conférence à l'UHP - Nancy I
	M. Bernard PARZYSZ	Professeur à l'IUFM d'Orléans-Tours

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thloria.

Remerciements

Je tiens tout d'abord à remercier Jean-François Dufourd, Bernard Parzysz, Jean-Pierre Peyrin et Jeanine Souquieres, membres du jury, pour l'intérêt qu'ils ont porté à ce travail et pour leurs remarques et commentaires. Un grand merci à Monique Grandbastien et à Josette Morinet-lambert pour m'avoir encadré (et supporté !) tout au long de ces années de thèse.

Celle-ci fut un vrai travail d'équipe et je voudrais une fois encore saluer et remercier toutes les personnes qui y ont collaborée : Jean-Claude Demoly, Christian Ginet, Jean-Pierre Giorgi, Norbert Gress, Françoise Jean, Philippe Lombard, Christine Manciaux, Marie-Hélène Munier, Bernard Parzysz, Bernard Pierrot, Serge Remy, Michel Samotyj, Michèle Thiry, Agnès Volpi.

Il y a aussi les proches que je tiens à remercier pour leur présence et leur soutient. Une pensée en particulier à ma mère, pour sa relecture du mémoire et la correction des nombreuses fautes qui s'y cachaient (on se demande bien comment !), et à ma soeur (et surtout à son inspecteur d'académie ! !).

Je n'oublie[0] pas non plus les *bisontins* (expatriés ou non) : Astrid, Catherine, Cécile, Delphine, Didier, Fabrice, Franck, Franck, Franck, Franck, Gilles, Hervé, Joël, Laura, Laurence, Nelly, Nicolas, Patricia et Séverine. Ni les *nancéiéns* (expatriés ou non) : Bernard, Denis, Etienne, Florent, Fred, Nolwen, Pascal, Pierre, Sophie, Thomas et Véro.

Enfin, comme d'autres l'ont fait avant moi et d'autres le feront certainement après, je tiens à offrir une tournée toute spéciale pour les participants aux HIBS, qui ont fait que bon nombre de soirées furent *"animées"*: Angelica, Anne-Sophie, Arnaud, Bertrand, Bruno, Carlos, Eric, Evelyne, Franck, Gina, Jean-Luc, Laurent, Lulu, Manu, Manu, Nico, Nico, Olivier, Olivier, La Peg, Philippe, Pierre, Raphaël et Yann.

Pour finir, je voudrais exprimer toute ma sympathie et ma reconnaissance aux moines de l'abbaye de Westvleteren et leur poser une question, bête mais essentielle : *"Vous n'auriez pas, par hasard, une annexe aux environs de Nottingham ?*

0. En fait, j'ai certainement oublié des gens dans ces remerciements. Qu'ils me le pardonnent ! !

Je dédie cette thèse
à Philippe Bernat

Les quelques passionnés d'informatique qui ont suivi la voie choisie par Philippe Bernat [...] savent que l'essentiel de leur quête n'est rien d'autre que d'avancer sur le chemin de cette "connaissance" *qui, pour eux comme pour beaucoup d'entre nous, s'appelle parfois* "culture", "progrès", "science", "mathématiques", "géométrie", *... mais qui, quel que soit le nom sous lequel on cherche à la délimiter, motive les heures passées à son amélioration et à son enseignement.*
Ils savent que, contrairement aux sciences occultes comme l'alchimie, qui se piquaient d'ésotérisme, d'élitisme et de secret, le moteur du progrès est précisément ici dans la transmission des connaissances et des méthodes. Celui qui cherche à plier la machine à son savoir-faire pour lui donner un semblant d'esprit ne cherche rien d'autre qu'à comprendre lui-même, en définitive, les arcanes de la pensée, afin de pouvoir transmettre aux autres ce qui pourrait le mieux les aider à comprendre et à découvrir à leur tour. [...] Autodidacte de l'informatique, Philippe Bernat a réussi dans cette quête, car il a su satisfaire sa passion et nous laisser les traces d'un travail créateur que je souhaite à chacun de pouvoir apprécier à sa juste valeur. D'autres - ses fils peut-être - ne manqueront pas, je l'espère, d'en tirer profit ; ils apporteront à leur tour leur pierre à l'édifice, tout en sachant que celui-ci, malheureusement, ne saurait jamais être complètement achevé, mais que sa construction ne saurait encore moins s'interrompre, poursuivie sans cesse des parents aux enfants, des maîtres aux élèves ...
Parce que - tout comme celles que nous suivons avec attachement du regard - les vagues ne meurent jamais... Qu'elles hésitent et se fragmentent contre les rochers, qu'elles se brisent en ricochant sur les falaises, qu'elles semblent arrêter leur course en déferlant par dessus les jetés ou qu'elles s'étirent lentement, comme épuisées, sur le sable de la plage ; toujours celles que nous croyons voir disparaître ne sont qu'une infime parcelle des vraies qui les animent et les dirigent... De celles qui, par delà les brisants et les caps bordant notre vue, poursuivent inlassablement leur course, au delà des jetées, au delà des falaises ou des continents, au delà de l'horizon, ... là-bas, au loin ... A jamais.

Philippe Lombard
Repère IREM n°28
Juillet 1997

Table des matières

Table des figures

Introduction

Le domaine de recherche

Le travail de recherche présenté dans cette thèse s'inscrit dans la problématique de la conception des *Environnements Interactifs d'Apprentissage avec Ordinateur* (EIAO) et s'appuie sur le constat suivant : s'il est vrai que de nombreux EIAO sont développés dans les différents laboratoires, force est de constater que peu d'entre eux les quittent pour une utilisation effective dans les classes.

Outre les raisons liées à la nature expérimentale des approches ou des architectures mises en œuvre dans certains de ces prototypes de recherche, nous pouvons cependant apporter d'autres explications. Parmi celles-ci, on remarque une certaine inadéquation des contenus embarqués dans les logiciels pédagogiques, vis-à-vis des programmes d'enseignement et leurs faibles possibilités d'adaptation à la pédagogie personnelle des enseignants. Cela s'explique par une approche technique d'informaticien, plus axée vers la programmation que vers la pédagogie et la didactique, et c'est la source de déconvenues lors de la mise en place d'actions favorisant l'intégration des EIAO dans les pratiques pédagogiques des enseignants (cf. par exemple le plan *"Informatique Pour Tous"*, les écoles d'été inter-IREM, ...).

Nous pensons, comme de nombreux auteurs l'ont fait remarquer, qu'il est nécessaire de prendre en compte les enseignants dès la conception d'un EIAO afin d'obtenir leur adhésion.

Pour cela, deux approches sont possibles. La première consiste à donner un rôle de concepteur et de réalisateur aux enseignants et de leur laisser développer eux-même un EIAO. Cela demande du temps et des compétences techniques que les enseignants doivent acquérir. L'émergence des systèmes d'aides à la conception de logiciels pédagogiques (langages ou systèmes auteurs, Ateliers de Génie Éducatif, ...) permet, dans une certaine mesure, de réduire ces difficultés. Cette première approche ne fait cependant pas l'objet de cette thèse.

La deuxième approche consiste à favoriser l'intégration des enseignants dans une équipe de production pluridisciplinaire et de créer des logiciels pédagogiques ouverts, paramétrables par l'enseignant qui les utilisera selon ses propres besoins. Reste à définir comment permettre ce paramétrage : sur quoi doit-il porter (*Quelle est la demande ?*), sur quoi peut-il raisonnablement porter (*Est-ce techniquement possible ?*), comment le mettre en œuvre, ...

Si les nouvelles technologies, avec les avancées dans les méthodes et concepts de développement et l'évolution de la micro-informatique sont maîtrisées par le concepteur informaticien, leur application dans le cadre d'activités pédagogiques relève de l'imagination et des compétences de transposition des enseignants. Le concepteur ne peut plus anticiper les besoins à partir d'un cahier des charges, sinon il court le risque de développer un produit trop technologique, sans réalité didactique, offrant pléthore de fonctionnalités par rapport aux objectifs de la formation.

C'est pourquoi, nous pensons que cette intégration ne doit pas se limiter à une prise en compte de l'enseignant en tant que simple utilisateur, c'est-à-dire prescripteur (adaptation, confi-

guration, ...) mais aussi et SURTOUT en tant qu'auteur dès la conception d'un EIAO (choix et définition des concepts, de leurs modes de présentation, définition des situations d'apprentissage, ...). C'est cette prise en compte de l'*enseignant auteur* que nous allons défendre dans cette thèse.

Le domaine d'application

Nous nous proposons d'aborder cette problématique à partir de la réalisation d'un micromonde pour l'enseignement de la géométrie dans l'espace. Le choix du domaine d'application de ce travail se justifie par les raisons suivantes.

Premièrement, les travaux antérieurs de l'équipe *Informatique et Formation*, en particulier ceux de Philippe Bernat concernant l'enseignement de la géométrie dans le plan (avec les environnements d'apprentissage `Calques 2` et `Chypre`) et dans l'espace (avec `Dessiner l'Espace`), nous permettent de disposer d'une expérience dans le domaine de la conception d'environnements d'apprentissage et d'une expertise dans le domaine de l'enseignement de la géométrie.

Deuxièmement, la géométrie dans l'espace est un domaine qui pose de nombreux problèmes d'enseignement et où l'outil informatique peut indéniablement apporter un plus. Les principales difficultés sont de nature visuelle et reposent sur la perte d'une dimension lors de la projection d'une figure tridimensionnelle sur une surface plane (la feuille de papier). Un environnement d'apprentissage avec ordinateur permettant la libre exploration d'une figure géométrique peut fournir une aide à l'élève dans ses activités de lecture et d'interprétation de la figure. Cela nous a amené à envisager le développement d'un micromonde de géométrie dynamique et à porter notre attention sur une interface propre à supporter cette libre exploration.

Troisièmement, la géométrie dans l'espace est un domaine dans lequel il n'y a pas de consensus des enseignants sur les connaissances à mettre en œuvre et sur leurs modes de présentation. Face aux difficultés inhérentes à ce domaine d'apprentissage, il n'existe pas de solution unique et optimale à laquelle les enseignants puissent adhérer : chacun adapte selon ses propres besoins les méthodes et outils issus de son expérience ou de "conventions culturelles". De ce fait, cela permet de disposer d'un terrain d'expérimentation idéal pour mettre en évidence le besoin d'une négociation des contenus pédagogiques et d'une paramétrisation d'un EIAO selon ces besoins.

Les objectifs du travail

Notre objectif principal est de développer un micromonde pour l'enseignement de la géométrie dans l'espace, micromonde appelé ***Calques 3D*** dans la suite de ce mémoire. Ce logiciel doit permet un accès intuitif et modulable à ses fonctionnalités : *intuitif* car il doit pouvoir être mis entre les mains d'élèves sans nécessiter un apprentissage long ; *modulable*, car il doit permettre à l'enseignant prescripteur de sélectionner les concepts et primitives qui resteront accessibles à l'élève, en fonction de la pédagogie pratiquée dans sa classe. Les objectifs pédagogiques de ***Calques 3D*** sont triples :

1. **Observation** : permettre à l'élève de voir et de comprendre la troisième dimension en changeant le référentiel (repère orthonormé, plancher, cloisons, ...) ou la perspective (cavalière, point de fuite, ...), en modifiant le point de vue de l'observateur, en affichant des rétroactions visuelles sur les objets, ...
2. **Construction** : permettre la construction dynamique de figures géométriques à partir d'objets élémentaires (points, lignes, plans, ...) et de primitives de construction (intersection, parallèle, perpendiculaire, ...).

3. **Exploration** : permettre à l'élève d'explorer et de découvrir les propriétés géométriques de la figure (déformation de la figure en déplaçant directement les points-base, extraction d'éléments de la construction dans des calques séparés, ...).

Pour ce faire, nous avons mis l'accent sur les trois axes de recherche suivants.

La conception d'un environnement d'apprentissage de type micromonde nécessite la *réalisation d'une interface* centrée sur l'utilisateur (l'apprenant et l'enseignant) et leurs activités respectives (apprentissage et paramétrisation).

L'intégration des enseignants dans le processus de conception doit favoriser la mise en place d'une méthodologie pour l'*acquisition de leur expertise pédagogique* et le développement d'outils pour la *négociation sur la sélection de ces connaissances* et leurs modes de présentation à l'interface.

Enfin, nous nous sommes intéressés à *l'organisation et la représentation des connaissances* ainsi négociées afin de permettre leur implantation et de garantir l'évolutivité du logiciel.

Le cadre du travail

La conduite d'un tel projet suppose la mise en relation de participants experts dans les domaines suivants, relatifs aux EIAO : développement d'outils et de techniques informatiques, analyse des techniques d'enseignement d'une discipline (didactique), mise en œuvre des outils dans le processus d'enseignement (sciences de l'éducation), ... Cette mise en relation est souvent difficile à réaliser : il faut pouvoir trouver des partenaires disponibles, établir des objectifs disciplinaires parfois contradictoires, planifier et diriger la collaboration de manière à aboutir à un système qui satisfasse les différents participants.

Notre travail au sein de l'équipe *Informatique et formation* du LORIA a permis de réunir une partie des conditions autour d'une équipe d'enseignants-chercheurs en informatique. Celle-ci est composée de Monique Grandbastien, Philippe Bernat et Josette Morinet-Lambert.

Nous avons associé à cette équipe permanente d'informaticiens plusieurs enseignants et didacticiens motivés, de manière à bénéficier de leur savoir-faire, de leurs compétences et de leur expérience en termes d'enseignement de la géométrie : Jean-Claude Demoly, Christian Ginet, Jean-Pierre Giorgi, Norbert Gress, Françoise Jean, Philippe Lombard, Christine Manciaux, Marie-Hélène Munier, Bernard Parzysz, Bernard Pierrot, Michel Samotyj, Michèle Thiry, Agnès Volpi. La plupart d'entre eux sont membres de groupes de recherche sur l'enseignement de la géométrie (MAFPEN[1], IREM[2], APMEP[3], ...). Ils ont participé au projet en deux groupes de travail : le premier pour l'intégration du logiciel dans l'enseignement général de la géométrie, le deuxième pour son intégration dans l'enseignement technique. Cette participation s'est effectuée principalement sur leur temps libre, car nous n'avons pas pu disposer de décharges horaires suffisantes pour couvrir la totalité des réunions et des expérimentations. Cela illustre un problème récurrent dans les projets de recherche pluridisciplinaire : le manque de moyens et de reconnaissance institutionnels pour formaliser ces collaborations et leur garantir efficacité et durée.

1. Mission Académique à la Formation des Personnels de l'Education Nationale.
2. Institut de Recherche sur l'Enseignement des Mathématiques.
3. Association des Professeurs de Mathématiques de l'Enseignement Public.

Contenu du mémoire

Le premier chapitre de ce mémoire est consacré aux questions plus générales qui se sont posées sur la conception d'un Environnement Interactif d'Apprentissage Avec Ordinateur. Nous insistons en particulier sur la question de l'interface qui, au niveau des différents modules de l'architecture des EIAO, est certainement le plus important pour un micromonde car elle conditionne la liberté d'exploration d'un domaine par l'apprenant.

Dans le deuxième chapitre, nous mettons en évidence les principales difficultés de l'enseignement de la géométrie dans l'espace et les points sur lesquels l'introduction de l'ordinateur peut apporter des éléments de solution. A partir de l'analyse de travaux en didactique des mathématiques et des micromondes existants, cette réflexion nous permet de mettre en place un premier cahier des charges "théorique" qu'il convient d'étayer de manière "empirique" par une collaboration avec des experts du domaine.

Le troisième chapitre décrit la méthodologie suivie pour le développement de ***Calques 3D***, c'est-à-dire un processus basé sur le prototypage incrémental et l'intégration des enseignants auteurs au centre même de la conception du logiciel.

Le quatrième chapitre décrit le formalisme que nous utilisons pour permettre l'acquisition de l'expertise pédagogique des enseignants, l'organisation des connaissances en vue d'une implantation et la négociation des besoins entre les participants au processus de conception.

Le cinquième chapitre décrit, du point de vue des utilisateurs élève et enseignant, les caractéristiques principales de ***Calques 3D***. Nous y présentons l'interface d'accès et les fonctionnalités mises en œuvre pour atteindre les objectifs pédagogiques attribués au logiciel. La dernière partie de ce chapitre est consacré à la paramétrisation du logiciel par l'enseignant prescripteur.

Le sixième chapitre, plus technique, est consacré à l'implantation de ***Calques 3D***. Nous présentons les choix conceptuels retenus pour la représentation interne des figures géométriques et la gestion de l'interaction et de la dynamique au niveau de l'interface. Nous élaborons un bilan sur le prototype réalisé et les problèmes informatiques relatifs à ce développement.

Nous présentons en conclusion de cette thèse les moyens mis en œuvre pour expérimenter et valoriser le produit auprès du public enseignant, ainsi que les réactions obtenues. Nous mettons en évidence les apports de ce travail et abordons les perspectives de recherche qu'il permet d'envisager.

Première partie

État de l'art

Chapitre 1

Conception et réalisation des logiciels d'apprentissage de type micromonde

1.1 Introduction

Ce chapitre a pour objectif de mettre en évidence la problématique de la conception des *Environnements Interactifs d'Apprentissage avec Ordinateur* (EIAO), en particulier celui des micromondes. Après avoir brièvement présenté les grands paradigmes des EIAO (section 1.2) et explicité le concept de micromonde (section 1.3), nous abordons les deux points que nous considérons comme essentiels pour la conception d'un environnement d'apprentissage de type micromonde : la conception de l'interface (section 1.4) et la question des aides et rétroactions (section 1.5).

Nous concluons ce chapitre par l'énoncé de critères à prendre en compte pour la conception d'un micromonde.

1.2 Des EAO aux EIAO

L'évolution des ordinateurs et la vulgarisation de leur usage ont permis leur introduction dans la plupart des activités humaines, y compris dans l'enseignement.

Bon nombre des premiers logiciels pédagogiques, développés dans le cadre de l'*Enseignement Assisté par Ordinateur (EAO)*, se sont contentés de "traduire" sous une forme informatique les supports traditionnels de l'enseignement : cours, recueils d'exercices, ... La rigidité des premiers *tutoriels*, comportant des parcours séquentiels uniques ou ramifiés, des bases d'exercices figées et des capacités de résolution réduites (comparaison des réponses de l'apprenant à des réponses pré-enregistrées) , ... a suscité plus de critiques que de satisfactions.

Certaines des insuffisances de ces systèmes peuvent être identifiées par leur incapacité à résoudre les problèmes posés à l'apprenant, leur faible degré d'adaptation à celui-ci, leur faible identification et maîtrise des connaissances du domaine et de l'expertise pédagogique, ... Une critique plus détaillée est effectuée par [Self 85].

L'apparition des techniques d'*intelligence artificielle*, en particulier l'émergence des *systèmes experts* a donné l'espoir d'apporter une réponse aux critiques des EAO, notamment par la capacité de résolution de problèmes. Le domaine s'est donc recentré autour du thème de

l'*Enseignement Intelligemment Assisté par Ordinateur (EIAO)*, dont le paradigme dominant a été sans conteste les *tuteurs intelligents* (*Intelligent Tutoring System*).

La notion de *tuteur intelligent* est centrée sur :

- une représentation explicite des connaissances du domaine et des mécanismes de raisonnement (dotant ainsi le système de la capacité de répondre à des questions et de résoudre des exercices dont la solution n'a pas été explicitement prévue),
- sur la primauté de l'apprentissage individuel (dotant explicitement le système d'informations sur le degré de maîtrise de l'apprenant, permettant ainsi une adaptation dynamique à son interlocuteur),
- sur la référence à un tuteur humain pour modéliser l'interaction entre l'apprenant et le système (dotant le système, à partir d'une explicitation des stratégies tutorielles, de la capacité d'engendrer dynamiquement ses interventions).

Pour de plus amples détails sur les tuteurs intelligents, le lecteur pourra se reporter à [Wenger 87], [Nicaud 88] et [Quéré 91].

L'architecture d'un tuteur intelligent s'appuie généralement sur quatre modules travaillant en coopération dans un objectif d'apprentissage [Nicaud 88] :

- Le *module du domaine d'apprentissage* est le dépositaire des connaissances de l'expert et sait résoudre les problèmes liés au domaine. Il est souvent élaboré sur le modèle des systèmes experts.
- Le *module de l'élève* contient les informations relatives à l'état des connaissances, exactes ou erronées, de l'élève. Sa construction s'établit principalement par comparaisons avec les connaissances de l'expert.
- Le *module du pédagogue* contient les stratégies d'enseignement parmi lesquelles le système devra choisir la plus adaptée à la situation de l'apprenant. Son élaboration est rarement effectuée de manière autonome, une partie des connaissances pédagogiques (ou didactiques) se retrouvant dans le modèle du domaine et dans la gestion de l'interaction.
- Le *module de l'interface* définit les interactions possibles entre l'apprenant et le système.

Ce quatrième module n'a cessé de prendre de l'importance dans la conception des logiciels éducatifs, au fur et à mesure que se développait l'usage des micro-ordinateurs et des stations de travail individuel. L'évolution rapide des matériels, logiciels et interfaces de communication homme-machine permettait d'envisager une interactivité système-utilisateur croissante, entraînant l'émergence d'une nouvelle voie dans le développement des logiciels éducatifs : une voie centrée sur l'apprentissage comme résultat de l'expérience de l'apprenant et de sa libre interaction avec le système et non plus sur le désir de reproduire dans un système informatique le comportement de l'enseignant face à l'élève. Ainsi, [Balacheff 94a] affirme que l'enjeu de l'intelligence artificielle dans le champ de la didactique n'est pas que l'ordinateur se comporte comme un enseignant mais qu'il soit capable de créer des conditions favorables à la construction par l'apprenant de connaissances en référence à un objet d'enseignement, et ceci en lui assurant des *interactions et rétroactions pertinentes*.

Cette mise en avant de l'*interactivité* a amené [Baron 91] à justifier une nouvelle interprétation du sigle EIAO par *Environnements Interactifs d'Apprentissage avec Ordinateur*, permettant ainsi de regrouper sous un même sigle des approches autres que celles des tuteurs intelligents

au sens strict, avec d'autres objectifs en termes de situations d'apprentissage et d'activités proposées aux apprenants : c'est le cas des environnements d'exploration ou de découverte plus ou moins guidée dont le *micromonde* est le représentant le plus connu.

1.3 Les micromondes : influence de l'utilisateur

La notion de micromonde a été mise en avant par S. Papert avec le célèbre langage LOGO et avec un point de vue *constructiviste* affirmé, c'est-à-dire un apprentissage basé sur la construction du savoir par l'élève et non pas sur sa transmission par l'enseignant [Papert 81] [Papert 87] [Lawler 84]. Le micromonde développé sur ordinateur est vu comme un objet de transition qui amène l'apprenant à découvrir et à explorer la connaissance à travers une interaction avec les objets du logiciel, lui permettant ainsi de progresser vers des concepts plus abstraits.

D'une manière générale, et par opposition aux tuteurs intelligents, les principes qui caractérisent les micromondes sont :

1. *Construction et non instruction.* L'élève apprend de manière plus efficace lorsqu'il construit son propre savoir par lui-même *en agissant*, au lieu de se le voir enseigné à travers des cours ou un système qui se contente de transmettre des connaissances.
2. *Contrôle par l'apprenant et non contrôle par le tuteur.* L'enchaînement des actions est déterminé par l'apprenant. Cela implique que l'apprenant doit disposer d'un contrôle significatif, sinon exclusif, de l'interaction d'apprentissage. Tout au plus, le système est pris comme un guide et non comme un tuteur.
3. *Individualisation de l'apprentissage déterminée par l'apprenant et non par le tuteur.* Les micromondes, comme les tuteurs intelligents, mettent en avant le fait que l'individualisation des retours d'informations (*rétroactions*, ou *"feedbacks"*) sont la clef de l'apprentissage. Cependant, ils différent quant à l'origine de cette adaptation à l'apprenant. Alors que les tuteurs intelligents sont responsables de l'adaptation des informations qu'ils fournissent à l'apprenant, celle-ci est généralement perçue dans les micromondes comme une fonction de l'interaction entre apprenant et environnement. Le mécanisme d'individualisation peut varier substantiellement entre différents micromondes mais l'adaptation est souvent sous le contrôle au moins partiel de l'apprenant.

Grossièrement parlant, nous pouvons dire que l'*"intelligence"* investie dans un micromonde est distribuée dans l'ensemble des outils qu'il fournit et non centralisée dans un module tutoriel.

La caractéristique la plus importante d'un micromonde réside dans le fait qu'il offre un espace d'exploration à l'élève. Ce dernier n'est pas soumis à un examen rigide, à une évaluation sanctionnée par un enseignant (ou un tuteur intelligent) mais peut prendre le temps d'essayer diverses solutions, de commettre des erreurs, d'aller de l'avant et de revenir sur ses productions. Selon Papert, un micromonde fournit ainsi un espace où l'apprenant peut exercer sa créativité *librement*.

Cette liberté d'usage et la confiance qu'elle procure à l'apprenant, nécessitent d'être prises en compte dans la conception d'un micromonde.

1.4 L'interface d'un micromonde

Bien que les premières définitions des micromondes ne comportaient pas explicitement de liens avec les principes d'interface homme-machine (IHM), ce facteur a pris rapidement une importance telle que micromonde et interface vont généralement de pair. Afin de faciliter l'exploration libre d'un domaine d'apprentissage par l'utilisateur, l'interface d'un micromonde doit

se référer autant que possible à un univers familier de ce dernier, que ce soit au niveau de la convivialité ou des possibilités d'adaptation et de personnalisation.

Dans le contexte de la conception de `Cabri 3D`, [Qasem 97] procède à une analyse approfondie de la question de l'interface d'un micromonde pour l'enseignement de la géométrie dans l'espace, question que nous étudions ci-après.

1.4.1 Manipulation directe et engagement direct

Dans le domaine des IHM, les interfaces peuvent se regrouper en deux catégories, selon la métaphore utilisée pour représenter le monde de l'utilisateur [Nanard 90] : la métaphore dite *du monde abstrait ou de la description* et la métaphore dite *du monde réel*.

Dans le cas d'une interface fondée sur la *métaphore du monde abstrait*, l'utilisateur décrit les actions qu'il veut réaliser et les objets qui vont subir l'action, mais n'agit pas directement sur les objets. Cette description se fait de manière syntaxique (en langue naturelle ou selon un lexique réduit) et les objets sont désignés symboliquement, par leur nom par exemple. C'est le cas des interfaces par ligne de commandes des systèmes d'exploitation UNIX ou DOS (l'action de copie d'un fichier par exemple porte sur le nom du fichier et non sur le fichier lui-même) ou d'un micromonde comme LOGO (exploration du monde par des commandes textuelles).

Les interfaces fondées sur la *métaphore du monde réel* ont pris leur essor avec l'introduction d'interface graphique dans les ordinateurs (Star de Rank Xerox, LISA d'Apple ...). Elles sont constituées d'une représentation du monde de l'utilisateur, faisant directement référence aux objets du monde physique. L'utilisateur ne fait pas référence aux objets mais leur applique directement les actions. Une action de l'utilisateur sur l'interface provoque une exécution immédiate de l'action, et un changement de la représentation pour rendre compte du nouvel état du monde. Par exemple, l'action de copie d'un fichier s'effectue en déplaçant l'icône représentant le fichier d'un emplacement à un autre. C'est ce que l'on appelle aussi *interface de manipulation directe* [Schneiderman 82]. B. Schneiderman caractérise ces interfaces par trois principes :

- la présentation permanente des objets, qui permet une identification immédiate entre l'objet et sa représentation visible,
- l'utilisation des actions physiques directes au lieu de commandes à la syntaxe compliquée,
- l'utilisation d'opérations rapides, incrémentales et réversibles dont l'impact sur les objets est immédiatement perceptible.

Ces critères permettent d'évaluer une interface et de la caractériser comme relevant de la manipulation directe ou non mais, en aucun cas, il ne s'agit d'une définition stricte : la manipulation directe reste un concept complexe qui ne possède pas de définition unitaire. C'est pour cette raison que [Nanard 90] introduit la notion d'*engagement direct*, condition à obtenir pour qu'une interface puisse être considérée comme telle.

L'impression d'*engagement direct* correspond à ce que l'usager ressent lorsqu'il peut agir directement et librement sur les représentations des objets et percevoir de façon immédiate leur réaction. Cette impression est obtenue lorsque l'interface est suffisamment transparente pour faire oublier à l'utilisateur son statut de simple intermédiaire entre lui-même et les objets qu'il manipule, favorisant ainsi l'immersion de l'apprenant dans le domaine d'apprentissage. En d'autres termes, elle doit cesser d'être une *interface* pour devenir un *espace* où l'apprenant doit avoir le sentiment d'agir sur les objets mêmes et de percevoir leurs réactions.

La facilité d'utilisation d'un environnement dépend fortement de la capacité de l'utilisateur à exprimer sans difficultés ses idées dans le formalisme imposé par l'interface et de sa capacité à

interpréter et comprendre les résultats introduits par le système. Cela signifie que la construction d'une interface doit se baser sur une analyse du type d'utilisateurs visés et de la nature de l'utilisation qui sera faite. Si les micromondes, à la différence des tuteurs intelligents, ne disposent pas d'un modèle de l'apprenant explicite, cela ne signifie pas que l'utilisateur n'est pas pris en compte dans ces environnements : cette prise en compte est en fait effectuée au moment de la conception de l'interface et se traduit par certains choix d'implantation au niveau des métaphores d'interaction, des modes de présentation des objets du domaine ou des actions disponibles. C'est ce qui est désigné, dans le domaine des interfaces homme-machine, sous le terme de conception centrée sur l'utilisateur (*User-Centered Design*, [Preece 94]).

Dans cette optique, il nous semble que trois objectifs doivent être atteints : obtenir une conformité d'usage en assurant une continuité entre le domaine d'apprentissage et sa représentation à l'interface, assurer la simplicité de l'utilisation du logiciel (et donc de l'interface) et respecter l'environnement d'usage du logiciel.

1.4.2 Conformité d'usage

La conception d'une interface, partie visible de l'environnement, doit permettre à l'utilisateur de s'approprier aisément les concepts du domaine d'apprentissage sous-jacent : la confrontation entre les réactions attendues par l'utilisateur et le comportement effectif de l'environnement peut amener l'utilisateur non seulement à apprendre à mieux se servir du système mais aussi à acquérir de nouvelles connaissances dans le domaine d'apprentissage. Cette appropriation passe par l'élaboration, lors de l'interaction entre le système et l'utilisateur, d'un *modèle conceptuel* [Norman 86], c'est-à-dire une représentation mentale constituée par l'ensemble des concepts que l'utilisateur met en place pour s'expliquer le comportement du système. Pour que l'apprentissage soit effectif, il faut donc que ce modèle conceptuel soit suffisamment compatible avec un modèle issu de l'apprentissage traditionnel. Cette conformité d'usage, condition nécessaire pour obtenir l'*acceptabilité* du logiciel, se traduit à plusieurs niveaux.

Au niveau des commandes disponibles, l'utilisateur doit pouvoir non seulement retrouver des commandes lui permettant de mettre en œuvre des activités liées au domaine d'apprentissage mais aussi les métaphores. Ainsi, dans le cas de l'enseignement de la géométrie dans l'espace, les concepts supportés par le système sont bien entendu les concepts géométriques (objets géométriques, propriétés et relations, ...) mais ils doivent avoir effectivement un lien avec le domaine d'utilisation du système qui est l'exploration et l'apprentissage. Par exemple, des commandes permettant la construction de polyèdres, la définition de sections planes de solide ou leur visualisation en projection frontale ont bien une place dans un environnement d'apprentissage de la géométrie mais d'autres, comme l'extrusion de solides, sont d'un intérêt pédagogique moins évident car plus représentatifs d'un modeleur 3D ou d'un éditeur de CAO.

Au niveau de la représentation du domaine, celle-ci doit être suffisamment claire et évocatrice de manière à permettre une correspondance, voire une analogie, entre objets manipulés à l'interface et objets du domaine qu'ils représentent [Nanard 90]. Cette analogie doit s'exprimer bien évidemment au niveau graphique (les représentations graphiques des objets doivent correspondre aux images traditionnelles), mais aussi au niveau lexical : les termes utilisés pour nommer les commandes ou les objets, pour décrire les propriétés de ceux-ci, ... doivent être identiques et évocateurs des termes utilisés habituellement par l'utilisateur.

Enfin, le comportement des objets à l'interface doit être cohérent par rapport à la théorie du domaine. Dans un environnement d'apprentissage de la géométrie, l'utilisateur est amené à construire des figures géométriques dans l'espace puis à les observer et à les explorer pour en découvrir les propriétés. A chaque étape d'une activité, l'utilisateur se base sur les réactions

du système pour analyser ses productions. Le comportement du système peut être par moment contradictoire avec les attentes de l'utilisateur : cela doit traduire la présence d'erreurs dans la construction de la figure ou des attentes erronées de la part de l'utilisateur mais, en aucun cas, un comportement du système différent de la théorie.

D'une manière plus générale, les connaissances embarquées dans un système doivent pouvoir être adaptées aux connaissances de l'apprenant, en particulier en prenant en compte leur évolution au fur et à mesure de l'apprentissage. Ce critère d'*adaptabilité* est important pour les enseignants car, à défaut de concevoir un environnement dédié à une situation d'apprentissage bien précise, la pluralité des situations possibles qu'un environnement peut supporter impose de pouvoir adapter l'environnement. Dans un micromonde, les connaissances sont difficilement modifiables (car généralement implantées "statiquement" dans le logiciel). L'enseignant ne "rajoute" donc rien au micromonde, sauf des activités à réaliser autour. L'adaptation à ses besoins s'effectue principalement, au niveau de l'interface, par *sélection* des connaissances embarquées. Cela nécessite une paramétrisation du logiciel par l'enseignant et, par conséquent, l'identification lors de la conception des éléments à paramétrer.

1.4.3 Simplicité d'usage

L'objectif d'un environnement d'apprentissage est l'apprentissage d'un domaine et non l'apprentissage d'un environnement. Cette affirmation peut sembler évidente mais force est de constater que de nombreux logiciels pédagogiques ne trouvent pas preneur auprès des utilisateurs (élèves ou enseignants) à cause d'une interface trop difficile ou ambiguë. L'utilisation de métaphores appropriées peut permettre de garantir l'*utilisabilité* d'un environnement d'apprentissage. Nous ne parlerons pas ici des éléments classiques d'une interface graphique (comme les menus déroulants, les icônes, les fenêtres déplaçables et modifiables, ...) dont l'utilité, mais aussi les problèmes inhérents, ne sont plus à démontrer (cf. [Coutaz 88]), mais de métaphores pouvant être considérées comme propres à un micromonde de géométrie dans l'espace.

a) Désignation directe

Le principe de *"Voir et montrer"* a été à l'origine défini pour caractériser l'utilisation de boutons cliquables (*icônes*) pour appeler une commande, au lieu d'utiliser une ligne de commande (*"Se souvenir et frapper au clavier"*). Avec l'évolution des interfaces graphiques, le même principe peut être étendu aux objets du domaine. Dans un environnement comme `Géospace` [Authier 98] par exemple, la réalisation d'une commande de construction invite l'utilisateur à choisir les arguments dans une liste affichée dans une boîte de dialogue. Cette méthode est commode pour l'introduction des arguments au clavier mais elle est contradictoire avec la manipulation directe : au lieu de montrer les objets eux-mêmes, l'utilisateur utilise leur nom, ce qui l'oblige d'une part à leur donner obligatoirement un nom, d'autre part à se souvenir de la correspondance entre noms et objets. L'utilisation du principe de *"Pointer et créer"*, c'est-à-dire la désignation des arguments d'une commande directement à l'interface grâce à la souris permet une réelle manipulation directe : c'est ce que nous appelons *désignation directe*. Pour ce faire, les objets doivent non seulement avoir une représentation graphique visible en permanence à l'interface, mais cette représentation doit aussi être une entité active que l'utilisateur peut pointer directement et qui doit réagir à cette action : la représentation graphique d'un objet ne doit pas être réduite au simple statut d'un dessin.

Ce principe de la désignation directe nécessite cependant une réflexion au niveau de sa mise en œuvre. En effet, des micromondes comme `Cabri Géomètre` [Baulac 92] ou `Calques 2`

[Bernat 94a] disposent de commandes différentes pour construire des points libres ou appartenant à un ou deux objets. La construction de l'intersection de deux droites par exemple nécessite d'activer la commande *"Intersection de deux droites"* puis de désigner à la souris les deux objets dont l'utilisateur veut construire l'intersection. Cette opération est différente de celle réalisée dans un environnement papier-crayon, où il suffit de dessiner un point à l'intersection désirée. C'est pour cela que `Cabri II` [Laborde 94] met en œuvre un tel mécanisme, plus proche des habitudes de l'utilisateur et permettant de prendre en compte son intention : en activant la commande *"Construire un point"*, l'utilisateur peut construire indifféremment un point libre (en cliquant dans le plan), un point à l'intersection de deux objets (en cliquant à l'intersection visuelle des deux objets), ... Cependant, l'implémentation de ce type d'interaction dépend de la nature des représentations utilisées et de leur degré de suggestivité : si en géométrie plane, les positions des objets suggèrent effectivement l'existence d'une intersection, il n'en est pas de même dans l'espace où une intersection peut être perçue "visuellement" mais sans avoir d'existence réelle au niveau de la figure.

b) Stratégie opérationnelle des commandes

La désignation directe des objets du domaine à la souris a suscité une discussion sur les stratégies de mise en œuvre des commandes, c'est-à-dire sur l'ordre à établir entre la commande (l'action) et les objets : soit la stratégie dite *"objets/action"* (désignation ou sélection des objets puis activation d'une commande), soit la stratégie inverse dite *"action/objets"*. La première stratégie est certainement la plus familière aux utilisateurs car elle est mise en œuvre dans de nombreux logiciels bien connus comme les logiciels de dessin ou de traitement de texte (pour mettre du texte en gras dans `Word` par exemple, l'utilisateur doit d'abord le sélectionner puis activer la commande "gras"). Cette familiarité a conduit certains auteurs de logiciels pédagogiques (comme `Geometer's Sketchpad` [Jackiw 95] dans le domaine de la géométrie plane) à conserver cette stratégie. Cependant, son impact sur l'apprentissage est controversé : en effet, cette stratégie requiert de la part de l'utilisateur un effort cognitif important pour se remémorer la liste des arguments d'une commande et les désigner à l'avance. Aucune aide ne peut être apportée au cours de l'action puisque celle-ci n'est pas connue par le système avant son activation. Cette stratégie réduit donc, surtout pour l'apprenant novice, les possibilités d'exploration d'un micromonde.

La stratégie *"action/objets"* permet au contraire d'offrir un retour d'information propre à aider l'apprenant dans son activité, et ceci en permanence pendant l'exécution de l'action. Dans des micromondes comme `Calques 2` ou `Cabri Géomètre`, lorsqu'une commande est activée, seuls les objets pouvant servir d'arguments à l'action peuvent être sélectionnés et réagissent lorsque la souris s'en approche à une certaine distance, indiquant ainsi l'*espace d'exploration* disponible pour l'action. Cette réaction à l'approche de la souris donne une information sur les arguments possibles et décharge l'utilisateur de l'obligation de s'en souvenir. S'il existe, l'ordre de désignation des objets peut être mis en évidence de la même manière.

c) Concision de l'interface

Souvent citée comme concept clef de la simplicité, l'élimination de la redondance à l'interface, en entrée comme en sortie, est cependant un critère à prendre avec précaution car il ne faut pas réduire les possibilités d'expression. En effet, dans quelle mesure pouvons nous dire que la commande *"Milieu d'un segment"* est redondante avec la commande *"Milieu d'un bipoint"*? De même, plusieurs représentations d'une même information peuvent être nécessaires pour faciliter

une tâche ou permettre l'apprentissage d'un concept. Pendant le déplacement d'un point dans `Cabri 3D` [Qasem 97], la position de l'objet dans l'espace est indiquée par l'affichage de ses coordonnées et par des marques sur les axes du repère orthonormé.

C'est pour cela que [Coutaz 88] préfère parler de minimisation de la redondance ou *concision* de l'interface, plutôt que d'élimination pure et simple. Elle attribue deux effets à la concision : éviter les surcharges d'information de sortie et réduire le nombre d'actions physiques nécessaires à la spécification des expressions d'entrée. Pour le premier point, c'est un travail d'ordre cognitif qu'il s'agit de faire, en trouvant un juste milieu entre informativité et surcharge cognitive : la limitation du nombre de fenêtres disponibles simultanément à l'utilisateur, la position des informations à l'écran, ... sont autant de questions qu'il convient d'aborder consciencieusement et de valider avec les utilisateurs. Pour le deuxième point, il est possible d'introduire des abréviations ou raccourcis permettant l'accès rapide aux commandes les plus fréquemment utilisées. Par exemple, les produits `Cabri Géomètre` utilisent un raccourci-clavier pour appeler la dernière commande mise en œuvre par l'utilisateur, ce qui s'avère utile lorsque plusieurs objets de même type doivent être construits. `Calques 2` préfère utiliser des commandes *cycliques*, c'est-à-dire qui bouclent sur elles-mêmes jusqu'à abandon par l'utilisateur et permettent la construction en série d'un même objet.

Dans le même ordre d'idée, l'interaction non modale est considérée comme restrictive à la notion de manipulation directe car elle limite la capacité d'interaction de l'utilisateur avec le système en le contraignant à agir à travers une boîte de dialogue, et non directement sur les objets, et introduit un goulot d'étranglement dans la libre exploration d'un univers. S'il est vrai que certaines opérations peuvent se passer d'une interaction non modale (par exemple la création de point - voir ci-dessus - ou la modification des vecteurs unitaires d'un repère par manipulation directe plutôt que par la saisie de données numériques), cela peut être plus difficile dans d'autres cas de figure. Un exemple significatif de ce problème est la modification des représentations graphiques des objets. Celle-ci passe souvent par la modification d'un certain nombre d'attributs visuels : choix du mode de tracé de l'objet (ligne continue ou discontinue pour les droites, points sous la forme d'un cercle plein ou vide, ...), de l'épaisseur ou de la couleur du tracé, du nom de l'objet, ... Une boîte de dialogue a l'avantage de permettre une modification de l'ensemble des attributs d'un objet en une seule opération mais rend fastidieuse la modification d'un seul attribut pour plusieurs objets.

1.5 Aides et rétroactions

Dans un tuteur intelligent, les modèles de l'utilisateur et des activités qu'il doit accomplir permettent, plus ou moins aisément, de mettre en évidence et de signaler une erreur commise par l'apprenant. Un micromonde ne possède pas de tels modules : il ne peut donc produire ni messages d'erreur ni indications liées à l'activité en cours. S'il est effectivement impossible de *détecter et analyser* les erreurs de l'utilisateur, il est cependant possible de lui permettre d'en prendre conscience par lui même en lui fournissant des retours d'information pertinents (*rétroactions*). Le retour d'information dans un micromonde est le résultat d'une action réalisée par l'utilisateur, rendant compte de l'état du système après son intervention.

L'utilisateur qui exécute une action quelconque doit pouvoir obtenir un retour d'information immédiat, lui permettant de juger de l'adéquation de son action. Il doit percevoir au niveau de l'interface le changement que son action a produit sur les objets du domaine. Si l'effet produit est différent de l'effet escompté, l'utilisateur peut alors prendre conscience de l'existence d'une erreur et être amené à revoir ses actions. Cette erreur peut devenir une partie de l'expérience de

l'apprenant.

La perception de l'erreur dépend de la nature de la rétroaction donnée par le système et des connaissances de l'utilisateur. Dans un micromonde de géométrie par exemple, lorsque l'utilisateur déplace l'une des deux droites supposées parallèles, l'absence de réactualisation de la seconde peut être considérée comme une erreur dans la construction de la relation de parallélisme entre les deux droites. Quand les objets du système sont moins connus ou appartiennent à un domaine où il existe des difficultés de compréhension comme c'est le cas dans la géométrie dans l'espace, l'interprétation du comportement du système est plus problématique[4].

Cette réalité pose la question de la représentation à l'interface des objets du domaine, qui se doit d'être aussi proche que possible de celle habituellement utilisée par l'utilisateur pour lui permettre d'agir efficacement avec le système mais aussi être assez riche et suggestive pour répondre à des problèmes d'interprétation des résultats de ses actions.

1.6 Conclusion

La conception d'un micromonde nécessite un effort particulier dans le domaine de l'interface, effort certainement plus important que dans n'importe quel autre type de logiciel pédagogique. En effet, l'apprentissage dans un micromonde est entièrement basé sur l'interaction entre le système et l'apprenant, sur les rétroactions pertinentes que ce dernier peut obtenir suite à ses actions.

La conception d'un tel environnement nécessite le respect de trois critères importants :

- l'*acceptabilité* du logiciel par l'enseignant qui souhaite l'utiliser conformément à ses pratiques,
- l'*utilisabilité* du logiciel par l'élève qui garantit une prise en main facile, dans son environnement habituel,
- l'*adaptabilité* du logiciel en fonction du niveau de l'élève.

4. Dans la géométrie dans l'espace, on parle en effet souvent des difficultés des personnes *à voir* dans l'espace

Chapitre 2

Des logiciels pour l'enseignement de la géométrie dans l'espace

2.1 Introduction

Le chapitre précédent nous a permis de mettre en évidence la problématique de la conception d'un environnement d'apprentissage de type *micromonde*, en particulier au niveau de l'interface, et de proposer un ensemble de grands principes et de critères *informatiques* à respecter pour ***Calques 3D***. Ce chapitre a pour objectif d'amener à compléter ces critères d'un point de vue pédagogique, en répondant aux questions suivantes :

- Quels sont les problèmes que pose l'enseignement de la géométrie dans l'espace et auxquels l'ordinateur peut proposer une réponse appropriée (section 2.2) ?
- Quelles sont les différences entre un environnement d'apprentissage de la géométrie et d'autres environnements permettant de manipuler des figures géométriques, comme en CAO par exemple (section 2.3) ?
- Quelles sont les fonctionnalités et propriétés des environnements d'apprentissage existants, (section 2.4) ?
- Quel cahier des charges pour ***Calques 3D*** (section 2.5) ?

2.2 Apports de l'ordinateur à la géométrie dans l'espace

L'enseignement de la géométrie s'effectue principalement à travers des activités qui consistent en l'étude et la résolution de problèmes liés à des *figures géométriques* dans le plan ou dans l'espace. Le *dessin* y est largement utilisé comme moyen privilégié, soit dans un rôle d'illustration ou de représentation de propriétés, soit dans le rôle de support à l'intuition et à l'élaboration d'une démarche de résolution. Des recherches en didactique comme celles présentées dans [Brunet 81], [Pallascio 92] ou [Laborde 96] ont démontré l'intérêt du recours aux représentations visuelles pour aider à l'apprentissage de différents concepts en géométrie.

Nous reprenons ici la distinction proposée par [Parzysz 88] entre la *figure géométrique*, c'est-à-dire l'être géométrique défini par un énoncé, et les *représentations* de cette figure. Ces représentations peuvent être le *dessin* (représentation graphique plane) ou la *maquette* (ou *le modèle*, représentation solide tridimensionnelle).

Il est important d'insister sur le lien qui existe entre ces deux notions, et les difficultés qu'il introduit. Cela tient au fait que les propriétés géométriques de la figure sont représentées par des relations spatiales sur le dessin mais la signification qu'un observateur donnera aux relations spatiales d'un dessin ou d'une maquette ne sera pas nécessairement identique à celle de l'objet géométrique.

En géométrie plane, le dessin est une représentation très voisine de la figure : une identification quasi-immédiate peut se produire entre les objets étudiés et leur représentation sur le papier. En géométrie dans l'espace, cette correspondance est moins immédiate : l'équivalent du dessin est bien la *maquette* (c'est-à-dire une matérialisation physique tridimensionnelle) et non un dessin en deux dimensions. Pour une figure tridimensionnelle, le dessin n'est qu'une projection sur une surface plane de la maquette, celle-ci n'étant qu'une représentation de la figure. La plupart des problèmes que pose l'enseignement de la géométrie dans l'espace viennent de cette distance entre le concept et sa représentation finale, car chaque étape dans le processus de représentation (*réification*) entraîne une perte d'information.

Mettre en œuvre un logiciel pour l'enseignement de la géométrie dans l'espace nécessite de prendre en compte ces difficultés afin de proposer des solutions. Nous les avons déclinées selon les quatre points suivants.

2.2.1 Représentation graphique des figures

Une figure géométrique est dessinée pour être visualisée. Il est donc nécessaire de s'intéresser à ce qu'un observateur est sensé voir dans le dessin d'une figure et à ce qu'il perçoit en regardant ce dessin.

Dans les manuels de géométrie en effet, les dessins sont utilisés pour accompagner un énoncé de problème géométrique, pour illustrer un théorème ou pour présenter des concepts. Ces dessins sont donc construits en prenant en compte le but précis pour lequel ils sont introduits et reflètent totalement l'intention de l'auteur qui est d'évoquer chez le lecteur le plus facilement possible ce but. Le lecteur sait plus ou moins ce qu'il doit chercher à voir dans le dessin, ce qui est de nature à lui faciliter la tâche.

Le choix des représentations graphiques des objets de géométrie dans l'espace est influencé par la volonté de permettre une meilleure perception de la figure. Il n'existe pas de représentation unique des objets géométriques. En consultant des manuels de géométrie et surtout grâce au travail collaboratif avec des enseignants de géométrie pour le développement de ***Calques 3D*** (cf. chapitres suivants), nous avons pu constater que les enseignants utilisent différentes manières pour représenter un même objet géométrique. Chacune de ces différentes représentations prend sa justification soit dans l'avantage qu'elle confère pour illustrer ou mettre en évidence une propriété particulière (par exemple, mise en valeur du parallélisme de deux segments), soit dans des habitudes culturelles qu'il est difficile d'ignorer (par exemple, un plan pouvant être représenté par un parallélogramme ou par un secteur angulaire).

Il est donc important qu'un environnement d'apprentissage de la géométrie dans l'espace apporte aux utilisateurs cette même *variété des représentations visuelles d'une figure*. Le choix de telle ou telle représentation ne peut pas être imposé par le logiciel mais dicté par la situation d'apprentissage. Ainsi, plus qu'à l'*utilisateur élève*, nous pensons que c'est à l'*utilisateur enseignant* de privilégier une représentation particulière dans une situation donnée.

2.2.2 Rôle de la perspective

Toute représentation plane (dite *représentation en perspective*) d'un objet tridimensionnel est l'image obtenue par projection de cet objet sur un plan (que l'on désignera sous le terme de *plan de projection*). Deux types de projections sont à distinguer :

- la *projection centrale* (ou perspective conique), où le centre de projection se trouve à une distance finie du plan de projection. Elle peut être réalisée avec un ou plusieurs centres de projection (points de fuite).
- la *projection parallèle* (ou perspective cylindrique), où le centre de projection est rejeté à l'infini. Elle n'est pas caractérisée par son centre de projection mais par la direction de la projection : on parlera de projection parallèle orthogonale si la direction de projection est normale au plan de projection, de projection parallèle oblique dans le cas contraire. Parmi ces perspectives cylindriques, les plus usuelles de l'enseignement de la géométrie et du dessin technique sont : la perspective cavalière, la perspective axionométrique, les vues de face et la géométrie descriptive.

La perspective conique donne un certain réalisme dans la représentation spatiale, d'une part en réduisant la taille des objets distants de l'observateur (deux objets identiques mais situés à des distances différentes de l'observateur sont représentés avec des tailles différentes), d'autre part en facilitant la lecture de la profondeur grâce à la perspective introduite par le ou les points de fuite. Cependant, elle est d'un usage assez difficile pour la réutilisation d'une figure en vue de la résolution d'un problème : la non conservation du parallélisme et la perte de la proportionnalité des aires et des longueurs suppriment un certain nombre de références de lecture qui jouent un rôle important dans la déduction de nouvelles propriétés géométriques.

La perspective cavalière est généralement considérée comme l'outil didactique le plus pertinent à l'apprentissage de la géométrie dans l'espace ([Audibert 88], [Brunet 81]), car elle est la plus facile à mettre en œuvre et conserve, à la différence de la perspective conique, certaines propriétés (alignement, parallélisme, rapport des aires et des longueurs).

Cependant, elle présente certaines difficultés, notamment pour la représentation des corps ronds [Destainville 95]. Prenons le cas de la sphère. Celle-ci est généralement représentée par la projection du cercle qui constitue son contour apparent. Or, en perspective cavalière, la direction de projection n'étant pas normale au plan de projection, la projection de ce contour apparent sur le plan de projection donne une ellipse. Cette représentation est en contradiction flagrante avec la représentation culturellement admise (*image stéréotype*, qui est un cercle). De plus, l'absence de points de fuite rend difficile la lecture des positions relatives et des distances.

Le passage d'une perspective à l'autre pour bénéficier des avantages de chacune d'elles, impossible dans le cas d'une représentation graphique sur le papier, semble être un apport non négligeable de l'ordinateur à l'enseignement de la géométrie dans l'espace. Cependant, il est hasardeux d'anticiper les besoins des enseignants sur un point aussi important : ce choix devra être laissé aux enseignants participant au développement du logiciel ou, à la limite, faire l'objet d'une paramétrisation dans le logiciel.

2.2.3 Interprétation des figures

Les limites du dessin en géométrie dans l'espace sont dues à la représentation plane d'une figure dans l'espace, c'est-à-dire à sa *projection*. De manière réciproque, la lecture et la compré-

hension d'une figure, c'est-à-dire l'abstraction du dessin vers la figure (définie en géométrie sous le terme de *relèvement*) est une entreprise qui s'avère être aussi difficile.

Contrairement au dessin dans la géométrie plane, sur lequel l'apprenant peut mesurer et vérifier directement, cette facilité n'est pas conservée dans le cas de la géométrie dans l'espace. La perception d'une propriété spatiale sur le dessin ne signifie pas qu'elle représente une propriété géométrique sur la figure. Par exemple, l'appartenance d'un point au contour apparent d'une sphère ou le parallélisme de deux droites sur le dessin ne signifient pas que ces propriétés soient valides. De plus, la perspective utilisée pour la projection de la figure influe énormément sur sa lisibilité, que ce soit à cause de la non-conservation de propriétés ou de l'absence de points de fuite.

Ce problème doit nous amener à réfléchir aux outils à réaliser dans un environnement d'apprentissage pour aider à l'interprétation d'une figure. Si le *changement du point de vue*, c'est-à-dire l'observation de la figure selon un autre angle de vue, est un atout évident, d'autres outils favorisant *la visualisation* doivent être imaginés et implantés.

2.2.4 Dessin statique vs. représentation dynamique

La construction chez l'apprenant d'une image mentale de la situation spatiale est nécessaire pour initier un processus de résolution ou d'exploration de cette situation.

L'enseignement initial de la géométrie dans l'espace fait une grande part à l'utilisation de maquettes, profitant de leur caractéristique d'objet réel. Elles servent à l'élaboration des images mentales et constituent un moyen de transition entre objets physiques et notions mathématiques. Leur intérêt réside dans leur représentation tridimensionnelle mais surtout dans le fait qu'elles offrent la possibilité de choisir un point de vue et de le changer.

Cependant, la maquette n'est pas toujours d'une grande utilité, notamment lorsque son caractère solide ne permet pas d'observer des propriétés apparaissant à l'intérieur de l'objet[5]. De plus, il est difficile d'envisager la construction de maquettes pour toutes les configurations possibles d'une figure.

C'est pourquoi, progressivement, ces maquettes sont abandonnées au profit du dessin sur un support papier, mettant cette fois à contribution ces images mentales et nécessitent une plus grande activité cognitive.

Ce qui fait défaut en travaillant sur une représentation plane d'une figure géométrique, c'est la possibilité d'*expérimenter* cette figure, de *mettre en évidence* ou d'*explorer* ses propriétés géométriques, de faire des conjectures et de pouvoir les confirmer ou les infirmer. Ainsi [Audibert 85] constate que les élèves ont souvent besoin de nombreux dessins pour les aider à résoudre les problèmes de géométrie dans l'espace. Il s'agit ainsi de retrouver les avantages du changement de points de vue sur la maquette mais, plus encore, d'exploiter la figure selon une autre configuration.

De même, dans le cadre d'une expérimentation sur le rôle d'une maquette par exemple, [Rommevaux 91] fait la remarque que les difficultés ne se situent pas nécessairement au niveau de la "vision", mais en amont : les élèves ne conçoivent pas *la diversité des sous-figures planes extraites* du cube.

L'ordinateur peut contribuer à apporter une réponse au problème de l'*exploration* des propriétés d'une figure géométrique en offrant des possibilités de *déformation* de cette figure, amenant vers le développement d'un environnement de *géométrie dynamique* dans l'espace

5. Même si des maquettes transparentes peuvent être envisagées.

[Goldenberg 98]. Le deuxième point, c'est-à-dire la possibilité d'*extraire des figures planes* d'une figure tridimensionnelle, est certainement plus crucial pour le processus d'exploration et devra être étudié en profondeur

2.3 Logiciels d'apprentissage et éditeur de CAO/DAO

L'évocation de la géométrie dans l'espace amène immanquablement à penser aux travaux et réalisations dans le domaine de la Conception Assistée par Ordinateur (CAO) et il convient de positionner ce domaine vis-à-vis de l'enseignement de la géométrie dans l'espace.

L'objectif d'un système de CAO est de permettre la *modélisation géométrique* de "formes" tridimensionnelles. Le domaine de modélisation géométrique a pris de plus en plus d'importance au cours des trente dernières années, en parallèle avec le développement des capacités informatiques et des modèles sous-jacents. Initialement axés sur une représentation "fil de fer" (*wireframes*) des objets de l'espace, ils ont rapidement évolué vers des représentations plus complexes permettant l'introduction des surfaces ou des volumes, le positionnement des objets dans l'espace ou l'assemblage d'objets entre eux, ... Le lecteur pourra se reporter à [Rogers 90] ou [Foley 90] pour un aperçu des approches utilisées dans la modélisation géométrique et la modélisation des solides.

Cependant, les systèmes de modélisation 3D sont insuffisants pour une approche pédagogique. [Qasem 97] cite trois raisons principales à cette inadéquation :

1. *Domaine de représentation.* La majorité des modèles existant pour la modélisation géométrique classique répond à une problématique de modélisation du monde physique et, dans ce sens, est destinée à la modélisation des solides (objets physiques fermés). Il n'est donc pas possible de représenter des objets mathématiques infinis (comme le plan ou les droites) qui sont principalement utilisés dans l'enseignement de la géométrie dans les classes du second degré.
2. *Opérations de construction.* Les systèmes de modélisation géométrique mettent en œuvre des opérations qui sont essentiellement booléennes (ou ensemblistes) : opérations d'extrusion (soustraction booléenne), d'unification (addition boolèenne) , ... Cela permet la réalisation de constructions complexes en un laps de temps relativement court, mais ces constructions dispensent l'utilisateur de "spécifier" les relations géométriques reliant les objets entre eux. Or, l'un des objectifs de l'enseignement de la géométrie est bien d'amener l'apprenant à expliciter ces relations lors d'un processus de résolution de problème.
3. *Rigidité des modèles.* La modélisation géométrique permet l'élaboration d'objets complexes par combinaison d'autres objets (élémentaires ou non). Cette combinaison, par l'application d'opérateurs de construction, nécessite une spécification exacte de la position des objets dans l'espace, et non une spécification des relations existant entre eux. Ainsi, toute modification d'une composante du modèle implique la mise à jour des positions des autres objets. Il n'est donc pas possible de construire une figure géométrique dont une des composantes disposerait d'un ou plusieurs degrés de liberté dont l'observation de la variation permettrait d'explorer le comportement du reste de la figure.

L'évolution des logiciels de modélisation géométrique au cours des dernières années a réduit de manière sensible ces limitations. En particulier, ils permettent maintenant de spécifier des propriétés et des relations entre objets (approches *"variationnelle"* ou par *"features"*). Cependant, à ces trois raisons, nous pouvons en ajouter une quatrième qui reste vraie malgré cette évolution :

4. *Objectifs d'utilisation.* L'objectif d'un système de modélisation géométrique est, comme son

nom l'indique, de permettre à l'utilisateur d'élaborer, étape par étape, une représentation d'un objet physique par des opérations géométriques. Cela présuppose de sa part une bonne connaissance de la géométrie (opérateurs, nature et propriétés des objets) et, plus encore, une bonne compréhension de l'espace (lecture des scènes spatiales, mise en œuvre des images mentales). Or, l'enseignement de la géométrie a pour objectif d'aider l'apprenant à se familiariser avec les concepts et à élaborer ces images mentales. Cela signifie que les systèmes de modélisation privilégient (au niveau de l'interface par exemple) la mise en œuvre des opérateurs, au détriment de la lecture du modèle. La prise en main d'un tel système est souvent très lourde, alors que celle d'un environnement d'apprentissage doit permettre une prise en main rapide, de manière à focaliser l'attention de l'apprenant sur le domaine d'apprentissage, et non sur l'environnement.

La problématique d'un environnement d'apprentissage et d'exploration de la géométrie dans l'espace est fondamentalement différente de celle d'un environnement de CAO ou de DAO. Les raisons ci-dessus rendent un tel environnement peu adapté à l'enseignement de la géométrie dans l'espace. Cela ne signifie pas qu'il n'y a rien à en tirer : de part l'ancienneté des domaines et les nombreuses réalisations, certains concepts (les vues multiples d'une scène spatiale par exemple) ou certaines mises en œuvre (de manipulation directe par exemple) peuvent être adaptés et implantés dans une finalité d'apprentissage.

2.4 Les micromondes de géométrie dans l'espace

La géométrie est un champ disciplinaire qui a fait l'objet de plusieurs travaux de recherche et de réalisations, mais force est de constater que la géométrie dans l'espace a été moins abordée que la géométrie plane. Cette constatation n'est pas surprenante, nous pouvons en effet la mettre en parallèle avec les difficultés propres à ce domaine, difficultés soulignées ci-dessus.

Mais ce n'est certainement pas la seule raison. En tant que domaine principalement visuel, un logiciel mettant en œuvre la géométrie dans l'espace est fortement dépendant des capacités technologiques (matériels et environnements de développement), que se soit au niveau des capacités graphiques, calculatoires ou d'interface homme-machine.

Dans ce paragraphe, nous présentons quelques logiciels dédiés à la géométrie dans l'espace. Certains sont destinés à son enseignement (c'est-à-dire destinés principalement à l'enseignant pour une utilisation devant ses élèves), d'autres à son apprentissage (destinés à être directement manipulés par l'élève).

L'objectif ici n'est pas de faire la liste complète des environnements d'apprentissage de la géométrie (la production ayant tendance à s'accélérer ces dernières années), ni de donner une description approfondie des quelques logiciels présentés, mais d'en faire ressortir les caractéristiques principales, les types d'activités qu'ils permettent de prendre en charge, mais surtout leurs limites, amenant ainsi à un *cahier des charges* pédagogique et informatique de ***Calques 3D***. Pour ces raisons, nous nous sommes limités à certains imagiciels et micromondes emblématiques.

Nous avons classé ces logiciels en deux catégories selon leur degré d'avancement correspondant à des interfaces utilisateurs plus conviviales, en rapport avec les avancées technologiques.

2.4.1 La première génération

a) SEDIMA

`SEDIMA` [Houben 86] est certainement le premier logiciel d'apprentissage de la géométrie dans l'espace. Il s'agit d'un *imagiciel* conçu, à l'usage seul des enseignants, pour une démonstration

devant leurs élèves. Il permet de visualiser un cube en perspective cavalière et de générer images et animations autour de cet objet : le développement du cube, les différentes sections planes possibles d'un cube (triangle, losange, rectangle, ...).

b) Euclide dans l'espace

`Euclide dans l'espace` [Allard 88] est une extension du langage `Euclide` [Allard 86, Artigue 89] à la géométrie dans l'espace, ce dernier fournissant un ensemble de procédure LOGO permettant de traiter la géométrie plane.

Il permet de construire des figures géométriques utilisant points, droites, sphères et plans. Ces figures peuvent être visualisées en perspective cavalière ou conique, en mode "fil de fer" ou opaque, et selon différents points de vue.

Avant de pouvoir commencer la construction de la figure, l'utilisateur doit établir un *plan de construction* puis l'exprimer à l'aide du langage LOGO ou utiliser les bibliothèques de polyèdres déjà existantes.

c) Dessiner l'espace

`Dessiner l'Espace` [Bernat 89] (suivi ultérieurement de `Pratiquer l'Espace` [Bernat 91]) a été spécialement conçu pour accompagner une brochure d'exercices destinés à des élèves du second degré [Morlet 89].

Le logiciel et les exercices ont pour but d'aider les élèves à mieux comprendre des constructions de l'espace présentées en perspective cavalière. Sur la base d'une figure initiale fournie (cube, pyramide, ...), l'élève dispose d'un certain nombre de primitives de construction (parallèle à une droite, point sur droite, ...) pour l'élaboration de la figure, cf. figure 2.1.

Il est possible de faire tourner la figure en modifiant le point de vue de l'utilisateur ou encore de consulter un historique traçant les étapes de la construction.

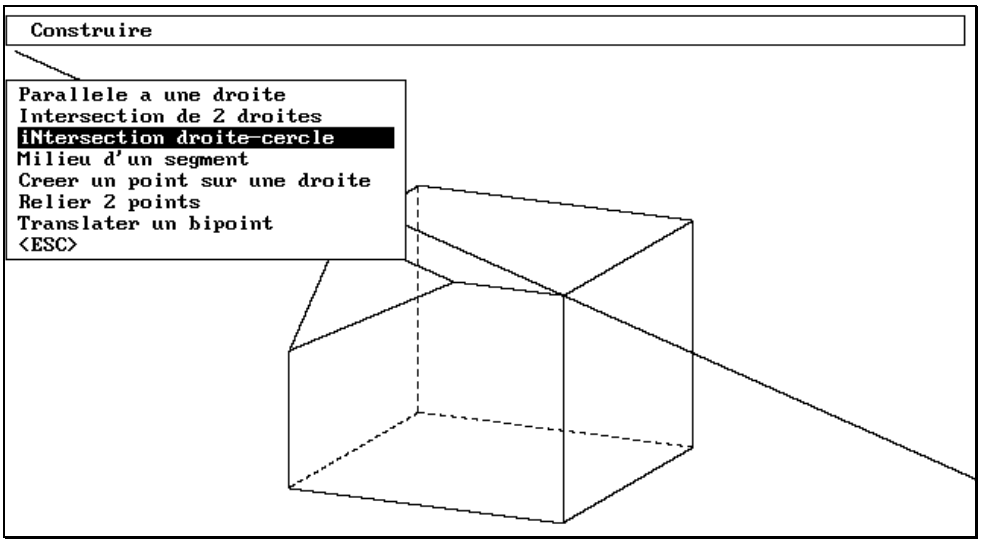


Figure 2.1 – *Pratiquer l'Espace*

Le point fort de ce logiciel réside dans sa simplicité de prise en main (les commandes sont immédiatement accessibles) et d'utilisation (la désignation directe des éléments de la figure à

l'aide de la souris institue une impression d'action directe sur l'objet représenté). A ce titre, il est le premier logiciel de géométrie dans l'espace offrant une interface de manipulation directe.

Ces deux logiciels ont fait l'objet de publications variées ([Bailleux 91, Bouteiller 93], [Cuppens 91], [DLC15 94], [IREM 94], ...) qui prouvent l'intérêt que leur a porté la communauté enseignante. Mais ils souffrent de certaines insuffisances. En particulier, ces logiciels ne respectent pas les standards actuels d'interface et ne proposent pas (directement) de possibilités d'ouverture à l'enseignant (il est excessivement difficile de définir une nouvelle figure de base).

2.4.2 La deuxième génération

Les logiciels suivants ont tous été développés sous Windows, à l'exception de `Cabri 3D`, développé sous Macintosh.

a) Géospace

`Géospace` [Authier 98][6] permet la construction de figures dans l'espace à partir d'objets prédéfinis (éléments du repère orthonormé, c'est-à-dire l'origine, les trois axes, les trois plans , ...) et d'objets élémentaires, géométriques (points, droites, plans, polygones, polyèdres convexes, sphères, cônes, ...) mais aussi mathématiques (vecteurs, variables numériques, fonctions, ...). Il est possible de faire tourner ces objets, les déplacer ou changer leur taille. `Géospace` autorise un accès direct au registre analytique de la géométrie (création d'objets à partir de leur représentation analytique, affichage de calculs liés aux mesures de la figure, ...).

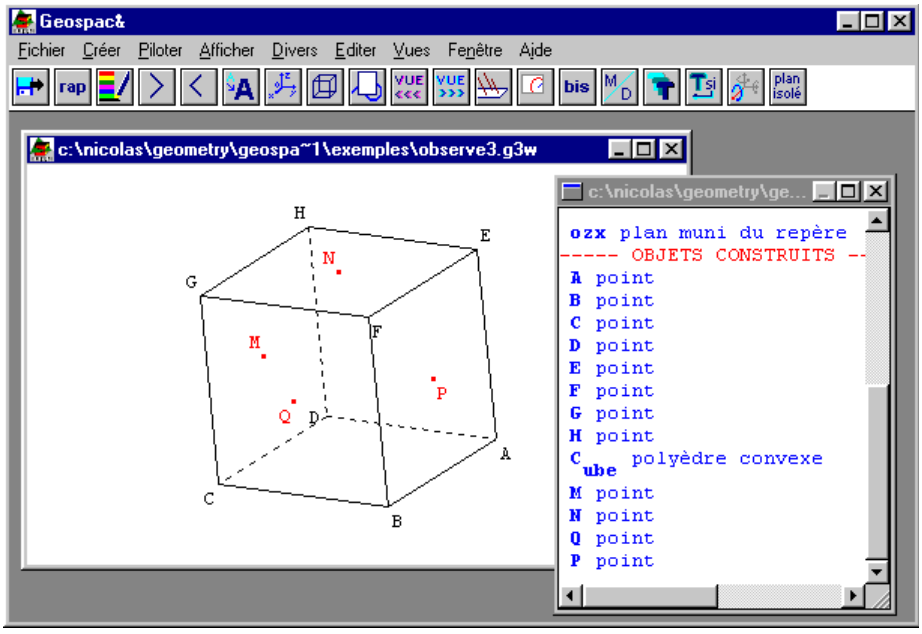


Figure 2.2 – *Géospace*

L'utilisateur travaille sur une représentation en perspective de sa figure, cf. figure 2.2. Celle-ci peut être observée selon différents points de vue et plusieurs modes de représentation : "fil de fer", transparent, opaque.

6. Nous présentons ici la version du logiciel développé pour Windows. Il existe une version plus ancienne (datant de 1992) pour DOS et distribué par le Ministère de l'Éducation Nationale.

Le point fort de `Géospace` est la possibilité de l'utiliser comme "langage-auteur" d'imagiciels : l'enseignant peut définir des commandes contenant des opérations de construction ou de manipulation, leur associer des touches, définissant ainsi un "scénario" qu'il peut rejouer ultérieurement devant ses élèves.

Cependant, l'un des problèmes du logiciel est le mode de construction choisi. Tous les objets doivent être nommés au moment de leur création, les constructions ultérieures étant définies à partir de ces noms et d'expressions analytiques contenant valeurs, calculs et appels de fonctions. Ce mode permet une définition stricte d'une figure mais est malaisé à mettre en œuvre, l'utilisateur devant se rappeler quel est l'objet associé au nom. De plus, tous les objets ne possèdent pas de représentation visuelle. Si les points, les droites, ... possèdent des attributs visuels qui peuvent être modifiés, les plans, les repères, les vecteurs et les transformations ne sont pas visibles à l'écran. Il faut les utiliser via d'autres constructions pour pouvoir observer le *résultat* de la construction.

Ce logiciel pédagogique est le résultat d'un travail de recherche dans le domaine de la didactique des mathématiques, impliquant une équipe pluridisciplinaire (informaticiens, didacticiens, enseignants). Comme en témoigne les compte-rendus d'expérimentation et les exemples de séquences pédagogiques[7], il semble être d'un usage relativement courant chez les enseignants impliqués dans l'intégration de l'outil informatique pour l'enseignement de la géométrie.

b) KAPPA

`Kappa` [Muzellec 96] est un atelier interactif de géométrie dans l'espace, mettant à la disposition du collégien ou du lycéen plusieurs modes de visualisation et de construction en trois dimensions. Il permet la construction des figures à partir des objets et des relations classiques. Il dispose aussi de quelques fonctions de transformation comme la rotation, la réflexion, ... et d'un accès direct au registre analytique (définition de figures à l'aide d'expressions vectorielles dans un éditeur d'expressions analytiques). Les figures peuvent être observées selon plusieurs modes : "fil de fer", transparent, opaque, avec ou non remplissage des faces, amenant ainsi vers la *géométrie du solide*.

Un des aspects intéressant de `Kappa` est la métaphore utilisée pour la manipulation directe des points libres. `Géospace` propose en effet le déplacement de ces points dans un plan parallèle au plan de projection (c'est-à-dire l'écran), garantissant ainsi une facilité de manipulation mais entraînant une difficulté de compréhension de l'opération puisque le résultat du déplacement sera différent suivant le point de vue courant de la projection. `Kappa` propose une métaphore plus "naturelle" en plaçant *a priori* les points libres dans le plan (xOy), et permettant leur déplacement soit selon ce plan horizontal, soit selon une parallèle à l'axe (Oz) du repère.

Une autre fonctionnalité intéressante est la possibilité d'obtenir le patron de tout polyèdre convexe et de l'imprimer afin de construire une maquette du solide.

Certaines lourdeurs peuvent être observées au niveau de l'interface, en particulier en ce qui concerne la levée des ambiguïtés de désignation ou le changement du point de vue. S'il est possible de déformer une figure (en déplaçant un point libre de celle-ci), la figure est malheureusement affichée sans effets tridimensionnels (élimination des arêtes cachées, ...).

c) L'Atelier de géométrie 3D

L'`Atelier de Géométrie 3D` [Lepine 97] met à la disposition de l'élève des outils de construction (points liés à des objets, droites, plans, cercles) et de transformation (symétrie,

7. Voir http://www2.cnam.fr/creem/general/publi.htm

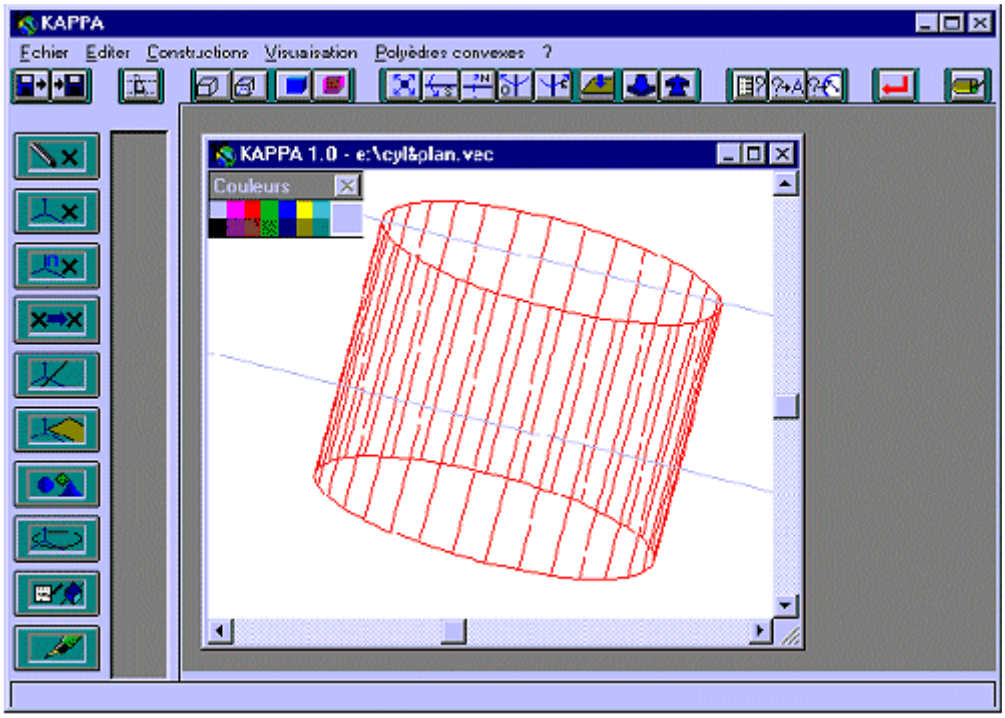


Figure 2.3 – *KAPPA*

rotation, ...). La figure s'observe en perspective cavalière et il est possible de changer le point de vue manuellement ou automatiquement (vue de face, de côté, ...).

Il existe un lien très fort entre ce logiciel et son pendant pour la géométrie plane (`L'atelier de géométrie 2D`) : l'utilisateur peut y exporter (c'est-à-dire *extraire*) des objets situés dans un plan préalablement désigné et utiliser les fonctionnalités qu'offre ce dernier logiciel pour réaliser un travail dans le plan (construction, exploration, ...). Les modifications ainsi apportées seront automatiquement mises à jour dans l'espace.

Il permet aussi la définition d'imagiciels pouvant servir d'aide à des exercices définis par un enseignant. Il s'agit en fait de l'enregistrement d'une construction que l'élève peut rejouer pas à pas. Cependant, la mise en œuvre de ces imagiciels nécessite la rédaction d'un fichier de commandes qui peut s'avérer fastidieuse.

L'un des gros problèmes du logiciel réside dans la métaphore utilisée pour la construction des points dans l'espace. En effet, selon les concepteurs de l'`Atelier de Géométrie 3D`, *"un point de l'espace étant défini par trois coordonnées et la souris ne se déplaçant que suivant deux directions, on ne peut placer un point libre dans l'espace"*. Cela permet de lever les ambiguïtés de manipulation directe mais limite énormément les possibilités de construction et d'exploration d'une figure. Les points doivent être liés à un objet (appartenant à un plan ou à un segment, à l'intersection de deux droites, ...). C'est pour cette raison que toutes les constructions doivent s'appuyer sur un des *dessins de base* (c'est-à-dire une matérialisation de l'espace) proposé par le logiciel : le repère orthonormé, le cube ou le tétraèdre (ces deux derniers étant les seuls objets volumiques prédéfinis).

D'autres lourdeurs sont à noter, en particulier en ce qui concerne la désignation directe des objets lors des tâches de construction (pour créer un segment, il faut cliquer sur le premier point, faire glisser la souris puis relâcher le bouton sur le deuxième). Les rétroactions visuelles

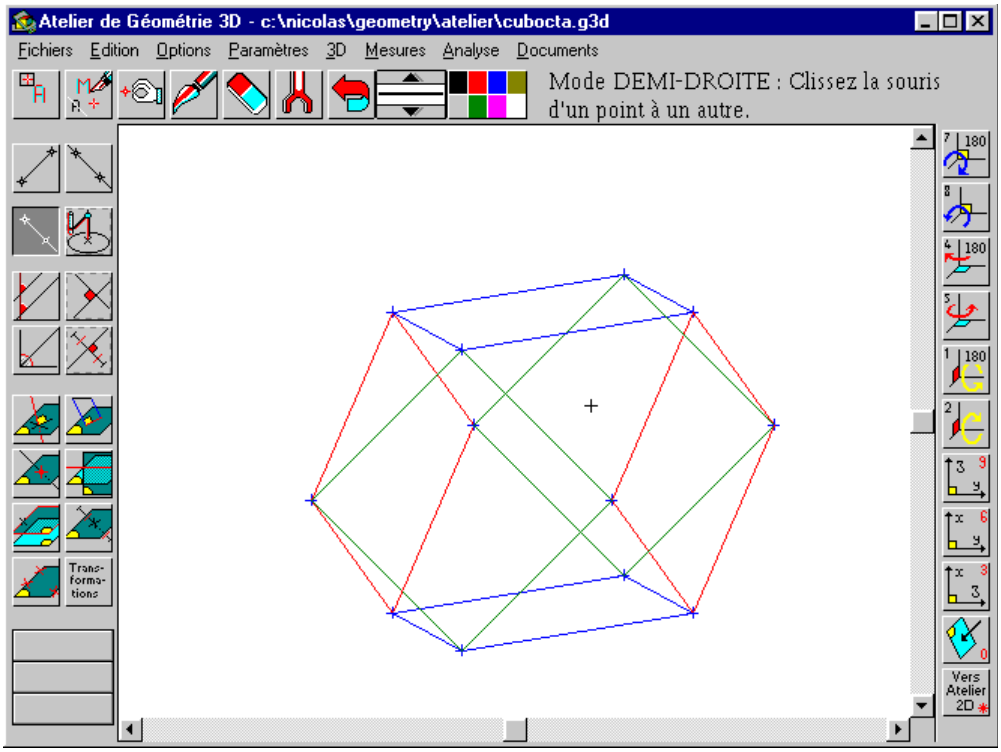


Figure 2.4 – *L'Atelier de Géométrie 3D*

sont aussi insuffisantes dans certaines situations : les constructions planes par exemple nécessitent l'*activation* préalable d'un plan avant d'effectuer l'opération, mais aucune information ne permet, après coup, de savoir quel plan a été activé.

d) Cabri 3D

`Cabri 3D` [Qasem 97] Un des aspects intéressants de `Cabri 3D` est le travail fait au niveau des effets visuels pour améliorer la lisibilité des dessins : dégradés de couleur pour suggérer l'éloignement d'un objet (par exemple une droite), intensités de couleurs différentes pour représenter les parties cachées (au lieu d'un attribut binaire, traduit par exemple par les "classiques" pointillés), ...

L'ajout d'un module de rendu réaliste, utilisant les capacités de la bibliothèque graphique `QuickDraw 3D` (l'équivalent Macintosh d'`OpenGL`), permet d'obtenir une figure dont la qualité de rendu, le dégradé des couleurs et la diffusion de l'éclairage restituent à l'utilisateur l'impression de volume et de profondeur. L'opacité des faces, de même que l'impossibilité de déformation de la figure, gênent cependant l'observation et l'exploration de celle-ci, l'utilité pédagogique d'un tel module reste sujet à caution.

De même, certains choix, faits apparemment dans un objectif de manipulation directe, nous semblent discutables. Par exemple, `Cabri 3D` représente les plans sous la forme d'un parallélogramme avec une épaisseur mettant en évidence la partie la plus proche de l'utilisateur. Cette épaisseur peut être modifiée (ou éliminée) par manipulation directe en saisissant la bordure du plan, ce qui peut entraîner une ambiguïté avec la déformation de la figure (déplacement d'un point libre). D'autre part, une épaisseur importante rend ce plan trop "similaire" à un parallélépipède.

`Cabri 3D` s'inscrit dans la continuité des travaux sur `Cabri Géomètre` et la géométrie dynamique et, à ce titre, a bénéficié de l'expérience d'une équipe de recherche pluridisciplinaire. A notre connaissance, il a cependant conservé un statut de prototype et n'a jamais fait l'objet d'expérimentations en dehors du laboratoire où il a été développé.

2.4.3 D'autres environnements d'apprentissage ?

Notons que des micromondes de géométrie dynamique dans le plan, comme `Cabri Géomètre` [Baulac 92], [Laborde 94] ou `Calques 2` [Bernat 94a], peuvent être utilisés pour "simuler" la géométrie dans l'espace : ils offrent en effet la possibilité d'étendre l'ensemble des commandes disponibles en construisant des *macro-commandes* permettant de mettre en œuvre les étapes élémentaires de la construction d'une figure dans l'espace et la "projection" de celle-ci sur le plan (définition du repère et de l'axe de projection, simulation des polyèdres, ...). C'est ainsi par exemple que [Pavel 99] utilise, à défaut d'un environnement dédié, les macros de `Cabri Géomètre` pour concevoir la partie géométrie dynamique d'un environnement distribué d'apprentissage de la géométrie descriptive.

Ces "simulations" de la géométrie dans l'espace impliquent nécessairement des contraintes supplémentaires quant à la manipulation et l'observation des figures géométriques. Elles n'offrent donc pas toutes les possibilités qu'un environnement dédié peut fournir.

2.5 Conclusion : vers un "cahier des charges" pour *Calques 3D*

L'analyse du domaine et l'étude des environnements d'apprentissage existants, présentées dans les sections précédentes, nous permettent de disposer d'une base de réflexion pour l'élaboration d'un premier cahier des charges pour le développement d'un prototype, c'est-à-dire une première phase d'expression des besoins. En particulier, nous pouvons mettre en évidence le besoin d'un certain nombre de catégories de fonctionnalités.

La première concerne les fonctionnalités de *construction* d'une figure géométrique. Si nous pouvons constater, dans les environnements existants, une certaine "universalité" des objets élémentaires du domaine (les points, les droites, les plans, ...), les différences entre systèmes interviennent principalement au niveau de la mise en œuvre des procédures (stratégies de construction, arguments, contexte requis pour la validation de la construction, ...). D'autres différences apparaissent au niveau des objets plus complexes, en particulier les objets volumiques : certains logiciels n'offrent pas les corps ronds, d'autres requièrent la construction explicite des polyèdres (à partir de points, segments, ...). De même pour le registre analytique de la géométrie : doit-on permettre la construction d'objets à partir de données numériques ?

La deuxième catégorie regroupe les fonctionnalités de *visualisation*, permettant d'aider l'élève à procéder à une abstraction de la figure. Là aussi, les possibilités sont vastes. Proposer des modes de présentation des objets géométriques variés et adaptés, autoriser l'élève à changer le point de vue ou la perspective utilisée, ... sont des exemples classiques mais pouvant s'avérer pertinents. D'autres restent cependant à proposer. En particulier, les annotations ou le marquage visuel de propriétés géométriques, par le système ou par l'utilisateur (élève ou enseignant) peuvent aider à pallier les difficultés de vision dans l'espace.

La troisième catégorie regroupe les fonctionnalités d'*exploration* de la figure. La déformation d'une figure à partir de ses points libres est une caractéristique commune aux environnements de géométrie dynamique, que ce soit dans le plan ou dans l'espace. L'extraction de figures planes

est un procédé utilisé relativement souvent dans la démarche de résolution d'un problème en géométrie dans l'espace. D'autres fonctionnalités d'exploration peuvent être proposées : l'historique de la construction d'une figure peut permettre d'en retirer des informations importantes, des fonctions de vérification automatique de propriétés (alignement de points, parallélisme de droites, ...) peuvent avoir un rôle important dans la démarche d'exploration.

Enfin, la dernière catégorie regroupe les *éléments d'ouverture à l'utilisateur*, qu'il s'agisse de l'élève ou de l'enseignant : sélection des modes de présentation, accès ou non à certaines commandes, ...

Cette première ébauche d'un cahier des charges de ***Calques 3D*** n'est que "théorique", c'est-à-dire élaborée à partir des problèmes de la géométrie dans l'espace. Il nous reste maintenant à la compléter de manière plus "empirique" en intégrant des enseignants dans le processus de conception, ceci afin de prendre en compte leurs besoins réels et leurs approches personnelles de l'enseignement de ce domaine.

Deuxième partie

Acquisition et formalisation de connaissances pour la réalisation de logiciels pédagogiques

Chapitre 3

Un cycle de production de logiciels éducatifs

3.1 Introduction

Dans la partie précédente, nous avons proposé une première ébauche d'un cahier des charges pour ***Calques 3D***. Cette ébauche n'est que "théorique", dans le sens où elle repose essentiellement sur l'analyse des problèmes de l'enseignement de la géométrie dans l'espace et sur les environnements d'apprentissage existants. Il convient de compléter cette ébauche de manière "empirique" en intégrant des enseignants dans le cycle de production d'un logiciel éducatif.

Après avoir donné un aperçu des limites des outils de conception permettant à l'enseignant de réaliser lui-même un logiciel éducatif (section 3.2) et des caractéristiques du *génie éducatif* (section 3.3), nous décrivons la démarche mise en place dans le cadre de ce projet et qui propose une redéfinition du rôle des enseignants dans un tel processus (section 3.4). Nous concluons ce chapitre en mettant en évidence le besoin d'un cadre permettant de gérer la négociation des besoins.

3.2 De l'enseignant concepteur à l'enseignant auteur

Historiquement, plusieurs types d'outils ont été mis à la disposition des enseignants pour leur permettre de réaliser *par eux-mêmes* les logiciels à vocation éducative (didacticiels, imagiciels, simulateurs, micromondes, tuteurs intelligents, ...), leur donnant ainsi des attributions d'*enseignant concepteur*.

Les premiers d'entre eux sont les *langages de programmation* classiques. S'ils permettent la conception de n'importe quel type d'environnement pour n'importe quel domaine d'apprentissage, ils nécessitent des compétences informatiques importantes. Cette approche a bien reçu les faveurs de quelques enseignants mais elle reste cependant marginale et ne garantit pas la prise en compte de l'évolution des techniques ou des modèles informatiques.

Les *langages-auteurs* ont tenté d'apporter, dès le milieu des années 70, une solution à ce problème de compétence en proposant des fonctionnalités spécifiques à l'écriture de logiciels pédagogiques : description des écrans, analyse de réponses, contrôle du cheminement, ... Ils imposent cependant un processus d'écriture encore lourd, notamment si les primitives ne satisfont pas les enseignants concepteurs.

L'évolution des techniques informatiques (en particulier au niveau des IHM) a permis une amélioration substantielle de ce principe par l'apparition des *systèmes-auteurs*, véritables généra-

teurs de logiciels éducatifs. En général, ils sont constitués d'un ensemble d'éditeurs dédiés (éditeur de contenus, éditeur de séquencements, ...) et fonctionnent selon deux modes (mode d'édition et mode d'exécution). Outre les problèmes de lourdeurs et de prérequis issus les langages-auteurs, ces systèmes souffrent de limitations importantes liées à leur degré de spécialisation : certains systèmes sont plus orientés pour une réalisation de séquences pédagogiques (`ECSAI`), d'autres pour des simulations ou des démonstrations (`AUTHORWARE`,), ... Le lecteur pourra se reporter à [Lapujade 96] pour un panorama des systèmes d'aide à la conception de logiciels éducatifs, ou à [Murray 99] pour une analyse approfondie de systèmes-auteurs dédiés à la réalisation de tuteurs intelligents.

Il convient donc de préciser la place des enseignants dans le processus de conception d'un logiciel pédagogique. Nous pensons qu'il n'est pas possible de fournir un environnement unique permettant à n'importe quel enseignant de produire n'importe quel type de logiciel éducatif : les formalismes de représentation des connaissances, les modes d'interactions, ... doivent pouvoir être choisis en fonction d'un besoin particulier et condamnent le recourt systématique à des fonctionnalités de génération automatique. De plus, si les enseignants ont bien un rôle important à jouer dans l'expression des besoins, dans la description *au niveau conceptuel*, des connaissances à mettre en œuvre, leur modélisation au niveau *logique ou physique* reste du ressort de l'informaticien. Plutôt qu'un rôle de concepteur, les enseignants doivent disposer d'un rôle d'*auteur* dans la production d'un logiciel éducatif.

Nous rejoignons ici un principe qui semble de plus en plus acquis dans la communauté EIAO ([Vivet 91], [Guin 94], ...) : l'importance d'une collaboration pluridisciplinaire pour la conception des logiciels pédagogiques, où tous les acteurs du domaine doivent pouvoir être pris en compte : enseignants, informaticiens mais aussi spécialistes en IHM, didacticiens, ...

3.3 Du génie logiciel au génie éducatif

En génie logiciel, le *cycle de vie* d'une application informatique permet de décomposer le processus de développement selon une série d'activités couplées entres elles, partant du besoin initial pour arriver au produit final. Quels que soient les modèles de cycle de vie utilisés, ils consistent en un minimum de cinq phases [Davis 88] :

- la phase de *définition ou spécification* qui permet, à partir d'un cahier des charges, d'obtenir une description détaillée du comportement externe du système,
- la phase de *conception* qui permet de décomposer le système en modules indépendants,
- la phase de *réalisation* qui concerne le développement de chaque module et leur intégration dans le système,
- la phase *de test* qui permet d'évaluer et de valider les caractéristiques du système[8],
- la phase *d'exploitation* qui permet la diffusion, l'utilisation et la maintenance du système.

Des cycles de vie séquentiels, comme le modèle *en cascade* [Boehm 76], s'ils permettent une gestion efficace des projets où les différentes phases sont facilement organisables et contrôlables,

8. Dans le cycle de vie d'un logiciel informatique, il existe plusieurs sortes de tests, à différents niveaux du développement (tests unitaires des modules isolés, tests d'intégration d'un ensemble de modules, tests systèmes du logiciel dans des conditions opérationnelles, ...) mais qui, pour la plupart, ne concernent que l'informaticien. La phase de *test et évaluation* décrite ici est à prendre dans le sens d'une expérimentation et d'une validation du prototype selon des scénarios, définis et réalisés par l'utilisateur du produit, c'est-à-dire l'enseignant. Cette phase peut être assimilée à la *maquette exploratoire* du maquettage (ou *prototypage rapide*) [Gaudel 96].

souffrent cependant de certains défauts. Le développement d'un système y est considéré comme un problème purement technique de traduction et d'enchaînement de phases, ce qui d'une part est rarement le cas dans la pratique et, d'autre part, rend l'informaticien seul maître à bord. Les phases de conception et de réalisation sont distinctes ce qui rend difficiles les retours-arrières. A chaque modification ou mise à jour du logiciel, l'indépendance des phases entraîne une remise en question partielle voire complète de celles-ci. Enfin, l'intervention des usagers n'est possible qu'en début (spécification) et en fin de cycle (test), ce qui augmente les risques d'inadéquation du logiciel aux besoins effectifs.

En outre, l'introduction des méthodes objets, par leur pouvoir de modularité et de réutilisation, et le développement de logiciels éducatifs (souvent désigné sous le terme de *génie éducatif* ou *génie didactique*), rendent ces modèles caduques. En effet, *génie éducatif* et *génie logiciel* ne sont pas deux domaine en opposition mais complémentaires : ils partagent le même objectif (la conception d'un logiciel), la même problématique (la prise en compte des besoins du client), ... A ce titre, tous deux peuvent faire l'objet des remarques suivantes.

1. Dans beaucoup de cas, il est impossible de disposer d'un cahier des charges initialement suffisant, soit parce que les enseignants n'ont aucune idée de toutes les possibilités que la technologie peut offrir dans la médiatisation d'un domaine d'apprentissage, soit parce qu'il n'existe pas de consensus sur les connaissances, les méthodes, les outils, ... représentatifs du domaine. Cela conduit donc à une description partielle ou incomplète du produit qu'ils désirent et amène l'informaticien à faire des choix d'ordre purement pédagogiques.
2. L'existence d'une équipe pluridisciplinaire (enseignants, informaticiens, ergonomes, ...) pour la conception d'un EIAO introduit des objectifs différents de conception dans ce processus. Ces différences font qu'une difficulté importante de ce domaine réside dans les problèmes de compréhension entre les différents acteurs : compréhension du langage mais surtout des motivations et intentions. Même si chacun se délimite strictement son domaine d'intervention, ceux-ci ne sont pas totalement disjoints et chacun attend de l'autre qu'il assure les compléments indispensables, d'où une synergie insuffisante.

Par conséquent [Bruillard 94], la coordination de cette équipe ne doit pas être assurée par l'informaticien mais bien par l'enseignant (ou le didacticien). Ainsi, les phases centrales du développement d'un logiciel éducatif ne sont plus les phases de conception et de réalisation mais les phases de spécification et de test. D'autres méthodes de conception permettant d'atteindre cet objectif doivent donc être envisagées.

3.4 Une méthodologie de conception pour *Calques 3D*

Nous décrivons ici la méthodologie de conception que nous avons appliquée afin de développer ***Calques 3D*** : un projet basé sur la réalisation incrémentale d'un prototype et l'organisation de groupes de travail incluant enseignants de mathématiques et informaticiens.

3.4.1 Un développement incrémental par prototypage

Le cycle de production que nous proposons pour la conception de ***Calques 3D*** est un cycle de production en spirale, où la réalisation d'un prototype opérationnel joue un rôle central.

Dans ce cycle de production, illustré dans la figure 3.1, nous séparons nettement les deux domaines relevant d'un travail pluridisciplinaire : la *pédagogie* où interviennent les enseignants

(auteur et prescripteur), et le *génie logiciel* où intervient l'informaticien. C'est à ce niveau que nous pouvons différentier les rôles et prérogatives des enseignants :

- l'*enseignant auteur* qui collabore directement au processus de développement,
- l'*enseignant prescripteur* qui est l'utilisateur final du logiciel.

Nous retrouvons dans ce schéma les cinq phases du cycle de vie du génie logiciel, réorganisées de manière à prendre en compte la spécificité de notre approche.

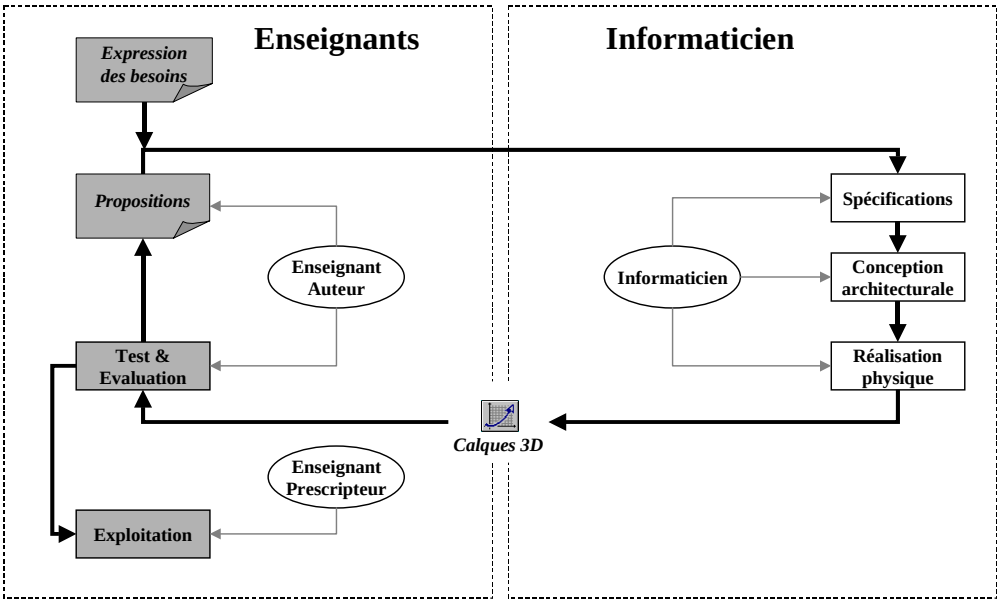


Figure 3.1 – *Un développement incrémental par prototypage*

A partir d'un premier "cahier des charges" défini théoriquement, désigné sous le terme d'*expression des besoins*, l'informaticien peut mettre en œuvre les phases de *spécification formelle*, de *conception architecturale* et de *réalisation physique* afin d'obtenir une première version du logiciel. Celle-ci est alors donnée aux enseignants auteurs qui mettent en œuvre une phase de *test et évaluation*, à l'issue de laquelle des propositions sont faites sur les mises à jour et ajouts à effectuer. Ces propositions sont prises en compte par l'informaticien pour relancer un nouveau cycle de conception, jusqu'à l'obtention d'une version stable donnée à *exploitation* aux enseignants prescripteurs.

L'activité de prototypage prend tout son sens dans ce processus. Elle a pour but de mettre *rapidement* dans les mains d'un expert un prototype opérationnel et pertinent, c'est-à-dire un prototype qui rende compte de la structure et du comportement du logiciel stable. Cela rend visible le processus de développement du logiciel à l'ensemble des partenaires du projet, tout en réduisant les risques et en améliorant sa manipulation. Lorsque le logiciel ne franchit pas les tests d'acceptation, les partenaires peuvent "voir" et se mettre d'accord sur les effets de l'imperfection.

De plus, le prototypage permet de focaliser le développement sur certaines parties du logiciel : aborder un point d'interface, mettre en œuvre une fonctionnalité donnée, améliorer ou revoir la stratégie opérationnelle d'une fonction, ... Cette concentration sur un point du développement nécessite l'utilisation d'un formalisme de conception qui respecte trois critères [Krief 92] :

- la *modularité* du logiciel, c'est-à-dire sa décomposition en modules *indépendants*,
- la *réutilisabilité* qui permet de récupérer les modules existants et de les intégrer dans le prototype en cours,

– l'*abstraction des données* qui permet de représenter, manipuler et faire cohabiter des connaissances hétérogènes (interface, noyau fonctionnel, ...).

Nous désignons cette méthodologie de développement sous le terme de *cycle de production par prototypage incrémental*, au sens de [Boehm 88]. C'est un processus *interactif* et *itératif* de choix de décision entre l'expert du domaine (l'enseignant) et le concepteur. C'est à ce niveau que se justifie la différence entre l'*enseignant prescripteur* et l'*enseignant auteur*. L'attitude de ce dernier doit être critique par rapport au prototype : il fournit d'une part l'ensemble des informations nécessaires à l'implantation du domaine d'apprentissage et d'autre part un regard averti sur le logiciel en cours d'utilisation.

3.4.2 Intégration des enseignants auteurs dans le cycle de production

Le projet ***Calques 3D*** s'est déroulé en trois phases, réparties sur les trois dernières années.

a) Élaboration d'une première maquette

En 95/96, nous[9] avons réalisé une première version d'une maquette de micromonde pour l'enseignement de la géométrie dans l'espace. Il nous semblait indispensable que le processus de développement incrémental s'appuie sur une première ébauche de l'environnement d'apprentissage, disposant d'un minimum de fonctionnalités réellement opérationnelles mais offrant au maximum un aperçu des potentialités. L'objectif de cette ébauche est double. D'une part elle sert de base de discussion pour la définition du contenu pédagogique et l'opérationnalisation des fonctionnalités , amenant ainsi l'environnement du statut de maquette au statut de prototype. D'autre part, elle joue un rôle non négligeable de motivation par l'illustration des potentialités qu'elle offre. Nous nous sommes rendu compte que les enseignants appréciaient de disposer rapidement d'une concrétisation de leur implication dans le développement d'un logiciel.

L'élaboration de cette première maquette s'est faite de manière "théorique", c'est-à-dire à partir de l'étude des environnements existants d'apprentissage de la géométrie dans l'espace (cf. chapitre 1) et de l'analyse des difficultés de l'enseignement de ce champ disciplinaire (cf. chapitre 2), mais sans réelle intégration des enseignants dans le processus. C'est ce que nous désignons sous le terme d'*expression des besoins* dans la figure 3.1. Durant cette phase, nous avons surtout mis l'accent sur l'introduction des premiers outils d'édition (premiers objets géométriques élémentaires et primitives de construction), la mise en œuvre des processus de visualisation des figures géométriques (perspectives, points de vue de l'observateur, modes de présentation des objets, ...) et l'élaboration de l'interface utilisateur (menus, fenêtres d'édition et de visualisation, manipulation directe dans l'espace, ...).

b) *Calques 3D* pour l'enseignement général

En 96/97, nous avons réuni un premier groupe de travail, constitué de dix enseignants de mathématiques du second degré, afin de valider les choix effectués lors de la première phase et les élargir pour intégrer d'autres pratiques. Sur la base du prototype existant, nous avons organisé le développement de ***Calques 3D*** en fonction des objectifs de cet enseignement en collège et lycée (géométrie filaire, perspective cavalière, approche constructiviste, ...) et des difficultés de cette formation (perte de la troisième dimension à l'écran, application des propriétés de la géométrie plane dans l'espace, ...), afin de définir et de mettre en œuvre des fonctionnalités adaptées.

9. Philippe Bernat, qui a initié et encadré le projet jusqu'à son décès en 1997, était à la fois enseignant de mathématiques et informaticien, ce qui nous permettait de disposer d'une expertise non négligeable.

Les trois objectifs de ***Calques 3D*** (construction, observation et exploration) ont été discutés et comparés avec les attentes des enseignants sur le rôle d'un logiciel pour l'apprentissage de la géométrie dans l'espace. Il est ressorti que la priorité devait être donnée à l'observation *directe* des figures géométriques, point faible du domaine. Ainsi, nous avons décidé d'un commun accord de ne pas prendre en compte le registre analytique de la géométrie dans le développement du micromonde : cela concerne bien sûr la construction des objets géométriques (construction par manipulation directe et non par saisie des équations ou des coordonnées) mais aussi l'affichage et la manipulation de ces représentations analytiques (affichage des équations de plans, calcul vectoriel, ...). Bien entendu, ce choix est temporaire, les versions futures de ***Calques 3D*** pourront prendre en compte ce registre.

Dans le même ordre d'idée, l'un des tous premiers points de discussion a été la question de la ou les perspectives à implanter dans le logiciel (cf. section 2.2.2). Dans la première version du prototype, nous avions implanté la perspective cavalière[10] et la perspective conique (un ou plusieurs points de fuite). Ces deux choix se justifiaient d'une part pédagogiquement par leurs avantages respectifs pour la lecture d'une figure et d'autre part informatiquement par l'unicité du modèle mathématique sous-jacent, permettant une mise en œuvre simple et efficace de la projection. Les discussions avec les enseignants ont permis de confirmer l'importance de la perspective cavalière dans l'enseignement mais ils nous ont aussi exprimé leurs réticences quant à l'utilisation d'une projection conique. Celle-ci a donc été purement et simplement supprimée du logiciel.

c) *Calques 3D* pour l'enseignement technique

En 97/98, un deuxième groupe s'est constitué, cette fois avec quatre enseignants provenant de lycées techniques. L'objectif de ce deuxième groupe était de valider les choix faits précédemment et de proposer de nouvelles fonctionnalités permettant l'utilisation du logiciel dans ce contexte, en particulier selon l'approche *systémique* de l'enseignement technique, mais surtout d'essayer d'arriver à une convergence des fonctionnalités du logiciel et de *terminer* le prototype.

Face à notre inexpérience dans ce champ disciplinaire particulier, une de nos grandes incertitudes concernait l'adéquation de ***Calques 3D***, telle que développée pour l'enseignement général, vis-à-vis des objectifs et prérequis de l'enseignement technique qui nous semblaient relativement éloignés. Or, la place potentielle d'un tel logiciel dans l'enseignement technique s'est avérée non négligeable. Car même si les activités pédagogiques sont proposées sous une "étiquette technique" (découpage et soudage de tuyaux, élaboration de charpentes de bâtiments, ...), elles se rapportent bien à des problèmes élémentaires de géométrie dans l'espace (sections planes de cylindre, construction de polyèdres usuels, ...). Les enseignants de ce groupe de travail ne cherchaient donc pas à obtenir une version éducative d'un logiciel de modélisation géométrique mais bien un logiciel pour l'enseignement de la géométrie dans l'espace, aucun dispositif de cette nature n'existant entre l'enseignement "papier-crayon" traditionnel et l'utilisation professionnelle d'environnement de CAO.

C'est dans ce contexte que nous avons pu approfondir la notion d'extraction de figures dans l'espace (cf. section 2.2.4). Dans l'enseignement général, ce mécanisme consiste surtout (voire uniquement) à extraire des figures planes d'une figure tridimensionnelle, de manière à transposer

10. Pour être précis, il s'agit en fait d'une perspective *axionométrique*, autre cas particulier de projection cylindrique. La définition de la perspective cavalière est en effet trop variable et contraignante (définie par un *angle des fuyantes* et un *rapport de réduction*). De plus, elle n'est plus viable dans un environnement où il est possible de modifier le point de vue de l'observateur. Pour des raisons historiques et de familiarité du concept, et avec l'accord des enseignants, nous l'avons cependant qualifiée de *cavalière*.

la résolution d'un problème de l'espace vers le plan pour y réutiliser théorèmes et propriétés, souvent mieux maîtrisés par l'élève. L'approche systémique de l'enseignement technique nécessite de l'apprenant l'analyse d'une figure complexe (par exemple une charpente ou un moteur) afin d'en isoler les éléments constitutifs : cette extraction n'est donc pas uniquement plane mais aussi spatiale (cf. figure 3.2).

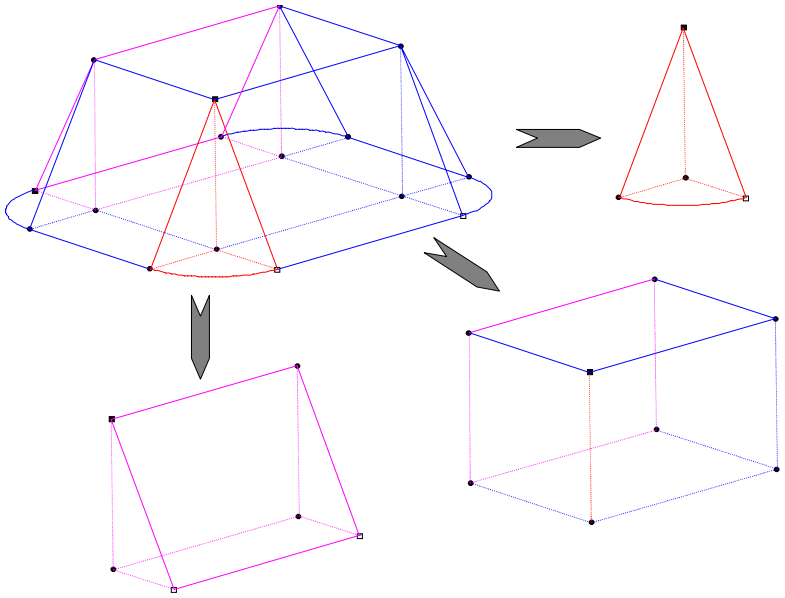


Figure 3.2 – *Approche systémique de l'enseignement de la géométrie dans l'espace.*

d) Organisation des séances de travail

Le travail avec les différents groupes s'est déroulé de façon similaire les deux dernières années. Nous avons organisé des réunions mensuelles d'une demi-journée. Dans la mesure où les délais de mises à jour du prototype le permettaient, les enseignants disposaient d'une version récente du logiciel avant chaque séance de travail, de manière à leur permettre de tester cette version et de valider des choix implantés. La réunion suivante s'organisait autour de l'analyse des critiques de ces tests et sur les propositions et discussions de nouvelles implantations. Certains des compte-rendus rédigés dans le cadre de ces réunions sont disponibles dans l'annexe A, à la fin de ce document.

Il est important de noter que la disponibilité préalable du prototype n'a pas pu être garantie à chaque fois. En effet, certaines modifications ou implantations nouvelles, malgré l'utilisation d'un environnement de développement robuste et d'un langage de programmation permettant modularité, réutilisation et abstraction des données, nécessitaient de longues et fastidieuses phases de réalisation. L'évolution du comportement apparent du logiciel, seul critère d'avancement valable aux yeux des enseignants auteurs, n'est pas forcément identique à l'évolution des représentations internes ou du codage de ce comportement. L'exemple le plus significatif de ce problème est celui de la représentation graphique du plan (dont nous discuterons plus en détail dans le chapitre suivant, section 4.4.3). L'implantation d'éléments visuels permettant d'aider sa lecture et son interprétation a nécessité le recours à des algorithmes classiques (intersection de polygones, enveloppe convexe, ...) qu'il a fallu adapter et optimiser de manière à garantir une rapidité de calcul et une fluidité de l'affichage. C'est le respect de ces deux critères, introduits par l'aspect interactif

et dynamique de la géométrie mise en œuvre dans ***Calques 3D***, qui justifie une implantation attentive mais longue et qui contraste avec un résultat qui semble trivial à l'écran.

Dans ces situations, il a fallu procéder à des démonstrations *in situ* plutôt que de laisser les enseignants procéder à des tests par eux-mêmes.

3.5 Vers un cadre pour l'acquisition de l'expertise pédagogique

Cette approche du génie éducatif basée sur une conception incrémentale par prototypage a pour principal avantage une meilleure implication des enseignants auteurs, non-spécialistes en informatique, dans le processus de développement d'un logiciel éducatif. Ceci permet de diminuer sensiblement les risques d'inadaptation de celui-ci à leurs besoins réels.

Or, tous les problèmes du génie éducatif ne sont pas résolus par cette approche, en particulier celui de l'acquisition et de l'organisation des connaissances issues de l'expertise des enseignants. Nous avons signalé que le prototype servait de support de discussion entre les différents partenaires du processus de conception, en particulier entre l'enseignant auteur et l'informaticien. Cependant, discussion ne signifie ni acquisition de connaissances, ni spécification des besoins. Ce qui manquait lors des premières séances de travail, c'est un cadre pour formaliser la *négociation* des besoins, c'est-à-dire l'accord sur les connaissances à intégrer dans un logiciel pédagogique et sur la manière de les mettre en œuvre. Un cadre qui tienne compte à la fois du registre pédagogique et du registre informatique (figure 3.3).

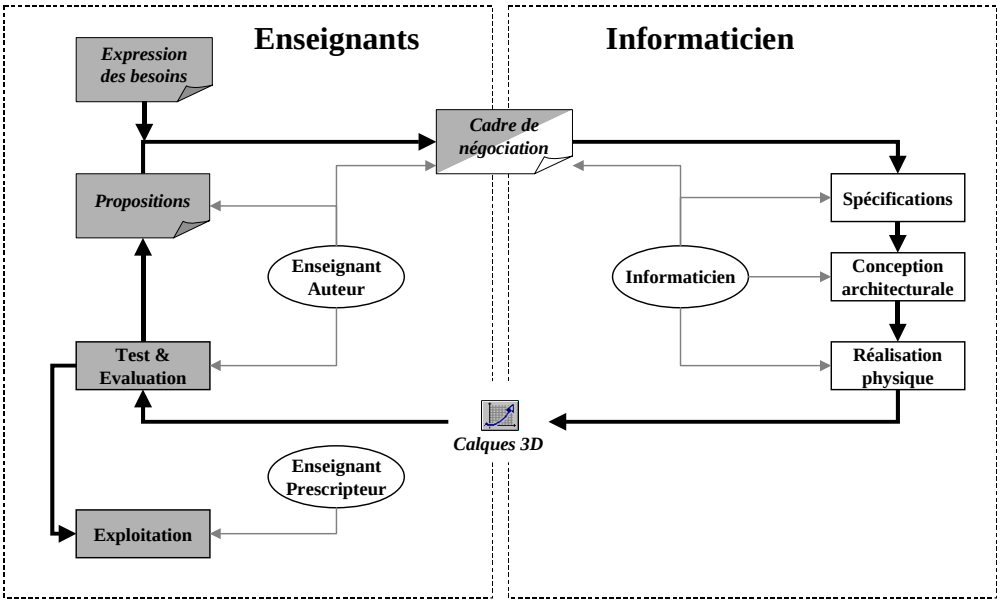


Figure 3.3 – *Un cadre pour l'acquisition de l'expertise pédagogique*

Ce problème fait l'objet du chapitre suivant.

Chapitre 4

Acquisition de l'expertise des enseignants pour *Calques 3D*

4.1 Introduction

Dans le chapitre précédent, nous avons présenté la démarche suivie pour le développement de ***Calques 3D***, c'est-à-dire un processus basé sur le prototypage incrémental et l'intégration des enseignants au centre même de la conception du logiciel. Cette intégration est indispensable car le traditionnel *cahier des charges* est un point de départ du processus de conception mais insuffisant. Il doit être complété par une phase d'acquisition de l'expertise pédagogique des enseignants et de négociation des contenus embarqués. Cette phase nécessite une prise en compte plus approfondie des situations d'apprentissage. Nous retrouvons ainsi les quatre propositions faites par [Bruillard 94] dans le cadre d'une ébauche de cadre méthodologique pour la conception des EIAO :

- partir de l'élève en situation d'apprentissage et caractériser de telles situations d'apprentissage comme fondement de la conception,
- définir un système de spécification des situations permettant de préciser et de contraindre le rôle des acteurs et des outils,
- baser l'évaluation sur ces spécifications, donnant des implications a priori sur les outils et centrer l'évaluation sur l'analyse globale des situations plutôt que sur des performances techniques des outils,
- structurer une démarche méthodologique intégrant l'ensemble des acteurs dans le processus de conception des EIAO.

Ce cadre méthodologique a, par exemple, servi de base de réflexion pour l'élaboration d'un *modèle de situation d'interaction* [Dubourg 95] dans le cadre du projet `REPÈRES`.

De manière similaire, nous avons cherché à définir un cadre permettant aux enseignants de décrire leurs souhaits, aux informaticiens d'en identifier les réalisations et, surtout, qui permette de rendre compte des différents moments d'intervention des participants.

4.2 Vers un cadre de négociation

Nous avons cherché à mettre en place un formalisme unique pour l'ensemble des participants au processus incrémental de production. Le rôle que nous avons attribué à ce formalisme est

triple :

- permettre aux enseignants la *description de contextes d'utilisation* d'un logiciel pédagogique d'une manière proche de leur expérience personnelle et de leur pratique quotidienne,
- permettre aux concepteurs d'initier un processus d'ingénierie des besoins en favorisant l'*extraction des connaissances pédagogiques* des enseignants,
- permettre aux différents participants du processus de conception de disposer d'un support à la *négociation du contenu pédagogique* du logiciel.

4.2.1 Le concept de *"contextes d'utilisation"*

Lorsque nous parlons de prise en compte de l'usager enseignant pour la conception d'un EIAO, nous envisageons un processus en deux temps. Le premier concerne l'implication de l'enseignant, en tant qu'expert du domaine, dans le processus même de conception (que nous désignons par *enseignant auteur*). Le second concerne son implication dans l'utilisation en situation d'apprentissage et l'adaptation du logiciel à ses besoins propres (l'*enseignant prescripteur*).

L'objectif de la prise en compte de l'*enseignant auteur* n'est pas seulement de comprendre et d'expliciter le contenu du domaine d'apprentissage mais aussi de formaliser *comment* ce contenu est enseigné par les enseignants et *pourquoi*, dans une situation donnée, certaines méthodes de présentation du contenu sont utilisées plutôt que d'autres.

Le développement d'un logiciel pédagogique n'est pas une fin en soi mais doit répondre à un réel besoin pédagogique pour une utilisation effective. Mais plus que la description d'une simple *utilisation* du logiciel, nous estimons indispensable la description de son *intégration* dans un processus plus large d'apprentissage ou d'enseignement. Un logiciel comme ***Calques 3D*** n'a pas vocation à remplacer toute ou partie des situations d'apprentissage mais plutôt à servir de support à la réalisation d'activités dans certaines de ces situations (visualisation collective monitorée par l'enseignant, réalisation d'exercices en mode tutoré, ...).

La prise en compte de l'*enseignant prescripteur* peut faire l'objet des mêmes remarques. Dans la plupart des cas, l'adaptabilité d'un logiciel à ses besoins spécifiques se fait au niveau de l'interface (configuration des menus par manipulation directe des commandes, boîtes de dialogue de "préférence", ...). Or, un tel paramétrage devient fastidieux dans le cas d'un logiciel trop complexe, d'autant plus qu'il n'est généralement pas le fruit du hasard ou de choix "ad-hoc" de la part de l'enseignant, mais bien plus révélateur d'une cohérence au niveau d'une situation d'apprentissage.

Pour toutes ces raisons, nous avons décidé de concevoir des documents décrivant un *contexte d'utilisation d'un logiciel pédagogique*, c'est-à-dire une description non pas de *ce que doit faire ce logiciel* mais plutôt de ce que les *utilisateurs veulent pouvoir faire avec ce logiciel.*

Dans la pratique, cela doit se traduire par la description d'activités à réaliser avec ***Calques 3D***. De plus, nous voulons que les enseignants ne se sentent pas éloignés de leur façon de faire lorsqu'ils rédigent des documents à destination de leurs classes (exercices, cours, ...). C'est pour cela que l'utilisation d'un langage formel nous semble à éviter, seule une description en langage naturel pouvant convenir.

4.2.2 Extraction de l'expertise pédagogique

L'acquisition de connaissances auprès d'experts est bien connue comme étant un exercice relativement difficile, que l'on dispose ou non des compétences d'un spécialiste de l'ingénierie de la connaissance (voir par exemple [Feigenbaum 83] et [Stefik 95]).

L'interrelation étroite existant entre expérience professionnelle et implication humaine et qui caractérise l'enseignement rend le processus encore plus difficile pour l'expertise pédagogique ([Murray 97, Murray 99], [Grandbastien 99]).

C'est afin d'atteindre cet objectif que nous devons préciser le champ d'action de ce cadre.

Premièrement, le cadre de description, même en offrant un maximum de liberté à l'enseignant grâce à une rédaction informelle, doit cependant restreindre les idées de manière à forcer une réflexion sur le processus de développement et l'identification des connaissances. Cela doit pouvoir se faire en contraignant cette description selon un cadre plus ou moins rigide, juste milieu entre l'ambiguïté des descriptions informelles et la complexité des spécifications formelles.

Deuxièmement, il ne s'agit pas seulement d'obtenir la description d'activités à réaliser AVEC mais plutôt AUTOUR de ***Calques 3D***, c'est-à-dire aussi bien des activités utilisant le logiciel que des activités "papier-crayon" traditionnelles, aussi informatives que les précédentes pour l'intégration des connaissances dans le logiciel. [Major 92] fait remarquer que la prise en compte de telles activités est souvent une demande importante de la part des enseignants. L'implantation, étapes par étapes, des connaissances pédagogiques issues de l'analyse de ces diverses activités, permet d'assurer non seulement l'évolution du logiciel (ajout de nouvelles fonctionnalités, renforcement de l'existant, ...) mais aussi de conserver la relation de ces ajouts à des fondements pédagogiques. De plus, ces descriptions par l'enseignant nous permettront de mieux cerner la place que peut occuper le logiciel dans l'enseignement traditionnel de la géométrie.

4.2.3 Négociation du contenu pédagogique

L'enseignement d'un domaine ne consiste pas en une simple transmission de la théorie mais le plus souvent en une transposition de celui-ci (*transposition didactique*), imposée par des contraintes d'ordre pédagogique (niveau des élèves, cadre de la formation, états des prérequis, ...). De même la médiatisation d'un domaine d'apprentissage résulte souvent elle aussi en une transposition similaire (*transposition informatique* [Balacheff 91, Balacheff 94b]), due cette fois à des contraintes d'ordre informatique (nature du logiciel éducatif, limites technologiques, contraintes ergonomiques, ...). Il n'est pas risqué d'affirmer que ces transpositions sont du ressort des spécialistes et donc uniquement discernables par ceux-ci. Ces transpositions doivent être discutées et justifiées de manière à conserver l'adhésion de tous les participants au logiciel. Or, arriver à un compromis qui satisfasse les points de vue des différents participants nécessite souvent de relâcher des contraintes, d'un côté ou de l'autre.

La conception d'un logiciel pédagogique n'est donc pas un simple transfert d'un domaine disciplinaire sur un nouveau support. Il s'agit plus d'une *co-construction*, faisant intervenir différents partenaires (enseignants, informaticiens, didacticiens, ...), d'un nouveau dispositif d'apprentissage dont les possibilités et les potentialités ne sont pas forcement connues à l'avance. L'une des conséquences d'un tel dispositif peut être par exemple d'amener les enseignants à imaginer de nouvelles approches pédagogiques, de nouvelles problématiques d'apprentissage, ... comme ce fut le cas par exemple pour la géométrie dynamique.

En d'autres termes, la transposition informatique peut être assimilé à une phase de *création pédagogique* qui se doit d'être partagée et négociée entre les participants au processus de conception d'un logiciel pédagogique. C'est pourquoi nous qualifions cette phase de *négociation*

plutôt que de *discussion*, même si ce terme a une connotation péjorative en sous-entendant un conflit.

Notons enfin que la négociation n'est pas restreinte aux seuls contenus pédagogiques mais aussi au cadre lui-même. En effet, nous ne prétendons pas élaborer un cadre de négociation qui soit adéquat dès sa première mouture. La proposition doit donc être une base de travail que les enseignants doivent pouvoir utiliser sans préjugés, quitte à la modifier si elle ne convient pas à leur compréhension.

4.2.4 Identification des connaissances à acquérir

Nous avons vu que l'objectif principal de ce cadre de négociation est de servir de support à l'acquisition de connaissances. Il convient maintenant de préciser quelles sont ces connaissances. Il ne s'agit pas ici d'en proposer une nouvelle typologie mais plutôt de préciser celles qui, aux vus des objectifs pédagogiques de ***Calques 3D***, peuvent nous être utiles. En d'autres termes, l'idée est ici de préciser la forme de ce cadre de négociation en identifiant les *champs* que les enseignants auront à remplir pour décrire un contexte d'utilisation.

Sans recourir à une analyse cognitive approfondie, nous pouvons estimer que la réalisation d'une activité posée dans l'enseignement de la géométrie implique plusieurs types de connaissances :

- Des *connaissances statiques* : faits et concepts du domaine.
- Des *connaissances procédurales* : méthodes permettant la mise en œuvre des connaissances statiques.
- Des *connaissances stratégiques* : stratégies utilisées pour organiser l'activité et atteindre les objectifs ; indices à relever dans les productions de l'élève pour choisir telle ou telle stratégie.

L'identification des connaissances statiques ne pose pas de problèmes a priori, puisque ***Calques 3D*** est un micromonde, sans capacité de raisonnement, de diagnostic ou d'aide. Les concepts du domaine se résument donc à une *liste des objets géométriques* présents dans l'activité.

De même, l'identification des connaissances procédurales est aussi immédiate, même si l'expérience a prouvé qu'elle introduit un risque de conflits et donc la nécessité de choix. Les objectifs de ***Calques 3D*** sont de permettre à l'apprenant de construire, d'observer et d'explorer une figure géométrique (cf. section 2.5). Pour ce faire, il faut donc disposer de *fonctions de construction*, de *fonctions d'observation* et de *fonctions d'exploration*.

Par contre, l'identification des connaissances stratégiques pose un peu plus de problèmes car, non seulement elles recouvrent un grand nombre de connaissances de natures différentes, mais l'essentiel des implicites et non-dits des enseignants s'y retrouvent.

L'organisation de l'activité par l'enseignant implique la sélection d'un ensemble de concepts et de méthodes parmi les éléments disponibles dans les connaissances. L'identification des critères de sélection permet de mettre en place des *outils de configuration* du contexte d'utilisation.

Les concepts du domaine sont rarement enseignés de manière théorique mais réifiés selon une forme dépendant de la situation d'apprentissage. Ces réifications donnent lieu à un choix d'un *mode de présentation* privilégié du concept.

Les critères utilisés pour l'activation d'une stratégie pédagogique sont souvent liés à des *erreurs* produites par l'élève, à ses capacités à surmonter les *difficultés* inhérentes à l'activité.

Ces difficultés peuvent être de nature différente et faire l'objet d'*aides* adaptées tout aussi variées [Blondel 96] : l'accès à un lexique peut être proposé face à des difficultés de vocabulaire ; une démonstration ou un exemple type peut remédier à des difficultés de savoir-faire, un contre-exemple à des difficultés d'explication, ...

4.3 Les contextes d'utilisation

Dans cette section, nous présentons la méthode suivie afin d'élaborer ce cadre de négociation que nous avons appelé *"contexte d'utilisation de logiciels pédagogiques"*. A chaque étape de l'élaboration, nous mettons en évidence ce que nous attendions des enseignants, les informations que nous espérions en retirer et les besoins de négociation qui en ont découlés.

4.3.1 Élaboration des documents

L'élaboration des *contextes d'utilisation* s'est effectuée en plusieurs étapes successives.

1. Nous sommes partis d'un modèle général issu de travaux en Sciences de l'Éducation : la définition d'une *séquence pédagogique* proposée dans la série *"Fenêtre active"*, publiée par le CDRP [11] de Lorraine. Cette collection regroupe un ensemble de fascicules à l'intention des enseignants et couvrant un large éventail de disciplines (histoire [Colotte 88], français [Guillet 88], mathématiques [Bénézra 89, Chouanière 91], ...). Elle propose une méthodologie de construction d'activités pédagogiques utilisant l'ordinateur comme outil privilégié (mais non exclusif).
   Cette méthodologie est le résultat du travail d'une équipe d'enseignants, regroupée au Centre de Ressources Informatique (CRI) de l'académie de Nancy-Metz entre 1986 et 1992, dont l'approche consistait à se poser les questions : *"Quels apprentissages voulons-nous provoquer chez l'élève ? Pour cela, quelles activités proposons nous ? De quels outils disposons nous et quels sont les mieux adaptés à chaque objectif et à chaque élève ?"*. Ces enseignants ont d'abord travaillé ensemble sur l'acte pédagogique (cf. [Meirieu 87]), les différents modes d'apprentissage et sur le rôle que pouvait jouer l'outil informatique. Ils se sont ensuite fixé pour objectif de produire, d'expérimenter et de diffuser des séquences pédagogiques illustrant des utilisations variées et pertinentes de l'ordinateur dans l'enseignement des disciplines. Ce sont ces séquences qui constituent la matière de la série *"Fenêtre active"*.
2. La deuxième étape de l'élaboration, correspondant à une simplification de ce modèle théorique, a consisté à prendre en compte nos objectifs de développement (figure 4.1). En effet, dans ce modèle initial, l'accent était mis surtout sur la description de la séquence en tant que module unifié. En fonction de nos propres objectifs, et parce qu'une séquence pédagogique contient des informations qui ne sont d'aucune utilité pour le développement de ***Calques 3D***, il nous semblait plus pertinent de centrer l'extraction des connaissances sur la description d'activités appartenant à une séquence plutôt que sur la description de la séquence elle-même.
3. La troisième étape a consisté à instancier ce modèle avec le domaine d'apprentissage couvert par ***Calques 3D***, c'est-à-dire la géométrie dans l'espace. Bien entendu, le logiciel n'a pas vocation à couvrir la totalité de ce vaste domaine : les objectifs pédagogiques initialement attribués à ***Calques 3D*** le définissent comme un micromonde de construction,

11. Centre Régional de Documentation Pédagogique

d'observation et d'exploration de figures géométriques. La démonstration d'un problème ou la rédaction de sa solution sont typiquement des exemples d'activités pédagogiques qui resteront à la charge de l'enseignant, car non supportées directement par le micromonde.

Les documents proposés pour la description d'activités à réaliser autour de ***Calques 3D*** sont une "traduction" libre de cette *séquence pédagogique*. Dans sa forme actuelle, ce cadre de description se compose de deux parties : une fiche pour la description générale de la séquence (*le contexte*), une fiche pour la description de chaque *activité* de la séquence. L'exemple qui illustre cette section est une séquence réelle rédigée par l'un des membres des groupes de travail : la figure 4.2 décrit la séquence pédagogique, les figures 4.4, 4.5 et 4.6 décrivent chacun des trois exercices qui la compose.

4.3.2 Description de la séquence pédagogique

Cette expression ne désigne pas une organisation selon une unité temporelle (une heure de cours par exemple), mais selon une unité de contenu, c'est-à-dire à un ensemble d'heures constituant un tout (par exemple un "chapitre" pour certaines disciplines, une "unité" pour d'autres).

Elle est composée de quatre temps (figure 4.1) :

- le choix des *objectifs pédagogiques* de la séquence,
- la constitution d'un ensemble d'*activités d'apprentissage et d'enseignement* correspondant à ces objectifs,
- la définition de l'*évaluation sommative*, c'est-à-dire le bilan des acquis des élèves par rapport à ces objectifs après le temps d'apprentissage, et de l'*évaluation formative*, c'est-à-dire qui porte sur le processus d'acquisition lui-même (enseignement et apprentissage) et qui fonde la mise en place de la remédiation,
- la proposition d'*activités de remédiation* pour permettre à l'élève n'ayant pas atteint un objectif de se rattraper.

La réalisation d'une séquence pédagogique dépend de trois variables :

- le contenu disciplinaire de la séquence, c'est-à-dire une explicitation des connaissances du domaine et des savoir-faire à mettre en œuvre (i.e. géométrie filaire et géométrie du solide),
- les élèves à qui s'adresse la formation : d'une part, le choix du public est conditionné par les prérequis de la séquence imposée ; d'autre part, la nature du public influe sur la nature des activités proposées dans la séquence (type d'activité, vocabulaire, rythme, ...),
- l'enseignant qui gère le fonctionnement de la séquence (pédagogie personnelle, contraintes d'enseignement, situation d'apprentissage, ...).

Chacune des "données" de la séquence contient des informations qui peuvent être utiles pour le développement de ***Calques 3D*** et que nous avons reportées dans la fiche de description (figure 4.2). Par exemple, la description du public visé (niveau, formation continue ou professionnelle, ...) permet d'associer le vocabulaire utilisé dans les fiches au contexte de cette formation (par exemple, on parlera de *"droite perpendiculaire à un plan"* dans certains cas, de *"normale à un plan"* dans d'autres) ou les prérequis attendus, en terme de savoirs et de savoir-faire. La description de la durée de la séquence, de l'articulation de cette séquence avec le reste de la formation permet de prendre en compte, dans une certaine mesure, des contraintes pédagogiques de l'enseignant.

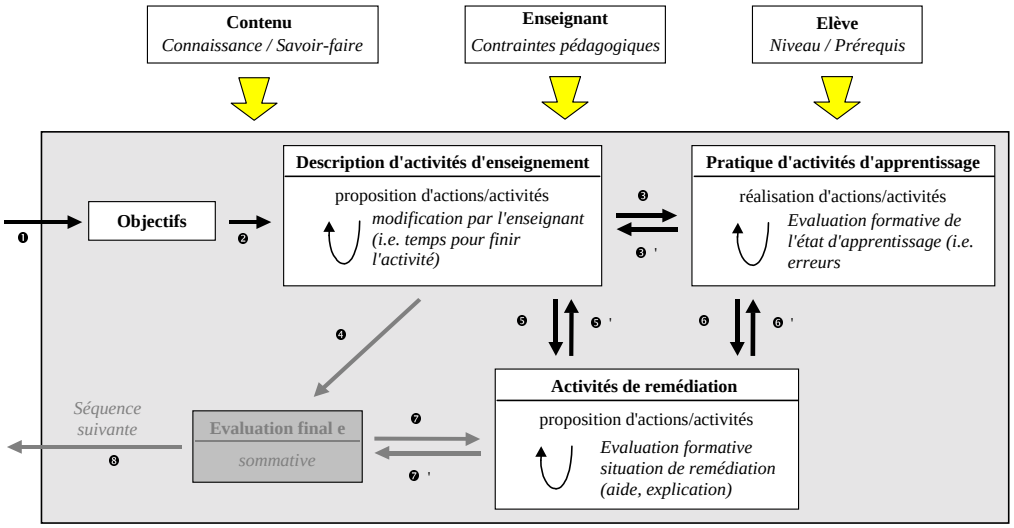


Figure 4.1 – *Définition d'une séquence pédagogique*

Nous nous sommes attachés à expliciter les informations qui restaient relativement générales ou imprécises. En particulier, nous avons fait détailler les objectifs pédagogiques de la séquence, préciser l'utilisation des phases d'évaluations sommative et formative et, surtout, recentrer la description d'une séquence sur les activités qui la composent.

a) Objectifs pédagogiques

Fournir un cadre pour la description des objectifs pédagogiques d'une séquence s'est avéré être un problème plus difficile que nous l'imaginions. Par exemple, le texte suivant, extrait de la séquence d'illustration (figure 4.2), consiste plus en une description ou un résumé de la séquence qu'en de réels objectifs pédagogiques :

> *"Le but de ces trois exercices est [...] d'étudier les différents cas d'intersection de deux droites dans l'espace pour :* (1) *trouver une condition nécessaire pour que deux droites de l'espace soient sécantes,* (2) *trouver une condition nécessaire pour que deux droites de l'espace soient parallèles et* (3) *trouver les différents cas d'intersection de deux droites."*

Ce problème a trouvé un développement intéressant lorsque nous avons travaillé avec le groupe d'enseignants du technique. La forme d'enseignement pratiqué dans les lycées techniques renforçait nettement le contrat pédagogique liant enseignant et apprenants via un contrôle continu et une forme d'auto-évaluation formative[12]. Les objectifs pédagogiques de toute séquence étaient donnés à l'élève en termes de *capacité*, c'est-à-dire de compétences à acquérir, de savoir-faire à maîtriser, ...

Les objectifs de la séquence sont décrits selon les quatre catégories suivantes :

- *information* : capacité d'observation et d'analyse de l'élève,
- *opération* : capacité de pratique d'une opération et d'auto-initiation à un problème,

12. Ce système, plus proche des *curriculum* anglo-saxons, qui n'existait plus que dans les lycées techniques, a été définitivement abandonné au profit du contrôle final des connaissances et de l'évaluation sommative, standard de l'éducation nationale française.

Contexte

Nom de la séquence : intersection de droites dans l'espace

Mots-clefs : droites coplanaires, intersection de droites

Description : Le but des trois exercices est de réfléchir et d'apporter une réponse aux "questions" ci-dessous. Il s'agit d'étudier les différents cas d'intersection de deux droites de l'espace pour (1) trouver une condition nécessaire pour que deux droites de l'espace soient sécantes, (2) trouver une condition nécessaire pour que deux droites de l'espace soient parallèles, (3) trouver les différents cas d'intersection de deux droites

Auteur : C.M.

Date de création : 01/03/99

Public : Secondes générales

Cadre de la formation : formation initiale technique ; formation continue

Durée : 2 heures

Situation : travail par groupe de deux élèves dans le cadre des modules, les élèves étant répartis d'après l'évaluation de rentrée

Prérequis

Concept :

Savoir-faire. : connaître et utiliser les règles de la représentation en perspective

Articulation dans la séquence :

Objectif ***(selon les 4 catégories suivantes)***

1. Information : Analyser une figure en isolant les éléments significatifs

2. Opération : Elaborer et organiser une démarche

3. Maîtrise : Conjecturer

4. Expertise : Identifier les différents cas d'intersection de droites dans l'espace

Documents fournis :

- La figure de l'exercice 1 est donnée dans le fichier "tétra.c3d"
- Le cube de l'exercice 3 est donné avec les sommets correctement nommés dans le fichier "cube.c3d"

Activités caractéristiques de la séquence

Activités préparatoires :

Pour l'enseignement :

Pour l'apprentissage : Exercice 1, Exercice 2, Exercice 3

Pour la remédiation :

Activités de synthèse : Conclusion : donner une réponse aux questions posées plus haut

Figure 4.2 – *Un exemple de séquence pédagogique*

- *maîtrise* : capacité de critique et de validation d'une réalisation,
- *expertise* : capacité de compte-rendu et de synthèse.

La description des objectifs selon cette catégorisation permet de disposer d'une base de discussion pour les objectifs et fonctionnalités du logiciel. Par exemple, l'objectif *"Analyser une figure en isolant les éléments significatifs"* (figure 4.2) souligne une fois de plus le problème majeur de la géométrie dans l'espace : la lecture d'une figure. Ce problème est à mettre en relation avec une pratique importante de l'enseignement de la géométrie dans l'espace, qui consiste à se replacer localement dans le cadre de la géométrie plane pour y réutiliser théorèmes et propriétés, souvent mieux maîtrisés par l'élève. Une pratique similaire existe aussi dans le contexte de la géométrie plane et avait déjà été prise en compte dans `Calques 2` [Bernat 94a] sous la forme des *calques*. Cela nous a donc amené à réfléchir à une transposition de cet outil dans un univers en trois dimensions.

b) Évaluations sommative et formative

En décidant, dès le début de ce travail, de concevoir ***Calques 3D*** comme un micromonde de géométrie dynamique, nous avons volontairement favorisé la libre exploration d'un domaine par l'élève au détriment de la vérification de ses productions ; à ce titre, ***Calques 3D*** ne dispose d'aucun mécanisme d'analyse ou de modélisation des erreurs. La phase d'*évaluation sommative* finale (en grisé sur la figure 4.1) a été délibérément mise de côté lors de la conception des fiches de description.

Par contre, l'*évaluation formative* a été non seulement conservée mais renforcée en tant que partie intégrale des activités. Elle présuppose une analyse a priori de chaque activité par l'enseignant de manière à lui permettre d'évaluer a posteriori la validation de ses objectifs par l'élève. L'explicitation des résultats de cette analyse est d'une importance capitale pour nous car elle permet de mettre en évidence les difficultés introduites par l'activité et les erreurs généralement commises par les élèves lors de sa réalisation. C'est à partir de ces informations que des *fonctionnalités d'aide* pourront être proposées pour pallier les problèmes.

c) Activités de la séquence

Par rapport au schéma original, nous avons clairement séparé la description des activités donnée par l'enseignant (*activité d'enseignement*) de leur pratique par l'élève (*activité d'apprentissage*) ; la première correspond au savoir-faire de l'enseignant et à une prévision de l'organisation de la séquence en dehors de celle-ci, permettant ainsi un contrôle de la situation ; la deuxième correspond réellement aux activités que l'élève aura à effectuer. La différenciation des trois types d'activités (*apprentissage*, *enseignement* et *remédiation*), auxquelles nous pouvons rajouter des activités *préparatoires* et *de synthèse*, permet de mettre en évidence les particularités de chacune et donc d'identifier des *contextes d'usage* différents. Nous pouvons en effet supposer que l'utilisation du logiciel pour la démonstration d'un concept par l'enseignant ou pour la construction d'une figure par l'élève va entraîner des ensembles différents de fonctionnalités, des modalités d'accès différentes à ces fonctionnalités ... Cela induit donc la nécessité d'un mécanisme de configuration permettant l'adaptation du logiciel à ces différents contextes d'usage.

4.3.3 Description des activités de la séquence

La deuxième étape dans l'élaboration des fiches de description a consisté à adapter la notion d'*activité* au contexte particulier de ***Calques 3D***. Il s'agit bien entendu de prendre en compte la géométrie dans l'espace comme domaine d'application mais aussi de mettre l'accent sur les types d'activité qui peuvent être mises en œuvre autour du logiciel.

Nous avons cherché à caractériser ce que signifiait *pratiquer une activité* en géométrie. Une des réponses possibles, proposée par [Chouanière 91] dans le cadre de la résolution de problèmes en géométrie, consiste à identifier les différentes étapes élémentaires que l'élève doit accomplir pour finaliser la tâche (figure 4.3).

1. Identification des objets géométriques issus de l'énoncé
2. Construction de la figure demandée
3. Vérification de la figure en identifiant les propriétés
4. Conjecture et résolution du problème demandé
5. Démonstration de la solution
6. Rédaction de la solution

Figure 4.3 – *Les phases de la résolution de problèmes en géométrie*

Il est évident que toutes ces phases de la résolution d'un problème ne relèvent pas des objectifs pédagogiques du logiciel. Ainsi, la phase de *rédaction* de la solution est définitivement hors de propos. Les phases de *conjecture* et de *démonstration* ne sont pas prises en charge directement par le logiciel mais celui-ci peut y contribuer indirectement : la première peut être assimilée à une phase de conjecture visuelle, la deuxième assimilée à la construction (et à la vérification) de la figure demandée.

En d'autres termes, et par rapports aux objectifs d'une séquence pédagogique (cf. page 47), nous nous intéressons à une description d'activités qui relèvent principalement des catégories *information* et *opération* de la séquence.

Déclinée dans le contexte de ***Calques 3D***, la description de chacune de ces étapes par l'*enseignant auteur* nous permet de proposer un début d'identification des éléments à acquérir. C'est pour cette raison que la fiche de description d'une activité se présente sous une forme très opérationnelle, en décrivant ce qui pourrait être les fonctionnalités de ***Calques 3D***. Ces fonctionnalités ont été regroupées en trois catégories :

- l'identification des objets géométriques,
- l'identification des fonctions, déclinées selon les objectifs de ***Calques 3D*** : fonctions de construction, fonctions de visualisation et fonctions d'exploration,
- l'identification des aides pouvant lever ou réduire les difficultés inhérentes à l'activité.

a) Listes des objets géométriques

La première phase de l'acte pédagogique (*identification des objets géométriques*), ainsi que la phase de *construction de la figure*, par l'identification des objets intermédiaires, permettent à l'enseignant auteur de décrire les objets géométriques mis en œuvre dans l'activité. Nous avons souhaité que cette description se fasse au niveau des concepts (identification des types d'objets :

Caractéristiques de l'activité ***(niveau d'objectif 1 et 2)***

Nom de l'activité : Exercice 1 .....

Nature : Apprentissage .....

Énoncé :

Ouvrir le fichier "tétra" et observer la figure sur l'écran de l'ordinateur afin de répondre aux questions posées ci-dessous.
Une pyramide ABCD est représentée en perspective cavalière.
Le point I est le milieu du segment [AB].
Le point J est le milieu du segment [AC].
Les points K et L sont deux points du segment [AD], distincts de son milieu et de ses extrémités.
Compléter les tableaux ci-dessous par oui ou par non.

Le point K est un point : (1) du plan (ACD)..... , (2) du plan (BCD) .....
Les droites (IK) et (BD) sont : (1) dans un même plan ..... , (2) sécantes .....
[...]

Remarques : pour mieux analyser la figure, on pourra :

Effectuer des rotations de la figure.
Changer le référentiel (en ajoutant ou en enlevant le "plancher", les "cloisons", le repère).
Changer la couleur de certains éléments.
Isoler les droites étudiées dans un calque.

A la fin de cette étude, fermer le fichier sans enregistrer les modifications.

Liste des objets géométriques	**Présentation de l'objet**
Points, droites, segments, plans	.....
Tétraèdre	.....
Fonctions de construction	**Autorisée/Non autorisée**
Aucune fonction de construction autorisée	.....
Fonctions de manipulation	**Autorisée/Non autorisée**
Gommer	Autorisée .....
Copier dans un calque	Autorisée .....
Changer la couleur d'un objet	Autorisée .....
Fonctions de visualisation	**Autorisée/Non autorisée**
Choix du référentiel	Autorisée .....
Modification du point de vue	Autorisée .....
Projection frontale	Autorisée .....
Visualisation des calques	Autorisée .....

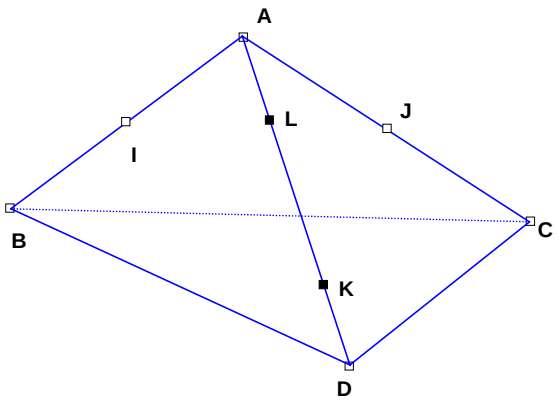


Figure 4.4 – *Première activité de la séquence*

point, droite, plan, cube, ...) mais aussi au niveau des modes de présentation des objets (attributs visuels et textuels : formes et couleurs du tracé, définition de l'objet, ...).

Deux remarques s'imposent à ce niveau. Premièrement, l'introduction d'un nouveau type d'objet dans une fiche de description peut ne pas aboutir immédiatement à un concept bien établi. Un tel problème est illustré dans le cas de l'exercice 1 de la séquence (figure 4.4) où un objet géométrique dénommé *tétraèdre* est identifié. Au moment de la rédaction de cette fiche de description, cet objet du domaine n'est pas défini dans les connaissances de ***Calques 3D*** et il convient de se poser plusieurs questions : *"Qu'est-ce qu'un tétraèdre ? S'agit-il d'un objet élémentaire de* ***Calques 3D****, au même titre que le point ou la droite ? S'agit-il au contraire d'une figure élaborée à partir d'objets élémentaires et quel est son statut conceptuel ? ..."*. Les seules fiches de description ne permettent pas de répondre directement à ces questions mais servent de support à une négociation entre les participants pour l'explicitation du concept.

Deuxièmement, nous pouvons noter que, dans les exemples d'activité fournis, les modes de présentation des objets géométriques sont rarement indiqués. Parmi les fiches obtenues, seules de rares exceptions imposaient certains attributs visuels pour tel objet particulier. Cela s'explique par le processus incrémental du développement qui permet la réutilisation des éléments préalablement décrits. Ainsi, pour éviter une description trop laborieuse, nous avons supposé que l'absence d'information indiquait qu'aucun mode de présentation n'est privilégié pour cette activité.

b) Fonctions de construction, de visualisation et d'exploration

La phase de *construction de la figure*, en plus d'introduire les objets intermédiaires de la figure, permet à l'enseignant auteur de décrire les moyens de mettre en œuvre les relations géométriques dans la figure (la fonction), ainsi que les objets nécessaires à l'application de la relation (les arguments de la fonction).

Dans l'exercice 2, la description de la fonction *"Construction du milieu d'un segment"* permet de préciser la relation à mettre en œuvre (milieu) et les arguments (un segment). Par contre, la fonction *"Construction d'un plan"* ne permet pas immédiatement l'identification de ses arguments : s'agit-il de pouvoir construire un plan défini par trois points, défini par une droite et un point, perpendiculaire à une droite et passant par un point, ...?

La phase de *vérification de la figure* permet de réfléchir aux fonctions permettant d'aider l'élève à vérifier par lui même l'exactitude de sa production. Outre la "vérification par essais-erreurs" consistant à construire des objets intermédiaires (par exemple construire l'intersection de deux droites pour "vérifier" leur coplanarité), ces fonctions sont principalement des fonctions de visualisation : modifier le point de vue de l'observateur, changer la perspective, marquer "visuellement" certains éléments significatifs de la figure (changer la forme ou la couleur d'objets, dessiner des marques de parallélisme ou d'égalité de longueur, matérialiser les coordonnées des points, ...).

Mais la définition de fonctions plus directes n'est pas à négliger. En effet, l'absence du registre analytique de la géométrie dans le cahier des charges initial de ***Calques 3D*** n'exclut pas la mise en œuvre de fonctionnalités permettant la vérification de propriétés géométriques élémentaires (appartenance, parallélisme, orthogonalité, ...), le logiciel pouvant répondre par l'affirmative ou la négative à des questions du genre *"La droite D1 est-elle parallèle à la droite D2 ?"* ou *"Le point M est-il le milieu du segment [AB] ?"*.

D'un autre côté, cette phase de vérification peut être laissée à l'entière initiative de l'élève, auquel cas il peut être intéressant de lui fournir des fonctions l'aidant dans la réalisation de cette

tâche. Par exemple, dans le cas de l'exercice 2 (figure 4.5), s'il est d'un intérêt peu évident de faire identifier automatiquement par le logiciel la nature du quadrilatère IJNM, des fonctions permettant sa visualisation en *projection frontale* ou son *extraction dans un calque* peuvent amener l'élève à procéder par lui-même à son identification, sans pour autant lui fournir la réponse exacte. Là encore, c'est uniquement le *contexte d'usage* du logiciel qui peut permettre la mise en avant de telle ou telle approche.

Calques 3D n'est pas un logiciel d'aide à la démonstration, la phase de *conjecture/résolution du problème* ne peut pas être prise en compte directement. Mais certaines fonctions de visualisation (extraction d'objets dans un calque, affichage d'un lieu géométrique, ...) ou d'exploration (déformation de la figure, ...) peuvent fournir une première conjecture sur les propriétés de la figure. De même, dans certaines activités, la *démonstration* peut être associée à la construction de la solution, permettant ainsi l'identification d'autres fonctions de construction.

Enfin, de manière à associer les diverses fonctions au contexte d'usage, la disponibilité des diverses fonctions lors de la réalisation de l'activité est aussi à préciser. Le premier exercice de la séquence par exemple est une activité amenant l'élève à observer une figure et à en identifier certaines propriétés. A ce titre, aucune fonction de construction n'est rendue accessible à l'élève, seulement des fonctions de visualisation et d'exploration. Dans d'autres situations, certains fonctions seulement peuvent être inhibées.

c) Difficultés et aides

L'identification des erreurs classiques commises par les élèves, la description des difficultés inhérentes à l'activité ont été des informations relativement difficiles à obtenir explicitement.

Comme pour les objectifs pédagogiques de la séquence, les descriptions étaient souvent très générales et peu représentatives de l'activité (*"difficulté de lecture d'une figure"*, "", ...). Dans d'autres situations, généralement liées à la description d'activité avec le logiciel seul, les difficultés décrites ne concernaient que l'utilisation de ***Calques 3D*** (*"Les élèves ont du mal à déplacer un point dans l'espace"*, *"Impossibilité de nommer tous les sommets d'un cube"*, ...). Ces informations sont utiles en soi mais ne correspondent pas particulièrement à nos attentes.

4.4 Négociation : quelques exemples de problèmes

Pour illustrer ce besoin d'un accord sur les connaissances à implanter dans ***Calques 3D***, sur leur nature et leurs influences sur le développement, nous allons détailler trois exemples extraits de nos réunions de travail : le problème de la matérialisation de l'espace, le problème des objets composites et le problème de la présentation du plan.

4.4.1 Matérialisation de l'espace : le choix d'un référentiel

Nous avons souligné (cf. chapitre 2.2) que l'une des principales difficultés de la géométrie dans l'espace résidait dans la perte de la troisième dimension. Entre autres incidences de cette perte, la plus importante est certainement la difficulté du repérage de la position des objets géométriques dans l'espace et leur position relative les uns par rapport aux autres.

La matérialisation de l'univers géométrique, sous une forme ou une autre, est un moyen pouvant s'avérer efficace pour diminuer ou éliminer ce problème de la localisation dans l'espace. Dans la littérature, l'exemple le plus courant d'une telle matérialisation de l'univers est le repère

Caractéristiques de l'activité ***(niveau d'objectif 1 et 2)***

Nom de l'activité : Exercice 2 .....

Nature : Apprentissage .....

Énoncé :

Construire un tétraèdre ABCD.
Soit I le milieu de [AB], J le milieu de [AC], M le milieu de [BD] et N celui de [DC].
Quelle est la nature du quadrilatère IJNM ?
Démontrer la conjecture.
Remarques :
Ouvrir un nouveau fichier pour y réaliser la construction.
Pour mieux analyser la figure, on pourra.
isoler le quadrilatère dans un calque.
Imprimer la figure obtenue.

Liste des objets géométriques	**Présentation de l'objet**
idem Exercice 1	.....
Fonctions de construction	**Autorisée/Non autorisée**
Construction d'un segment	Autorisée .....
Construction d'un point sur segment	Autorisée .....
Construction du milieu d'un segment	Autorisée .....
Construction d'un plan	Autorisée .....
Intersection de deux droites	Non autorisée .....
Intersection de deux plans	Autorisée .....
Nommer un objet	Autorisée .....
Fonctions de manipulation	**Autorisée/Non autorisée**
idem Exercice 1	.....
Fonctions de visualisation	**Autorisée/Non autorisée**
idem Exercice 1	.....

Erreurs classiques identifiées dans l'activité :

- .....
- .....

Difficultés de l'activité et aides à apporter pour réduire la difficulté		
Type de la difficulté	**Nature de l'aide**	**Libellé, contenu, texte, ...**
.....	.....	.....
.....	.....	.....

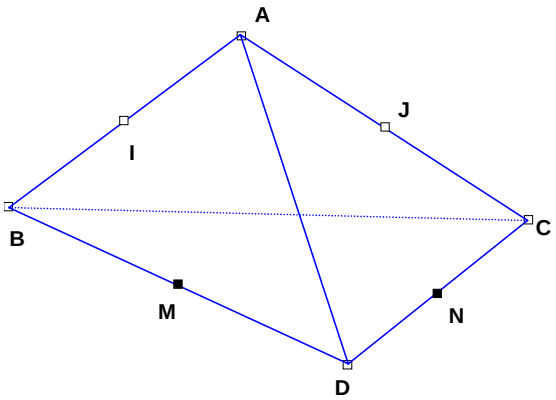


Figure 4.5 – *Deuxième activité de la séquence*

Caractéristiques de l'activité ***(niveau d'objectif 1 et 2)***
Nom de l'activité : Exercice 3 .....
Nature : Apprentissage .....

Énoncé :

Soit un cube ABCDEFGH.
On appelle Z le milieu de [AB] et V celui de [HG].
On notera a la longueur de son arête, calculer les longueurs ZE, ZF, FV et VE.
Peut-on conclure que ZEVF est un losange?
Justifiez votre réponse.
Remarques :
Ouvrir le fichier "cube.c3d".
Pour mieux analyser la figure, on pourra isoler ZEVF dans un calque et modifier le point de vue de la figure.

Liste des objets géométriques	**Présentation de l'objet**
idem Exercice 2.....	.....
Fonctions de construction	**Autorisée/Non autorisée**
idem Exercice 2.....	.....
Fonctions de manipulation	**Autorisée/Non autorisée**
idem Exercice 2.....	.....
Fonctions de visualisation	**Autorisée/Non autorisée**
idem Exercice 2.....	.....

Erreurs classiques identifiées dans l'activité :

- .....
- .....

Difficultés de l'activité et aides à apporter pour réduire la difficulté		
Type de la difficulté	**Nature de l'aide**	**Libellé, contenu, texte, ...**
.....	.....	.....
.....	.....	.....

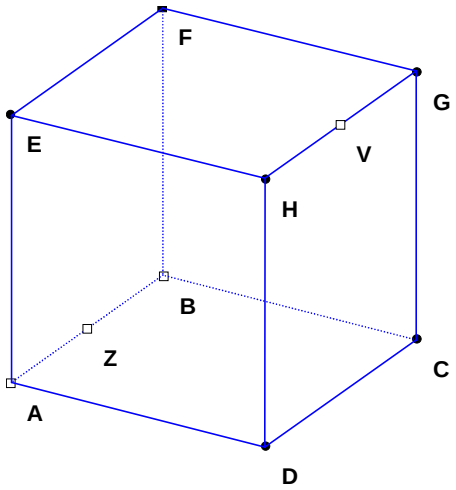


Figure 4.6 – *Troisième activité de la séquence*

orthonormé, constitué d'une origine et de trois axes orthogonaux. C'est aussi le choix qui a été fait dans certains micromondes comme `Cabri 3D` [Qasem 97] ou `Dessiner l'Espace` [Bernat 89].

Il semblait donc indispensable d'implanter dans ***Calques 3D*** cette matérialisation de l'espace. Or, à la première présentation du prototype aux enseignants auteurs, ce choix a été immédiatement rejeté pour deux raisons :

- La première raison est de nature pédagogique : le repère orthonormé est une représentation surtout utilisée dans le contexte de la géométrie analytique. Or, nous avions fixé dans les objectifs de ***Calques 3D*** que la partie analytique de la géométrie ne ferait pas partie du développement du prototype.
- la deuxième raison, provenant de l'usage du repère et de l'expérience des enseignants, est que le repère orthonormé ne fournit pas une base de référence suffisante pour favoriser la lecture des objets dans l'espace.

Nous avons donc cherché à matérialiser l'univers selon d'autres paradigmes permettant une meilleure lecture de la position relative des objets dans l'espace. Les discussions avec les enseignants et les tests successifs du logiciel nous ont permis de mettre au point un certain nombre de propositions : utilisation d'un plancher horizontal, matérialisation des trois plans du repère orthonormé, ... Aucune de ces propositions n'est indispensable, aucune n'est obligatoire, toutes ont une utilisation dépendant, au mieux d'un contexte d'utilisation bien précis, sinon du choix personnel de l'utilisateur.

Cette ambiguïté d'utilisation d'un concept pose une question importante et générique : que faut-il réaliser pour permettre ce choix ? Dans le cas d'un référentiel privilégié par une situation d'apprentissage, ce choix revient à l'enseignant prescripteur qui doit pouvoir imposer sa décision à l'élève : une configuration au niveau de la session de travail est donc nécessaire. Lorsque la nature exacte du référentiel importe peu, le choix est laissé à l'initiative de l'élève : il doit pouvoir modifier facilement ce choix. Dans les deux cas, une modalité d'accès adaptée est nécessaire.

Ce problème de la responsabilité du choix est récurrent et doit être négocié entre les enseignants, qui peuvent ne pas apprécier de voir un logiciel leur "voler" certaines de leurs prérogatives, et les informaticiens car la mise en œuvre de l'interface d'accès à ce choix peut se révéler difficile ou irréaliste.

4.4.2 Objets élémentaires et objets composites

La définition des concepts, c'est-à-dire des objets géométriques, s'est déroulée de manière relativement aisée lorsqu'il s'agissait des objets élémentaires (le point, la droite, le plan, ...). Mais les problèmes de définition, ou plutôt d'*accord* sur ces définitions, ont commencé à surgir lorsque nous avons abordé le cas des objets volumiques usuels. En effet, comme nous l'avons signalé dans la section 4.3.3 à propos du tétraèdre, un grand nombre de ces objets peuvent être construits à partir des objets élémentaires et des relations géométriques (intersection, parallèle, ...).

La question qui se pose est de savoir à partir de quel moment une figure complexe peut être (ou doit être) considérée comme une entité et faire partie des objets élémentaires (c'est-à-dire un objet du domaine).

Pour répondre à cette question, il a fallu identifier des "catégories" d'objets pouvant être le résultat de l'association d'objets élémentaires. Par discussion avec les enseignants, nous avons établi que le facteur discriminant était la préservation au maximum des propriétés géométriques de l'objet qu'ils "simulent" : par exemple, la préservation des faces ou des arêtes d'un cube pour

y construire d'autres objets (milieu d'une arête, point appartenant à une face, ...). Dans ce sens, nous avons isolé deux catégories parmi les objets volumiques usuels :

- les corps ronds comme la sphère, le cylindre, le cône, ... difficilement constructibles à partir d'objets élémentaires (l'enveloppe est en général impossible à obtenir),
- les polyèdres comme le cube, la pyramide, le tétraèdre, ... qui peuvent être le résultat satisfaisant d'une construction, même complexe.

Là aussi, une négociation a été nécessaire, car les points de vue informatique et pédagogique sur ce fait étaient en contradiction. D'un point de vue informatique, éviter le "syndrome de l'usine à gaz" et favoriser la réutilisation est un objectif important : si ces objets peuvent être construits par l'utilisateur (enseignant prescripteur ou élève), pourquoi vouloir les intégrer avec un coût de développement important ? Si la construction est fastidieuse, la sauvegarde d'une figure permet sa réutilisation. De même, il est impossible de couvrir la totalité des objets tridimensionnels : à défaut, la mémorisation de constructions permet d'envisager de manière satisfaisante l'évolution du logiciel.

Cependant, d'un point de vue pédagogique, l'énoncé *"Soit ABCD un tédraèdre"* n'est pas forcément équivalent à l'énoncé *"Soient quatre points A, B, C et D. Soient les segments joignant ces points deux-à-deux"* car ils n'induisent pas les mêmes propriétés sous-jacentes (en particulier la notion de face). Du point de vue de la modélisation objet, ce problème se pose en d'autres termes : faut-il implanter une classe *tétraèdre* ou non ?

Après négociation et analyse des contextes d'utilisation impliquant ces objets, un compromis a été établi. Le cube sera le seul polyèdre faisant l'objet d'un implémentation en tant qu'objet élémentaire. Les corps ronds les plus usuels (sphère, cylindre, cône) seront eux aussi implantés. A plus long terme, un mécanisme de réutilisation permettant la composition d'objets élémentaires *par l'utilisateur* au sein d'une entité distincte sera implantée : les *macro-constructions*.

4.4.3 Problème de la présentation du plan

La discussion avec les enseignants a permis de mettre en évidence un certain nombre de conventions de présentation, que nous avons essayées puis adoptées ou rejetées :

- Le plan est généralement présenté sous la forme d'un rectangle, les angles droits de cet objet permettant de mieux en appréhender la position dans l'espace.
- Quand cela est possible, deux des côtés du plan sont parallèles à l'un des bords du dispositif d'affichage (feuille de dessin, écran), permettant l'introduction (artificielle) de la "verticale" ou de l'"horizontale".
- L'introduction de l'épaisseur du plan, peu répandue dans les manuels actuels, permet d'indiquer l'orientation du plan dans l'espace.
- L'utilisation d'un plan, conjointement à un autre objet, permet de suggérer l'appartenance de ce dernier au plan (par exemple une pyramide "posée" sur un plan horizontal).

Il apparaît ainsi que ces choix de présentation ne sont pas anodins mais reflètent bien la volonté de mettre en évidence *visuellement* certaines propriétés *implicites* de l'objet (position, appartenance, ...).

D'une manière générale, cette discussion sur les choix de présentation du plan peut être mise en relation avec les discussions sur les aides à l'observation, en particulier avec la mise en place d'indicateurs visuels permettant d'aider à la compréhension d'une figure : les *éléments visuels de compréhension*. Nous avons accordé une grande importance à ces éléments car ils facilitent

la lecture. De nature variée (projetantes[13] des points, grands cercles des sphères, marques d'orthogonalité ou de parallélisme, ...), ils sont difficiles à définir car souvent reliés à des *contextes d'usage* particuliers. Seules les discussions avec les enseignants et leur expérience des difficultés permettent de les rendre explicites.

De plus, leur implantation dans le logiciel peut s'avérer plus longue et plus difficile que tout le reste. En effet, si les choix de présentation faits pour un objet peuvent être rapidement généralisés (tous les points, quelle que soit la manière de les construire, disposent des mêmes attributs visuels), il n'en est pas de même pour ces *éléments visuels de compréhension* (un point sur un plan n'induit pas les mêmes propriétés géométriques qu'un point sur un cercle et ne requiert pas le même traitement) qui doivent faire l'objet de négociations au cas par cas.

4.5 Organisation des connaissances

Dans les sections précédentes, nous avons montré que les connaissances pédagogiques sont de nature variées et se retrouvent implantées à de nombreux endroits dans un logiciel pédagogique. Nous cherchons maintenant une description permettant l'organisation de ces connaissances, de manière à permettre de prendre en compte l'évolution des besoins.

Une organisation classique des tuteurs intelligents [Nicaud 88], séparant connaissances du domaine, connaissances pédagogiques et connaissances de l'élève ne convient pas à notre contexte. ***Calques 3D*** est un micromonde et non un tuteur intelligent, ce qui explique l'absence de connaissances sur l'élève. De plus, il apparaît évident que connaissances pédagogiques et connaissances du domaine sont étroitement liées, et ceci à juste raison.

Nous nous sommes alors orientés vers une organisation permettant d'une part de prendre en compte les contraintes informatiques, d'autre part d'exprimer les besoins des enseignants auteurs.

Cette analyse nous permet de décrire les diverses connaissances et d'introduire des caractéristiques dans un objectif de *conception* et d'*utilisation*. Ainsi, chacun des partenaires du projet peut avoir accès aux connaissances, selon ses propres besoins et l'objectif qui le concerne (figure 4.7).

Dans un objectif de conception, l'*enseignant auteur* décrit et fait évoluer les connaissances du domaine et leurs différents points de vue (ou modes de présentation). L'*informaticien* implante les connaissances du domaine sous la forme d'une représentation interne unifiée et traduit à l'interface les points de vue sur ces connaissances. Notons cependant que, si dans un registre pédagogique, la réification des connaissances en leurs modes de présentation permet d'assurer la cohérence de l'ensemble, cela n'est plus vrai dans un registre informatique. La représentation interne unifiée doit jouer ce rôle de garant de la cohérence.

L'organisation des connaissances est relativement similaire, lorsqu'on l'analyse d'un point de vue *utilisation* du logiciel. L'*enseignant prescripteur* sélectionne les connaissances du domaine et les points de vue selon lesquels il va présenter ces connaissances à l'élève. Ce dernier, dans les limites imposés par l'enseignant prescripteur, configure les traductions à l'interface de ces points de vue.

Nous voyons ici se profiler une organisation correspondant à un degré d'abstraction des connaissances : des plus abstraites (le domaine) aux moins abstraites (l'interface). Cette organisation nous amène tout naturellement vers le modèle en quatre couches décrit dans [Bernat 95].

13. La projetante d'un point est la droite reliant le point et son projeté orthogonal sur un plan, en l'occurrence sur chacun des plans définis par le repère orthonormé.

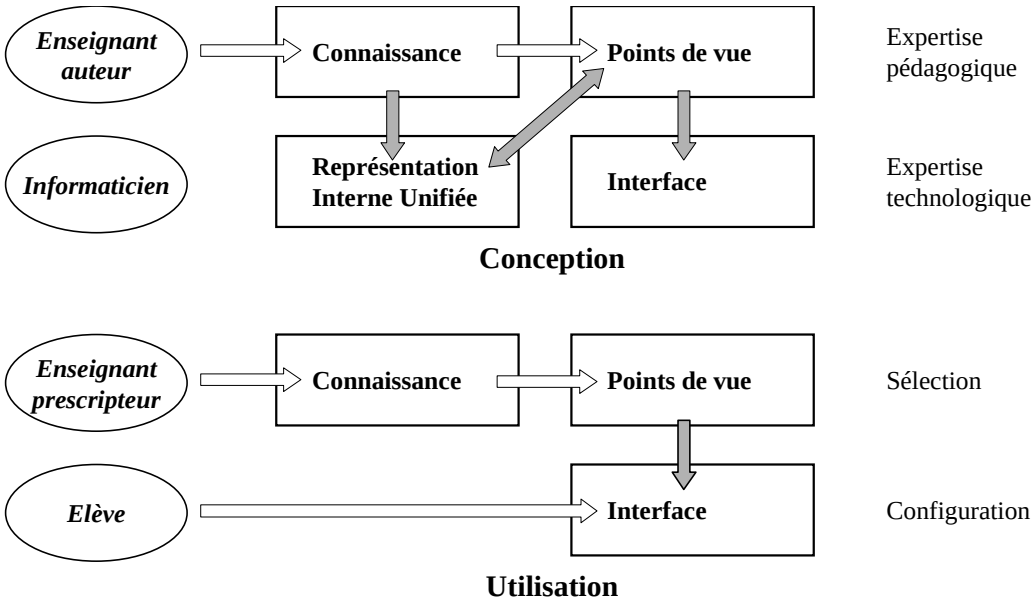


Figure 4.7 – *Évolution des besoins et accès aux connaissances*

Ce modèle se compose des couches *Domaine*, *Représentation Interne*, *Présentation* et *Interface* (figure 4.8), et permet une description des connaissances au niveau approprié.

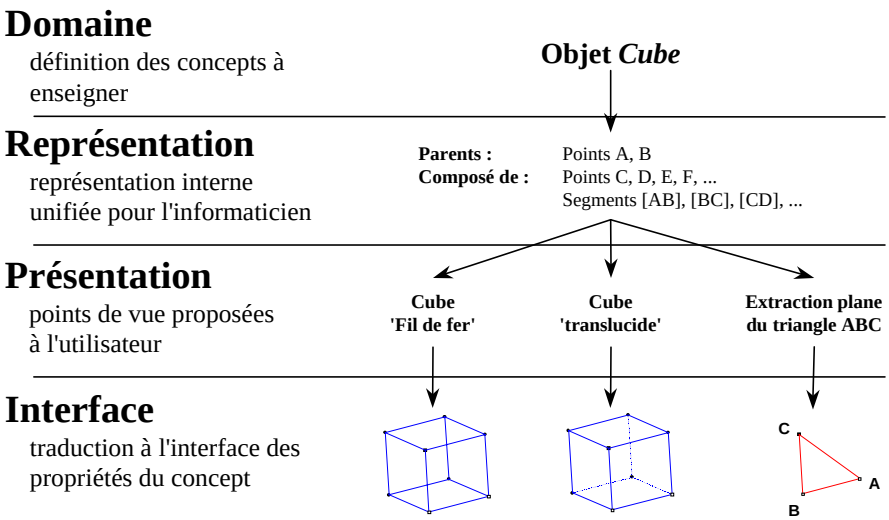


Figure 4.8 – *Organisation des connaissances selon un modèle en quatre couches*

4.6 Conclusion sur l'utilisation des cadres

L'utilisation de ces fiches de description avec les deux groupes de travail qui ont collaboré au développement de ***Calques 3D*** s'est avérée relativement positive, même si nous pouvons regretter de ne pas avoir pu organiser plus de réunions de travail et discuter d'avantage des cadres remplis.

4.6.1 Problèmes

L'un des problèmes majeurs rencontré a été une certaine incompréhension quant à l'objectif des fiches de description. Les activités initialement proposées étaient imaginées autour du seul logiciel, en omettant des activités "papier-crayon" classiques aussi bien que des activités souhaitées. Cela explique la référence à des fichiers du logiciel dans certaines fiches de description (par exemple les fichiers "cube.c3d" ou "tetra.c3d" dans la rubrique "documents fournis" de la figure 4.1).

Une fois identifié, ce problème fut facile à corriger mais cette anecdote illustre bien un problème communément observé dans beaucoup de scénarios d'acquisition de connaissances ([Gaines 87]) : il est nécessaire d'atteindre un consensus sur les objectifs du processus, de manière à ce que tous les participants puissent se comprendre.

En plus de cette incompréhension, nous avons dû faire face à un certain nombre d'interprétations erronées des termes utilisés dans les fiches. Par exemple, nous avions initialement défini les trois catégories de fonctions par *construction*, *observation* et *manipulation*. Les enseignants ont alors assimilé cette dernière notion avec tout mouvement d'objet à l'écran (c'est-à-dire au niveau interface) alors que, pour nous, elle regroupait toute action ayant pour conséquence une réorganisation d'une figure à l'écran (c'est-à-dire une modification de la représentation interne de l'objet). Du point de vue conception, cela signifiait que l'extraction d'une sous-figure dans un calque ou la modification du point de vue de l'observateur, même si la mise en œuvre de l'opération nécessitait une *manipulation directe* par l'utilisateur, n'était pas considérée comme une opération de *manipulation* mais d'*exploration*, alors que la translation d'un point dans l'espace (pour allonger un segment par exemple) l'était. C'est pour résoudre ce problème que nous avons réorganiser les fonctions selon les catégories de *construction*, d'*observation* et d'*exploration*.

Un autre problème qui s'est posé fut l'ambiguïté introduite par notre demande à la fois d'activités nouvelles à réaliser avec ***Calques 3D*** et d'analyse des difficultés et erreurs des élèves sur celles-ci. Les enseignants ont trouvé particulièrement difficile de prévoir ces difficultés sans véritable expérimentation avec les élèves, et donc sans expérience dans l'utilisation de ce type de logiciel. En leur demandant ce travail, notre objectif était de leur faire établir ceci sur la base de leur expérience passée dans des contextes similaires, ce qui s'est avéré relativement irréalisable dans la pratique. Cela nous a conforté dans notre postulat que la conception d'un logiciel pédagogique, sans processus incrémental ni prototype opérationnel, n'a aucun sens.

Étant donné que nous nous intéressions à la description aussi bien d'activités basées sur le logiciel qu'à d'autres l'étant moins directement, nous avons essayé d'élaborer ces fiches de description afin d'acquérir les informations relatives à ces deux types de contexte. Cependant, nous devons reconnaître que nous avions sous-estimé l'impact que l'expérimentation d'un prototype pouvait avoir sur les enseignants, dans un domaine où un réel manque se faisait sentir. Ils ont immédiatement perçu un certain nombre de situations intéressantes pour l'utiliser (ce qui, dans un sens, est un bon point !) mais ont éprouvé plus de difficultés pour décrire comment intégrer ces situations au sein d'activités plus traditionnelles (cours, démonstration, ...). Or, c'est bien cette pluralité des activités que nous avions tenté de favoriser, de telle manière que nous puissions proposer des directions de développement propres à améliorer la conception du prototype, ainsi que suggérer des possibilités effectives d'intégration de ***Calques 3D*** dans leurs cours, et non pas simplement d'utilisation.

4.6.2 Avantages

L'utilisation des *contextes d'utilisation* a permis d'obtenir une meilleure explicitation des raisons pour lesquelles les enseignants choisissent de présenter tels ou tels concepts selon une série d'activités. Ce point nous semble important dans la mesure où c'est une étape obligatoire dans le processus d'intégration d'un logiciel pédagogique, que se soit pour une présentation des concepts usuels ou, au contraire, pour l'enseignement de nouveaux concepts.

Une des raisons principale de l'intérêt des enseignants dans l'utilisation des nouvelles technologies est de conserver ou susciter la motivation des élèves. Celle-ci est souvent réalisée de manière intuitive ou par expérience, mais la clef est bien de réussir à "traduire" cette intuition en utilisation concrète du logiciel. Dans notre situation, ce problème a été grandement simplifié en demandant aux enseignants auteurs de décrire explicitement les étapes de leurs processus d'enseignement. En particulier, cela nous a permis (à nous mais aussi aux enseignants) d'obtenir une image claire sur la manière d'utiliser ***Calques 3D*** pour convaincre l'élève de certaines propriétés géométriques (dynamiques ou non) qui ne peuvent être présentées autrement (par exemple le problème des pôles d'une sphère, "traditionnellement" placés de manière incorrecte sur son contour apparent).

Focaliser l'acquisition de l'expertise pédagogique des enseignants sur la description d'*activités autour du logiciel* plutôt que sur la description d'une séquence, au sens strict du terme, a permis aux enseignants de disposer d'un cadre pour la formulation d'activités applicables selon un large éventail de stratégies d'enseignement, mais surtout nous a permis de leur faire comprendre l'importance d'une description centrée sur le point de vue de l'élève. Sans cela, nous nous sommes rendus compte d'une certaine tendance à céder à l'attrait des possibilités du logiciel et à imaginer des usages et fonctionnalités potentiels qui, sans être dénués d'intérêt, ne disposaient d'aucun fondement pédagogique. De plus, beaucoup de ces idées dépassaient les frontières technologiques des possibilités de ***Calques 3D*** et il aurait été difficile, sinon impossible, de les implanter.

Réalisation de Calques 3D

Troisième partie

Réalisation de *Calques 3D*

Chapitre 5

Description de *Calques 3D*

5.1 Introduction

Dans les deux premières parties de ce document, nous avons présenté les problèmes que peut poser l'enseignement de la géométrie dans l'espace (difficultés de lecture d'une figure géométrique, multiples représentations des objets géométriques, ...) et la méthodologie suivie pour développer un logiciel pédagogique qui satisfasse les différents acteurs du projet du point de vue de l'*acceptabilité*, de l'*utilisabilité* et de l'*adaptabilité* (cf. chapitre 1).

Dans ce chapitre, nous présenterons l'interface d'accès de ***Calques 3D*** et les fonctionnalités du logiciel qui permettent d'atteindre les objectifs fixés, à savoir :

- permettre la **construction** dynamique d'une figure à partir d'objets géométriques élémentaires et de fonctions de construction,
- favoriser l'**observation** de cette figure géométrique en fournissant aux utilisateurs des fonctions de visualisation,
- permettre l'**exploration** de ses propriétés, en particulier par la manipulation directe de ses éléments.

A chaque fois que cela se justifie, nous mettrons en évidence les choix offerts à l'utilisateur et leurs modalités d'accès.

Rappelons que ***Calques 3D*** est un logiciel développé sous Windows. A ce titre, nous avons essayé de respecter les directives de Microsoft concernant la création d'une interface graphique : conserver une homogénéité dans les différents éléments de l'interface facilite la prise en main de l'application par tout nouvel utilisateur connaissant cet environnement.

5.2 L'interface graphique de *Calques 3D*

Avant de présenter les éléments de l'interface de ***Calques 3D***, il est nécessaire de décrire les formats des données tels qu'ils sont accessibles à l'utilisateur.

1. Chaque figure géométrique construite avec ***Calques 3D*** est associée à une structure de données interne (un graphe, cf. chapitre suivant), sauvegardée dans un fichier. L'extension par défaut de ce fichier est "*.c3d".
2. Chaque fichier peut être visualisé sous trois vues différentes : *univers* (figure globale), *calque*(sous-partie de la figure) et *historique* ("énoncé" de la figure). Chaque vue est unique

pour une figure donnée, elle est présentée dans une fenêtre indépendante qui dispose d'éléments d'interface (menus, icônes, ...) permettant d'accéder aux fonctions de manipulation des données, fonctions spécifiques selon le type de vue concerné.

3. Enfin, dans l'univers des applications Windows, l'interface de ***Calques 3D*** est ce qu'on désigne sous le terme de *MDI (Multiple Documents Interface)*, c'est-à-dire qu'il est possible d'avoir plusieurs figures différentes ouvertes en même temps dans l'espace de travail. Par contre, une figure donnée ne peut avoir qu'une seule vue de chaque type (via une fenêtre).

Calques 3D est un micromonde. En tant que tel, son rôle est de fournir à l'utilisateur (enseignant ou élève) des outils lui permettant la libre exploration de l'univers qu'il modélise. Cela signifie en particulier qu'il n'y a aucune contrainte quant à l'initialisation du logiciel. Donc, à l'ouverture de ***Calques 3D***, aucun document, aucune fenêtre n'est ouverte. L'utilisateur peut sélectionner un fichier de travail (par exemple une figure préalablement construite par l'enseignant et fournie à l'élève) ou demander la création d'une nouvelle figure.

5.2.1 L'espace de travail

La figure 5.1 présente un aperçu de l'espace de travail de ***Calques 3D***, tel qu'il apparaît au cours d'une session de travail avancée, et permet de mettre en évidence les différents éléments de l'interface d'accès. De manière identique à ce qui existe dans les applications Windows, on y retrouve : la barre de menus de l'application, les fenêtres correspondant aux différentes vues, les barres d'icônes, permettant l'accès rapide, intuitif et privilégié à des fonctionnalités de ***Calques 3D***, la barre de statut dont le rôle est de fournir une aide contextuelle immédiate à la tâche en cours. Ces éléments vont être brièvement présentés dans cette section.

a) La barre des menus

La barre de menus de l'application présente toutes les commandes du micromonde, organisées par types de fonctionnalités (figure 5.1). Outre les menus, devenus standards, permettant la manipulation des fichiers, l'édition et la manipulation des données de l'application et l'accès aux aides du logiciel, on peut y noter les menus propres à ***Calques 3D***. L'organisation de ceux-ci reflète, autant que possible, les objectifs pédagogiques de ***Calques 3D***, déterminés théoriquement (cf. chapitre 2) ou empiriquement (cf. chapitre 4) :

- le menu `Visualisation` regroupe toutes les commandes qui permettent ou facilite l'observation de la figure géométrique : choix du référentiel, de la perspective, projections particulières, ...
- le menu `Objet` donne accès aux objets élémentaires disponibles (point, droite, sphère, ...), le menu `Construction` aux primitives de construction (intersection, parallèle, ...) : ce sont les commandes qui permettent la construction d'une figure géométrique. Elles sont séparées pour des raisons d'ergonomie.
- le menu `Exploration` regroupe toutes les commandes permettant l'exploration de la construction (déformation, extraction dans un calque, vérification de propriétés, ...).

Chaque commande des menus donne accès à une *tâche* particulière : tâches de construction, d'observation ou d'exploration (figure 5.3).

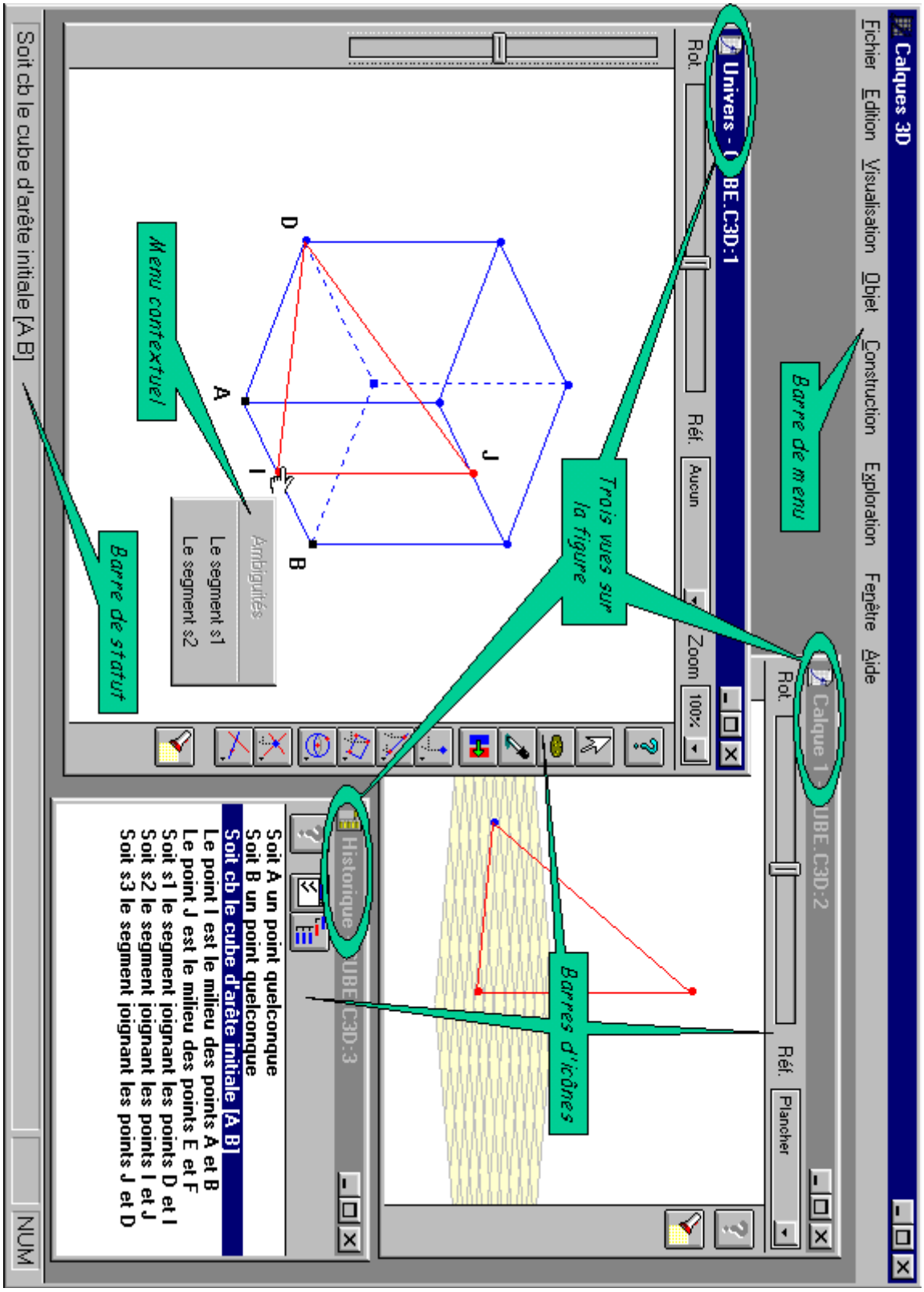


Figure 5.1 – *L'espace de travail de* ***Calques 3D***

b) Les fenêtres de *Calques 3D*

L'utilisateur dispose de trois types de fenêtres différentes pour l'exploitation de chaque vue de la figure géométrique (figure 5.1) :

- la fenêtre *Univers* est une fenêtre graphique. Elle est ouverte à l'initialisation d'un nouveau document de travail ou à l'ouverture d'un fichier existant. Elle permet la construction, la visualisation et la déformation d'une figure géométrique, sans aucune restriction. C'est la fenêtre principale pour l'application ;
- la fenêtre *Calque* est utilisée pour visualiser une partie de la construction qui a été extraite de la figure globale, de manière à observer plus facilement certaines propriétés. Visuellement, elle ressemble à la fenêtre Univers, mais il est impossible d'y réaliser des constructions. Seules la visualisation et la déformation de la figure y sont autorisées (cf. section 5.5.2) ;
- La fenêtre *Historique*[14] présente, sous la forme d'une liste textuelle, tous les objets présents dans la figure courante : les objets visibles mais aussi les objets invisibles, c'est-à-dire ceux gommés par l'utilisateur ou ceux correspondants à des cas particuliers de construction (cf. section 5.6).

c) Les barres d'icônes

Dans ***Calques 3D***, les icônes sont utilisées à la fois comme raccourcis des fonctions proposées dans les menus et comme indicateur de l'état de fonctionnement du logiciel : la commande active est indiquée visuellement par le mode "sélectionnée" de l'icône correspondante (figure 5.2(a)).

Afin de favoriser l'engagement direct et la proximité visuelle, les barres d'icônes ont été rattachées à chaque fenêtre, plutôt qu'au cadre de l'application elle-même.

Enfin, certaines icônes donnent l'accès à un sous-menu (figure 5.2(b)), permettant de modifier la commande qui y est rattachée. On diminue ainsi la surcharge visuelle causée un trop grand nombre d'icônes dans l'espace de travail.

d) La barre de statut

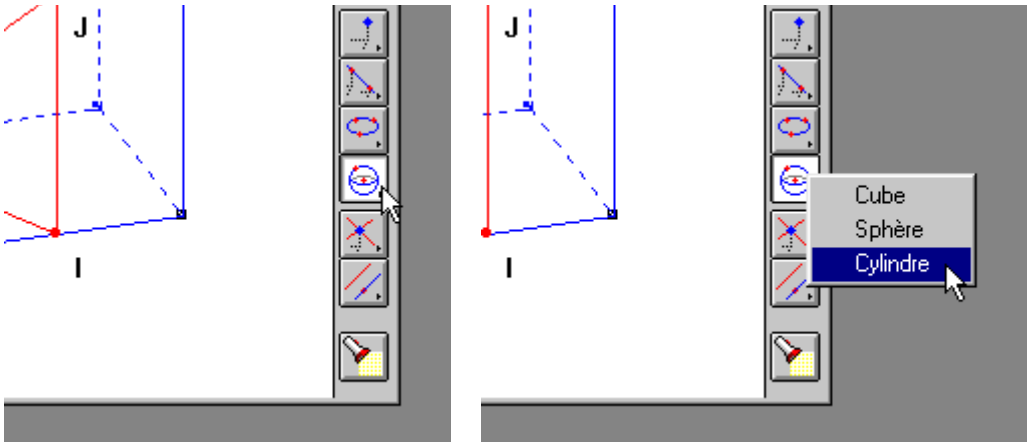
La barre de statut (figure 5.1) a pour fonction de fournir à l'utilisateur des messages d'aide et des indications de rétroaction. Ces messages sont importants car certaines fonctionnalités sont complexes quant à leur mise en œuvre : elles se réalisent en plusieurs étapes et il est donc nécessaire de fournir une information sur l'état de la tâche, c'est-à-dire sa nature, ce que l'utilisateur a déjà fait et ce qu'il lui reste à faire pour l'achever.

Nous reviendrons plus en détail sur ce point dans la section consacrée aux aides et explications (section 5.6).

e) Le menu contextuel

Dans ***Calques 3D***, certaines tâches sont complexes et nécessitent de développer d'autres possibilités, comme pour l'extraction d'objets vers un calque (cf. section 5.5.2) ou pour la levée des ambiguïtés lors de la désignation d'un objet (figure 5.1). Les différentes options sont alors présentées à l'utilisateur dans un menu contextuel, accessible par le bouton droit de la souris.

14. Le terme "historique" est relativement impropre car cette liste ne tient pas compte de toutes les opérations réalisées sur la figure (en particulier les objets détruits par l'utilisateur). Il s'agit bien d'une description des objets de la figure à un instant donné, c'est-à-dire de l'énoncé de la construction. Les enseignants ont cependant préféré le nom "historique" à "énoncé" car ce dernier fait plus référence au texte d'un exercice donné à l'élève.



(a) Icône "sélectionnée" : la commande *"Construire une sphère"* est active.

(b) Sous-menu de l'icône *Volumes*.

Figure 5.2 – *Exemple de barres d'icônes*

Ce menu n'est pas toujours disponible car il dépend de la tâche en cours et ne dépend pas directement de la fenêtre comme dans la plupart des applications Windows..

5.2.2 Organisation des tâches

Les fonctionnalités spécifiques à ***Calques 3D*** constituent l'ensemble des outils mis à la disposition de l'utilisateur en matière de construction géométrique : nous parlons de *tâches* assignées à l'ordinateur. Les tâches obéissent à certains principes énoncés ci-dessous :

1. Pour la réalisation d'une tâche, ***Calques 3D*** applique une stratégie dite *action/objets* : l'utilisateur doit en premier lieu choisir une action à réaliser puis seulement après les objets sur lesquels cette action s'appliquera. Ceci permet d'affiner l'aide fournie à l'utilisateur pour la réalisation de la tâche (spécifier le texte de la barre de statut, restreindre le champ des objets désignables, ...).
2. Les tâches ne sont valides que sous certaines conditions liées à la présence des objets géométriques nécessaires à leur application. Par exemple, construire une droite nécessite la présence d'au moins deux points dans la fenêtre de travail : la tâche *"construction d'une droite"* ne pourra être appliquée que si ces deux points existent. Dans le cas contraire, la commande correspondante est inactive (menu et icône grisés).
3. Les tâches sont répétitives, c'est-à-dire qu'elles bouclent indéfiniment sur elles-mêmes jusqu'à ce que l'utilisateur décide qu'il a fini de les utiliser. Cela permet d'appliquer une même opération plusieurs fois de suite sans avoir à réactiver la commande correspondante (par exemple la construction de points libres à la volée).
4. Il n'est possible de réaliser qu'une seule tâche à la fois : si l'utilisateur souhaite en accomplir une autre (par exemple passer d'une tâche de construction à une tâche d'exploration de la figure), il doit auparavant achever ou abandonner la tâche courante avant de pouvoir passer à la suivante.
5. Le statut répétitif des tâches et l'unicité de la tâche active permet d'assurer en permanence l'existence d'une tâche accessible à l'utilisateur (que nous désignons sous le terme de *tâche*

active). Cette "permanence des tâches" nécessite l'introduction d'une *tâche par défaut* qui correspond à l'absence de toute tâche activée explicitement par l'utilisateur et sert de point d'entrée du logiciel. Dans ***Calques 3D***, elle ne donne accès qu'à la modification des attributs visuels des objets de la figure (cf. section 5.4.3).

6. Il est possible de quitter la tâche active à tout moment de son exécution, entraînant de fait l'interruption de toute activité en cours. Si un objet est en cours de construction, celui-ci n'est tout simplement pas construit. Quitter une tâche peut se faire soit explicitement en utilisant la touche ESC pour revenir à la *tâche par défaut*, soit en activant une autre tâche.
7. Enfin, il est possible d'annuler le résultat de la dernière tâche effectuée, si celle-ci a entraîné une modification de l'état de la figure (construction ou suppression d'un objet, modification des attributs visuels, ...). Un seul pas d'annulation est autorisé dans ***Calques 3D***, et cette annulation est définitive.

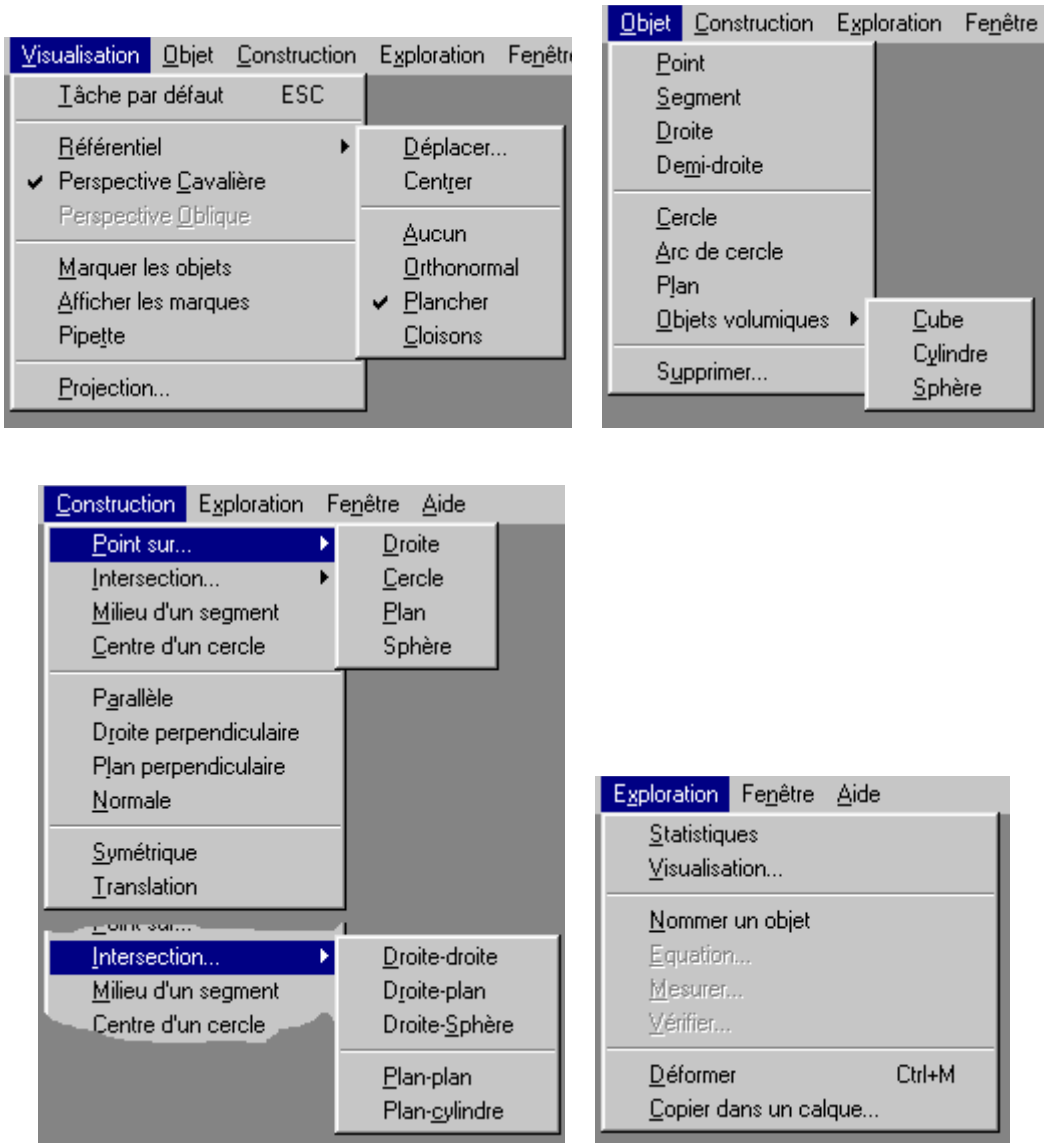


Figure 5.3 – *Les menus de* ***Calques 3D***

5.3 Les fonctions de construction

Les objets géométriques peuvent être regroupés en plusieurs catégories : les points, les courbes (droites, cercles, ...), les surfaces (plans, ...) et les volumes (sphères, cylindres, ...). Respectant le cahier des charges initial, les volumes pleins ne sont pas abordés en tant que tel dans ***Calques 3D*** : la sphère et le cylindre sont considérés comme des surfaces (assimilés à leur enveloppe), le cube comme une *composition d'objets* (points et segments, cf. section 4.4.2). Cela implique que les notions d'*intérieur* et d'*extérieur* ne sont pas traitées dans le logiciel, seule la propriété topologique *sur* est abordée.

Pour des raisons de commodité langagière, nous avons cependant conservé le terme de *volume* pour désigner ces trois objets.

5.3.1 Objets élémentaires et primitives de construction

La *géométrie du point*, mise en œuvre dans ***Calques 3D***, implique que le point est le seul objet à construire (et à déplacer) en toute liberté dans l'espace. Tous les autres objets géométriques peuvent être définis :

- soit à partir de deux ou plusieurs points (la sphère est définie par deux points : son centre et un point sur l'enveloppe ; le plan passe par trois points ; ...). C'est ce que nous appelons *objets élémentaires*. Ils sont regroupés dans le menu `Objet`.
- soit comme ayant une relation géométrique avec un objet (point sur une droite, ...) ou plusieurs objets (droite intersection de deux plans, ...). C'est ce que nous appelons *primitives de construction*, de manière à mettre en évidence la relation géométrique plutôt que l'objet en tant que tel. Ces primitives sont regroupées dans le menu `Construction`.

Les deux tableaux 5.1 et 5.2 présentent la liste complète des objets élémentaires et des primitives de construction disponibles dans ***Calques 3D***.

Objet	Description
`point`	point libre dans l'espace
`droite`	droite passant par 2 points
`segment`	segment joignant 2 points
`demi-droite`	demi-droite passant par 2 points
`cercle`	cercle passant par 3 points
`arc de cercle`	arc-de-cercle passant par 3 points
`plan`	plan passant par 3 points
`cube`	cube défini par 2 points (arête initiale) et par l'orientation d'une face
`sphère`	sphère définie par 2 points (centre et rayon)
`cylindre`	cylindre droit défini par 3 points (extrémités et rayon)

Table 5.1 – Liste des objets élémentaires

Pour les primitives de construction, nous en indiquons les arguments (c'est-à-dire les objets cibles, nécessaires à sa mise en œuvre), ainsi que l'ordre de désignation de ceux-ci (sous la forme `objet1, objet2, ...`). Notons que les arguments représentent une catégorie d'objets géométriques : `droite` désigne les droites, segments et demi-droites, `cercle` désigne les cercles et arcs de cercle, ...

Primitive	**Description**	**Arguments**
`intersection...`		
	point intersection de deux droites	`droite, droite`
	point intersection d'une droite et d'un plan	`droite, plan`
	point intersection d'une droite et d'une sphère	`droite, sphère`
	droite intersection de deux plans	`plan, plan`
	ellipse intersection d'un plan et d'un cylindre	`plan, cylindre`
`point sur...`		
	point sur droite	`droite`
	point sur cercle	`cercle`
	point sur plan	`plan`
	point sur sphère	`sphère`
`milieu`	point milieu d'un bipoint	`point, point`
`centre`	centre d'un cercle	`cercle`
`parallèle`	droite parallèle à une droite et passant par un point	`droite, point`
`perpendiculaire`	droite perpendiculaire à une droite et passant par un point	`droite, point`
	plan perpendiculaire à une droite et passant par un point	`plan, point`
`normale`	droite normale à un plan et passant par un point	`plan, point`
`symétrique`	symétrique d'un point par rapport à ...	
	... un point (symétrie centrale)	`point, point`
	... une droite (symétrie axiale)	`point, droite`
	... un plan (symétrie plane)	`point, plan`
`translation`	translation d'un point selon un vecteur	`point, point, point`

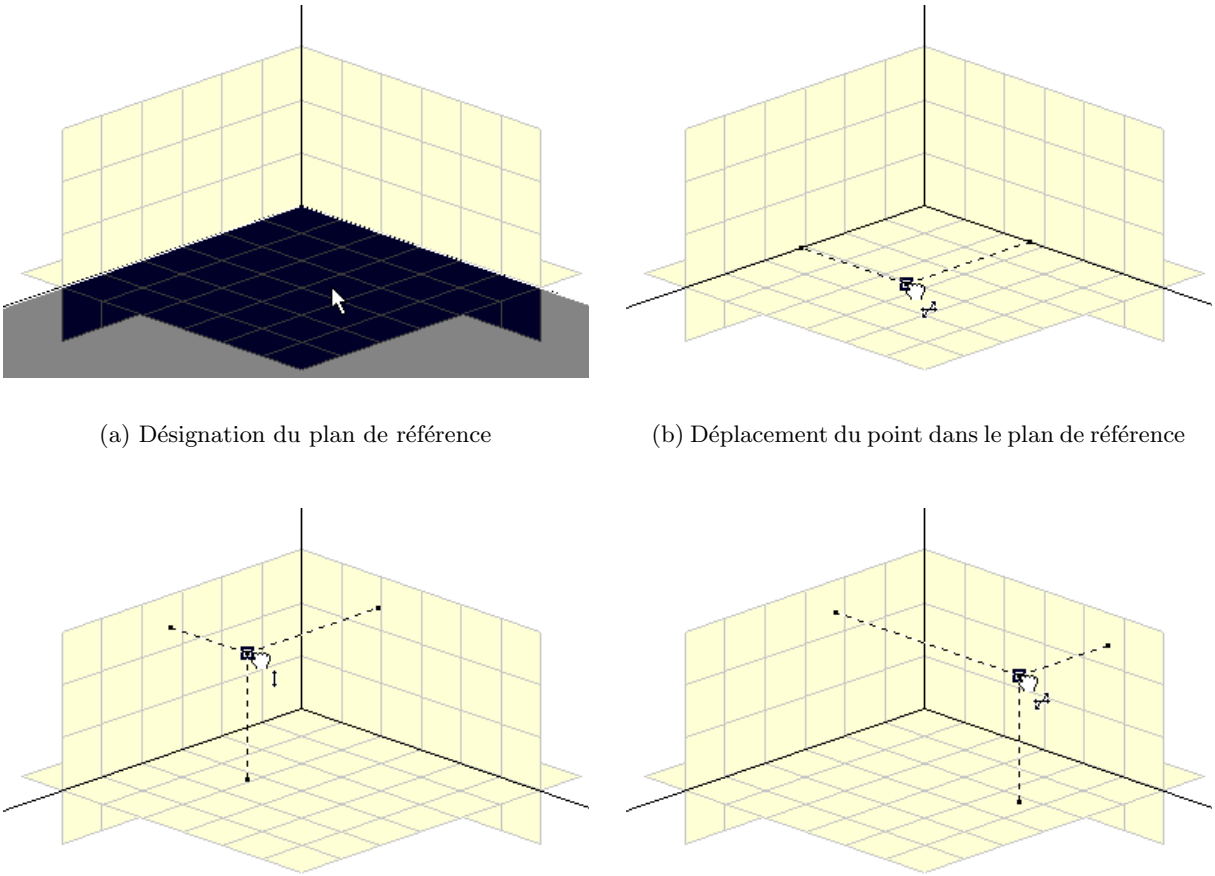
Table 5.2 – Liste des primitives de construction

Certaines primitives comme `intersection...` ou `point sur..` sont des primitives *génériques*, c'est-à-dire que le même concept géométrique mis en œuvre (intersection ou appartenance) se décline différemment, soit selon l'objet résultat (point d'intersection de deux droites, droite d'intersection de deux plans), soit par la nature des arguments (la symétrie peut être centrale ou axiale). Dans ***Calques 3D***, nous avons choisi d'associer une commande distincte à chaque instance. Cela contraint l'utilisateur à savoir ce qu'il veut construire mais cela favorise la gestion de la désignation et des ambiguïtés, et donc de l'ergonomie du logiciel.

5.3.2 Création d'un point libre

La difficulté de la création des points dans l'espace réside dans l'absence de dispositif d'interface adéquat : nous ne disposons en effet que d'une souris, pointeur en deux dimensions, pour désigner, sur un écran en deux dimensions, une position dans un espace en trois dimensions !

Pour contourner cette difficulté, nous avons choisi de décomposer la création d'un point dans l'espace en une succession d'étapes élémentaires se ramenant chacune à une création dans le plan. Pour cela, nous avons choisi d'imposer la présence des cloisons comme matérialisation du référentiel de construction (cf. section 5.4.2).



(a) Désignation du plan de référence

(b) Déplacement du point dans le plan de référence

(c) Modification de l'éloignement du point par rapport au plan

(d) Déplacement du point dans un plan parallèle au plan de référence

Figure 5.4 – *Construction d'un point libre dans l'espace*

La création d'un point libre, illustrée dans la figure 5.4, s'effectue en plusieurs étapes. La première consiste à désigner à l'aide de la souris l'une des trois cloisons comme support de création. La cloison désignée sert alors de plan de référence pour la création, un point apparaît alors, momentanément assujetti au déplacement du curseur. Il est possible, soit de déplacer le point selon un plan parallèle à la cloison (curseur *déplacement plan*), soit de modifier sa distance à la cloison en maintenant la touche SHIFT pressée (curseur *déplacement linéaire*). En alternant ces deux modes de déplacement, il est possible de positionner un point n'importe où dans l'espace visible. Il suffit alors de cliquer une dernière fois pour valider la position finale du point.

Au cours du déplacement du point, ses *projetantes* (c'est-à-dire les projections du point sur les trois cloisons) sont matérialisées, de manière à permettre à l'utilisateur une perception *rapide et immédiate* de sa position dans l'espace. C'est aussi une des raisons pour laquelle nous avons choisi de contraindre la création d'un point à l'utilisation du référentiel "cloisons".

a) Les limites de ce mode de création

Cette méthode de création a l'avantage pédagogique de montrer la décomposition d'un déplacement spatial en déplacements plans et linéaires. Elle présente cependant quelques contraintes dues à la nature des cloisons. En effet, que doit-il se passer lorsque l'utilisateur crée un point derrière les cloisons ?

Cette question a souvent été abordée au cours du développement de ***Calques 3D*** mais il n'y a pas eu de consensus sur ce qu'il fallait faire dans cette situation. Nous avons donc choisi arbitrairement de limiter la construction des points à la partie "intérieure" des cloisons : il est donc impossible de modifier la position de l'observateur ou de changer le référentiel.

5.3.3 Construction de la figure

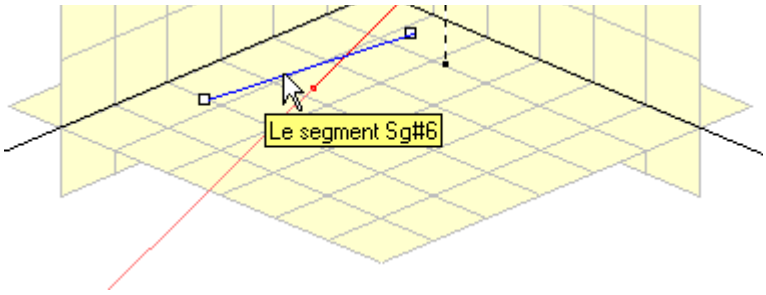
Après la création de points libres, il est aisé de construire les autres objets géométriques de la figure.

a) Désignation des objets cibles

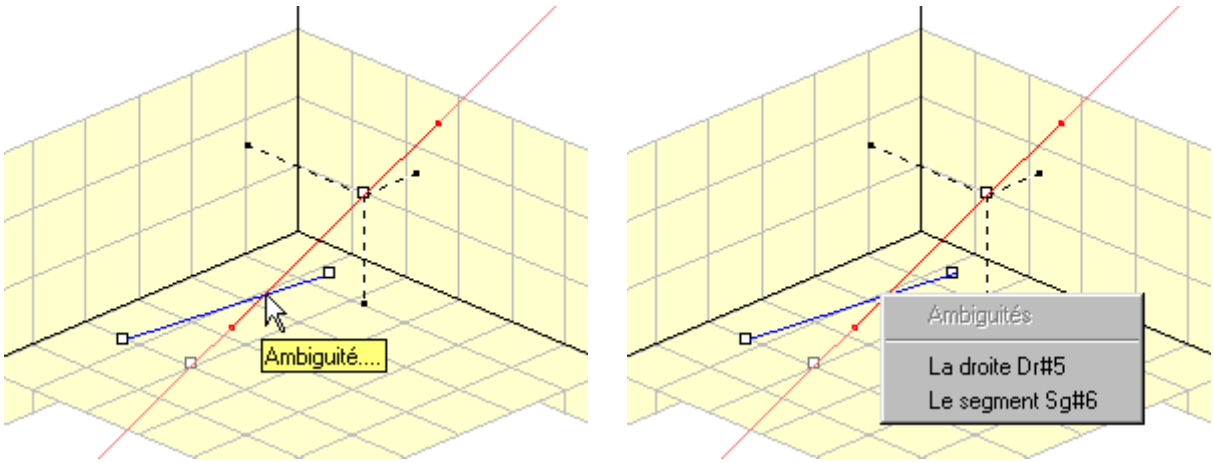
Calques 3D applique le paradigme *action/objets* : la création d'un objet géométrique (autre que le point libre) nécessite donc la désignation successive des objets servant à sa définition, les *objets cibles*. Plusieurs aspects sont à prendre en compte pour cette désignation.

1. Les objets cibles sont d'un type bien précis (par exemple les deux points pour construire une droite, un point et une droite pour construire une droite parallèle) : la tâche de construction ne permet de désigner que les objets du type requis, les autres ne sont pas "détectés" par le curseur, réduisant ainsi l'*espace d'exploration*. Ces objets sont appelés objets *désignables*.
2. Toutes les tâches de construction impliquent un *ordre de désignation* des objets cibles. Cet ordre est imposé quand les objets cibles interviennent dans la construction selon une complexité croissante et une certaine dépendance. Par exemple, pour construire une demi-droite, on désigne deux objets cibles de type `point` : le premier désigne toujours l'origine, le deuxième, associé au premier, concourt pour désigner la direction de la demi-droite. Cet ordre reste cependant discutable quand les objets cibles sont indépendants (construire une parallèle à une droite nécessite de désigner d'abord la droite - la direction - puis un point - l'origine -) : c'est un choix d'implémentation du logiciel qui peut être justifié par des critères ergonomiques.
3. Lorsqu'un objet peut être désigné lors d'une tâche de construction, une étiquette d'information apparaît au voisinage du curseur. Celle-ci contient le type et le nom de l'objet sous le curseur (figure 5.5(a)).
4. Enfin, il peut arriver que, lorsque plusieurs objets désignables se trouvent à proximité du curseur, une ambiguïté de désignation survienne : le logiciel est incapable de déterminer avec précision lequel des objets est souhaité par l'utilisateur. ***Calques 3D*** signale alors l'ambiguïté et sa levée se fait au moyen d'un menu contextuel qui donne la liste des objets présents sous le curseur. Le choix de l'un d'entre eux valide la désignation (figure 5.5(b)).

La liste des objets cibles des primitives de construction et leur ordre de désignation sont donnés dans le tableau 5.1



(a) Désignation et étiquette d'information



(b) Désignation et levée de l'ambiguïté

Figure 5.5 – *Désignation des objets cibles d'une construction.*

b) Rétroactions visuelles

Durant l'exécution d'une tâche de construction, un certain nombre d'indicateurs et de rétroactions visuels sont introduits de manière à mieux rendre compte de ce qui est en train de se construire et de matérialiser l'espace de construction. Ces indicateurs sont des éléments qui favorisent la compréhension de la figure : nous les appelons *éléments visuels de compréhension* . Ils ont un statut temporaire, c'est-à-dire qu'ils apparaissent lorsque leur besoin de fait sentir et disparaissent soit lorsque l'objet est construit, soit lorsque l'on quitte ou annule la tâche.

La nature exacte de ces éléments dépend de la tâche active et de l'objet en cours de construction mais certains d'entre eux sont communs à l'ensemble des tâches de construction. C'est le cas par exemple de la rétroaction de sélection : les objets désignés par l'utilisateur sont indiqués par une marque spécifique (généralement un carré en inverse vidéo).

Les projetantes des points, qui apparaissent lors de leur création, sont un exemple d'indicateur qui dépend de la tâche. Un autre exemple plus complexe, mettant en évidence l'espace de construction accessible à l'utilisateur, concerne la création d'un cube (figure 5.6).

Elle se fait en désignant successivement deux points distincts qui définissent l'arête initiale du cube. Une prévisualisation de cette future arête est affichée. Après désignation du deuxième point, un cercle (centré sur le deuxième point, perpendiculaire à l'arête et de rayon sa longueur)

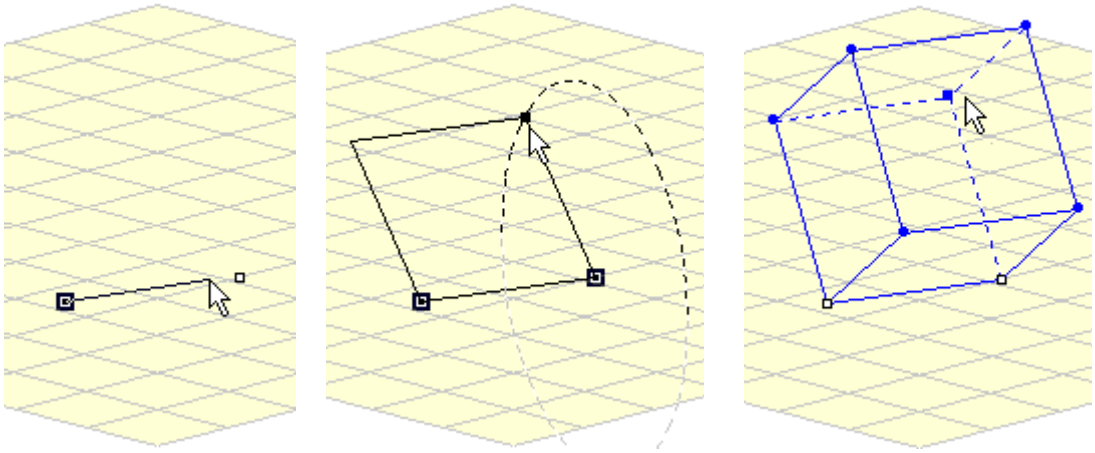


Figure 5.6 – *Rétroactions visuelles et construction d'un cube.*

apparaît alors : l'espace de construction est réduit à ce cercle. Déplacer un point le long de ce cercle permet de matérialiser et d'orienter la première face du cube. Le cube est alors obtenu par le développement de cette face, selon une direction fixe (obtenue par produit vectoriel des trois points).

c) Validité des objets cibles

Le dernier point concerne les propriétés ou relations géométriques requises entre les objets cibles pour permettre la construction de la figure. Cela correspond à la notion de cas particuliers en géométrie. Par exemple, pour que l'intersection de deux segments soit valide (figure 5.7), il faut bien évidemment que les segments aient été construits par l'utilisateur, mais aussi qu'ils soient coplanaires et non parallèles et que leur intersection soit dans les limites de chacun des deux segments. Or, à la création, il se peut qu'une ou plusieurs de ces conditions ne soient pas respectées. Dans ce cas, ***Calques 3D*** n'autorise pas la construction de l'objet. Un message indiquant les raisons de l'échec apparaît et la tâche est réinitialisée : il faut à nouveau désigner deux segments ayant les "bonnes" propriétés.

5.4 Les fonctions de visualisation

La visualisation d'une construction géométrique dans l'espace n'englobe pas uniquement les choix de présentation des objets géométriques. D'autres aspects sont pris en compte :

- choix de la position de l'observateur pour définir la projection de la figure sur l'écran,
- choix du référentiel pour matérialiser l'univers géométrique,
- choix les attributs visuels pour présenter à l'interface les objets géométriques et leurs propriétés et relations géométriques.

5.4.1 La projection des figures

Pour présenter le principe de la projection d'une scène spatiale sur un écran plan, nous allons prendre une analogie : celle de l'observateur virtuel (figure 5.8).

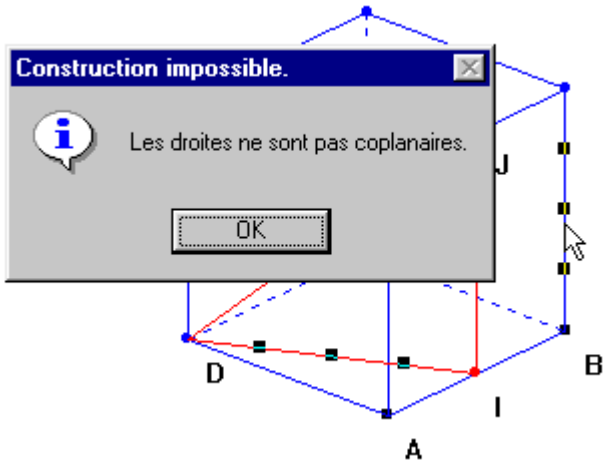


Figure 5.7 – *L'intersection des deux segments n'est pas valide*

Imaginons une sphère, centrée sur l'origine du repère de l'espace (figurée par le carré noir sur la figure) et suffisamment grande pour englober toute figure géométrique construite dans cet espace. Matérialisons sur cette sphère l'équateur, c'est-à-dire le grand cercle inclus dans le plan horizontal $(O\vec{x},O\vec{y})$ et un méridien, c'est-à-dire le grand cercle inclus dans le plan $(O\vec{x},O\vec{z})$. Plaçons maintenant un observateur virtuel quelque part sur l'enveloppe de cette sphère (figuré par le carré blanc) dont nous matérialisons aussi le demi grand cercle passant par ce point. Cet observateur a la particularité de toujours fixer de l'œil le centre du repère : c'est l'axe de projection. Sa position sur la sphère (et donc dans l'espace) peut être repérée de manière unique :

- par l'angle qu'il fait avec le méridien (c'est-à-dire la longitude θ),
- par l'angle qu'il fait avec l'équateur (c'est-à-dire la latitude φ).

C'est ce que voit l'observateur qui va être affiché à l'écran. Ainsi, la visualisation de la figure géométrique sous différents angles, se fait non pas en déplaçant l'ensemble de la scène mais en modifiant la position de l'observateur (modification du point de vue).

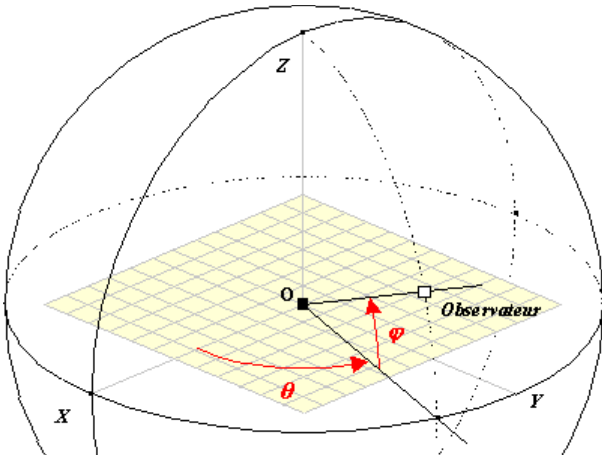


Figure 5.8 – *Observateur virtuel et point de vue*

Les modifications de la projection ne peuvent se faire qu'en modifiant l'un des deux angles. Cela contraint fortement l'exploration libre par l'utilisateur : il n'a pas la possibilité d'observer

la construction sous tous les angles possibles. En particulier les rotations autour de l'axe de projection sont interdites (pas de "tangage"). En revanche, cette restriction permet une analogie plus facile entre le déplacement naturel d'un utilisateur dans un espace et le "comportement" visuel du résultat obtenu : il modifie sa position naturellement en "tournant" autour d'un objet sans position ésotérique (la tête en bas ou tout autre axe de rotation de l'utilisateur).

a) Modification manuelle du point de vue

Par défaut, à l'ouverture d'une fenêtre *Univers*, le point de vue de l'observateur est positionné de manière à donner une visualisation légèrement plongeante sur la construction (correspondant aux valeurs d'angles $\theta = 45$ et $\varphi = 25$, soit relativement similaire aux valeurs prises traditionnellement pour la projection en perspective cavalière). Rappelons que l'intérêt du mode analytique n'a pas été retenu comme critère pédagogique pour le développement de ***Calques 3D*** : les *valeurs numériques* ne sont, en tant que telles, ni accessibles ni manipulables par l'utilisateur.

La position de l'observateur peut-être à tout instant modifiée grâce aux deux curseurs se trouvant dans les barres d'icônes des fenêtres *Univers* et *Calque* (figure 5.1). Le curseur horizontal simule une rotation du point de vue en modifiant la longitude (modification de l'angle θ) alors que le curseur vertical simule une modification de la latitude (modification de l'angle φ).

La modification du point de vue n'est pas une tâche en soit : elle peut être effectuée n'importe quand, sous n'importe quelle autre tâche. Il est alors possible que la souris soit déjà engagée dans une action dont l'interruption peut s'avérer préjudiciable. Pour laisser toute liberté à l'utilisateur, le point de vue peut aussi être modifié directement au clavier (pavé numérique).

Il est à noter que le résultat visuel de la rotation est inversé par rapport à l'action sur l'observateur : en déplaçant le curseur horizontal vers la droite par exemple, l'observateur est déplacé vers la gauche, donnant ainsi l'impression que c'est l'ensemble de la construction qui est déplacée, et non l'observateur virtuel. Là aussi, cela permet de conserver un certain degré de similitude entre l'action effectuée et le résultat obtenu.

b) Modification automatique du point de vue

Le pas de rotation de la modification manuelle est fixe. Cette contrainte implique que toutes les positions ne peuvent pas être effectivement atteintes. En particulier, il sera difficile par simple rotation manuelle de placer l'observateur selon des points de vue particuliers : observation d'un plan selon sa tranche (*vue de profil*), position dans l'axe d'une droite (*vue frontale*), ... Pour pallier cette contrainte, une commande permet de choisir ces positions particulières.

Par exemple, dans la figure 5.9, nous avons construit la normale au plan. Modifier manuellement le point de vue ne permet pas de vérifier correctement si la droite est bien normale au plan. L'utilisation de la commande `Projection...` permet de désigner le plan puis de choisir dans le menu contextuel `"Position en vue..."` pour obtenir l'observation du plan en vue frontale et s'apercevoir que la droite se résume à un point (droite de bout).

5.4.2 Représentation de l'univers

Nous avons choisi quatre types de matérialisation du référentiel, chacun pouvant être associé à une ou plusieurs situations didactiques privilégiées (figure 5.10) :

(a) la première représentation visuelle du référentiel est le traditionnel *repère orthonormé*, représentant le trièdre de référence : l'origine du repère, les trois axes et les vecteurs unitaires de chacun de ces axes. Ce référentiel est plutôt associé à une utilisation analytique

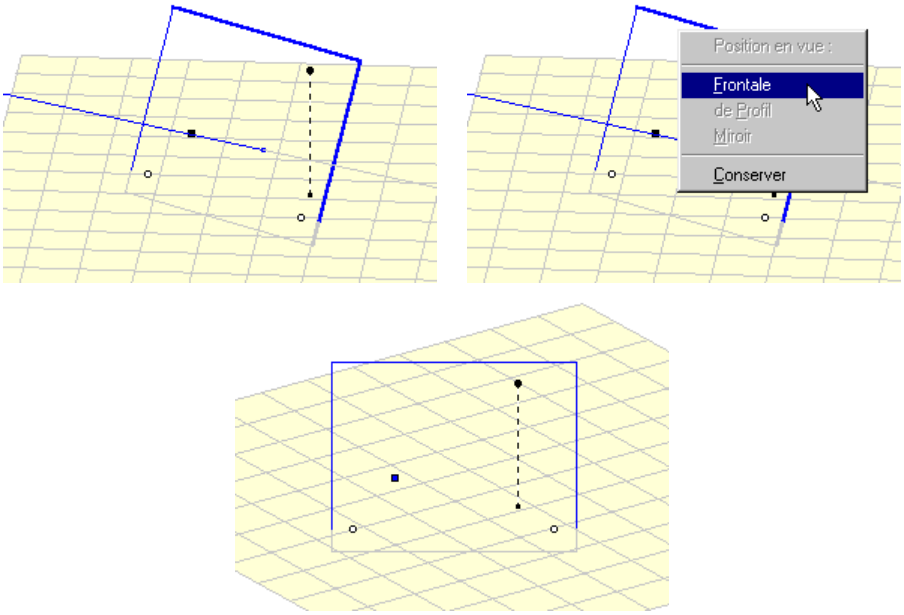


Figure 5.9 – *Observation d'une figure en projection quelconque et frontale*

du logiciel (calcul de distances, d'angles, affichage d'équations, ...) : ce n'est pas le mode privilégié par notre équipe d'enseignants.

(b) Le *plancher* est une matérialisation quasi-physique de l'espace sous la forme d'un plan borné horizontal. Il s'agit du référentiel privilégié de ***Calques 3D***, dans la mesure où il s'apparente le plus à ce qui existe dans le monde réel et donc favorise la lecture de la figure par l'élève. Il peut être utilisé principalement pour visualiser la position relative des objets géométriques par rapport au plan horizontal (*hauteur*) et leurs intersections avec celui-ci.

(c) Les *cloisons* sont une autre matérialisation du référentiel : en plus du plancher horizontal précédent sont représentés les plans frontal et gauche. Ce référentiel est utilisé lors de la création des points (cf. section 5.3.2) et peut aider à la visualisation des droites, des plans, ... par la matérialisation de leurs intersections avec chacune des cloisons.

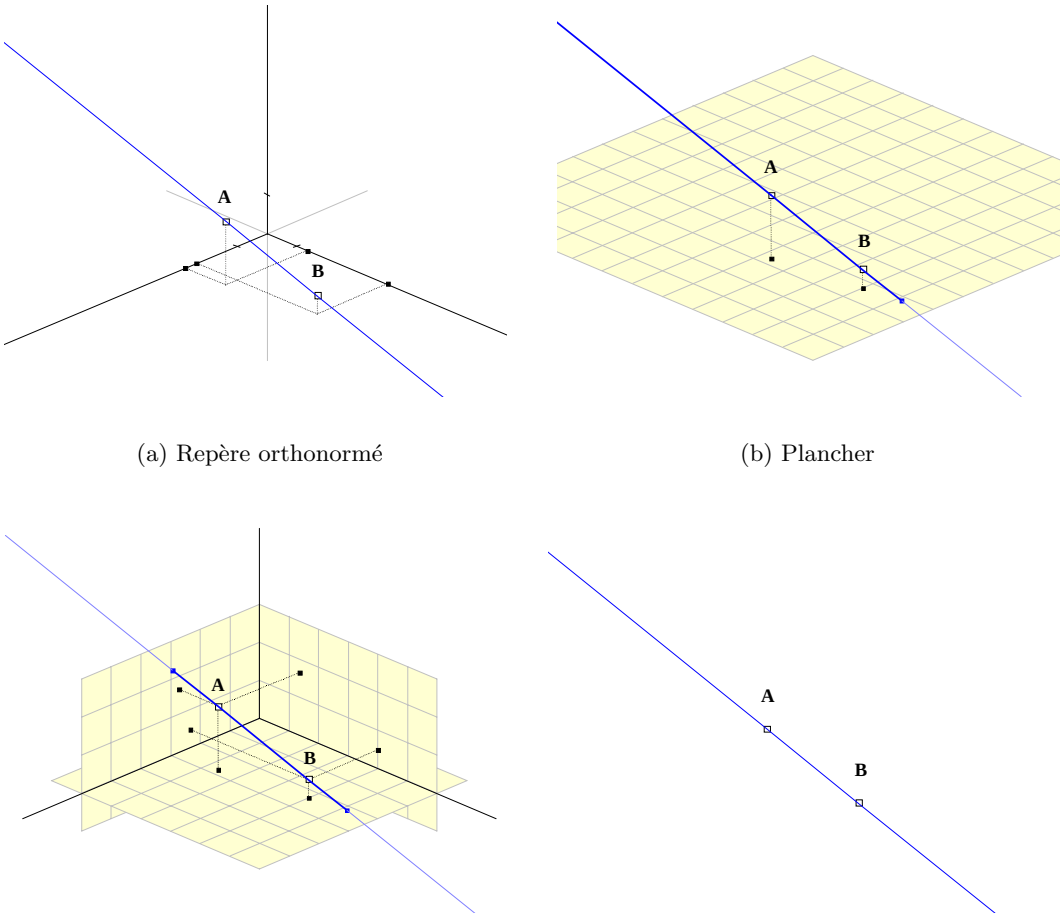
(d) Enfin, certaines situations peuvent ne nécessiter aucune matérialisation du référentiel. C'est le cas par exemple lorsque la présence d'un objet volumique (comme la sphère ou le cube), servant de figure de base à l'exercice, suffit à donner une bonne compréhension de l'espace mais où la présence d'un référentiel pourrait gêner la lecture de la construction.

L'utilisateur peut à tout moment changer le référentiel grâce à la liste déroulante `Réf.` [15] de la barre d'icône des fenêtres *Univers* et *Calque* (figure 5.1). Seule exception, la construction des points libres s'appuie obligatoirement sur les cloisons (cf. section 5.3.2).

5.4.3 Présentation des objets géométriques

Le choix des modalités de présentation graphique des différents objets géométriques s'est fait selon trois principes fédérateurs.

15. Abréviation de "Référentiel", utilisée pour des raisons de place sur la barre d'icône.



(a) Repère orthonormé

(b) Plancher

(c) Cloisons

(d) Aucun

Figure 5.10 – *La droite (AB) représentée sous différents référentiels*

Premièrement, nous avons essayé de conserver une homogénéité visuelle pour tous les objets d'un même type : un plan défini par trois points doit être présenté de la même manière qu'un plan parallèle à une droite. Dans ***Calques 3D***, cette homogénéité est assurée par l'introduction d'*attributs visuels* propres à chaque type d'objets qui se "traduisent" graphiquement à l'interface par une représentation générale unique. De plus, nous avons essayé de définir des valeurs pour ces attributs qui respectent les habitudes et les usages de l'enseignement traditionnel, même si elles ne sont pas exhaustives. Ces valeurs sont librement accessibles à l'utilisateur, comme autant d'éléments de paramétrisation du logiciel.

Deuxièmement, nous avons signalé dans la section 4.4.3 qu'il est important d'indiquer visuellement certaines relations géométriques pouvant lier les objets entre eux et non directement lisibles. Lorsque cela était possible et nécessaire, nous avons introduit des *marques* qui sont des *éléments visuels de compréhension* permettant le codage d'une propriété particulière. Ces marques, en compléments des attributs de l'objet, permettent d'en affiner sa représentation graphique.

Enfin, ***Calques 3D*** est un logiciel de géométrie dynamique permettant la manipulation directe des objets de l'univers. L'aspect visuel des objets géométriques doit faciliter cette ma-

nipulation, notamment pour la désignation des objets cibles lors de la construction ou de la déformation d'une figure. Cela nous a amené à porter une attention particulière aux objets infinis (comme la droite ou le cylindre, mais surtout le plan).

a) Modification des attributs visuels

Chaque objet a une représentation graphique générale contrôlée principalement par deux *attributs visuels* :

- sa *forme* qui définit l'aspect de son tracé dans la fenêtre graphique. Par exemple, un point peut être présenté sous la forme d'un carré ou d'un rond, ... ; une droite sous la forme d'une ligne continue ou pointillée, ...
- sa *couleur*, choisie dans un ensemble proposé par défaut (rouge, vert, ...).

D'autres attributs, liés à l'état de l'objet (*attributs d'état*), peuvent avoir une influence sur sa présentation, en particulier :

- l'attribut *sélectionné* qui indique que l'objet a été désigné par l'utilisateur dans une tâche (par exemple dans une tâche de construction),
- l'attribut *caché* qui indique que l'objet existe mais n'est pas visible à l'interface (par exemple pour les constructions intermédiaires),
- l'attribut *marqué* qui indique que l'utilisateur requiert l'affichage des *marques* de l'objet (par exemple pour afficher les projetantes d'un point).

La modification de ces attributs se fait à l'initiative de l'utilisateur, dans la *tâche par défaut*, en cliquant sur un objet avec le bouton droit de la souris. C'est l'un des rares modes d'accès à l'interface qui soit modal dans ***Calques 3D*** : l'utilisation d'une boîte de dialogue, même si elle bloque la "fluidité" de la manipulation directe, permet la modification de plusieurs paramètres en une seule et unique opération [16].

Figure 5.11 – *Modification des attributs visuels*

Chacun des objets possède par défaut des valeurs initiales. Celles-ci ont été choisies plus ou moins arbitrairement (par exemple un trait continu bleu pour les courbes, la couleur noir pour les plans, ...) mais peuvent aussi être modifiées dans les préférences de ***Calques 3D*** (cf. section 5.7)

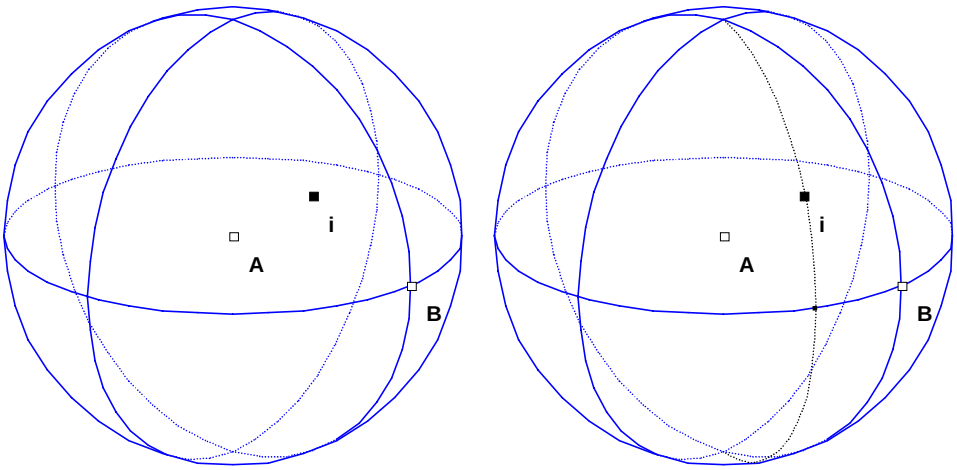
16. Pour permettre une modification uniforme et à la volée des attributs, nous avons ajouté une commande nommé `Pipette` qui permet de "copier" les attributs d'un objet et de les réappliquer à d'autres objets de la même catégorie.

b) Affichage des marques

Pour cela, ***Calques 3D*** fournit une commande `Marquer...` qui permet de désigner à la volée les objets dont on souhaite afficher les marques. Cette action peut aussi se faire localement à un objet dans la boîte de dialogue de modification des attributs (en cochant l'attribut *marqué*, figure 5.11).

Pour éviter de surcharger inutilement une figure géométrique, ces *éléments visuels de compréhension* ne sont pas affichés en permanence. De même que les rétroactions visuelles des tâches de construction, ils n'ont qu'un statut temporaire. L'utilisateur peut à tout moment demander l'affichage des *éléments visuels de compréhension* de tous les objets ayant été préalablement marqués.

Prenons un exemple concret. La figure 5.12 représente une sphère définie par son centre A et passant par le point B. Le point i est construit sur l'enveloppe de la sphère. Visuellement, il n'y a aucune indication de l'appartenance du point i à la sphère : il pourrait se trouver n'importe où dans l'espace. Bien sûr, il est possible de modifier le point de vue pour tourner autour de la sphère et conjecturer les propriétés du point i. Mais il est plus efficace et intéressant de *marquer* ce point et d'*afficher les marques* : le demi-grand cercle de la sphère passant par le point i est alors affiché, ce qui permet de confirmer visuellement l'appartenance du point à la sphère mais aussi sa position relative (*devant* la sphère).



(a) Sans affichage des marques. (b) Avec affichage des marques.

Figure 5.12 – *Représentation visuelle d'un point sur une sphère*

La plupart des objets géométriques disposent de marques équivalentes.

La liste complète des objets géométriques, de leur présentation graphique et des marques associées se trouve dans l'annexe C.

c) Le cas des objets infinis

L'un des problèmes les plus intéressants de la géométrie 3D est la présentation à l'interface des objets infinis comme la droite ou le plan.

Pour les droites, ce problème a été résolu de manière "traditionnelle" en utilisant le cadre de la fenêtre graphique pour limiter le dessin. Mais pour le plan, le problème est tout autre

et a nécessité de longues discussions avec les enseignants des différents groupes de travail (cf. section 4.4.3) pour arriver à une solution, sinon universelle, du moins acceptable.

La solution adoptée, ainsi que les conséquences de ces choix, sont illustrées dans la figure 5.13 dont l'énoncé est le suivant :

> *Soit P_1 le plan passant par trois points A, B et C. Soient I un point appartenant à ce plan et D la droite passant par I et perpendiculaire à P_1.*
>
> *Soient M un point appartenant à la droite D et P_2 le plan passant par M et perpendiculaire à la droite D.*

Nous représentons le plan sous la forme d'un rectangle fini qui englobe tous les points appartenant explicitement au plan (c'est-à-dire les trois points A, B et C servant à sa construction mais aussi le point I, construit sur le plan). De manière à ne pas confondre plan et polygone, une certaine marge est ajoutée pour déterminer ce contour. De plus, deux des côtés du plan sont toujours parallèles au plancher horizontal, ce qui permet de matérialiser son intersection avec celui-ci.

Bien évidemment, il ne s'agit que d'une représentation visuelle, le plan conservant son concept infini : lors d'une déformation (déplacement des point-bases du plan, déplacement d'un point appartenant au plan, ...), son contour est automatiquement adapté à la nouvelle configuration de la figure. Les choix faits pour la définition de ce contour assurent ainsi une continuité de son aspect visuel lors des déformations.

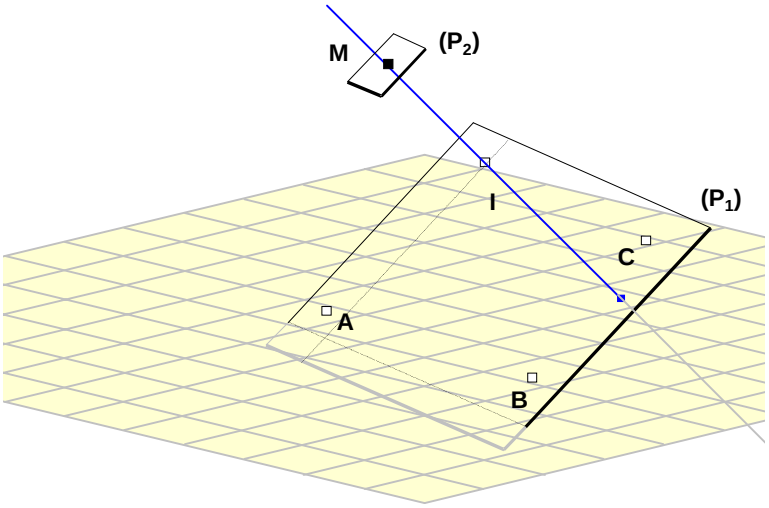


Figure 5.13 – *Représentation graphique des plans*

Ce mode de représentation du plan a deux caractéristiques visuelles intéressantes du point de vue pédagogique. Premièrement, le fait d'avoir deux des côtés du plan parallèle au plancher horizontal implique que toutes les droites perpendiculaires à ceux-ci (en particulier les deux autres côtés) représentent *la ligne de plus grande pente du plan*. Ce concept, très utilisé dans l'enseignement de la géométrie, est donc défini comme *éléments visuels de compréhension* pour les points construits sur un plan (c'est le cas du point I de la figure). Deuxièmement, les propriétés du contour apparent du plan permettent d'assurer que si deux plans sont parallèles (comme les plans P_1 et P_2) alors leur contour (c'est-à-dire les côtés des rectangles) le sont aussi, proposant ainsi une lecture directe de cette propriété.

5.5 Les fonctions d'exploration de la figure

Le dernier objectif de ***Calques 3D*** est de permettre à l'utilisateur l'exploration de la figure afin d'en découvrir ou vérifier les propriétés intéressantes. Deux fonctionnalités en particulier satisfont cet objectif : la déformation de la figure par le déplacement des point-bases et l'extraction d'une partie (ou de la totalité !) de la figure dans un calque.

5.5.1 Déformation de la figure

Comme nous l'avons déjà signalé, la géométrie mise en œuvre dans ***Calques 3D*** est une géométrie du point : toute construction se fait à partir de points préalablement créés. Cela a pour conséquence que la modification structurelle d'une figure ne peut se faire qu'à partir de déplacements des points initiant sa construction. Par exemple, il est impossible de saisir une arête d'un cube dans le but de lui faire subir une translation relative à l'univers. Il faut déplacer successivement les points ayant servis à sa construction, même si cela nécessite plusieurs déplacements pour "simuler" la translation du cube

Dans la section précédente, nous avons présenté les différentes constructions possibles de points. Cela permet la mise en évidence leurs *degrés de liberté* :

- les points libres, déplaçables sans contrainte dans l'espace,
- les points semi-libres, contraints par un objet support (point sur une droite, point sur une sphère, ...),
- les points induits par la logique même de leur construction (milieu d'un segment, point d'intersection, ...).

Seules les deux premières catégories sont des points pouvant être déplacés et entraîner une déformation de la figure. Nous les appellerons les *point-bases* de la figure.

Le principe de la déformation reprend en partie celui de la création d'un point : une décomposition du déplacement dans l'espace en déplacements plans et linéaires alternés. La seule différence est que la déformation d'un point ne nécessite pas la présence des cloisons, contrairement à sa construction. Dans ce cas, l'utilisateur ne dispose plus d'un plan de base visible pour initier le déplacement. Le choix de la décomposition est alors fourni grâce à un menu contextuel (figure 5.14).

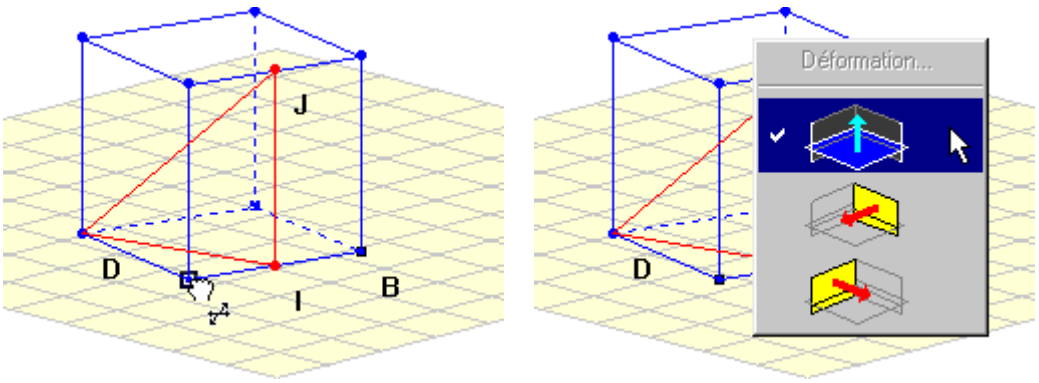


Figure 5.14 – *Décomposition des déplacements pour la déformation*

La présence d'un référentiel est recommandée pour une meilleure visualisation du déplacement et de ses implications structurelles grâce à la matérialisation des projetantes du point-base au cours de la déformation.

5.5.2 Les calques

L'objectif principal de ***Calques 3D*** est l'observation d'une construction géométrique dans l'espace. Dans certaines situations, il peut être souhaitable de visualiser un sous-ensemble de la construction sans interférences visuelles avec le reste de la construction. Or, gommer tous les objets non désirés pour arriver à cette fin peut s'avérer fastidieux, surtout s'il faut les rendre à nouveau visibles à la fin de la tâche d'observation. Pour cela, ***Calques 3D*** dispose d'un processus de filtrage d'une construction qui consiste à extraire de la construction principale les éléments que l'on désire observer comme un tout et de les copier dans une nouvelle fenêtre, désignée sous le terme de *calque*.

Cette extraction n'est pas une opération géométrique mais une opération visuelle, dans le sens où les propriétés intrinsèques d'un objet ou les relations liant plusieurs d'entre eux sont conservées, seule leur représentation graphique étant copiée dans une nouvelle fenêtre.

a) Extraction de sous-figures

Comme nous l'avons signalé précédemment, une fenêtre *calque* n'est en fait qu'un cas particulier de la fenêtre *Univers* dans laquelle il n'est pas possible de construire de nouveaux éléments : les seules fonctions disponibles sont celles qui permettent l'observation (modifier le référentiel, changer le point de vue, ...) et l'exploration (déformation, extraction, ...) de la figure géométrique. De manière à limiter la surabondance de fenêtres à l'écran, nous avons limité le nombre de calques disponibles à quatre.

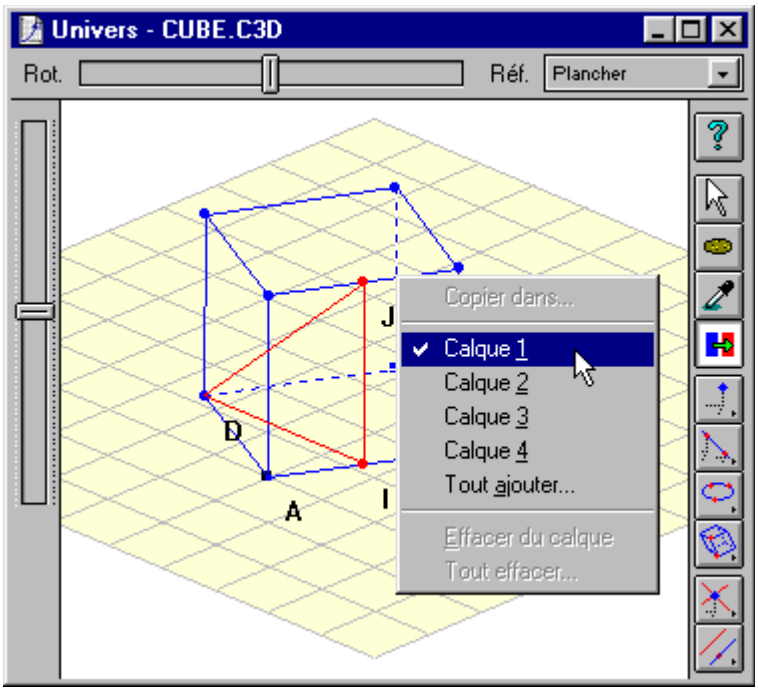


Figure 5.15 – *Extraction d'une partie de la figure.*

Il est possible d'opérer toutes les combinaisons possibles d'extraction d'objets :

- extraire un objet dans un calque, à partir de n'importe quelle fenêtre (calque ou univers),
- extraire tous les objets d'une fenêtre (calque ou univers) vers un autre calque,

– enlever d'un calque (uniquement, et pas de l'univers) un objet extrait au préalable,
– enlever d'un calque (et pas de l'univers) tous les objets extraits.

Ces différentes possibilités d'extraction se trouvent dans le menu contextuel, accessible grâce au bouton droit de la souris (figure 5.15). L'extraction des objets se fait en les désignant successivement à l'aide de la souris.

La tâche d'extraction concerne uniquement la désignation des objets. Il faut de plus demander explicitement l'affichage du calque correspondant.

b) Observation et déformation dans les calques

L'extraction d'un élément vers un calque donné consiste en un filtrage visuel de la construction : les objets sont en fait partagés entre différentes fenêtres. Cela implique en particulier que toute modification de l'aspect visuel d'un objet (changer la couleur ou la forme d'un objet, masquer un objet, ...) dans une fenêtre, *Univers* ou *Calque*, entraîne sa modification dans les autres fenêtres le contenant.

Par contre, les informations concernant l'univers géométrique (perspective, position de l'observateur, référentiel, affichage ou non des marques) sont liées à la fenêtre. Cela permet par exemple d'observer la construction selon un point de vue donné dans la fenêtre *Univers* et selon un autre dans une fenêtre *Calque*. La figure 5.16(a) illustre cette caractéristique : la figure de la fenêtre *Univers* (à gauche de l'écran) est observée simultanément dans une fenêtre *Calque* (à droite de l'écran), selon un point de vue différent et avec le *plancher* comme référentiel.

La déformation de la construction (par déplacement de ses point-bases) dans une fenêtre entraîne une modification similaire dans toutes les autres fenêtres où les parties déformées de la construction ont été copiées. Cela permet d'étudier les variations d'une sous-partie de la figure extraite dans un calque lors du déplacement d'un point-base dans l'univers.

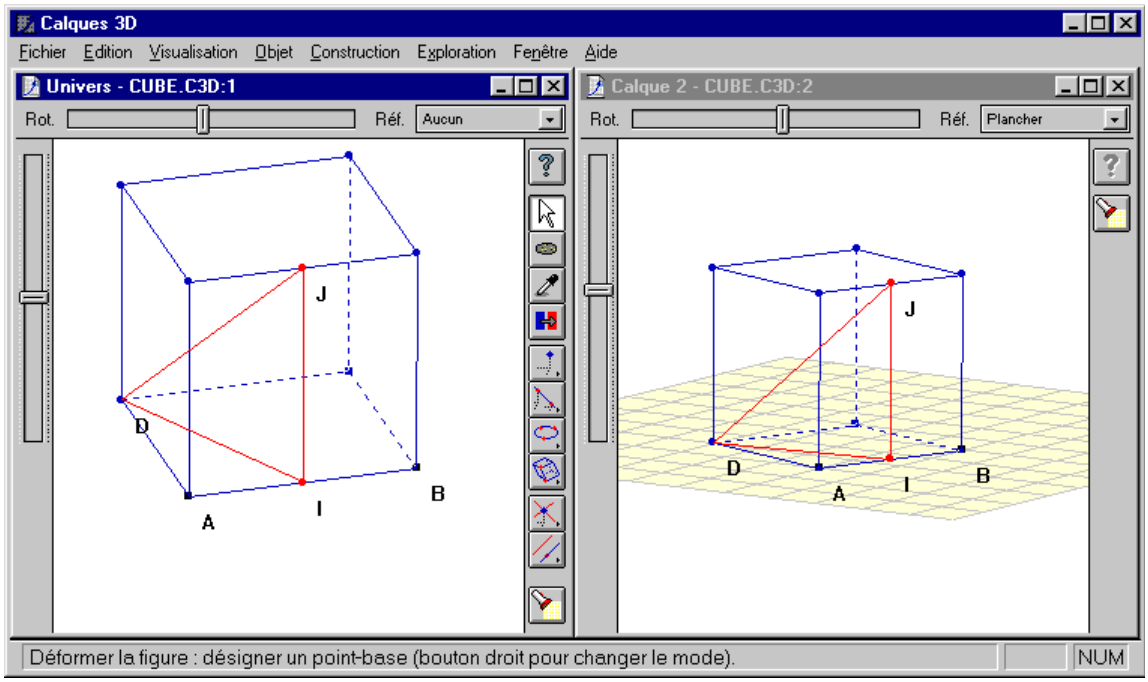
De même, il est possible de cumuler les avantages de la déformation, de l'extraction de sous-figures et de l'observation selon des points de vue particuliers en une seule opération. Une figure extraite dans un calque (par exemple le triangle DIJ de la figure 5.16(b)) peut être observée en projection frontale et *conservée* dans cette position. L'élève peut alors déformer la figure dans la fenêtre *Univers* : les déformations sont répercutées dans la fenêtre calque mais la figure est maintenue en projection frontale durant la déformation. Cela permet d'observer les variations planes de la figure, sans que d'éventuelles variations spatiales viennent en perturber l'interprétation.

Notons pour finir que la rapidité du rafraîchissement simultanée des différentes fenêtres au cours d'une déformation dépend de la complexité de la figure : plus le nombre d'objets déformés est important, plus l'affichage est long. Pour permettre une lecture efficace, la mise à jour des fenêtres peut être désynchronisée : la mise à jour de la fenêtre où à lieu la déformation est immédiate alors qu'elle est reportée à la fin du déplacement (lorsque l'utilisateur relâche le point-base) dans toutes les autres.

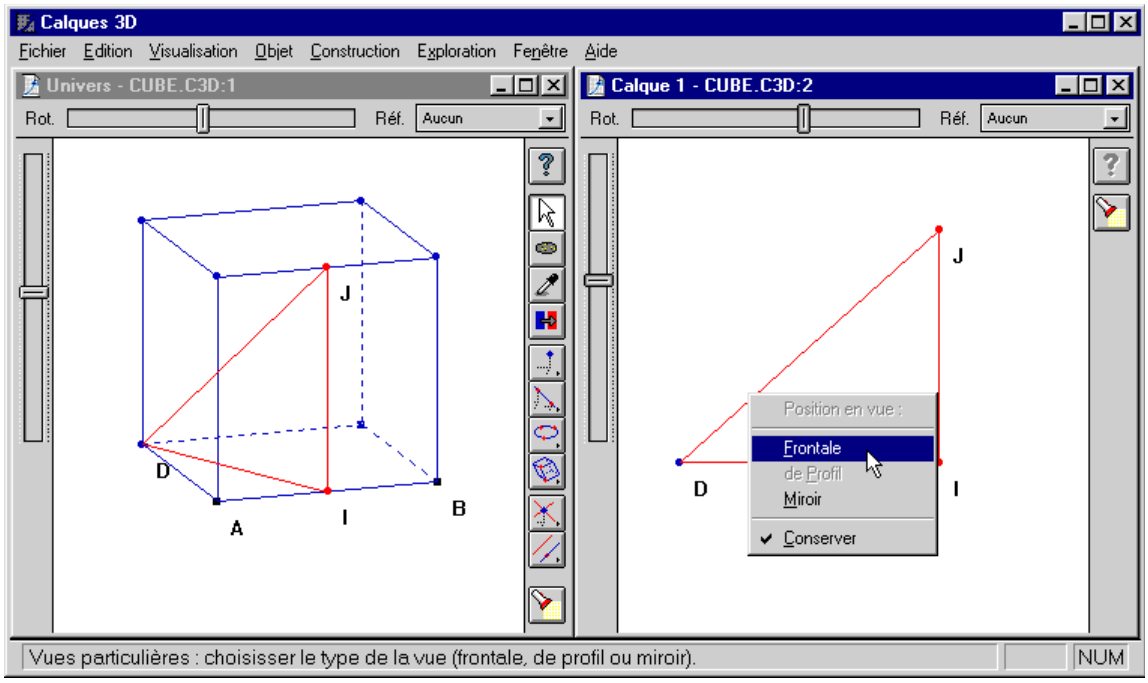
5.6 Aides et explications

5.6.1 Aide contextuelle et aide en ligne

La barre de statut, un des éléments récurrents des interfaces Windows, fournit une aide textuelle succincte et immédiate à la tâche ou à la commande en cours : c'est ce que nous appellons *aide contextuelle*.



(a) Observation de la figure sous deux points de vue différents.



(b) Observation d'une partie extraite en projection frontale.

Figure 5.16 – *Deux exemples d'utilisation des calques*

L'intérêt de cet outil est sa présence en permanence au bas de l'espace de travail[17], fournissant ainsi un repère permanent à l'utilisateur. Il y a cependant deux inconvénients à cet outil. D'une part sa position en bas de l'espace de travail peut introduire une distance trop importante entre ce que fait l'utilisateur et ce qu'il doit (ou peut) lire. En terme d'IHM, cet impact est non négligeable. D'autre part, la taille de cette zone d'aide est forcement très réduite : l'aide textuelle fournie doit donc être suffisamment courte et explicite pour pallier cette restriction.

Dans ***Calques 3D***, le contenu de cette aide dépend de plusieurs situations :

1. lorsqu'une commande de menu ou une icône est sélectionnée, une ligne de commentaire s'affiche dans la barre de statut, fournissant ainsi une rapide présentation des objectifs de la commande (figure 5.17(a)) ;
2. lorsqu'une tâche de construction est activée, la barre de statut fournit, étape par étape, des indications sur la manière de mener celle-ci à bien (figure 5.17(b)) ;
3. lorsque l'utilisateur désigne un objet, dans une tâche de construction par exemple, la barre de statut affiche le descriptif de l'objet en question. Associé à l'étiquette de désignation, ce texte permet une identification plus précise de l'objet cible (figure 5.17(c)).

Cette multitude des situations où des aides textuelles peuvent être affichées dans la barre de statut nécessite une organisation précise et invariante. Celle-ci est gérée par l'affichage permanent d'une information textuelle, quelle qu'elle soit. L'organisation des tâches dans le logiciel (cf. section 5.2.2) et l'existence d'une *tâche par défaut* nous ont amené à afficher par défaut l'aide sur la tâche active, fournissant ainsi à l'utilisateur une indication supplémentaire sur sa situation dans la réalisation d'une activité.

En tant qu'aide immédiate, le but de cet outil n'est pas de fournir une aide complète. C'est le rôle de l'aide en ligne (figure 5.18). Celle-ci se présente sous la forme d'un hypertexte `Windows` qui contient des explications détaillées sur les différentes fonctionnalités du logiciel. Dans sa version actuelle, le texte de cette aide a été rédigé sans le concours d'un enseignant utilisateur. Il devrait être repris, de manière à assurer une rédaction *pédagogique* et non *informatique*. D'un point de vue "ouverture du logiciel", il faudrait envisager de fournir à l'enseignant prescripteur des outils lui permettant de rédiger ou d'adapter ce texte à son propre contexte.

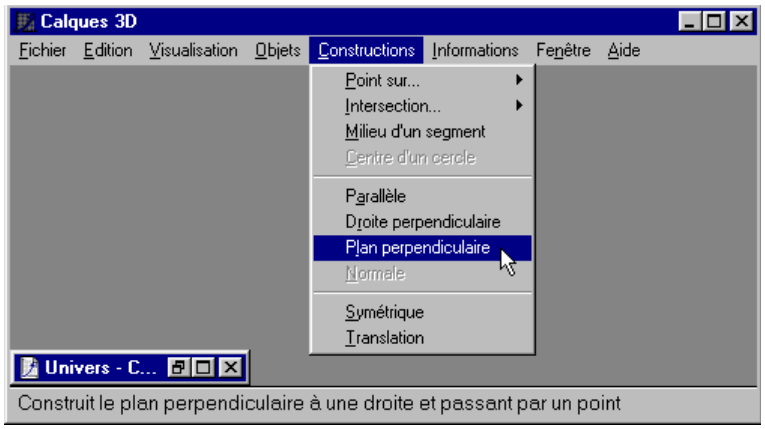
5.6.2 Information sur les cas particuliers

Calques 3D est un micromonde pour la construction, la visualisation et l'exploration de figures géométriques. En aucun cas, il ne s'agit d'un tuteur d'aide à la résolution de problèmes ou à la démonstration. En ce sens, la capacité explicative de ***Calques 3D*** est forcément réduite mais non inexistante.

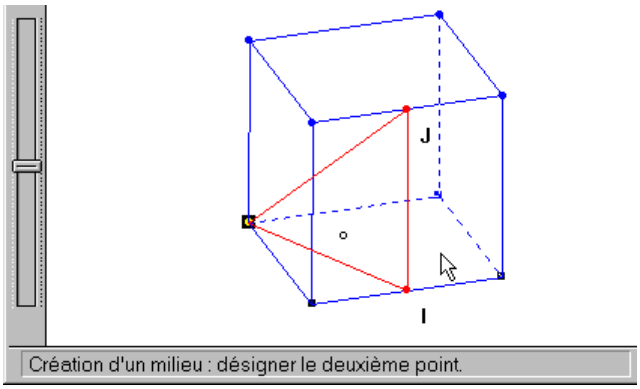
En effet, nous avons déjà souligné le fait que la géométrie dynamique introduisait de nouvelles propriétés à la géométrie classique, induisait de nouveaux comportements dûs principalement à l'aspect dynamique. Si les explications relatives à la géométrie euclidienne sont, et doivent rester, à la charge de l'enseignant, c'est par contre au logiciel de fournir celles relatives à la géométrie dynamique.

Prenons un exemple : l'intersection de deux segments, lorsque ceux-ci ne se coupent pas. Lors de la construction d'une telle intersection, nous avons vu que ***Calques 3D*** refusait tout simplement la création du point. Mais, même si les propriétés des deux segments permettent la création de l'intersection, il peut arriver, au cours de la déformation de la figure, que la nouvelle

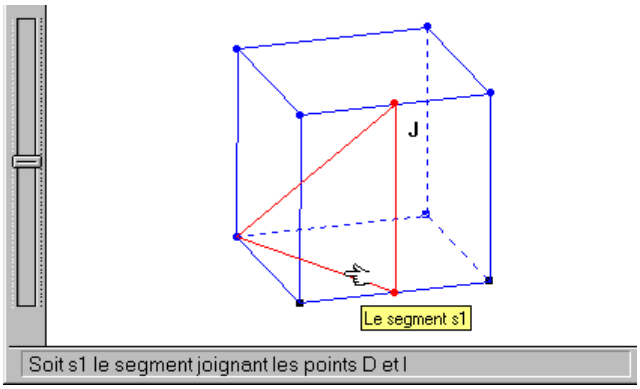
17. Même si, dans certaines applications, il est maintenant possible d'inhiber l'affichage de cette barre.



(a) Description d'une commande



(b) Aide sur la tâche active



(c) Aide sur la désignation

Figure 5.17 – *Le contenu de la barre de statut change suivant le contexte d'utilisation.*

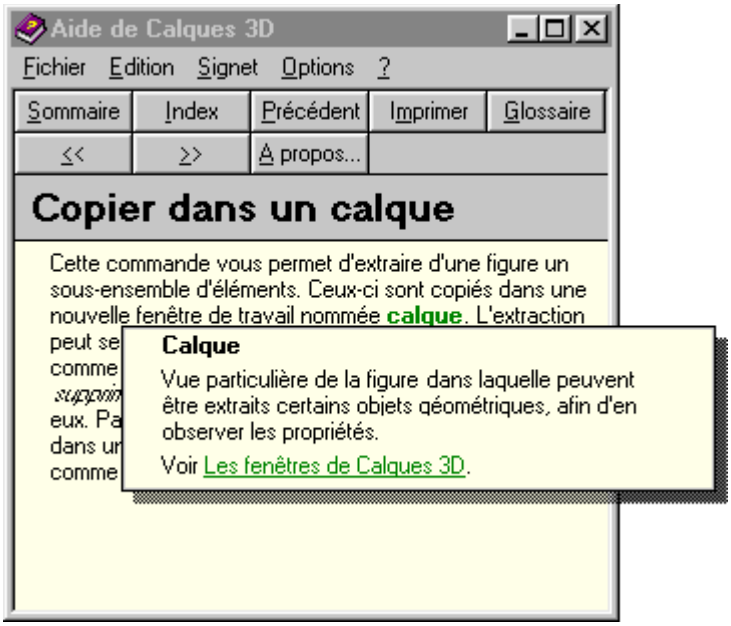


Figure 5.18 – *L'aide en ligne de* ***Calques 3D***

configuration ne vérifie plus les conditions d'existence du point. Dans ce cas, ***Calques 3D*** applique ce qui est devenu une constante dans tous les logiciels de géométrie dynamique : le point n'a plus d'existence et disparaît de la figure. Cependant, il ne s'agit pas d'une destruction mais d'une situation temporaire : si les segments sont repositionnés de telle manière que l'intersection existe à nouveau, le point réapparaît. De même, il ne s'agit pas non plus de l'effacement d'un objet : tous les objets dépendants de ce point disparaissent eux aussi, leurs conditions d'existence n'étant plus vérifiées.

Le problème est que, visuellement, effacement, destruction et disparition d'un objet sont équivalentes. Pour aider l'utilisateur à comprendre ce genre de comportement lié à l'aspect dynamique de la géométrie, nous avons amélioré le rôle de la fenêtre *Historique*.

Celle-ci présente à tout instant la liste de tous les objets de la construction géométrique (figure 5.19), que ce soient les objets visibles (en texte plein), les objets gommés par l'utilisateur (en texte grisé), ou les objets inexistants du fait des cas particuliers de la figure (en texte grisé, avec un drapeau rouge devant). En cliquant sur ces derniers objets, un message apparaît, indiquant la raison du cas particulier.

5.7 Paramétrisation par l'enseignant

Nous avons montré, au long de ce chapitre et des précédents, que de nombreux points nécessitaient des possibilités de choix pour l'utilisateur mais surtout des possibilités de configuration pour l'enseignants prescripteur. Si tous les éléments de choix ont bien été donnés à l'élève (choix des attributs visuels des objets, choix du référentiel, ...), ils ne sont pas tous disponibles pour une paramétrisation par l'enseignant. La raison principale de cette lacune réside dans la difficulté à mettre en place une interface adaptée, c'est-à-dire une interface qui permette à l'enseignant de configurer *facilement et rapidement* l'ensemble des paramètres du logiciel. Nous avons essayé plusieurs métaphores d'organisation et de présentation à l'interface de ces choix (cases à cocher, listes hiérarchiques, boîtes à signets, ...) mais, face à l'abondance des éléments paramétrables,

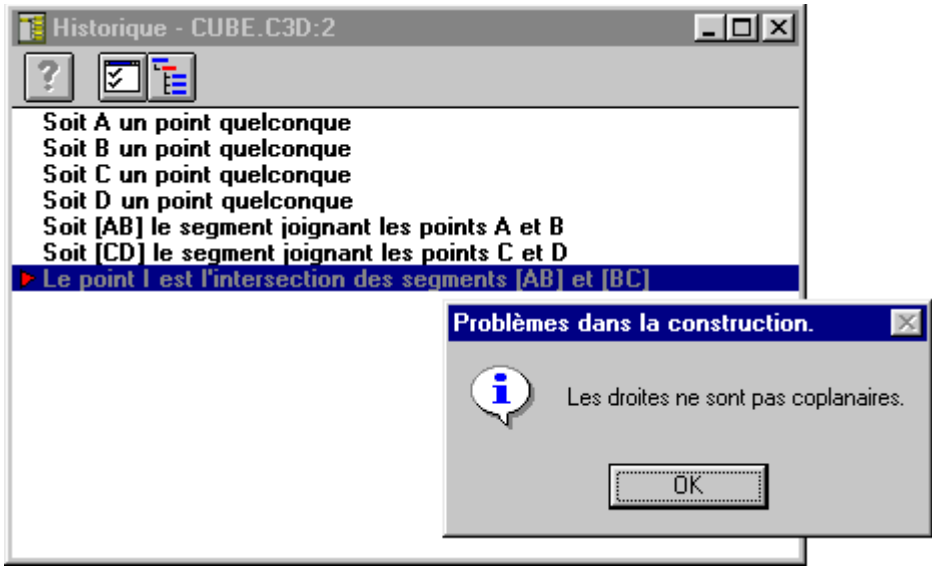


Figure 5.19 – *Fenêtre historique et explications sur la figure*

aucune n'a pu remplir complètement des conditions d'*utilisabilité* satisfaisantes.

Dans la version actuelle, ***Calques 3D*** propose un ensemble minimum d'éléments configurables. L'accès à la configuration se fait par l'intermédiaire d'une boîte de dialogue (cf. figure 5.20), qui peut être obtenue à tout moment d'utilisation du logiciel[18].

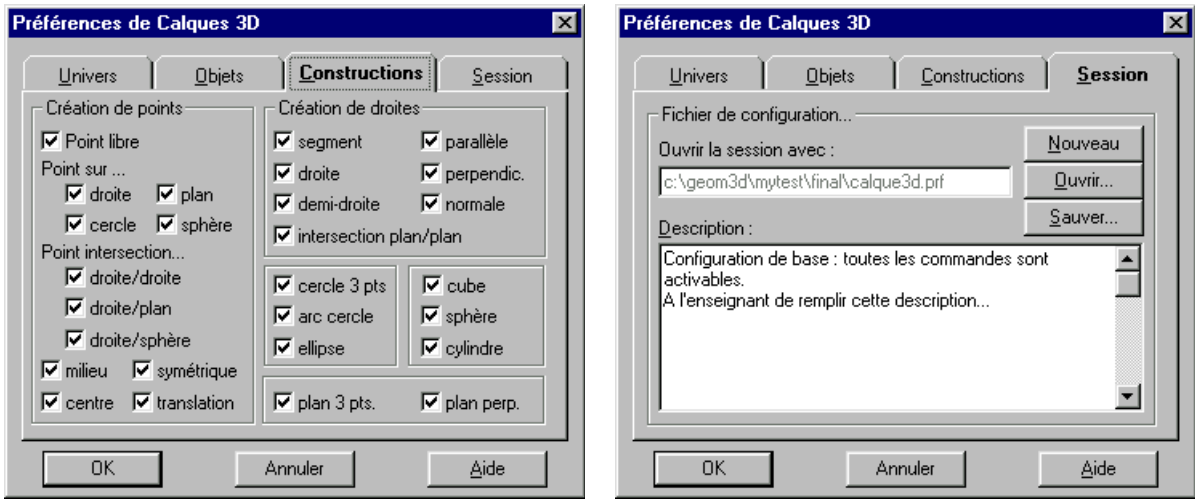


Figure 5.20 – *Modification des préférences de* ***Calques 3D***

Celle-ci regroupe les éléments configurables sous la forme de quatre signets :

- le signet `Univers` permet de définir et de restreindre les fonctionnalités d'observation d'une figure géométrique (choix d'un référentiel privilégié, disponibilité des fonctions de projection ou de modification du point de vue de l'observateur, ...),
- le signet `Objets` permet de fixer les modes de présentation des objets géométriques

18. Une autre lacune, qu'aucun des membres des groupes de travail n'a soulignée, concerne la nécessité d'identification de l'enseignant (par son nom et/ou un mot de passe par exemple).

(assignation d'attributs visuels particuliers pour les points en fonction de leur degré de liberté, définition des couleurs disponibles pour le tracé, ...),

– le signet `Constructions` permet de définir la disponibilité des différentes commandes de construction.

L'enseignant a aussi, grâce au signet `Session`, la possibilité de sauvegarder une configuration, de manière à pouvoir la charger à l'ouverture du logiciel par l'élève, ou de définir une nouvelle configuration sur la base d'une sauvegarde précédente.

Les modifications apportées dans la configuration du logiciel sont prises en compte immédiatement : il n'y a nul besoin de fermer et de relancer l'application.

5.8 Conclusion

La question des choix de présentation des objets et relations géométriques à l'interface est un problème particulièrement difficile qui doit faire l'objet de travaux à la fois dans le domaine des sciences cognitives, de la didactique et de l'ergonomie, de manière à analyser les métaphores de manipulation directe et les rendus visuels propres à favoriser l'engagement direct. Les réflexions que nous avons eues à ce propos nous ont permis de mettre en place divers mécanismes permettant, malgré l'aspect "transparent" de la géométrie filaire, de favoriser la lisibilité d'une figure géométrique (attributs visuels des points en fonction de leur degré de liberté, *éléments visuels de compréhension* pour matérialiser les propriétés et relations géométriques).

L'importance de l'ergonomie a été aussi mise en avant, d'une part en ce qui concerne la réactivité du système aux actions de l'utilisateur (nécessité de temps de réponse court, ce qui a entraîné des limitations de certaines fonctionnalités), d'autre part en ce qui concerne l'acceptabilité de l'environnement (aides visuelles sur les procédures de construction, réduction de la charge cognitive en limitant le nombre de calques simultanément ouverts).

Enfin, il est à noter que, même si ***Calques 3D*** est un prototype en cours de développement, il est néanmoins pleinement opérationnel. Dans sa version actuelle, il ne couvre pas l'intégralité du programme de l'enseignement de la géométrie dans le second degré, mais permet d'en aborder un bon nombre de parties, que ce soit dans le cadre de l'enseignement général ou technique.

Même si aucune expérimentation et évaluation *"in-vivo"* n'ont été réalisées, les enseignants auteurs impliqués dans le développement de ***Calques 3D*** ont affirmé leurs appréciations sur les choix d'implantation et de mise en œuvre effectués.

Chapitre 6

Implantation de *Calques 3D*

6.1 Introduction

Après avoir présenté les contraintes de la mise en œuvre, nous décrivons les choix de modélisation des figures géométriques (représentation interne des objets et des relations géométriques), des modes de présentation externes de ces figures et de gestion de l'interaction et de la dynamique (c'est-à-dire les liens entre représentation interne et représentation externe).

6.2 "Contraintes" de mise en œuvre

Le choix d'un langage de programmation et des outils annexes (bibliothèque, environnements de développement, ...) pour l'implantation de ***Calques 3D*** s'est fait selon un ensemble de critères dont nous avons donné un aperçu au long des chapitres précédents et que nous résumons ici.

Dans la section 4.5, nous avons présenté une organisation des connaissances qui utilise un modèle en quatre couches, issu de [Bernat 95]. Ce modèle, proche d'une méthodologie objet, permet en effet une approche modulaire du problème, c'est-à-dire de prendre en compte les différents besoins des participants au projet (besoins des enseignants, besoins des informaticiens, ...), et d'y associer les contraintes d'implantation. Il permet :

- la prise en compte du *domaine* d'apprentissage,
- la prise en compte des contraintes de développement en instaurant une *représentation interne unique* des objets du domaine qui permet d'assurer une cohérence entre les différentes couches de l'implantation,
- la prise en compte des choix de l'enseignant en permettant la mise en œuvre de différents modes de *présentation* des connaissances du domaine et qui nécessitent la gestion de ces points de vue (modification d'états par l'élève, ...),
- la prise en compte de la plateforme de visualisation pour la traduction à l'*interface* de ces points de vue. Elle nécessite la gestion du périphérique d'affichage (nombre de couleurs, taille de l'écran, ...), la gestion de l'ergonomie d'usage (organisation de l'espace de travail, ...), la gestion de l'interaction avec l'utilisateur (périphériques d'entrée, gestion des fenêtres, propagation des messages, ...).

Grâce à ses caractéristiques, la programmation objet offre un intérêt certain pour une conception basée sur le prototypage. Par la surcharge et la modularité, elle permet en effet une mise à jour rapide et efficace du code (représentation interne, ajout de points de vue, ...).

De plus, elle offre une proximité intéressante entre *classes objet* et *objets géométriques*. L'héritage des classes est en effet à mettre en parallèle avec la spécialisation des objets géométriques et leurs propriétés. Ainsi, *un point sur une droite* étant une spécialisation d'un *point quelconque*, il n'est pas anormal de retrouver cette relation au niveau des classes qui implanteront ces objets géométriques.

Calques 3D est avant toute chose une évolution d'un logiciel existant : une adaptation à la géométrie dans l'espace de `Calques 2` [Bernat 94a], conçu pour l'apprentissage de la géométrie plane. Nous donc essayé de réutiliser autant que possible les travaux faits par Philippe Bernat sur ce logiciel ; cette réutilisation n'a pu se faire qu'au niveau des concepts, et non du code proprement dit. En effet, `Calques 2` a été développé avec une version objet du langage de programmation `Pascal`, qui n'est actuellement presque plus utilisé et n'offre plus des conditions de maintenance et de suivi suffisantes. Nous avons préféré développer ***Calques 3D*** avec le langage `C++` qui bénéficie de l'existence de nombreux environnements de développement performants, de bilbiothèques ou de boîtes à outils permettant une mise en œuvre rapide de certaines parties du logiciel (en particulier la gestion de l'interface).

Enfin, le dernier critère de mise en œuvre, et certainement le plus important, est le respect du dispositif d'usage disponible aux utilisateurs potentiels du logiciel (enseignants et élèves). Dans les collèges et les lycées français, les micro-ordinateurs de type `PC` disposant du système d'exploitation `Windows` sont les plus répandus : nous avons donc orienté nos choix de développement en direction de cet environnement. Ce choix préalable n'est pas sans conséquences car, à de trop rares exceptions, il n'y a pas de langage de programmation qui permette une conception véritablement indépendante d'un système d'exploitation donné[19]. Le modèle en quatre couches, en permettant une séparation des informations, garantit que seule la couche *interface* soit dépendante de la plateforme de développement. Cependant, par rapport à la partie interface, le noyau fonctionnel réutilisable représente une partie suffisamment peu importante du développement pour que le portage de l'application vers une autre plateforme ne soit pas réaliste.

De manière plus indirecte, le respect du dispositif d'usage nécessite la prise en compte de ses capacités techniques (mémoire disponible, vitesse d'exécution, ...). La conception de ***Calques 3D***, en particulier la partie géométrique, nécessite souvent le recours à des algorithmes classiques de *géométrie computationnelle* ou d'*informatique graphique*(intersection de polygones, *clipping* 2D et 3D, ...). Ces algorithmes sont bien décrits dans la littérature (citons entre autres [Rogers 90, Foley 90, Watt 93, Goodman 97, de Berg 97, O'Rourke 98]), voire déjà implantés dans des bibliothèques. Cependant, nous pouvons faire remarquer que, soit ces bibliothèques sont trop générales et peu modulaires pour une intégration telle quelle dans le logiciel, soit les algorithmes en question ne permettent pas de prendre en compte l'aspect dynamique de la géométrie introduite dans ***Calques 3D***. Pour ces raisons, il nous a fallu souvent réécrire et adapter ces algorithmes de manière à garantir une rapidité de calcul et une fluidité de l'affichage.

Il est intéressant de noter que, même si de nombreux logiciels de géométrie dynamique (dans le plan ou dans l'espace) ont vu le jour, il n'existe aucune base commune (bibliothèques, algorithmes, axiomatisation, ...) sur laquelle un informaticien peut se reposer pour développer un nouvel environnement. Cela tient en partie à l'aspect "confidentiel" du code de certains logiciels commerciaux, en partie aux approches relativement différentes qui peuvent exister d'un logiciel

19. Le langage objet `Java`, grâce aux *"applets"* ou à sa *"machine virtuelle"*, permet un tel développement multiplateforme. Cependant, au moment où nous avons commencé le développement de ***Calques 3D***, les capacités graphiques de ce langage étaient très faibles. De plus, de par sa nature interprétée et non compilée, il souffre de lenteurs pouvant s'avérer préjudiciables pour une exploration libre et des rétroactions immédiates, indispensables pour un micromonde.

à l'autre pour la mise en œuvre des propriétés de la géométrie dynamique.

En résumé, ***Calques 3D*** est développé avec l'environnement de développement intégré (*Integrated Development Environment, IDE*) `Borland C++` version 4.5. Il représente environ 25000 lignes de codes dont 60% sont consacrées au noyau fonctionnel, 40% à la gestion de l'interface.

6.3 Architecture générale de *Calques 3D*

L'architecture de ***Calques 3D*** est présentée dans la figure 6.1. Afin de donner un aperçu général du logiciel, nous allons en décrire brièvement chacune des composantes, les plus importantes seront discutées dans les sections suivantes de ce chapitre.

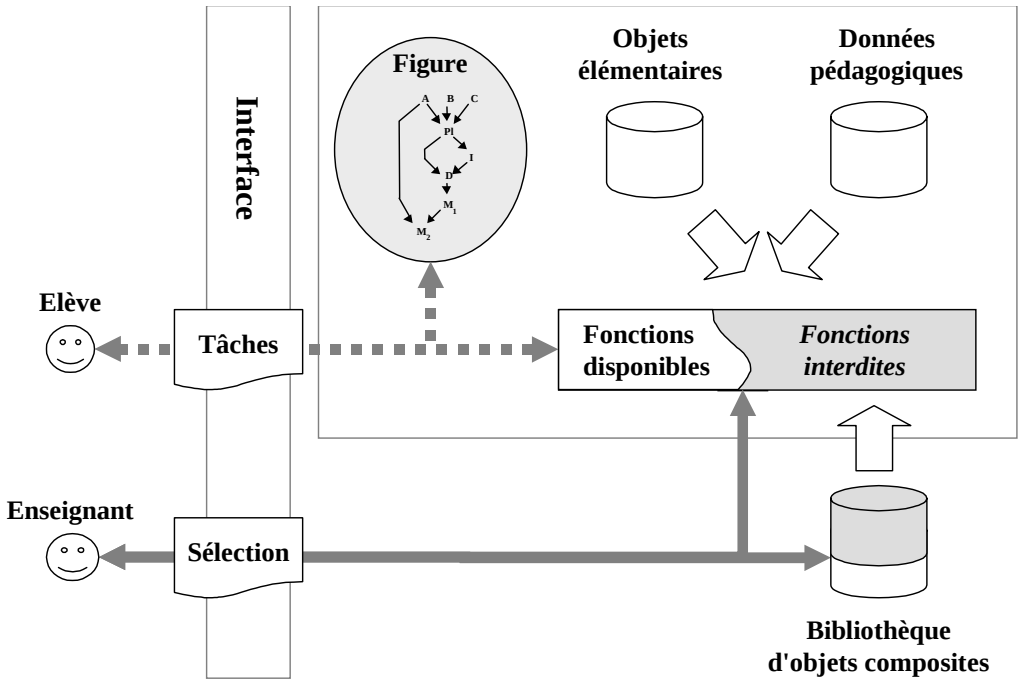


Figure 6.1 – *Architecture de* ***Calques 3D***

Schématiquement, ***Calques 3D*** est constitué de deux parties : d'une part l'ensemble des composantes intégrées qui constituent le corps du système ; d'autre part l'interface qui constitue le module de communication entre les utilisateurs et le système.

Le corps du système regroupe les composantes suivantes :

- une base de *données pédagogiques* qui contient les aides contextuelles, l'aide en ligne, les explications, ... La question des aides et explications a fait l'objet d'une discussion dans le chapitre précédent (cf. section 5.6) ;
- une base de modèles *d'objets géométriques élémentaires* qui seront mis en relation entre eux de manière à constituer une *figure géométrique*. La représentation interne de celle-ci (définition des objets et des relations géométriques) est décrite dans la section 6.4 ;
- un ensemble de *fonctions* qui permettent la manipulation de la figure géométrique, que se soit au niveau de sa représentation interne (fonctions de construction) que de ses représentations externes à l'interface (fonctions d'observation, fonctions d'exploration).

L'articulation entre la représentation interne d'une figure géométriques et ses différentes représentations externes est présentée dans la section 6.5.

L'interface d'accès à ***Calques 3D*** est différente suivant le statut de l'utilisateur (enseignant et élève).

L'utilisateur enseignant dispose d'une *interface de sélection* des contenus pédagogiques lui permettant la paramétrisation du logiciel en fonction de ses besoins. Cette sélection s'effectue principalement au niveau des fonctions qui seront offertes à l'élève (cf. chapitre précédent, section 5.7).

L'utilisateur élève dispose d'une interface qui permet l'affichage et la manipulation directe d'une figure géométrique. Les interactions de l'élève avec cette dernières sont conditionnées par les fonctions disponibles et gérées au niveau de l'interface par un mécanisme de *tâches* qui est discuté dans la section 6.6).

Enfin, ***Calques 3D*** dispose de possibilités d'extension de sa base d'objets géométriques sous la forme d'une *bibliothèque d'objets composites*, construits à l'initiative de l'enseignant et pouvant être mis à disposition de l'élève pour exploitation. Cette ouverture du logiciel est réalisée par un mécanisme *macro-constructions* sont traitées dans la section 6.7.

6.4 Représentation interne des figures

Nous abordons dans cette section la question des structures de données utilisées pour la modélisation informatique des figures géométriques, c'est-à-dire les *objets géométriques* eux-mêmes et leur mise en relation au sein d'une figure par un *graphe de dépendance*. Nous concluons cette section par un cas particulier d'objets géométriques : les *objets composites*.

6.4.1 Les objets géométriques

Tout objet géométrique possède des propriétés analytiques qui le caractérisent (exemple : coordonnées pour un point, origine et vecteur directeur pour une droite, ...), un ou plusieurs modes de présentation à l'interface (exemple : le tracé d'une droite en mode graphique, son énoncé en mode textuel, ...) et un ensemble de relations qui déterminent ses propriétés et son rôle dans une structure hiérarchique représentant la figure (exemple : la relation d'appartenance pour un point construit sur une droite, ...).

Pour représenter ces différentes informations, nous avons tout naturellement adopté la distinction introduite par [Baulac 90] entre le *type* et la *catégorie* d'un objet [20]. Le *type* d'un objet caractérise son appartenance à un ensemble déterminé par ses propriétés propres comme objet géométrique (par exemple le point, le plan, la sphère, ...). La *catégorie* d'un objet désigne chaque spécialisation d'un type donné, que se soit en ajoutant une relation particulière avec d'autres objets de la figure (par exemple la catégorie des droites passant par deux points, la catégorie des droites parallèles à une droite, ...), ou en définissant des propriétés supplémentaires qui les caractérisent par rapport au type initial (par exemple la catégorie des demi-droites, ...).

20. Y. Baulac utilise le terme de *classe* d'objet. De manière à ne pas introduire une ambiguïté avec la terminologie de la méthodologie objet, et ainsi garder la clarté de l'exposé, nous préférons utiliser le terme de *catégorie*.

a) Modélisation des relations géométriques

Faire la distinction entre *type* d'objets et *catégorie* d'objets n'est pas un choix neutre et a une influence non négligeable sur la modélisation des connaissances dans ***Calques 3D***, en particulier au niveau de la représentation des relations géométriques. En effet, il y a (au moins) deux possibilités pour modéliser ces relations et permettre la structuration d'une figure géométrique (figure 6.2).

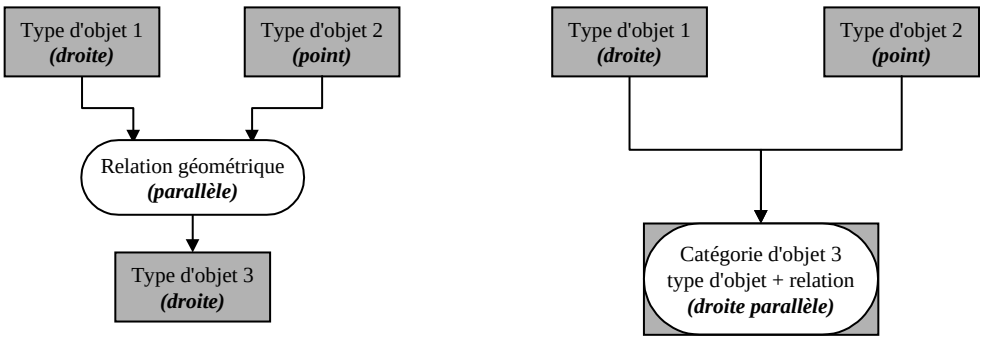


Figure 6.2 – *Deux méthodes de modélisation des relations géométriques* ***Calques 3D***

La première consiste à disposer de deux ensembles d'entités différentes, les *objets* et les *relations* géométriques, puis à les associer dans un réseau sémantique. Cette méthode a l'avantage de dissocier l'opérateur (la relation géométrique) de son résultat (la figure géométrique obtenue par la mise en relation des objets), ce qui permet facilement de changer la nature de la relation et d'observer le résultat de ce changement. De plus, l'ensemble des objets à modéliser est forcément restreint (il correspond à l'ensemble des *types* d'objets de la classification de Y. Baulac).

La deuxième consiste au contraire à inclure la relation géométrique au sein même de la définition des objets géométriques, introduisant ainsi la notion de *catégories* d'objet. Cela permet en particulier d'unifier les entités manipulées dans le logiciel, assurant plus facilement le contrôle du comportement et des réactions de ceux-ci à toute action de l'utilisateur ou du système, ainsi que la cohérence des relations géométriques mises en œuvre dans une figure.

C'est cette deuxième méthode que nous avons choisie dans ***Calques 3D***, cela grâce à la nature *déclarative* de la géométrie mise en œuvre dans les micromondes de géométrie dynamique, c'est-à-dire la construction d'une figure à partir d'un énoncé contenant les déclarations des objets à construire.

b) Les classes *"objets géométriques"*

La distinction entre type d'objet et catégorie d'un objet trouve ici toute sa justification. En effet, les objets d'un même type, quelles que soient leurs relations géométriques avec les autres objets de la figure, possèdent des caractéristiques communes (en général les représentations analytiques internes et les méthodes de tracé) alors que les catégories d'objets sont introduites pour mettre en évidence des particularités (par exemple différentes méthodes de calcul des représentations analytiques ou différentes réactions aux actions de l'utilisateur).

Les caractéristiques de tous les objets géométriques présentent une structure similaire : les champs et attributs mémorisent l'état de l'objet, les méthodes et opérateurs d'une classe constituent l'ensemble des instructions comprises par l'objet et peuvent être regroupées en quatre niveaux différents. La figure 6.3 ci-contre donne un aperçu commenté de la définition d'une classe *"objet géométrique"*, mettant en évidence les différents champs et méthodes, ainsi que leur

appartenance à l'un des niveaux de modélisation issus de l'organisation des informations (cf. section 4.5).

Au niveau **Domaine** (❶) correspondent :
- les attributs de définition, d'identification et d'information de l'objet (nom, textes pour l'aide et la désignation, énoncé, ...), ainsi que les méthodes d'accès correspondantes.

Au niveau **Représentation Interne Unifiée** (❷) correspondent :
- les attributs de représentation analytique de l'objet (coordonnées du point, origine et vecteur directeur de la droite, ...),
- la listes des parents de l'objet (c'est-à-dire les objets géométriques nécessaires à sa définition) et de ses descendants, qui définissent ainsi un graphe, dit *de dépendance*, assurant la structuration de l'objet dans la figure,
- les constructeurs (au sens programmation objet) de la classe qui, à partir des données d'interface, déterminent le graphe de dépendance de l'objet,
- la méthode de calcul de la représentation unifiée de l'objet en fonction de ses parents
- les attributs d'état de l'objet (sélectionné, marqué, valide, ...).

Au niveau **Présentation** (❸) correspondent :
- les attributs visuels de l'objet (forme, couleur, police, ...),
- les méthodes d'accès aux attributs visuels,
- les méthodes de calcul des différents modes de présentation de l'objet (c'est-à-dire les différentes présentations de la représentation unifiée),
- les méthodes de d'interprétation des événements externes (actions de l'utilisateur) et internes (actions du logiciel).

Au niveau **Visuel** ou **Interface** (❹) correspondent :
- les méthodes de traduction à l'interface des modes de présentation
- la méthode de calcul de la distance du curseur à l'objet

c) La hiérarchie des classes

Les classes représentant les différents objets géométriques peuvent être organisées sous forme arborescente, mettant en évidence l'héritage de leurs caractéristiques. Le sommet de l'arbre est la classe générique (ou *super-classe*) `TObject3D` qui définit l'ensemble des attributs et méthodes communs à tous les objets géométriques. Chaque type possède ses méthodes propres qui sont ensuite surchargées pour ses catégories. Cela assure un comportement homogène des objets de même type.

La figure 6.4 représente le sous-arbre correspondant aux droites. La classe `TDroite3D` (figure 6.5) détermine, pour tous les objets de type droite :
- la représentation analytique interne d'une droite par la donnée d'une origine et d'un vecteur directeur (❶),
- le calcul de sa représentation interne (méthode `CalculConceptuel()`, ❸) en fonction de ses parents (deux points, ❷)
- son tracé (pour le mode de présentation graphique, ❹) et son énoncé (mode textuel ❺), à partir de sa représentation interne,
- ...

```
class TObject3D  :            // Implémentation de l'objet géométrique
{
    string    *ObjectName;    // Nom de l'objet  ❶
    unsigned  ObjectId;       // ID unique de l'instance (dans la figure)

    // Attributs d'état de l'objet ❷
    bool      Validate,       // l'objet existe conceptuellement
              Visible,        // l'objet est visible
              Marked,         // l'objet est marqué
              IsSelected;     // l'objet est sélectionné

    int       Depth,          // Profondeur dans le graphe de dépendance
              InCalqueNb;     // Présence dans le calque n° x (xeme bit)
    FCoord    ZOrder;         // Ordre de tri pour ZBuffer

    // Graphe de dépendance ❷
    TSetObject3D  *DependList;            // Liste des successeurs
    TSetObject3D  *ParentList;            // Liste des parents

    TShape    *ObjectShape;   // forme et couleur de l'objet ❸

    // Accès aux informations textuelles de l'objet ❷
    virtual string GetObjectName();     // nom de l'objet
    virtual string GetObjectDef();      // énoncé de l'objet (Historique)
    virtual string GetObjectHelp();     // texte d'aide à la désignation
                                        // ('cette objet', 'la droite ###', ...)

    // Constructeurs et destructeur de l'objet ❷
    TObject3D();
    TObject3D(const TObject3D & );
    ~TObject3D();

    // Méthodes internes de gestion des objets et du graphe
    virtual TObject3D* CopyObject();
    virtual unsigned hashValue() const { return ObjectId; }
    virtual bool MaskObject(unsigned long mask);
    virtual bool AddInCalque(int CalcNum,bool add=1);
    virtual void HandleObjectError(int,bool);

    // opérateurs de comparaisons des objets
    bool operator == ( const TObject3D &other ) const ;
    bool operator < ( const TObject3D &other ) const ;
    virtual bool IsEqual(const TObject3D &other);       // égalité géométrique

    virtual int  CalculConceptuel() {return 0;};              ❷
    virtual int  ChangeProperties();                          ❸
    virtual void CalculVisuel(TVisualParam *){};              ❸
    virtual void Draw(TDC&,TVisualParam *vp,bool small=0){};  ❹
    virtual void DrawRetro(TDC&,TVisualParam *vp){};
    virtual bool IsInActiveArea(TPoint&) {return false;}
    [...]
};
```

Figure 6.3 – *Un extrait de la classe générique* `TObject3D`.

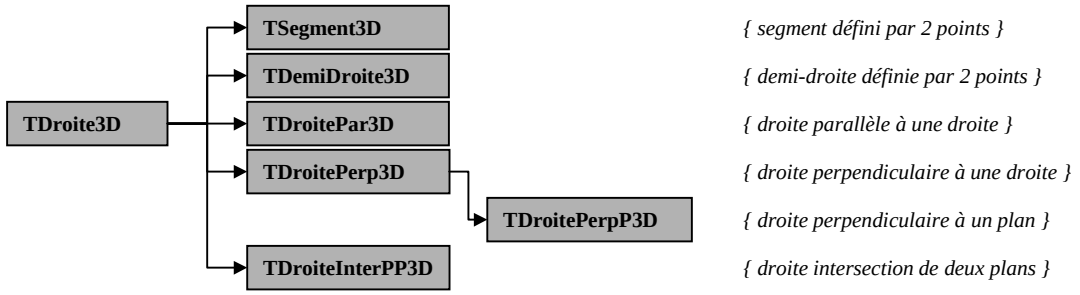


Figure 6.4 – *La classe `TDroite3D` définit les attributs et méthodes communes au type droite. Les classes héritées les spécialisent selon les différentes catégories de droites.*

Les sous-classes (`TSegment3D`, `TDroiteInterPP3D`, ...) qui héritent de `TDroite3D` représentent toutes les spécialisations possibles d'une droite (figure 6.5). La classe `TDroiteInterPP3D` par exemple définit les catégories de droites intersection de deux plans : elle spécifie la nature de la spécialisation mise en œuvre (introduction de l'intersection). En surchargeant les méthodes et attributs nécessaires, elle détermine :

- le nombre et la nature de ses parents (deux plans, ❷),
- la méthode de calcul (`CalculConceptuel()` surchargée ❸') permettant de mettre à jour la représentation analytique de la droite en fonction des parents et de la relation mise en œuvre,
- ...

Par ailleurs, toutes les classes "catégories d'objet" n'héritent pas directement de la classe "type d'objet" correspondante, comme c'est le cas par exemple pour `TDroitePerpPl3D` qui hérite de `TDroitePerp3D`, et non de `TDroite3D`. Cela s'explique par le fait que la spécialisation mise en œuvre à ce niveau ne porte que sur une partie réduite des propriétés ou des relations (uniquement au niveau des parents et du calcul de la représentation analytique dans ce cas de figure). Il est plus avantageux de faire porter l'héritage directement sur une sous-classe, de manière à minimiser les surcharges.

La hiérarchie complète des objets géométriques implantés dans ***Calques 3D*** est donnée dans la figure 6.6. Les classes `TPoint3D`, `TDroite3D`, `TPlan3D`, ..., directement héritées de la classe générique `TObject3D`, représentent les différents types d'objets (respectivement le point, la droite, le plan, ...) et définissent localement méthodes et attributs propres à chaque type. Elles ne sont pas abstraites mais correspondent aux objets géométriques élémentaires. Ainsi, la classe `TPoint3D`, en plus de définir le cadre de définition des points géométriques, représente par ailleurs *le point libre* (c'est-à-dire librement constructible et déplaçable dans l'espace). La classe `TDroite3D` implante la droite passant par deux points, la classe `TPlan3D` implante le plan passant par trois points, ...

De ces classes est dérivé l'ensemble des sous-classes représentant les catégories d'objets d'un même type. Ces catégories représentent les objets possédant certaines propriétés ou relations qui les particularisent vis-à-vis de leur type.

Les classes abstraites, c'est-à-dire les classes non instanciables sous forme d'objets géométriques dans une figure, sont indiquées en italique dans la hiérarchie. Elles ont été introduites en raisons des similitudes de traitement que possèdent leurs sous-classes. Par exemple, tous les objets du type *"point sur ..."* possèdent des caractéristiques identiques (leur appartenance à un objet), leurs différences résidant dans la nature du parent et le calcul analytique de leur re-

```
class TDroite3D : public TObject3D    // Implémentation des droites
{
    // Représentation interne unifiée ❶
    Vector4     CDirVector;           // Vecteur directeur de la droite
    Vector4     CBasePoint;           // Origine de la droite

    // Constructeur de la droite : passe par deux points
    TDroite3D(TWindow *AParent,TPoint3D *s1=0,TPoint3D *s2=0);  ❷

    TDroite3D() : TObject3D() {};
    TDroite3D(const TObject3D & );
    ~TDroite3D();

    virtual int  CalculConceptuel();                              ❸
    virtual string GetObjectDef();                                ❹
    virtual void CalculVisuel(TVisualParam *);
    virtual void Draw(TDC&,TVisualParam *vp,bool small=0);        ❺

    // Points sur droite dans les limites de l'objet
    virtual bool IsInLimit(FCoord x);

    // Méthodes internes de calcul des intersections avec la droite
    virtual int  ClippingDroite();
    Vector4*     IntersectPlan(Vector4& s1,Vector4& s2,Vector4& s3,bool lim);
    Vector4*     IntersectDroite(Vector4& s1,Vector4& s2);
    [...]
};
```

```
class TSegment3D : public TDroite3D     // Implémentation des segments
{
    TSegment3D() : TDroite3D() {};
    TSegment3D(TWindow *AParent,TPoint3D *s1,TPoint3D *s2);
    TSegment3D(const TObject3D & );

    virtual bool IsInLimit(FCoord x);
    virtual int  CalculConceptuel();
    [...]
};
```

```
class TDroiteInterPP3D : public TDroite3D   // Implémentation de la droite
                                            //     intersection de 2 plans
{
    TDroiteInterPP3D() : TDroite3D() {};
    TDroiteInterPP3D(TWindow *AParent,TPlan3D *s1,TPlan3D *s2); ❷
    TDroiteInterPP3D(const TObject3D & );                       ❸'

    virtual int  CalculConceptuel();
    [...]
} ;
```

Figure 6.5 – *Exemples de surcharge de la classe* `TDroite3D`.

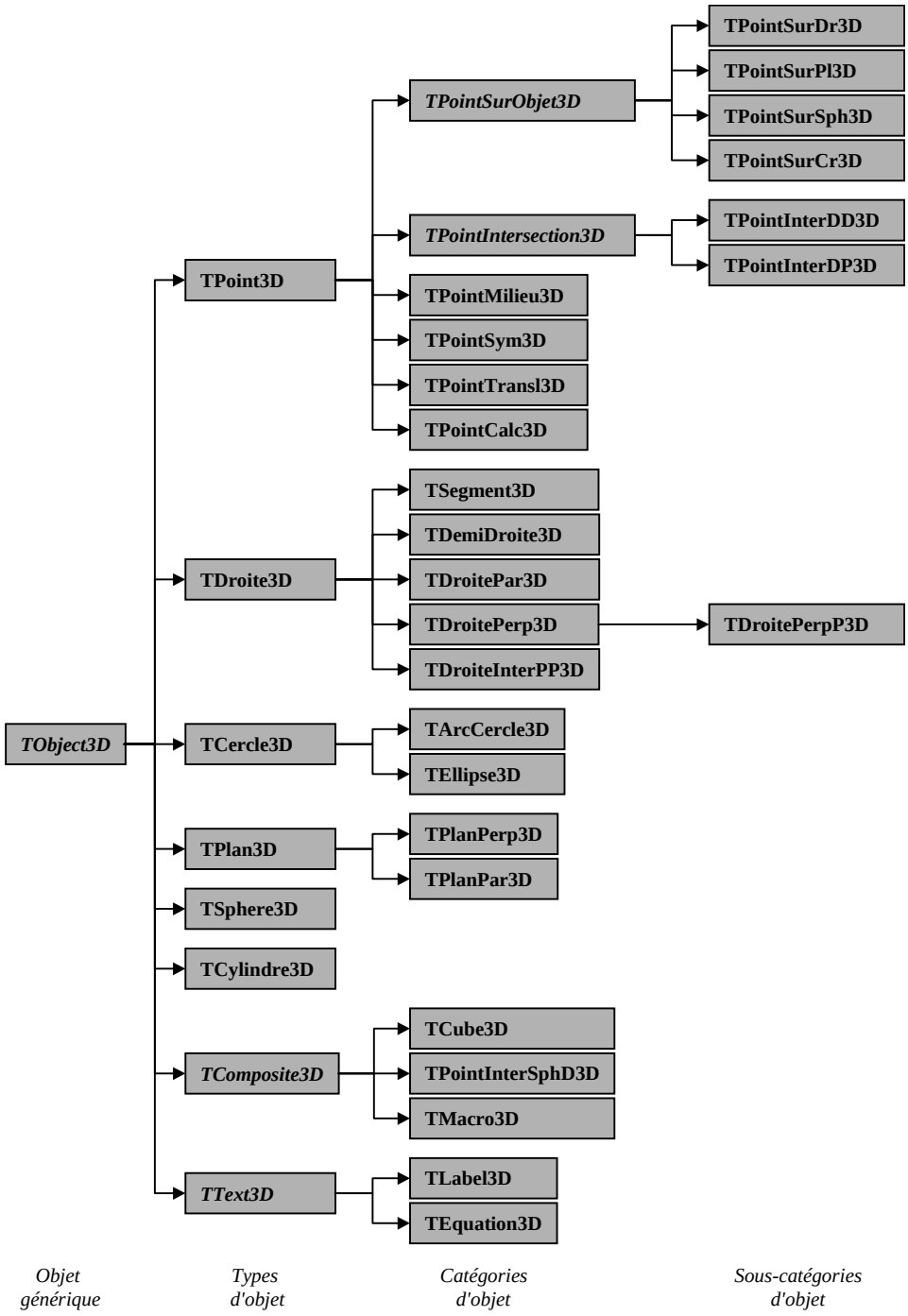


Figure 6.6 – *La hiérarchie complète des classes définissant les objets géométriques de* ***Calques 3D***

présentation interne. C'est pour faciliter la spécialisation de la relation *appartenance* que nous avons introduit la classe abstraite `TPointSurObjet3D` d'où dérivent les différents sous-catégories d'objets.

6.4.2 Le graphe de dépendance

Nous avons vu comment objets et relations géométriques sont modélisés dans ***Calques 3D***. Il reste maintenant à définir la modélisation de la figure géométrique en tant que telle, c'est-à-dire la mise en place (et la conservation) des relations entre les différents objets géométriques qui la composent. Cette modélisation se fait grâce au graphe de dépendance.

Chaque classe d'objets possède une liste d'objets parents et une liste d'objets descendants. Les objets parents (nature et nombre) sont déterminés par la catégorie de l'objet. Les objets descendants sont tous les objets de la figure qui dépendent *directement* de celui-ci. En d'autres termes, la relation de *dépendance structurelle* ainsi introduite est la réciproque de la relation de parenté : nous dirons qu'un objet B est un descendant d'un objet A (noté $A \leq_d B$) si A est un parent de B. Cette relation d'ordre partiel est identique à celle définie par Baulac dans la conception de `Cabri Géomètre` [Baulac 90]

La construction d'une figure dans ***Calques 3D***, en introduisant des relations entre différents objets, crée ce graphe de dépendance qui permet de relier chaque objet à ses parents et à ses descendants. Rappelons que la nature des relations introduites est directement inscrite dans la classe de l'objet et est exploitée par les méthodes de celle-ci.

Prenons l'exemple de la construction présentée dans la figure 6.7(a) et dont l'énoncé est le suivant :

> *Soit Pl le plan passant par les trois points A, B et C. Soient I un point appartenant à ce plan et (Dr) la droite passant par I et perpendiculaire à Pl.*
> *Soient M_1 un point appartenant à la droite (Dr) et M_2 le milieu de $[AM_1]$.*

La construction, étape par étape, de cette figure permet de définir les parents des différents objets géométriques ainsi mis en relation : les points A, B et C, étant des points libres, n'ont pas de parents ; le plan Pl a pour parents les trois points sus-nommés ; la droite (Dr) dépend à la fois du plan Pl et du point I qui, lui-même, dépend aussi du plan ; ...

Cette détermination des parents d'un objet permet de définir le graphe de dépendance de la figure 6.7(b).

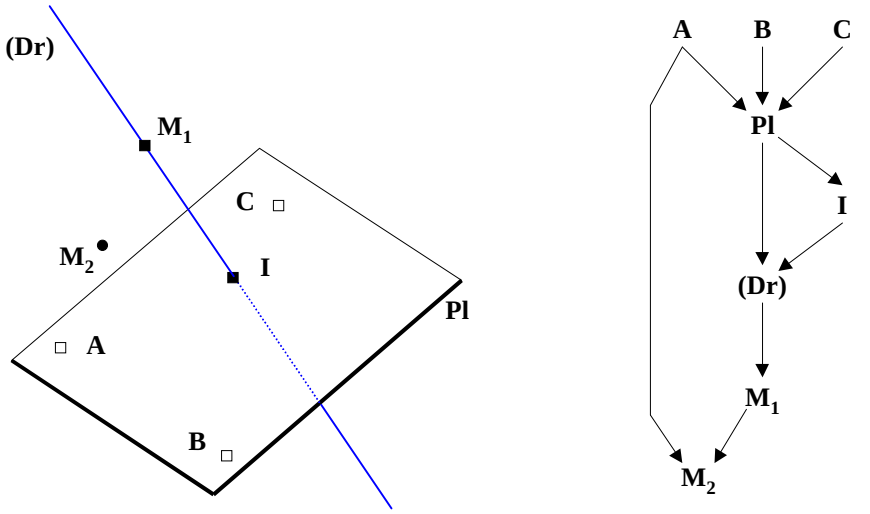
Conséquences

Le graphe de dépendance ainsi obtenu présente des particularités intéressantes.

En parallèle à la relation de dépendance, il existe une autre relation d'ordre qui intervient sur les objets géométriques d'une figure (cf. [Bernat 94b]). La relation de *précédence temporelle*, qui détermine si un objet A est construit avant un objet B et notée $A \leq_t B$, se traduit par l'ordonnancement de tous les objets de la figure dans une liste.

De part la stratégie de construction mise en œuvre dans ***Calques 3D*** (stratégie dite *action/objet*), il existe une compatibilité entre *dépendance structurelle* et *précédence temporelle* : un objet ne peut en effet être construit que si tous les objets nécessaires à sa définition (ses parents) ont été construits au préalable. Ainsi si un objet A a pour descendant l'objet B, alors l'objet A est créé avant l'objet B :

$$A \leq_d B \Rightarrow A \leq_t B$$



(a) Droite normale à un plan et passant par un point du plan.

(b) Graphe de dépendance de la figure.

Figure 6.7 – *Une figure géométrique et le graphe de dépendance correspondant.*

Cela assure en particulier le caractère non cyclique du graphe de dépendance, permettant ainsi le respect des étapes de construction et favorisant la portée, la propagation et la résolution des actions dans la figure.

Toute action effectuée sur un objet du graphe est traitée en premier lieu au niveau de cet objet. Puis, si la nature de cette action requiert sa propagation dans le graphe, un parcours du graphe de dépendance permet de déterminer le sous-graphe contenant les objets à réactualiser. Ainsi, seuls les objets ayant une relation de dépendance directe (les fils) ou indirecte (il existe un chemin entre) avec l'objet cible de l'action sont susceptibles de réagir eux-aussi à l'action, minimisant ainsi le processus de mise à jour.

Il est à noter que la détermination de ce sous-graphe ne peut se faire selon un simple parcours en largeur ou en profondeur d'abord. En effet, un objet ne doit être mis à jour que si tous ses parents ayant subit la propagation de l'action en ont préalablement traité les effets et ont été eux aussi mis à jour.

Reprenons l'exemple précédent (figure 6.7) et supposons que l'utilisateur déplace le point A. L'interprétation de l'action (déformation) nous permet de savoir qu'elle entraîne une modification structurelle de la figure par la modification de la représentation analytique du point A puis par la propagation de cette modification à tous les objets qui en dépendent, c'est-à-dire le plan Pl, la droite (Dr) et les points I, M_1 et M_2.

Dans cette configuration, un parcours en largeur fournirait un ordre de mise à jour erroné :

$$A \prec Pl \prec M_2 \prec I \prec (Dr) \prec M_1$$

Le point M_2 serait en effet traité avant le point M_1 qui, pourtant, est un des parents de M_2 et doit être traité avant lui. De même un parcours en profondeur serait susceptible de fournir un sous-graphe incorrect suivant l'ordre d'apparition des descendants d'un objet (l'ordre des

descendants du point A est-il Pl puis M_2 ou bien M_2 puis Pl?) :

$$A \prec Pl \prec M_2 \prec I \prec (Dr) \prec M_1$$
$$A \prec M_2 \prec Pl \prec (Dr) \prec I \prec M_1$$
$$\dots$$

Un parcours en profondeur sera satisfaisant à la seule condition que soit prise en compte explicitement la relation de précédence temporelle dans l'ordonnancement des descendants d'un objet. Lorsque le point A est déplacé, la réactualisation de la figure ne concerne que les objets suivants dans l'ordre chronologique.

De manière à optimiser la détermination du sous-graphe, il faut donc procéder à une *recherche topologique* du graphe de dépendance selon les deux relations d'ordre : la relation de dépendance permet d'extraire uniquement les objets concernés par la réactualisation alors que la relation de précédence permet d'en déterminer l'ordre de modification.

Dans ***Calques 3D***, nous avons implanté ce tri topologique en redéfinissant la notion de profondeur d'un objet dans le graphe de dépendance de la manière suivante :

- les points libres, n'ayant pas de parents et étant sommets du graphe, ont une profondeur égale à zéro ; dans le cas de la figure 6.7 par exemple :

$$prof(A) = prof(B) = prof(C) = 0$$

- les autres objets du graphe ont une profondeur égale à la profondeur maximale de ses parents, incrémentée de un, par exemple :

$$prof(Pl) = max(prof(A),prof(B),prof(C)) + 1 = 1$$
$$prof(Dr) = max(prof(Pl),prof(I)) + 1 = 2 + 1 = 3$$

Appliquée à l'exemple précédent, le point M_1 a une profondeur de 4 et le point M_2 une profondeur de 5 et non de 1, comme tous les descendants directs de A. Le tri topologique de ces objets nous permet d'en déterminer l'ordre de réactualisation correct, à savoir :

$$A \prec Pl \prec I \prec (Dr) \prec M_1 \prec M_2$$

Ainsi, après la mise à jour de la représentation interne du point A (et éventuellement de ses présentations à l'interface), le plan Pl est à son tour traité, puis le point I, ... Les points B et C ne sont pas concernés par cette action.

6.4.3 Un cas particulier : les objets composites

La hiérarchie des classes, représentée dans la figure 6.6, respecte une modélisation objet par spécialisation progressive. Il existe cependant quelques objets qui ne se placent pas "naturellement" dans ce type de hiérarchie. Nous en présentons deux exemples dans cette section : le cube et l'intersection d'une droite et d'une sphère.

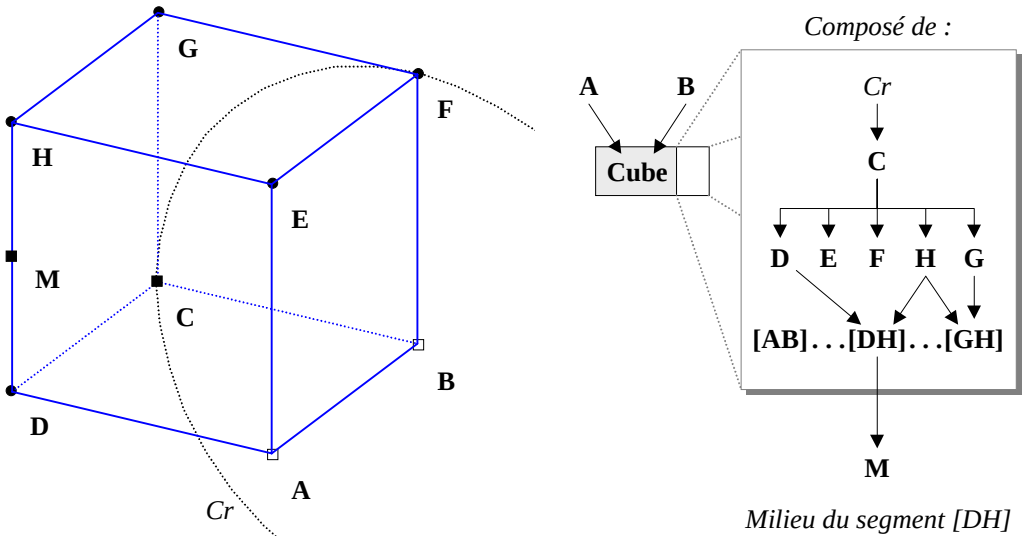
a) Le cas du cube

Selon les objectifs pédagogiques de ***Calques 3D***, nous nous sommes restreints à une modélisation de la géométrie filaire pour développer le micromonde. Cela signifie en particulier que le cube ne nous intéressait pas en tant que tel (c'est-à-dire en tant que volume) mais plutôt en tant

que *composition* d'objets : ses sommets (des points), ses arêtes (des segments) et, éventuellement, ses faces (des plans ou des polygones).

En d'autres termes, et d'un point de vue *constructiviste*, nous souhaitions avoir accès à ses éléments constitutifs pour pouvoir les utiliser comme éléments de base de construction, tout en conservant le cube comme un objet géométrique élémentaire (c'est-à-dire un objet du domaine, cf. section 4.4.2).

La solution adoptée a donc consisté à proposer un *objet composite*, c'est-à-dire à définir explicitement cette relation de composition en introduisant en interne tous les objets élémentaires qui le composent (des points d'une catégorie judicieusement choisie pour les sommets, des segments pour les arêtes), ainsi que les relations géométriques qui les relient (orthogonalité des arêtes, égalité des longueurs, ...). Cela ce traduit par la construction d'un graphe de dépendance *interne* à l'objet, que nous désignons par *graphe de composition*. Il possède exactement les mêmes propriétés que le graphe de dépendance de la figure géométrique (dépendance structurelle et précédence temporelle des objets, identification des parents et descendants d'un objet). Cependant, alors que le graphe de dépendance de la figure est construit par l'utilisateur, le graphe de composition d'un objet composite est imposé par la nature même de l'objet composite. Les objets constitutifs d'un composite conservent toutes leurs propriétés et caractéristiques : attributs visuels et attributs d'état, méthode de calcul de leur représentation interne à partir des parents, méthodes d'affichage à l'interface, ...



(a) M est le milieu de l'arête $[DH]$ du cube $(ABCDEFGH)$

(b) Graphe de dépendance de la figure

Figure 6.8 – *Un exemple d'objet composite : le cube*

Rappelons que dans ***Calques 3D***, un cube (figure 6.8(a)) se définit à partir de deux points quelconques (les points A et B de la figure) qui déterminent l'arête initiale du cube. Un troisième point C est construit comme appartenant à un cercle Cr (centré sur le point B, orthogonal à l'arête $[AB]$ et de même longueur) et permet de définir l'orientation de la première face du cube. Les autres sommets du cube (points D à H) sont des points calculés à partir de ces trois points initiaux. Les douze arêtes sont alors définies à partir de ces huit points. L'objet cube a donc

pour parents deux points et est composé de six points et de douze segments[21], structurés dans le graphe de composition interne (figure 6.8(b)). Notons que le cercle Cr est aussi un élément constitutif du cube et, à ce titre, est inclus dans le graphe de composition. Cependant, son statut n'est que visuel : il n'apparaît que lors du déplacement du point C et ne peut être utilisé comme support de construction.

La cohérence de la figure est assurée par un traitement du graphe de composition similaire à celui de dépendance. Ainsi, chaque élément constitutif d'un objet composite est totalement accessible à l'utilisateur et peut être utilisé comme parent d'un nouvelle objet dans la figure. Le graphe de dépendance se construit donc à partir de chacun des éléments constitutifs, et non de l'objet composite en tant que tel. Dans la figure 6.8(a) par exemple, le point M est construit par l'utilisateur comme étant le milieu de l'arête $[DH]$. Le graphe de dépendance de la figure tient bien compte de ce fait, en définissant l'objet M comme étant un descendant de l'objet $[DH]$, et non de l'objet *Cube*.

Cette particularité de la modélisation du cube explique pourquoi, contrairement aux autres objets volumiques comme la sphère ou le cylindre qui héritent directement de la classe générique, le type d'objet cube (la classe `TCube3D`) hérite d'une classe abstraite, dénommée `TComposite3D`, cf. figure 6.9.

Cette classe définit la structure de données et le comportement général des objets composites. Le graphe de composition interne est représenté par une liste des objets constitutifs (le champ `SubObjects`), qui seront créés (lors de l'appel du constructeur de la classe) et calculés (par la méthode `CalculConceptuel()`) automatiquement.

La classe `TCube3D` régit le traitement particulier du cube. Elle identifie les sommets (un tableau `sommet` de huit objets de type `TPoint3D`) et les arêtes (un tableau `arete` de 12 objets de catégorie `TSegment3D`) parmi les éléments constitutifs stockés dans le graphe de composition et les parents, de manière à pouvoir y accéder plus rapidement pour les calculs. Elle surcharge la méthode `CalculConceptuel()` de manière à définir le calcul des objets du graphe de composition (les six sommets restants et les arêtes) à partir des parents (les deux premiers sommets).

b) Le cas de l'intersection droite/sphère

Le résultat de l'intersection dans l'espace d'une droite et d'un volume comme la sphère ou le cylindre pose un problème. En effet, contrairement aux autres primitives de construction dont le résultat est "unique" (par exemple l'intersection de deux droites est un seul point), celle-ci donne un résultat dont la nature est *l'association de deux points*. Prenons l'exemple de la construction présentée dans la figure 6.10(a) et dont l'énoncé est le suivant :

> *Soient deux points A et B. Soient la sphère Sph de centre A et passant par le point B et la droite (Dr) passant par les deux points M_1 et M_1.*
> *Soient I_1 et I_2 les points d'intersection de la sphère Sph et de la droite (Dr).*
> *SoitI le milieu des deux points I_1 et I_2.*

La nature associative de cet objet a une importance pour les cas particuliers de la construction, c'est-à-dire lorsque la droite ne coupe pas la sphère (pas d'intersection) mais surtout lorsqu'elle est y tangente (l'intersection se résume à un seul point).

En effet, si visuellement le traitement de cette dernière configuration consiste à n'afficher qu'un seul point, l'aspect dynamique de la géométrie de ***Calques 3D*** requiert la levée de l'am-

21. La version actuelle de ***Calques 3D*** ne dispose pas d'un objet "polygone" permettant de définir les faces du cube.

```
class TComposite3D : public TObject3D       // Implémentation des objets composites
{
    typedef TISetAsVector<TObject3D> TSubObject3D;
    TSubObject3D    *SubObjects;            // Ensemble des objets constitutifs

    TComposite3D();
    TComposite3D(TWindow* AParent);
    TComposite3D(const TObject3D & );

    virtual int  CalculConceptuel();
    virtual void CalculVisuel(TVisualParam *);
    virtual void Draw(TDC&,TVisualParam *vp,bool small=0);
    virtual int IsCompositeValide(int);
    [...]
};
```

```
class TCube3D : public TComposite3D   // Implémentation du cube
{
    // Raccourcis vers les éléments constitutifs du cube
    TPoint3D    *sommet[8];
    TSegment3D  *arete[12];

    TCube3D() : TComposite3D() ;
    TCube3D(TWindow* AParent,TPoint3D* p1,TPoint3D* p2,FCoord val=0.);
    TCube3D(const TObject3D & );

    void InitCube();
    virtual int  CalculConceptuel();
    virtual int ChangeProperties();
    [...]
};
```

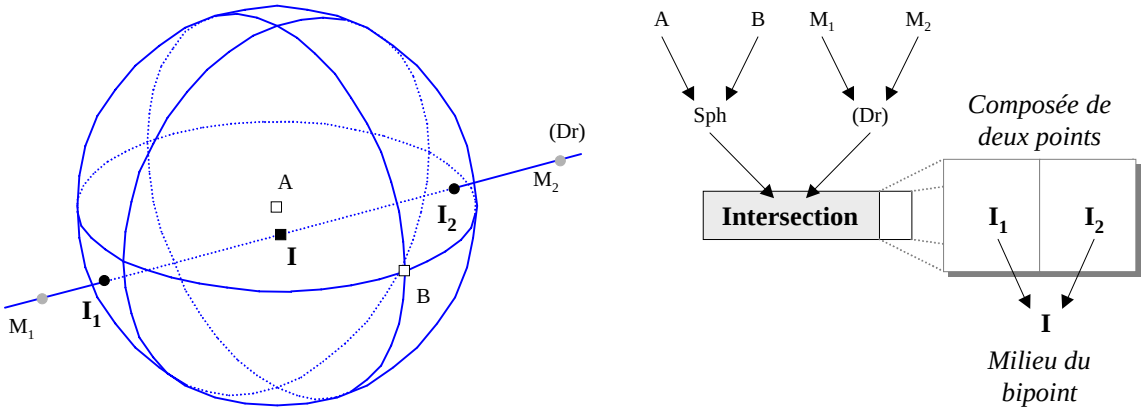
Figure 6.9 – *Un extrait des classes `TComposite3D` et `TCube3D`.*

biguïté de cette situation : *la représentation interne de l'intersection de la droite et de la sphère est-elle un seul point ou deux points de même valeur (c'est-à-dire de coordonnées identiques) ?*

La réponse à cette question est importante car d'autres objets peuvent être construits à partir de chacun des deux points intersection (comme c'est le cas pour le point I milieu de I_1 et I_2 dans l'exemple) et il convient d'assurer la cohérence de la figure, et surtout sa continuité lors du passage du cas général au cas particulier de l'intersection.

Dans ***Calques 3D***, nous avons choisi d'implanter la première solution, à savoir le fait qu'il existe deux points de même valeur. Ceci nous permet de conserver une homogénéité du comportement de l'objet (tant que l'intersection de la droite et de la sphère existe, il y a toujours deux points) et de représenter cette association d'objets par le mécanisme de composition introduit pour la gestion du cube(figure 6.10(b)). L'objet *intersection d'une droite et d'une sphère* est représenté par la classe `TInterSphDr3D` qui hérite de la classe abstraite `TComposite3D` et non de `TPointIntersection3D` comme toutes les autres intersections.

Ce choix se justifie par les objectifs pédagogiques du logiciel. En effet, l'objectif principal de ***Calques 3D*** est de permettre à l'élève de prendre conscience des relations qui existent entre des objets dans l'espace. L'accent est donc mis sur les outils qui permettent la construction de ces



(a) Les points I_1 et I_2 sont les intersections de la sphère AB et de la droite D. I est le milieu du bipoint (I_1,I_2)

(b) Graphe de dépendance de la figure

Figure 6.10 – *Un exemple d'objet composite : l'intersection d'une sphère et d'une droite*

relations, l'exploration de la figure (par déformation par exemple) et l'observation du résultat de ces opérations plutôt que sur des outils qui permettent la construction d'un objet possédant une propriété particulière. Pour l'intersection de la sphère et de la droite, cet objectif nous a amené à implanter l'*intersection des deux objets* plutôt que la construction d'*un point situé à l'intersection des deux objets*.

La cohérence de la figure dans le cas particulier de la tangence est assurée par le traitement local des cas particuliers, c'est-à-dire les préconditions associées à l'existence des objets géométriques. Dans notre exemple, lorsque la droite devient tangente à la sphère, les deux points intersection I_1 et I_2 restent bien définis mais sont confondus visuellement : leur milieu I est aussi confondu avec ces points, préservant ainsi la continuité de la figure. Par contre dans certaines constructions, les cas particuliers des relations géométriques mises en œuvre entraînent une discontinuité. Le plan médiateur du segment $[I_1I_2]$ passant par I est un exemple d'une construction discontinue : lorsque la droite est tangente à la sphère, le segment est de longueur nulle, et ne permet pas de définir ce plan médiateur.

6.5 Représentations externes des figures géométriques

Dans sa version actuelle, ***Calques 3D*** gère deux types de représentation externe d'une figure géométrique (cf. section 5.2.1) :

- une représentation textuelle, dénommée vue *historique*, et calculée à partir du graphe de dépendance de la figure,
- une représentation graphique correspondant au dessin de la figure à l'écran. Cette représentation se décompose elle-même en deux sous-catégories : une représentation intégrale (la vue *univers*) ou partielle (la vue *calque*) du graphe.

Chacune de ces vues (figure 6.11) est associée à un module qui gère la communication avec l'utilisateur (interception des événements d'interface) et l'affichage du résultat. Elles contiennent un certain nombre de paramètres locaux nécessaires à la caractérisation de l'affichage. Par

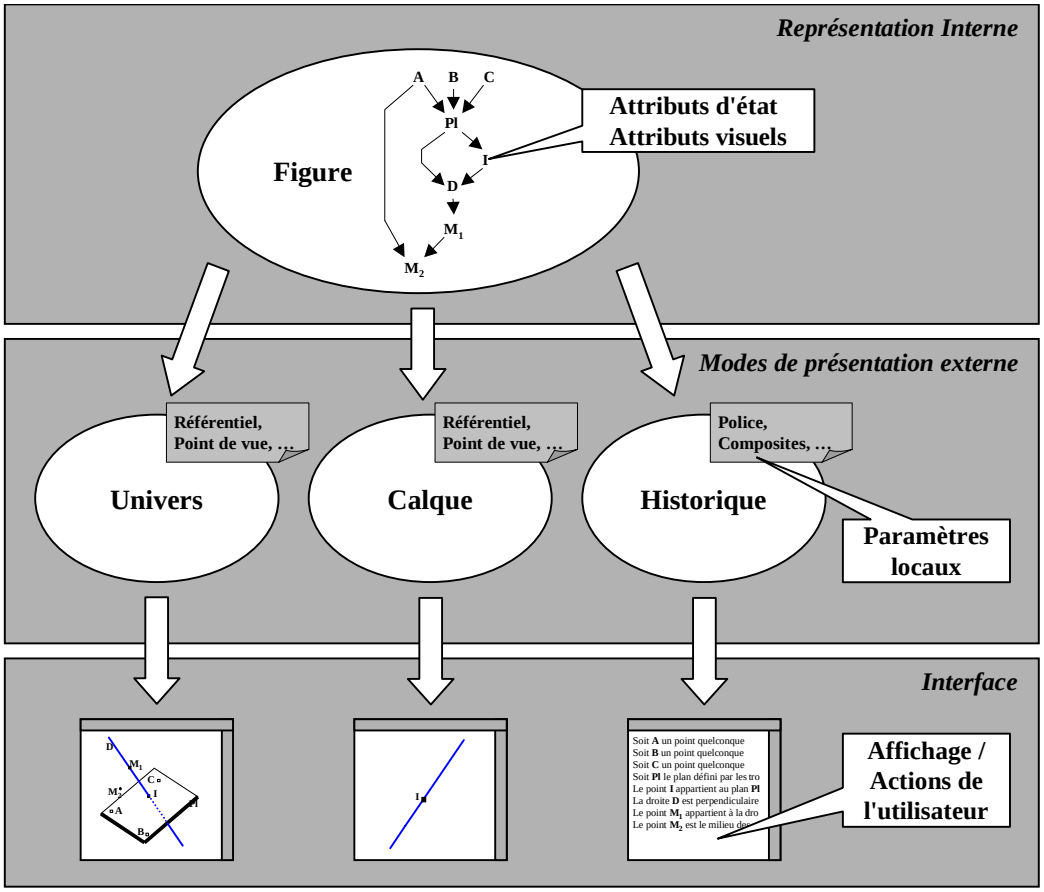


Figure 6.11 – *Représentation interne et représentation externe des figures géométriques dans* ***Calques 3D***

exemple, les vues graphiques (*Univers* et *Calque*) contiennent les paramètres nécessaires à la définition de l'espace dans lequel la figure est représentée (cf. section 5.4), à savoir :

- la représentation graphique du référentiel (plancher, repère orthonormé, ...)
- la position de l'observateur (pour la projection),
- la perspective (fuyante ou cavalière),
- l'affichage ou non des *éléments visuels de compréhension* des objets géométriques,
- ...

La vue *Historique* définit :

- la police d'affichage de l'énoncé,
- l'affichage ou non des sous-objets des objets composites,
- ...

Des *paramètres d'état* complètent la spécification des vues en permettant la définition de leur comportement (nombre de vues autorisées, autorisation de modification de la position de l'observateur, ...).

D'une manière générale, tous les paramètres d'affichage sont librement accessibles à l'utilisateur et font l'objet d'une paramétrisation possible par l'enseignant prescripteur. Les paramètres d'état de la vue sont généralement réservés à un usage interne par le logiciel et ne sont donc pas accessibles à l'utilisateur.

La figure géométrique est alors traduite à l'interface en prenant en compte les propriétés d'interface des objets (attributs visuels, attributs d'état, ...) et les paramètres de la vue correspondante. Chaque instance de vue dispose de ses propres valeurs de paramètres. Cela permet donc l'observation d'une même figure géométrique selon des réifications différentes à l'interface.

Schématiquement, l'affichage d'une figure géométrique suit toujours le même principe, à partir du parcours, objet par objet, de son graphe de dépendance :

- calcul de la représentation interne de l'objet dans l'espace (méthode `CalculConceptuel()`, cf. figure 6.3) en fonction de ses parents et de sa nature,
- mise à jour des attributs d'état (gestion des cas particuliers, présence de l'objet dans un calque, ...),
- pour les vues graphiques :
  - en fonction des attributs visuels et d'état de l'objet et des paramètres locaux de la vue, calcul de la représentation graphique de l'objet (méthode `CalculVisuel(...)`, c'est-à-dire la projection de la représentation spatiale à l'écran,
  - affichage de l'objet (méthode `Draw(...)`).
- pour les vues textuelles :
  - affichage de l'énoncé de l'objet (méthode `GetObjectDef()`) en fonction des attributs internes à l'objet et externes à la vue.

6.6 Gestion de l'interaction et de la dynamique

Dans cette section, nous présentons le mécanisme utilisé pour assurer la gestion de l'asynchronisme, caractéristique forte des micromondes, qui laisse une grande liberté d'actions à l'utilisateur. Ce mécanisme est basé sur une décomposition des *activités* offertes à l'utilisateur en *tâches* élémentaires, chaque tâche étant constituée d'une succession d'*étapes*.

6.6.1 Engagement direct

La modélisation par niveau est particulièrement intéressante pour spécifier des actions de l'utilisateur. Les actions physiques se situent en effet au niveau de l'interface et doivent permettre la manipulation d'objets du domaine d'apprentissage à travers des fonctions d'engagement direct.

Les actions physiques autorisées dans ***Calques 3D*** sont en nombre relativement limité. En plus du clavier, nous disposons d'une souris à un, deux ou trois boutons. Les actions possibles sont par exemple *clic*, *double-clic*, *drag* (déplacement de la souris avec le bouton gauche enfoncé). Ces actions peuvent être combinées avec des touches spécifiques du clavier : *Shift-clic*, *Control-clic*, ...

La sémantique de ces actions n'est pas universellement connue, même si des tentatives de normalisation apparaissent. Elle réside au niveau présentation du modèle par l'introduction de fonctions qui peuvent avoir pour nom *désigner*, *informer*, *déplacer*, ... Le lien entre actions syntaxiques au niveau interface et fonctions sémantiques au niveau présentation est défini par le développeur avec, comme objectif, le maintien de la cohérence de l'interface.

Cette cohérence est assurée si des actions de même sémantique au niveau présentation correspondent à la même fonction au niveau de l'interface. Ainsi toutes les fonctions *désigner un objet* peuvent correspondre à un *clic* sur un objet, que cette action porte sur un point, une droite ou un plan.

Pour différencier les actions, nous avons complété la notion de *tâche* telle qu'introduite dans [Bernat 94b].

Lors de l'utilisation de ***Calques 3D***, l'élève conduit des *activités* (cf. chapitre 4). Ces activités se caractérisent par un objectif pédagogique et par la mise à disposition de l'élève d'un ensemble d'outils permettant la réalisation de *tâches* particulières, elles-mêmes pouvant se décomposer en une succession d'*étapes* (ou tâches élémentaires) (figure 6.12).

Pour l'utilisateur, élève ou enseignant, les notions d'activité, de tâche et d'étape sont intuitives. Les activités n'ont pas d'autre matérialisation dans le logiciel que celle que l'enseignant prescripteur introduit par lui-même en préparant et donnant un exercice à ses élèves (voir par exemple les activités des séquences pédagogiques, section 4.3.3).

Pour l'élève, chaque tâche correspond à une fonctionnalité du logiciel, mise à sa disposition par l'enseignant. Elle est associée à une commande, accessible par la barre de menus ou par la barre d'icônes de l'espace de travail (cf. section 5.2.1). Le choix des tâches à exécuter, l'ordre de leur utilisation, ... sont laissés à la totale initiative de l'élève, dans les limites imposées par l'enseignants.

Les étapes nécessaires à la réalisation d'une tâche donnée sont prédéfinies dans le logiciel par les contraintes d'implantation (principalement des contraintes ergonomiques). Chaque étape permet de décomposer une tâche potentiellement complexe en unité simple où les actions physique de l'utilisateur seront interprétées et associées aux fonctions sémantiques correspondantes. Les rétroactions visuelles et l'aide contextuelle lui servent de fil conducteur et le soutiennent dans la réalisation, étape par étape, de la tâche.

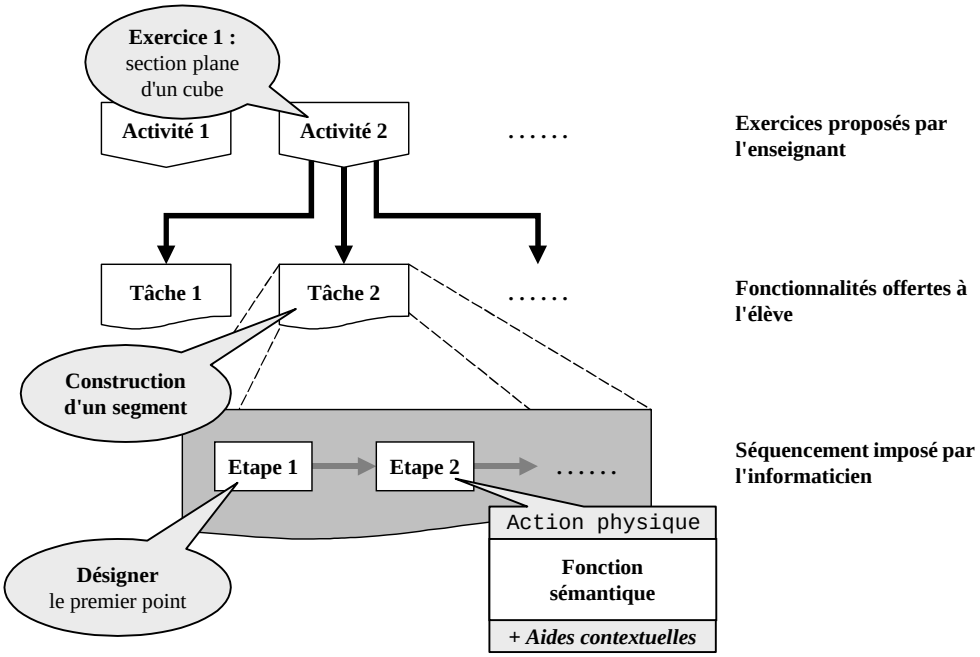


Figure 6.12 – *Activités, tâches et étapes*

6.6.2 Activités, tâches et étapes

Nous restreignons dans cette partie le terme de *tâche* aux seules fonctionnalités propres à ***Calques 3D***, permettant à l'utilisateur de mettre en œuvre des activités de construction

(création d'un point, d'une droite ; construction de l'intersection de deux objets ; ...), de visualisation (modifier la perspective ; changer les attributs visuels des objets ; ...) et/ou d'exploration (déformation d'une figure ; extraction dans un calque ; ...).

L'interface est modélisée par un ensemble de tâches. Une tâche définit l'ensemble des actions accessibles à l'utilisateur, les rétroactions disponibles en réponse aux actions et les aides que le logiciel peut apporter à l'utilisateur pour faciliter la compréhension de la tâche et son accomplissement. Nous appelons *tâche active* la tâche disponible à un instant donné. L'enchaînement des tâches au cours d'une activité d'apprentissage est totalement libre et laissé à la charge de l'utilisateur. Cela requiert donc une gestion de l'asynchronisme entre les différentes tâches d'une activité.

Chaque tâche est décomposée en un ensemble d'étapes élémentaires dont l'ordonnancement dans la tâche est défini par le développeur et représenté par un arbre ET/OU [22]. Nous appelons *étape courante* l'étape disponible à l'utilisateur à un instant donné : elle représente l'état courant de la résolution de la tâche et permet de caractériser ce qui a été fait et ce qui reste à faire pour achever la commande. Chaque étape définit la réponse aux différentes actions de l'utilisateur en y associant une *fonction* particulière.

Par exemple, la figure 6.13(a) représente le graphe d'enchaînement de la tâche *"Construction d'un segment"*, caractérisé par la succession des étapes suivantes :

- *désignation d'un point* correspondant à la première extrémité du segment,
- *désignation d'un point* correspondant à la deuxième extrémité du segment.

Chacune des deux étapes de cette tâche implique la même action physique de l'utilisateur à l'interface (*clic avec le bouton gauche de la souris*) qui se traduit par la même fonction (*Désigner*) et le même type d'argument (*un point quelconque*). La décomposition de la tâche en deux étapes apparemment identiques est nécessaire pour permettre l'attribution d'une sémantique différente au résultat des deux fonctions : l'identification de chacune des extrémités du segment à construire.

La tâche *"Construction du milieu"* (figure 6.13(b)) se caractérise par une alternative dans la succession des étapes, selon que l'on choisisse de construire :

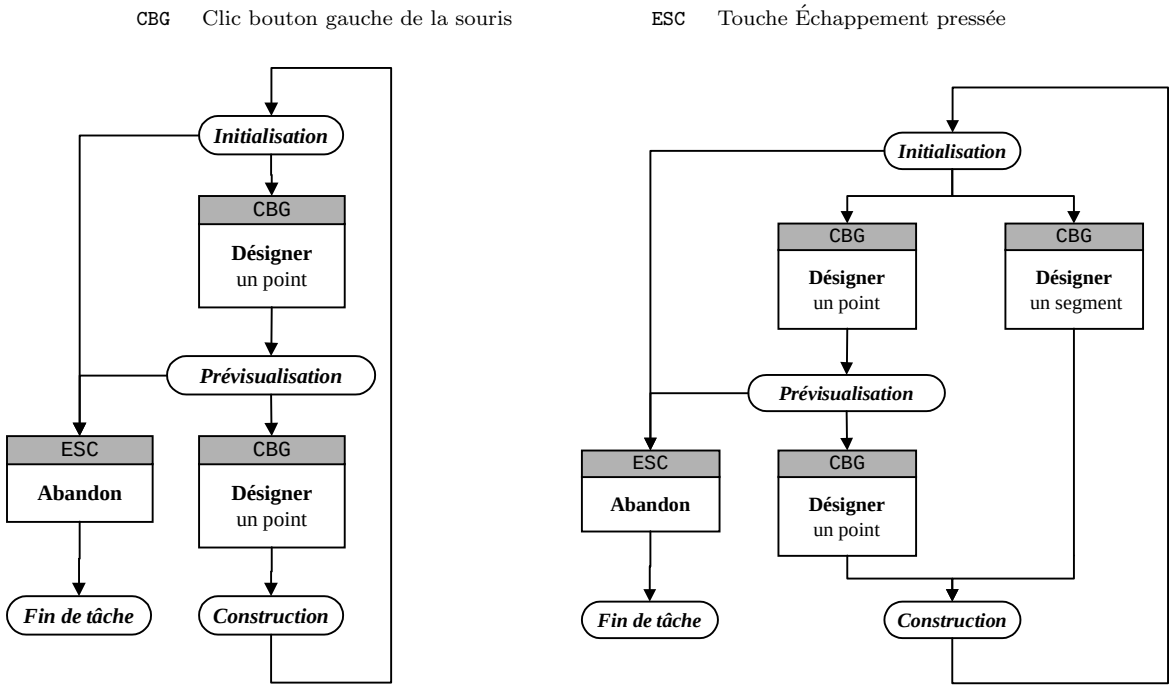
- soit le milieu d'un bipoint (en deux étapes : *désignation d'un point* correspondant au premier point du couple puis *désignation d'un point* correspondant au deuxième point du couple),
- soit d'un segment (en une étape, *désignation d'un segment*).

Ainsi, l'interprétation des actions au niveau présentation diffère selon l'étape courante. Un clic avec le bouton gauche de la souris, toujours associé à la fonction *Désigner*, pourra donc servir à désigner soit un point soit un segment selon la nature de l'étape courante.

La figure 6.14 donne un dernier exemple, plus complexe, de graphe d'enchaînement. Elle définit les différentes étapes de la tâche *"Déformation d'une figure"* et leur séquencement.

La réalisation de cette tâche (cf. section 5.5.1) consiste à *désigner* un point libre (*clic bouton gauche*), puis à le *déplacer*, soit dans un plan (*déplacement de la souris*), soit selon la normale à ce plan (*déplacement de la souris touche SHIFT pressée*). Ces deux actions de déplacement sont répétitives et peuvent être alternées jusqu'à ce que l'utilisateur *désigne* une position dans l'espace (*clic bouton gauche*). Cliquer avec le bouton droit de la souris permet d'obtenir un menu contextuel afin de *modifier les paramètres* de la déformation, à savoir le choix de la décomposition du déplacement selon le plan horizontal, frontal ou gauche.

22. Même si, dans la plupart des cas, nous avons essayé de limiter cette représentation à une simple liste chronologique, de manière à faciliter la compréhension de la tâche à réaliser par l'utilisateur et donc son appropriation.



(a) Graphe de la tâche *"Construction d'un segment"*

(b) Graphe de la tâche *"Construction du milieu"*

Figure 6.13 – *Deux exemples de graphe d'enchaînement des tâches*

a) Tâches, aides et rétroactions visuelles

La décomposition des tâches en étapes successives permet d'y associer, outre les réponses aux actions de l'utilisateur, des éléments d'aide adaptés. Dans ***Calques 3D***, ces éléments sont de plusieurs natures (cf. section 5.6).

Une aide contextuelle est fournie. Elle est contextuelle à la tâche ou, plus précisément, à l'étape courante sous la forme d'un texte (par exemple *"Construction d'un segment : désignez la première extrémité"*). Ce texte permet d'indiquer de manière succincte mais immédiate l'état courant de la tâche.

Les fonctions de *désignation* requièrent l'identification d'arguments, c'est-à-dire la désignation d'un objet géométrique d'un type bien précis. La décomposition en étapes permet localement de réduire le sous-ensemble des objets cibles et d'en avertir l'utilisateur : seuls les objets du type correspondant à l'étape sont désignables, signalés par une *étiquette de désignation* qui apparaît lors du passage du curseur à proximité de ceux-ci (cf. figure 5.5(a) du chapitre précédent, page 75).

Fortement liés à l'approche visuelle de l'apprentissage de la géométrie, les éléments d'aide les plus importants dans ***Calques 3D*** restent les *rétroactions visuelles*. En effet, un des principes de la manipulation directe est la réponse immédiate à toute action de l'utilisateur. La forme de cette rétroaction renforcera plus ou moins l'impression d'engagement direct. Sous une forme visuelle, elle peut également permettre à l'utilisateur de prévoir l'aspect du résultat final de l'action en cours.

Dans le cas de la tâche *"Construction d'un segment"* par exemple, une *prévisualisation* du

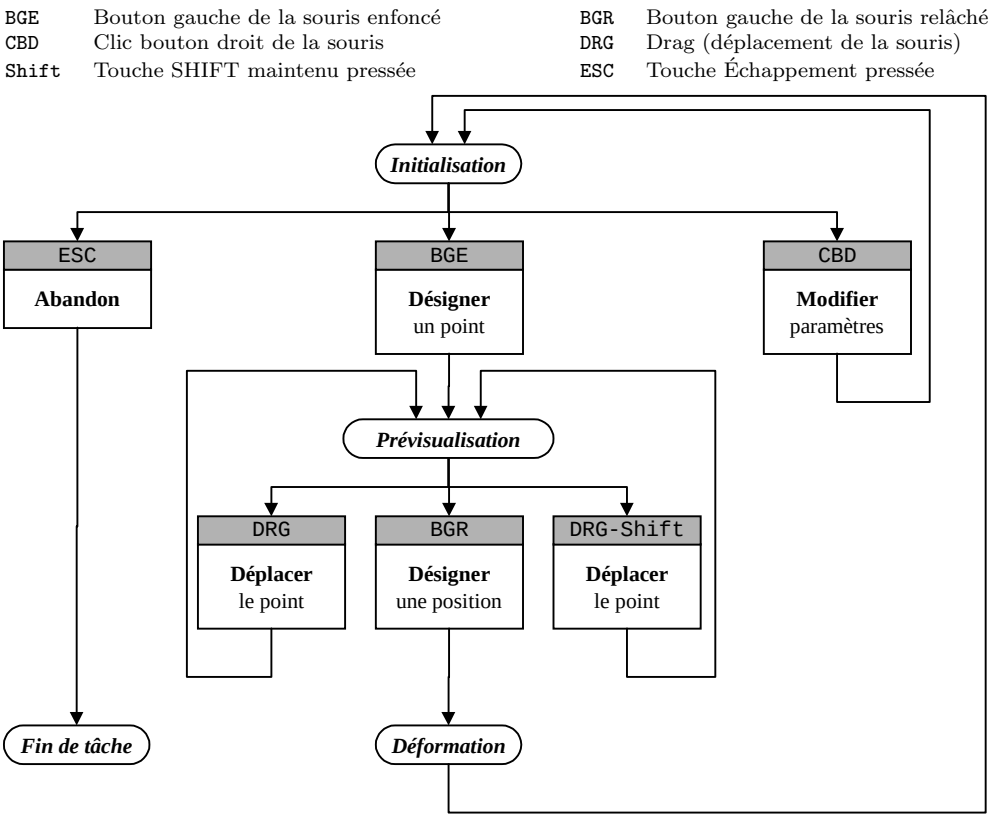


Figure 6.14 – *Graphe d'enchaînement de la tâche* "Déformation d'une figure"

segment, sous l'aspect d'une forme élastique de longueur variable, est affichée entre le premier point désigné et la position du curseur.

Le cas de la tâche *"Déformation d'une figure"* (figure 6.14) permet de mettre à nouveau en évidence la nécessité de séparer les différents niveaux de modélisation des objets du domaine. En effet, les rétroactions visuelles appartiennent au niveau *présentation* du modèle en quatre couches et, à ce titre, n'interviennent que sur la vue où a lieu l'action. Ainsi, lors du déplacement du point libre désigné, la prévisualisation consiste en un réaffichage *local* de la figure déformée. Ce n'est qu'une fois la fin du déplacement signalé que le logiciel procédera à la *déformation* proprement dite en répercutant les modifications dans toutes les vues, ce qui permet un gain de temps et de calcul pour l'affichage[23].

b) Asynchronisme et tâche par défaut

Comme nous l'avons signalé, la réalisation d'une activité par l'élève nécessite l'exécution d'un certain nombre de tâches. De par l'essence même d'un micromonde, leur nature, leur nombre, leur ordonnancement au sein de l'activité ne peuvent pas être prévus ni planifiés à l'avance, comme cela peut être le cas dans un environnement de type tutoriel.

Ce problème est d'autant plus crucial que, d'un point de vue interface utilisateur, ***Calques 3D*** propose un environnement multi-fenêtres : des activités de nature différente, et

23. Notons cependant qu'une option de ***Calques 3D*** permet de synchroniser, au cours du déplacement du point libre, la mise à jour de la figure dans toutes les vues ouvertes. Dans ce cas, la *prévisualisation* est remplacée directement par la *déformation*, résultat de la tâche.

donc des tâches différentes, peuvent être menées en 'parallèle' dans plusieurs fenêtres ouvertes simultanément.

De manière à permettre une gestion cohérente de l'asynchronisme et de l'ordonnancement des différentes tâches, il est nécessaire de restreindre l'espace de liberté qu'offre à l'utilisateur un environnement du type micromonde ou, du moins, d'en 'brider' les possibilités. Pour ce faire, nous avons conservé un certain nombre des choix d'implantation de `Calques 2` [Bernat 94a].

Les tâches sont ainsi rattachées directement aux fenêtres de l'interface, c'est-à-dire au niveau des différentes vues de l'application. Elles jouent ainsi le rôle d'une couche de 'traduction' entre les actions physiques de l'utilisateur à l'interface et la manipulation des objets du domaine. Cela explique qu'il ne peut y avoir qu'UNE SEULE tâche active à un instant donné et, surtout, qu'il y en a TOUJOURS une d'active.

Ce principe de permanence et d'unicité des tâches rend nécessaire l'introduction d'une *tâche par défaut*, point d'entrée des fonctionnalités du logiciel (cf. section 5.2.2). Elle offre à l'utilisateur la possibilité de réaliser la ou les actions les plus importantes aux vues des objectifs pédagogiques du logiciel. Dans `Calques 2` par exemple, elle offre la possibilité de déformer la figure géométrique et d'en exploiter les propriétés. Dans ***Calques 3D*** par contre, l'observation est plus importante que la déformation, la tâche par défaut ne met en œuvre que des fonctions relevant de cet objectif (modification de point de vue, changement du référentiel, ...).

c) Échappement et réversibilité

L'introduction des *tâches* pour la gestion des actions de l'utilisateur a une conséquence directe sur les choix des modes d'interaction dans ***Calques 3D***, et s'avère en particulier parfaitement adaptée à la stratégie dite *action/objet* : on choisit d'abord un type d'action (c'est-à-dire une tâche), on a alors accès à une aide contextuelle et les seuls objets accessibles à l'utilisateur sont ceux qui correspondent à l'action (c'est-à-dire à l'étape courante).

Cependant, l'inconvénient d'une telle stratégie est que le choix préalable d'une tâche à accomplir peut être assimilé à une sorte de dialogue modal entre le système et l'utilisateur puisque ce dernier, avant de pouvoir effectuer une autre tâche, doit achever la tâche en cours ou disposer d'un moyen d'en annuler l'exécution.

L'échappement est une solution pour mettre fin à ce dialogue. Sa mise en œuvre peut être explicite (par l'utilisation d'une commande dédiée, généralement la touche ESC) ou implicite (par l'activation d'une autre tâche) et peut a priori revêtir différentes formes :

1. *Annuler une étape* et remonter dans la hiérarchie des étapes de la tâche. Par exemple, dans la tâche *"Construction d'un segment"*, l'échappement de l'étape *désignation du deuxième point* ramène à l'étape *désignation du premier point* qui correspond à l'étape initiale de la tâche.
2. *Réinitialiser la tâche* et retourner à l'étape initiale de la tâche. Par exemple, dans la tâche *"Construction d'un plan"*, l'échappement de l'étape *désignation du troisième point* ramène à l'étape initiale, *désignation du premier point*.
3. *Abandonner la tâche active* et retourner à la *tâche par défaut*.

C'est ce dernier choix qui a été choisi pour le développement de `Calques 2` ([Bernat 94a], [Bernat 94b]) et conservé pour celui de ***Calques 3D***. Sa simplicité garantit le résultat et réduit l'activité cognitive de l'utilisateur relative à la compréhension de l'action.

De manière analogue, un micromonde comme ***Calques 3D*** doit permettre à l'usager une exploration libre afin d'en tester les possibilités. L'utilisateur doit avoir, à tout instant, la pos-

sibilité de faire repasser le système à l'état précédant au moins la dernière interaction. Cette propriété de *réversibilité* des commandes doit être directement accessible, sans exception et par une commande invariable dans sa forme[24].

Dans ***Calques 3D***, l'annulation du résultat d'une tâche n'est disponible qu'une fois celle-ci achevée, c'est-à-dire une fois de retour dans la tâche par défaut. De plus, on ne permet que l'annulation de la dernière tâche réalisée. Ces restrictions ont été introduites de manière à préserver une cohérence dans l'organisation des tâches au sein de l'application : éviter par exemple que l'utilisateur puisse annuler la construction d'objets requis par la tâche active alors que leur existence est une condition d'activation de la tâche. Par exemple, pour que la tâche de construction d'une droite soit activable, il faut que l'univers contienne au moins deux points. Si l'utilisateur construit ces deux points, active la tâche *"Construction d'une droite"* et annule la construction du dernier point, le système se retrouve dans un état incohérent où la droite ne peut plus être construite mais dont la tâche reste active.

d) Implantation des tâches

Comme nous l'avons signalé, les tâches sont rattachées au niveau de chaque fenêtre de l'application. Les fenêtres sont des objets dont l'un des champs pointe sur la tâche active. Le choix d'une tâche par l'utilisateur (typiquement par l'activation d'une commande dans un menu ou une barre d'icônes) instancie ce champ par la tâche courante.

Les tâches sont implantées par une hiérarchie de classes dont les méthodes mettent en œuvre l'interception des actions physiques et leur association aux fonctions sémantiques, et dont les champs permettent la conservation des informations nécessaires à la réalisation de la tâches (étape courante, listes des objets désignés par l'utilisateur dans les étapes précédentes, ...).

Le sommet de cette hiérarchie est la classe `TDefTache` qui définit le cadre des méthodes et champs de toutes les classes *tâches*. Cette classe sert aussi à implanter la *tâche par défaut*, mettant ainsi en évidence la position centrale qu'elle occupe dans les fonctionnalités du logiciel.

La figure 6.15 montre un extrait du profil de la tâche par défaut (classe `TDefTache`), ainsi que celui des tâches *"Construction d'une droite"* (`TDroite3DTache`) et *"Construction d'un segment"* (`TSegment3DTache`).

L'héritage permet de gérer de manière optimale la spécialisation ou la surcharge des informations et méthodes nécessaires. Ainsi la tâche `TSegment3DTache` ne se différencie de la tâche `TDroite3DTache` que par des rétroactions visuelles différentes (c'est-à-dire la forme élastique fournie à l'utilisateur comme prévisualisation de la construction : ébauche d'un segment ou d'une droite), par l'aide contextuelle associée aux étapes de ces tâches et, bien entendu, par la nature de l'objet construit par cette tâche. Il est donc naturel de faire hériter `TSegment3DTache` de `TDroite3DTache` et d'en surcharger les méthodes requises.

Les actions physiques de l'utilisateur (clic bouton gauche, déplacer le curseur, ...), effectuées au niveau de la fenêtre, sont interceptées par les méthodes correspondantes de la tâche (`Exec_Mouse_L(...)`, `Exec_Mouse_mv(...)`, ...) et associées aux fonctions correspondantes (`FindObject(...)`, `CreateObject3D()`, ...), suivant la nature de l'action et de la tâche.

Au cours des différentes étapes de la tâche active, les messages de l'aide contextuelle (`GetContextualHelp(...)`) et les rétroactions visuelles (`DrawFeedBack(...)`), sont déterminés et affichés.

24. J. Nanard ajoute qu'en plus cette commande doit être unique [Nanard 90]

```
class TDefTache             // Implémentation de la tâche par défaut
{
    int     TaskStep;       // étape de la tâche en cours
    TDefTache(TWindowView *AParent,uint taskID);
    virtual void CancelTache();

    int  FindObject(TPoint& loc,uint mask,bool ShowLabel=true);
    TObject3D* RetriveObject(int index);

    // réaction aux actions physiques de l'utilisateur
    virtual void Exec_Mouse_L(uint mod, TPoint& loc);
    virtual void Exec_Mouse_R(uint mod, TPoint& loc);
    virtual void Exec_Mouse_mv(uint mod, TPoint& loc);
    virtual void Exec_Mouse_LUp(uint mod, TPoint& loc);
    virtual void Exec_Keydown(uint mod, TPoint& loc);

    virtual void CreateObject3D() {};

    // rétroactions visuelles et aides contextuelles
    virtual bool SetTaskCursor(TResId &curID);
    virtual string GetContextualHelp();
    virtual void CalculFeedBack();
    virtual void DrawFeedBack(TDC&);
    [...]
};
```

```
class TDroite3DTache : public TDefTache     //Tâche 'construction d'une droite'
{
    TPoint3D *ptA,*ptB;     // les deux points à désigner

    TDroite3DTache(TWindowView *AParent,uint taskID);
    virtual void CancelTache();

    virtual void Exec_Mouse_L(uint, TPoint&);
    virtual void Exec_Mouse_R(uint, TPoint&);
    virtual void Exec_Mouse_mv(uint, TPoint&);
    virtual void CreateObject3D();
    virtual void DrawFeedBack(TDC&);
    virtual string GetContextualHelp();
};
```

```
class TSegment3DTache : public TDroite3DTache   // Tâche 'construction d'un segment'
{
    TSegment3DTache(TWindowView *AParent,uint taskID);
    virtual void DrawFeedBack(TDC&);
    virtual void CreateObject3D();
    virtual string GetContextualHelp();
};
```

Figure 6.15 – *Implantation des tâches*

6.6.3 Portée des actions et répercussion

La figure 6.16 schématise la problématique de la portée des actions et la répercussion de leurs effets sur deux cas de figure : la modification du point de vue (faire tourner la figure) et la déformation d'une figure (déplacer un point libre).

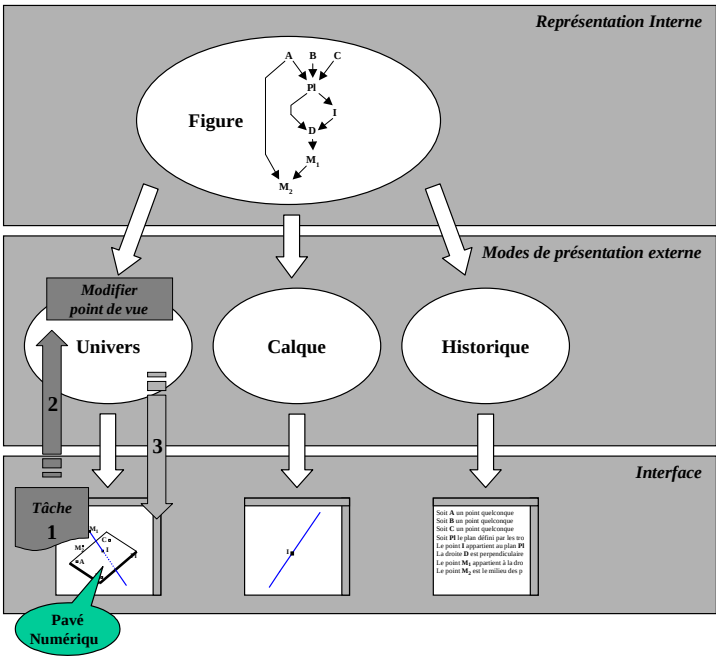
Dans le premier cas (figure 6.16(a)), l'utilisateur presse une touche du pavé numérique dans la vue *Univers*. Cette action physique au niveau de l'interface (❶) est interceptée par la *tâche active* qui l'interprète, au niveau présentation, comme une demande de modification du point de vue de l'observateur (❷). Cette fonction sémantique est effectuée à ce niveau (c'est-à-dire par une modification des paramètres locaux définissant la vue) mais ne nécessite pas une remontée de l'information aux niveaux supérieurs (représentation interne et domaine). Une fois la modification opérée au niveau présentation, elle est répercutée (❸) aux niveaux inférieurs (c'est-à-dire le niveau interface dans cette situation) et entraîne une mise à jour de l'affichage de la figure en fonction de la nouvelle position de l'observateur. Les autres vues ne sont pas concernées par cette modification, l'action n'ayant pas a être répercutée au niveau de la représentation interne car elle ne concerne qu'un affichage partiel.

Dans le deuxième cas (figure 6.16(b)), l'utilisateur ouvre une vue *calque*, active la tâche *"Déformation d'une figure"* et clique sur un point puis déplace le curseur. Là aussi, cette séquence d'action physique est interceptée (❶) par la tâche active (ou, plus précisément, par chacune des étapes de celle-ci, cf. figure 6.14). L'action *clic* est assimilée à la fonction *Désigner* du niveau présentation et l'action *Drag* à la fonction *Déplacer* (❷). La fonction *Déplacer* implique une modification structurelle de la figure : le message est propagé jusqu'au niveau *Représentation interne* (❸) où il a pour conséquence la détermination du sous-graphe concerné puis sa réactualisation (c'est-à-dire la mise à jour des représentations analytiques internes de chaque objet du sous-graphe). Une fois cette opération effectuée, le résultat de l'action est répercuté dans les niveaux inférieurs (❹). Au niveau présentation, il est propagé dans chacune des trois vues *Univers*, *Calque* et *Historique* (si elles existent à l'interface), entraînant une mise à jour des modes de présentation de la figure (calcul de la projection de la figure pour les vues graphiques, mise à jour de la présentation textuelle pour la vue *Historique*). Pour chacune des vues, le message atteint alors le niveau interface pour une mise à jour de l'affichage des présentations. Dans ce dernier cas de figure, toutes les vues sont concernées par l'action et la répercussion de son résultat.

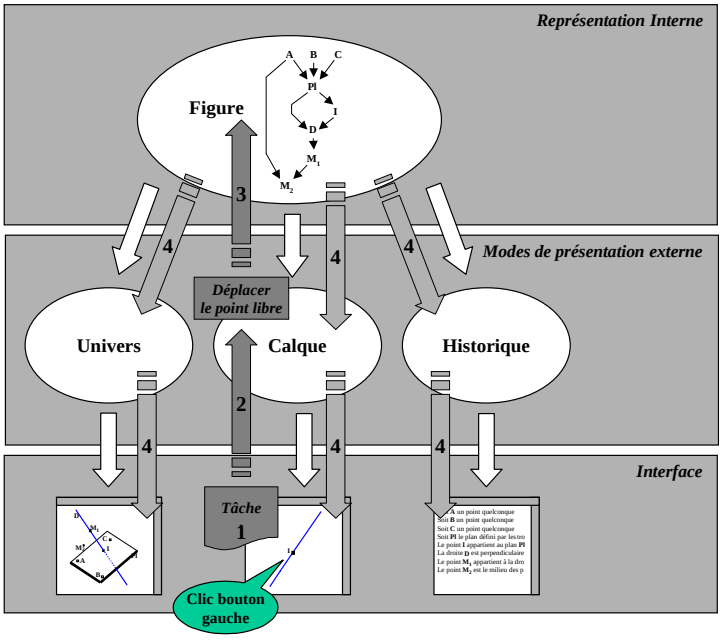
En d'autres termes, toute action ayant pour résultat une modification de la *représentation interne* d'une figure (déformation d'un point libre de la figure, suppression d'un objet, ...) sera répercutées dans toutes les vues alors qu'une action n'ayant pour résultat que la modification locale de l'affichage de la figure, soit directement (changement d'un attribut visuel) ou indirectement (modification du point de vue de l'observateur) n'aura de répercussion que dans la vue où l'action a eu lieu.

6.7 Réutilisation et généralisation à la création de figures complexes

L'évolution des logiciels de construction géométrique a principalement privilégié l'interface utilisateur de manière à permettre une prise en main plus facile et donc plus efficace pour l'apprentissage. Des élèves de tous niveaux peuvent maintenant réussir une construction grâce à l'utilisation de commandes visuelles (intuitives et simples d'usage), ce qui n'était pas le cas avec



(a) Portée de l'action *"Modifier le point de vue"*



(b) Portée de l'action *"Déformer la figure"*

Figure 6.16 – *Portée et répercussion des actions*

un environnement de "programmation" comme `Euclide` [Artigue 89] par exemple. Cependant, si les logiciels y gagnent en convivialité, ils y perdent en efficacité sur certains points.

`Euclide` est une extension de `Logo` dont il conserve toutes les fonctionnalités. En particulier, il permet d'écrire des procédures, c'est-à-dire de réutiliser certaines constructions réalisées préalablement. Or la réutilisation de constructions est un concept qui permet d'atteindre deux objectifs importants pour les micromondes : l'*adaptabilité* et l'*utilisabilité* (cf. chapitre 1).

La réutilisation de constructions permet en effet de faire évoluer un logiciel en parallèle avec les connaissances de l'utilisateur : c'est l'*adaptabilité* au niveau des compétences de l'utilisateur. Elle autorise aussi la définition de nouvelles commandes pour mémoriser et créer de nouvelles catégories de figures (raccourcis de construction).

De même, la construction de figures complexes, réalisée à l'aide des commandes de base du logiciel, peut devenir rapidement répétitive et fastidieuse. Or, la résolution d'un problème en géométrie passe souvent par la construction d'une figure ; si celle-ci requiert des étapes intermédiaires trop répétitives, l'utilisateur risque de perdre le fil de sa pensée. L'utilisation de "procédures" prédéfinies permet de traiter un problème selon une approche systémique, fréquemment rencontrée dans l'enseignement technique, et peut réduire sensiblement les étapes intermédiaires de la construction. C'est une condition importante pour que le logiciel soit utilisé : c'est le critère d'*utilisabilité*.

Prenons l'exemple de la pyramide à base carrée. ***Calques 3D*** ne dispose pas, dans ses objets élémentaires, d'un tel objet volumique. Cela est en partie lié au statut de prototype du logiciel, en partie volontaire dans le sens où la construction d'un tel objet, à partir des primitives existantes, n'est pas impossible. Cette construction est cependant relativement lourde car elle nécessite un certain nombre d'objets élémentaires (une sphère, deux plans, deux droites, ...). Un énoncé de construction d'une telle pyramide (figure 6.17) pourrait être le suivant [25] :

> *Soient A et C deux points quelconques, i le milieu du segment $[AC]$.*
> *Soit Sph la sphère de centre i et passant par A.*
> *Soit Pl_1 le plan passant par i et perpendiculaire au segment $[AC]$ et j un point appartenant à ce plan.*
> *Soient B et D les intersections de la sphère S et de la droite (ij).*
> *Soit (Nle) la droite normale au plan (ABC) et passant par i. Soit M un point de (Nle).*
> *$ABCDM$ est une pyramide à base carrée $ABCD$ et de sommet M.*

Si une activité requiert une telle pyramide comme support de construction, il peut être alors satisfaisant de pouvoir sauvegarder la figure, réalisée par l'enseignant, puis de la donner comme document à l'apprenant. Mais cela est moins pratique lorsque l'objectif est de la réutiliser dans d'autres constructions (comme objet intermédiaire par exemple).

D'où le concept de procédure. Cependant, la traduction d'un tel concept dans un environnement de manipulation directe diffère sensiblement selon les approches prises et, d'une manière générale, sont toutes réductrices par rapport au concept informatique de procédure. Citons en particulier :

- les *scripts* de `Geometer's Sketchpad` [Jackiw 95], qui permettent d'enregistrer une séquence de constructions et de la rejouer ultérieurement en faisant correspondre des objets sélectionnés avec les paramètres formels du script ;

25. Une autre possibilité serait de la construire à partir d'une des faces d'un cube, en gommant les sommets et arêtes inutiles.

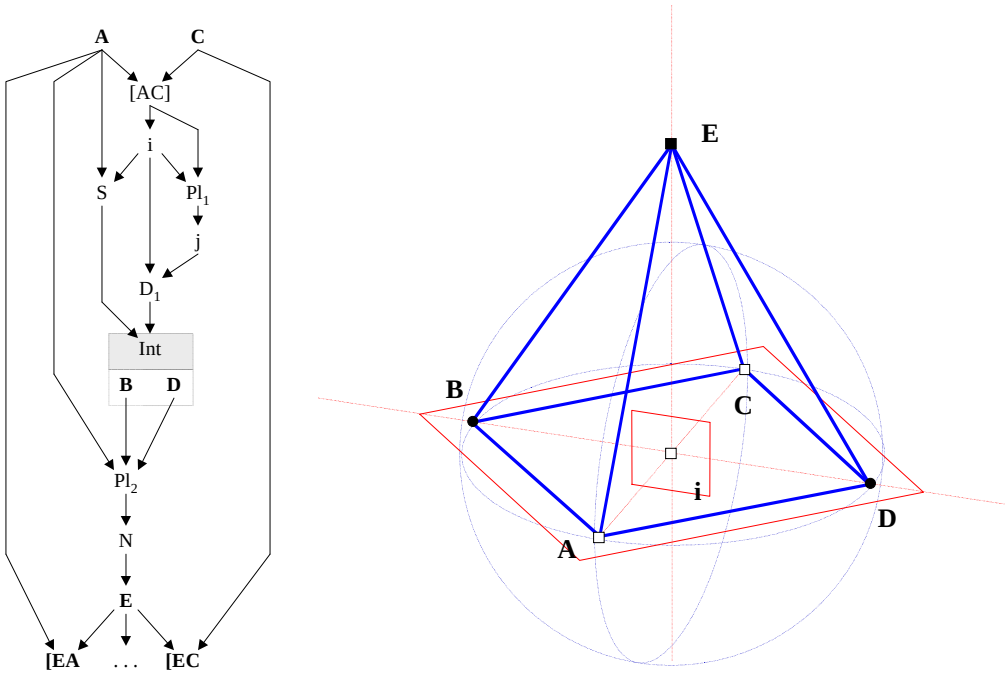


Figure 6.17 – *Construction d'une pyramide à base carrée*

– les *macro-constructions* de `Cabri Géomètre` [Baulac 92], qui permettent de définir a posteriori toute construction partiellement réalisée dans une figure comme une macro, c'est-à-dire l'extraction d'un sous-graphe connexe entre objets initiaux (paramètres) et objets finaux (résultats), puis à l'utiliser comme une primitive de construction.

Ces deux techniques de réutilisation apparaissent en fait comme des boîtes noires : l'utilisateur peut difficilement prévoir le résultat de sa construction, ce qui peut occasionner une gêne dans certains cas particuliers. Dans le cas de notre pyramide, son orientation (c'est-à-dire la position du sommet) dépend entre autre des deux points A et C. Un script ou une macro-construction de la pyramide, suivant l'ordre de désignation des deux points à instancier avec les paramètres de la procédure, donnera des résultats différents et imprévisibles.

Pour réduire cette imprécision, Philippe Bernat [Bernat 94a] a proposé une méthode qui se veut plus visuelle que les macro-constructions ou les scripts : les *greffes*. Cette technique consiste à insérer dans la figure la sauvegarde d'une construction complète précédemment réalisée puis à greffer chacun des points libres de ce greffon sur des points de la figure support. Cette technique consiste à une *agrégation structurelle* des graphes de dépendance des deux constructions : celui de la figure cible et celui de la figure à greffer.

C'est ce principe que nous avons repris pour ***Calques 3D***, en y réintroduisant *l'agrégation visuelle* des objets : la composition des objets du greffon dans une nouvelle entité plus générale, c'est-à-dire une nouvelle catégorie d'objets composites dont le graphe de composition interne est définit à l'initiative de l'utilisateur, et non plus déterminé automatiquement comme pour le cube. Ainsi, dans le cas de la pyramide, la greffe ne se contente pas d'ajouter à l'univers les objets intermédiaires nécessaires à sa construction de la pyramide mais les regroupe au sein d'un nouvel objet composite que l'utilisateur peut dénommer *"Pyramide"* (figure 6.18). L'utilisateur peut alors définir lui-même certaines propriétés textuelles de ce nouvel objet (son nom, sa définition, ...). Les éléments gommés de la figure (représentés en italique sur la figure)

sont considérés comme des constructions intermédiaires et n'apparaissent donc pas dans l'objet composite : ils ne sont ni affichés à l'interface, ni utilisables comme support de construction. Pour l'utilisateur, l'objet composite *"Pyramide"* sera constitué uniquement des sommets et des arêtes de la pyramide $ABCDM$ mais ses propriétés induites (base carrée, sommet modifiable uniquement le long de l'axe de symétrie, ...) sont conservées.

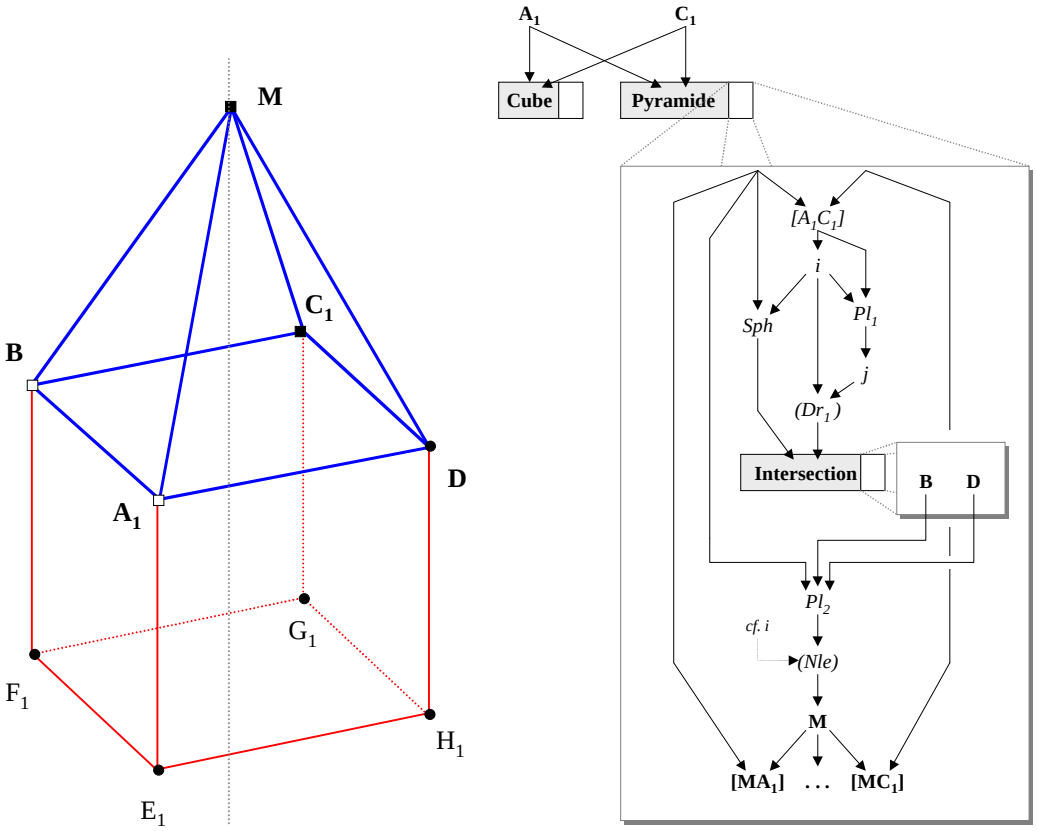


Figure 6.18 – *Greffe d'un objet composite* "pyramide" *sur un cube*

Le mécanisme de réutilisation s'effectue donc en trois étapes. La première consiste à *construire* une figure complexe à partir des objets et des primitives de construction disponibles et à gommer les éléments non significatifs de la figure. La deuxième consiste alors à définir cette figure comme une nouvelle catégorie d'*objet composite* qui sera ajoutée à une bibliothèque d'objets complexes, élaborée et complétée par les utilisateurs. La troisième étape consiste enfin à réutiliser un objet de cette bibliothèque et à le *greffer* sur les points de la figure de travail (par exemple sur un cube, cf. figure 6.18).

Le travail sur les greffes est venu très tardivement dans le développement de ***Calques 3D*** [26], car leur nécessité a été mise en évidence lors du travail avec les enseignants du technique et leur approche systémique de la géométrie (cf. chapitre 4). Il est alors apparu que la structure de données utilisée pour l'implantation des *objets composites* (cube, intersection droite/sphère, cf. section 6.4.3) permettait la mise en œuvre des greffes, sans remise en cause des choix précédents, que ce soit au niveau de la structure de données des objets, de leur représentation interne ou

26. Ce qui explique qu'elles ne soient pas mises en œuvre dans la version stable du prototype.

de la gestion de la dynamique d'une construction. Des adaptations ont été nécessaires en terme d'ajout (champs et méthodes) ou de mise à jour (par exemple, l'imbrication des objets composites n'avait pas été prévue à l'origine : le parcours du sous-graphe de *composition* se faisait de manière itérative au lieu d'être récursive).

Quatrième partie

Conclusion

Chapitre 7

Conclusion et perspectives

L'objectif de notre travail était l'étude, à travers la conception d'un environnement ouvert pour l'apprentissage de la géométrie dans l'espace, de la prise en compte de l'enseignant dans le processus de réalisation de logiciels pédagogiques.

Un premier résultat concerne la fourniture d'un produit et son impact auprès du public enseignant, ainsi que les actions de valorisation effectuées. Un deuxième concerne une proposition de méthodologie applicable pour d'autres produits similaires. Enfin, nous ouvrons les perspectives de ce travail sur deux points : le besoin d'une spécification plus formelle du domaine d'apprentissage et les évolutions futures du logiciel.

Bilan du travail

Réalisation et valorisation de *Calques 3D*

Nous avons réalisé ***Calques 3D***, un micromonde pour l'apprentissage de la géométrie dans l'espace. Nous avons proposé trois catégories de fonctionnalités : des fonctionnalités de *construction* pour permettre la réalisation de figures géométriques, des fonctionnalités de *visualisation* pour permettre l'observation de ces figures, et des fonctionnalités d'*exploration* pour permettre leur interprétation. Cependant, afin de répondre aux besoins et aux difficultés de ce domaine, l'accent a surtout été mis sur l'*observation directe* des figures géométriques en définissant et mettant en œuvre des *éléments visuels de compréhension*, c'est-à-dire des éléments qui favorisent la compréhension de la figure.

Comme nous l'avons déjà signalé, ***Calques 3D*** n'a été testé que par les enseignants impliqués dans le projet et avec des expérimentations dans leur classe respective, la plupart sous la forme de démonstrations collectives monitorées par l'enseignant. De manière à valoriser le logiciel et acquérir d'autres informations auprès d'un public différent, nous avons cherché à étendre sa diffusion auprès d'autres enseignants, dans des cadres plus institutionnels.

Une première action de valorisation de ***Calques 3D*** a eu lieu lors du colloque inter-IREM *"Géométrie et Informatique"*, organisé en Juin 1998 à Bussang. Au-delà d'une simple démonstration des possibilités du logiciel, la présentation a été complétée par l'exposé des problèmes que peut poser la réalisation d'un tel environnement, de la méthode de conception collaborative mise en place et des questions ouvertes. Le logiciel a reçu un accueil très favorable et c'est surtout l'exposé de la méthode de travail que les participants au colloque ont particulièrement apprécié, car elle illustre bien le fait que les informaticiens ont pour tâche essentielle la traduction technique de scénarios pédagogiques interactifs et attrayants que les enseignants conçoivent.

Une deuxième démonstration s'est déroulée lors du Plan National de Formation *"Développement de l'utilisation des technologies d'information et de communication dans la mise en œuvre des nouveaux programmes de mathématiques du collège"*, organisé à Nancy par l'Inspection Générale en janvier 1999. L'intérêt des participants s'est principalement focalisé sur les *éléments visuels de compréhension* mis en œuvre, apports non négligeables vis-à-vis des autres logiciels pédagogiques existants.

A l'issue de ces deux présentations, des contacts informels ont pu être pris avec certains enseignants souhaitant expérimenter le logiciel. Nous avons mis à leur disposition une version complète de ***Calques 3D***, en contrepartie de quoi nous leur demandions des retours sur l'expérimentation (réactions, problèmes, exemples d'activités, ...). Malheureusement, à défaut d'avoir pu organiser ces expérimentations dans un cadre plus strict (avec fiche de validation, observation des élèves *in situ*, ...), ces retours ont fait défaut.

Enfin, le logiciel a fait l'objet en mars 1999 d'une démonstration dans le cadre de la formation en informatique disciplinaire (module *"Intégration de l'informatique dans les pratiques pédagogiques"*) des stagiaires PLC2 à l'IUFM de Lorraine (site de Nancy). A la fin de cette formation les stagiaires ont à produire un rapport concernant une utilisation de l'outil informatique en classe. ***Calques 3D*** a été le sujet d'un tel rapport qui ne nous est pas encore parvenu à ce jour, malgré des demandes réitérées.

On peut déduire de ces diverses tentatives de diffusion qu'il est nécessaire d'établir des protocoles d'évaluation en spécifiant précisément les indicateurs permettant d'évaluer le produit et en établissant ces protocoles dans un cadre institutionnel afin de disposer de l'infrastructure propre à faire les relances et obtenir satisfaction.

Intégration de l'usager enseignant dans la conception des EIAO

Le travail de recherche présenté dans cette thèse s'inscrit dans la problématique de la conception des *Environnements Interactifs d'Apprentissage avec Ordinateur*. De manière à concevoir un environnement adapté aux différentes situations d'usage et adaptable par l'enseignant prescripteur, nous nous sommes intéressé à une méthodologie de développement qui permet de prendre en compte les besoins des usagers tout au long du processus de développement, et pas uniquement en amont de la production (par l'intermédiaire d'un cahier des charges par exemple). Cette prise en compte permanente des besoins s'est concrétisée d'une part par l'instauration d'un cycle de production basé sur le prototypage incrémental, d'autre part, par une redéfinition du rôle des enseignants au sein d'une équipe pluridisciplinaire et leur intégration, en tant qu'acteur, dans le processus de création.

De manière à contrôler le développement du prototype et à mettre en place un mécanisme d'acquisition de l'expertise pédagogique des enseignants, nous avons proposé un cadre d'expression des besoins, appelé *contexte d'utilisation d'un logiciel pédagogique*. Ce cadre permet aux enseignants auteurs de décrire des activités à réaliser autour du logiciel (activités de construction, de démonstration monitorée, ...) et d'y expliciter les choix qu'ils font au niveau de la présentation des connaissances mises en œuvre dans une telle activité. En retour, ce cadre sert de support à la négociation entre les participants au développement, sur l'implantation de ces connaissances et de leurs modes de présentation.

Malgré le fait que la forme des fiches proposées soit fortement liée à la géométrie (dans l'espace) d'une part, et à la conception d'un micromonde d'autre part, cette proposition dépasse le cadre de l'enseignement de ce domaine. Les mots-clefs ici sont *contexte d'utilisation* et *description d'activités*. Quels que soient la nature du logiciel pédagogique que l'on souhaite développer

et son domaine d'application, nous pensons qu'il est possible de concevoir des documents analogues aux nôtres et qui ciblent les trois objectifs que nous leur avons attribués : l'explicitation des connaissances implicites des enseignants, l'acquisition et l'organisation de ces connaissances à des fins d'implantation et la négociation sur cette implantation.

Perspectives du travail

Vers une spécification d'un domaine d'apprentissage

Les fiches de description de *contexte d'utilisation d'un logiciel pédagogique* sont un premier pas vers la formalisation de l'expérience et des pratiques pédagogiques d'un groupe d'enseignants. Il nous faudrait maintenant faire un deuxième pas vers une généralisation du *recueil* systématique des connaissances et des pratiques relatives à un domaine d'apprentissage. En effet, nous avons montré dans ce mémoire que l'enseignement de la géométrie dans l'espace impliquait des approches variées et parfois contradictoires. Plus que le recueil des connaissances du domaine, c'est-à-dire les objets géométriques sur lesquels il est possible d'obtenir un consensus entre les participants au projet, il est plus intéressant, voire nécessaire, de constituer un recueil explicite et "quasi" exhaustif de ces différences d'approches pédagogiques.

Dans l'état actuel de notre travail, la seule matérialisation de l'acquisition de l'expertise pédagogique mais surtout du résultat de la négociation de cette expertise reste le logiciel lui-même avec ses objets géométriques, ses primitives de construction, ses fonctions d'observation, ... Il n'y a pas eu en effet une exploitation systématique et exhaustive des connaissances contenues dans les contextes d'utilisation (concepts, aides, vocabulaire, pratiques pédagogiques, ...), faute de temps suffisant pour mener à bien ce travail.

Intégré dans le processus de développement, ce recueil aurait deux fonctions : le *partage* et la *réutilisation* des connaissances. Élaboré à partir de la description des activités fournies par les enseignants auteurs, il servirait de *mémoire* pour la négociation portant sur la sélection des connaissances embarquées dans le logiciel. Cette mémoire, évoluant au fur et à mesure du développement de ***Calques 3D***, pourrait en retour servir de *base de connaissances* pour la définition de nouvelles activités.

Nous pensons que tous les acteurs du processus de développement devraient pouvoir accéder à ce recueil et en enrichir le contenu. L'enseignant auteur doit pouvoir y décrire les concepts du domaine (par exemple le cube et ses propriétés) et les pratiques pédagogiques issues de la *transposition didactique* de ce domaine (par exemple les modes de présentation du cube). L'informaticien, sans aller jusqu'à une description du codage ou de la représentation interne des connaissances, doit pouvoir y décrire les informations relevant de la *transposition informatique* du domaine : par exemple, exprimer le fait que le cube est une *composition* d'objets élémentaires, ou décrire les modes d'interaction entre l'élève et le logiciel.

Cela implique que ce recueil ne doit pas se limiter à un simple glossaire ou thésaurus des connaissances mais doit prendre en compte des aspects plus généraux. Il doit permettre une description *à niveau connaissance* (au sens de [Newell 82] et [Nicaud 94]), indépendant d'un quelconque codage au niveau symbolique et comportant à la fois des connaissances sur le domaine (description des objets) et sur le contrôle (description des processus qui modifient les objets).

Une base de connaissances établie à ce niveau de langage représente une *ontologie*, c'est-à-dire *une spécification explicite d'une vue abstraite et simplifiée d'un monde que l'on doit se représenter pour un certain usage* [Gruber 95].

Cependant, l'état de l'art dans ce domaine et les outils aujourd'hui disponibles n'autorisent pas encore les enseignants à exprimer leurs connaissances de façon aisée et conviviale.

Evolutions de *Calques 3D*

En terme d'évolution du logiciel, plusieurs points peuvent être envisagés selon deux axes de travail : l'intégration de l'enseignant prescripteur d'une part et la couverture du domaine d'apprentissage d'autre part.

Du point de vue de l'intégration de l'enseignant prescripteur dans le logiciel, un des premiers objectifs d'amélioration de ***Calques 3D*** concerne l'accès à la paramétrisation du logiciel : cet aspect de l'utilisation d'un logiciel pédagogique est en effet essentiel et mérite un traitement aussi important que pour le logiciel proprement dit. Nous avons montré dans ce mémoire que ***Calques 3D*** proposait un ensemble de paramètres que l'enseignant pouvait sélectionner de manière à définir le comportement global du logiciel : choix des objets géométriques et des primitives de construction disponibles, préférence de modes de présentation, ... Cependant, l'interface actuelle d'accès à ces paramètres souffre de lourdeurs qui rendent cette tâche de paramétrisation relativement incommode.

Dans un premier temps, il convient de réfléchir à la mise en place d'une interface plus conviviale pour l'enseignant prescripteur. Cela pourra se faire en proposant une organisation des paramètres différente de celle mise en œuvre actuellement (c'est-à-dire qui respecte les trois objectifs *construction*, *observation* et *exploration*), et qui permettra une utilisation plus efficace des paradigmes de présentation des informations (c'est-à-dire les signets, les cases à cocher, les listes hiérarchiques, ...).

A plus long terme, il nous semble qu'un travail de fond doit être mené sur l'activité de paramétrisation en tant que telle. Le projet que nous avons présenté dans cette thèse a porté essentiellement sur la formalisation des besoins des enseignants pour l'*enseignement d'un domaine*. Il conviendrait d'envisager également un travail portant cette fois sur la formalisation des besoins relatifs à *la paramétrisation d'un logiciel pédagogique*.

Dans sa version actuelle, ***Calques 3D*** ne couvre qu'une partie seulement du programme de l'enseignement de la géométrie (qu'il s'agisse de l'enseignement général ou de l'enseignement technique) et son élargissement doit être une perspective d'évolution du logiciel à plus ou moins long terme. Cet élargissement du domaine peut être assuré en partie par l'utilisateur lui-même : le mécanisme des *greffes* proposé permet en effet la définition de nouvelles catégories d'objets, augmentant ainsi la liste des objets constructibles.

Cependant, les objets géométriques disponibles ne représentent qu'un aspect du domaine d'apprentissage de la géométrie, celui de la *construction* de figures géométriques. Les autres aspects, à savoir l'*observation* et l'*exploration* des figures ne sont pas à négliger. Dans cette optique, deux propositions peuvent être faites.

La première concerne l'introduction du registre analytique de la géométrie, présent dans de nombreux autres micromondes de géométrie dans l'espace mais que nous avions rejeté initialement pour incompatibilité avec les objectifs de ***Calques 3D***. Nous pensons que cette restriction reste opportune car les difficultés de l'enseignement de ce domaine relèvent plus de la visualisation que du passage d'un registre à un autre. Cependant certaines fonctionnalités analytiques (ou "pseudo-analytiques") pourront être envisagées comme, par exemple, la vérification automatique des propriétés géométriques (appartenance, parallélisme, ...) ou la définition de points assujettis aux référentiels (permettant de "fixer" certaines propriétés).

Deuxièmement, et pour rester dans le registre visuel qui nous semble le plus important pour l'apprentissage de la géométrie dans l'espace, nous envisagerons l'ajout de nouveaux modes de présentation des figures géométriques, comme par exemple le "rendu-réaliste" (permettant l'accès à la géométrie du solide) ou les *épures de Monge* (permettant l'accès à la *géométrie*

descriptive, comme le décrit Paulo Pavel dans sa thèse [Pavel 99]). Sur ce dernier point, un début de collaboration avec l'auteur nous a permis d'établir que ***Calques 3D*** disposait actuellement de tous les éléments nécessaires à une simulation acceptable de l'épure (par extraction d'objets dans des calques séparés et observation frontale selon le plancher horizontal et une cloison verticale). Ce travail permettrait d'améliorer l'efficacité et le potentiel pédagogique de ***Calques 3D***.

En conclusion, nous voudrions souligner l'enrichissement personnel apporté par cette expérience, en particulier la collaboration, au sein d'une équipe pluridisciplinaire, avec des enseignants et le développement d'un prototype opérationnel répondant à des besoins pédagogiques. Au delà de l'apport personnel, cette méthode de travail nous semble être la seule voie amenant à la réalisation et surtout à l'utilisation effective de logiciels pédagogiques.

Bibliographie

[Allard 86] J.C. Allard et C. Pascal. EUCLIDE, un langage pour la géométrie plane. Cédic-Nathan, 1986.

[Allard 88] J.C. Allard et C. Pascal. Euclide dans l'espace, un langage pour la géométrie dans l'espace. Cedic-Nathan, 1988.

[Artigue 89] M. Artigue, J. Belloc et S. Touaty. Une recherche menée dans le cadre du projet Euclide. Rapport, IREM - Paris VII, 1989.

[Audibert 85] G. Audibert. *Représentation de l'espace et empirisme dans le problème FIL*. Publication IREM-USTL, Montpellier, 1985.

[Audibert 88] G. Audibert. *La perspective cavalière*, volume 75. publication APMEP, 1988.

[Authier 98] A. Authier, C. Grolleau, M.L. Hocquenghem, S. Hocquenghem, F. Monnet, Y. Paquelier, P. Sérès, A.M. Serfati et A. Varoquaux. Géospace : logiciel de construction mathématique dans l'espace. CREEM (CNAM) - CRDP de Champagne-Ardenne, 1998.

[Bailleux 91] A. Bailleux. *Universités d'été "Enseigner la géométrie avec l'Ordinateur"*. IUFM de Douai, 1991.

[Balacheff 91] N. Balacheff. Contribution de la didactique et de l'épistémologie aux recherches en EIAO. C. Bellissant, éditeur, *Actes des XIIIe journées francophones de l'informatique*, pages 9–38, Grenoble, 1991.

[Balacheff 94a] N. Balacheff. Didactique et intelligence artificielle. N. Balacheff et M. Vivet, éditeurs, *Didactique et intelligence artificielle*. La Pensée Sauvage, Grenoble, 1994.

[Balacheff 94b] N. Balacheff. La transposition informatique, un nouveau problème pour la didactique. M. Artigue, éditeur, *Vingt ans de didactique des mathématiques en France*, pages 364–370. La Pensée Sauvage, Grenoble, 1994.

[Baron 91] M. Baron, R. Gras et J.F. Nicaud. Introduction. M. Baron et J.F. Nicaud, éditeurs, *EIAO'91 - Deuxièmes journées EIAO de Cachan*, pages 7–8. Editions de l'ENS de Cachan, 1991.

[Baulac 90] Y. Baulac. *Un micromonde de géométrie dynamique : Cabri-géomètre.* Thèse de Doctorat, Université Joseph Fourier, 1990.

[Baulac 92] Y. Baulac, F. Bellemain et J.M. Laborde. Cabri : the Interactive Geometry Notebook, 1992.

[Bernat 89] P. Bernat. Dessiner l'Espace. Topiques Editions, 1989.

[Bernat 91] P. Bernat. Pratiquer l'Espace. Topiques Editions, 1991.

[Bernat 94a] P. Bernat. Calques 2. Topiques Editions, 1994.

[Bernat 94b] P. Bernat. *Conception et réalisation d'un environnement interactif d'aide à la résolution de problèmes. CHYPRE : un exemple pour l'enseignement de la géométrie.* Thèse de Doctorat, Université Henri Poincaré - Nancy I, 1994.

[Bernat 95] P. Bernat et J. Morinet-Lambert. Spécificités et modélisation de l'interaction dans un EIAO. *EIAO'95 - Quatrièmes Journées EAIO de Cachan*, pages 208–220, Cachan, 1995. Eyrolles, Paris.

[Blondel 96] F.M. Blondel. *Diagnostic et aide en EIAO. Etude d'un environnement d'aide à la résolution de problèmes en chimie.* Thèse de Doctorat, Université Henri-Poincaré - Nancy I, 1996.

[Bénézra 89] A. Bénézra, F. Jean et B. Rothan. *Math plein écran.* Fenêtre active. CRDP de Lorraine, Nancy, 1989.

[Boehm 76] B. Boehm. Software Engineering. *IEEE Transaction on Computers*, C25(12) :1226–1241, 1976.

[Boehm 88] B. Boehm. A Spiral Model of Software Development and Enhancement. *IEEE Computer*, 12(5) :61–72, 1988.

[Bouteiller 93] Y. Bouteiller et M. Duperier. *Apprendre et pratiquer la géométrie avec l'ordinateur.* IREM d'Orléans, 1993.

[Bruillard 94] E. Bruillard et M. Vivet. Concevoir des EIAO pour des situations scolaires, approche méthodologique. *Recherche en Didactique des Mathématiques*, 12(1-2) :275–302, 1994.

[Brunet 81] R. Brunet et A. Chevalier. Rôle du dessin dans la résolution des problèmes de géométrie. *Compte-rendu du colloque GEDEOP*, pages 37–48. IREM-Université de Poitiers, 1981.

[Chouanière 91] B. Chouanière et T. Gille. *Mathématiques : problèmes concrets.* Fenêtre active. CRDP de Lorraine, Nancy, 1991.

[Colotte 88] G. Colotte, B. Dugand, J. Lopparelli et M. Mercier. *Chrono 6 : Histoire et Informatique.* Fenêtre active. CRDP de Lorraine, Nancy, 1988.

[Coutaz 88] J. Coutaz. *Interface Homme-Ordinateur : Conception et Réalisation*. Thèse d'État, Université Joseph Fourier, 1988.

[Cuppens 91] R. Cuppens. *Université d'Eté "Informatique et Enseignement de la Géométrie"*. IREM de Toulouse, 1991.

[Davis 88] A.M. Davis, E.H. Bersoff et E.R. Comer. A strategy for comparing alternative software development life cycle models. *IEEE Transactions on Software Engineering*, 14(10), 1988.

[de Berg 97] M. de Berg, M. van Kreveld, M. Overmars et O. Schwarzkopf. *Computational Geometry : Algorithms and Applications*. Springer-Verlag, 1997.

[Destainville 95] B. Destainville. *Ces problèmes qui font les mathématiques : enseigner la géométrie dans l'espace au collège et au lycée*, volume 99. Publication APMEP, 1995.

[DLC15 94] DLC15. *Faire des Mathématiques au lycée avec l'Ordinateur*. Direction des Lycées et Collèges. Bureau 15, Ministère de l'Education Nationale, 1994.

[Dubourg 95] X. Dubourg. *Modélisation de l'interaction en EIAO, une approche événementielle pour la réalisation du système REPERES*. Thèse de Doctorat, Université de Caen, 1995.

[Feigenbaum 83] E. Feigenbaum et P. McCorduck. *The Fifth Generation*. Addison Wesley, Reading, Mass., 1983.

[Foley 90] J.D. Foley, A. van Dam, S.K. Feiner et J.F. Hughes. *Computer Graphics : Principles and Practice*. Systems programming series. Addison Wesley, 2e édition, 1990.

[Gaines 87] B.R. Gaines. An Overview of Knowledge Acquisition and Transfer. *Journal of Man-Machine Studies*, 26(4) :453–472, 1987.

[Gaudel 96] M.C. Gaudel, B. Marre, F. Schlienger et G. Bernot. *Précis de génie logiciel*. Enseignement de l'informatique. Masson, 1996.

[Goldenberg 98] E.P. Goldenberg et A. Cuoco. What Is Dynamic Geometry ? R. Lehrer et D. Chazan, éditeurs, *Designing Learning Environments for Developing Understanding of Geometry and Space*, Studies in Mathematical Thinking and Learning. Erlbaum, Hillsdale, NJ, 1998. (to appear).

[Goodman 97] J.E. Goodman et J. O'Rourke, éditeurs. *Handbook of Discrete and Computational Geometry*. CRC Press LLC, 1997.

[Grandbastien 99] M. Grandbastien. Teaching Expertise is at the Core of ITS Research. *International Journal of Artificial Intelligence in Education*, 10, 1999.

[Gruber 95] T.R. Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human and Computer Studies*, 43(5/6) :907–928, 1995.

[Guillet 88] N. Guillet. *Ordi-Lettre : Français et Informatique*. Fenêtre active. CRDP de Lorraine, Nancy, 1988.

[Guin 94] D. Guin. Nécessité et richesse d'une interaction entre concepteurs des outils informatiques, didacticiens et formateurs dans l'enseignement des mathématiques. *Apports de l'outil informatique à l'enseignement de la géométrie*, pages 5–16. Commission Inter-IREM Mathématiques et Informatique, 1994.

[Houben 86] J.P. Houben et J. Vanhamme. SEDIMA. *Colloque Inter-IREM*, pages 53–57, 1986.

[IREM 94] IREM. *Apports de l'outil informatique à l'enseignement de la géométrie*. Commission Inter-IREM Mathématiques et Informatique, 1994.

[Jackiw 95] N. Jackiw. The Geometer's Sketchpad. Visual Geometry Project, Key curriculum Press, 1995.

[Krief 92] P. Krief. *Utilisation des langages objets pour le prototypage*. Collection Etude et Recherche en Informatique. Masson, 1992.

[Laborde 94] J.M. Laborde et F. Bellemain. Cabri-Géométre II, logiciel et manuel d'utilisation. Texas Instruments, 1994.

[Laborde 96] C. Laborde. Towards a New Role of Diagrams in Dynamic Geometry ? P. Brna, A. Paiva et J. Self, éditeurs, *Euro-AIED - European Conference on Artificial Intelligence in Education*, pages 350–356, Lisbon, Portugal, 1996. Edições Colibri.

[Lapujade 96] A. Lapujade et C. Ernst. Un panorama des systèmes d'aide à la conception de logiciels éducatifs. *Sciences et Techniques Educatives*, 3(3) :297–334, 1996.

[Lawler 84] R.W. Lawler. Designing Computer-Based Microworlds. M. Yazdani, éditeur, *New Horizons in Educational Computing*. Ellis Horwood, 1984.

[Lepine 97] J. Lepine et S. Wallerand. Atelier de géométrie 3D. TLC-Edusoft, 1997.

[Major 92] N. Major et H. Reichgelt. COCA : A Shell for Intelligent Tutoring Systems. *ITS'92 - 2nd International Conference on Intelligent Tutoring Systems*, volume 608, série *LNCS*, pages 523–530, Montréal, 1992.

[Meirieu 87] P. Meirieu. *Apprendre... oui, mais comment*. ESF, Paris, 1987.

[Morlet 89] C. Morlet. *Dessiner l'Espace*. IREM de Lorraine. Topiques Editions, 1989.

[Murray 97] T. Murray. Expanding the knowledge acquisition bottleneck for Intelligent Tutoring Systems. *International Journal of Artificial Intelligence in Education*, 8(3-4) :222–232, 1997.

[Murray 99] T. Murray. Authoring Intelligent Tutoring Systems : An Analysis of the State of the Art. *International Journal of Artificial Intelligence in Education*, 10 :98–129, 1999.

[Muzellec 96] D. Muzellec. Le logiciel KAPPA, la géométrie en trois dimensions, 1996.

[Nanard 90] J. Nanard. *La manipulation directe en interface homme-machine*. Thèse d'État, Université de Sciences et Technique du Languedoc, 1990.

[Newell 82] A. Newell. The Knowledge Level. *Artificial Intelligence*, 18(1) :87–127, 1982.

[Nicaud 88] J.F. Nicaud et M. Vivet. Les tuteurs intelligents : réalisation et tendances de recherches. *Techniques et Sciences Informatiques*, 7(1) :21–45, 1988.

[Nicaud 94] J.F. Nicaud. *Modélisation du raisonnement algébrique humain et conception d'environnements informatiques pour l'enseignement de l'algèbre*. Habilitation à diriger des recherches, LRI, Université Paris 11, 1994.

[Norman 86] D.A. Norman. Cognitive Ingeneering. D.A. Norman et S.W. Draper, éditeurs, *User Centered System Design*. Lawrence Erlbaum Associates, 1986.

[O'Rourke 98] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 2^{e} édition, 1998.

[Pallascio 92] R. Pallascio, R. Allaire et P. Mongeau. Représentation de l'espace et enseignement de la géométrie. *Topologie Structurale*, 19, 1992.

[Papert 81] S. Papert. *Le Jaillissement de l'esprit*. Flammarion, 1981.

[Papert 87] S. Papert. Microworlds : Transforming Education. R.W. Lawler et M. Yazdani, éditeurs, *Artificial Intelligence in Education*, volume 1, pages 79–92. 1987.

[Parzysz 88] B. Parzysz. Knowing vs. Seeing, problems of the plane representation of space geometry figures. *Educational Studies in Mathematics*, 19(1) :79–92, 1988.

[Pavel 99] P. Pavel. *GD.Visu@l, Environement Distribué Interactif d'Apprentissage Humaine de la Géométrie Descriptive*. Thèse de doctorat (soutenence en octobre 1999), Université du Maine - Le Mans, 1999.

[Preece 94] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland et T. Carey. *Human-Computer Interaction*. Addison-Wesley, 1994.

[Qasem 97] S. Qasem. *Conception et réalisation d'une interface 3D pour Cabri-Géomètre.* Thèse de Doctorat, Université Joseph Fourier , Grenoble I, 1997.

[Quéré 91] M. Quéré. *Systèmes Experts et Enseignement Assisté par Ordinateur.* Ophrys, Paris, 1991.

[Rogers 90] D.F. Rogers et J. A. Adams. *Mathematical Elements for Computer Graphics.* McGraw Hill, 2e édition, 1990.

[Rommevaux 91] M.P. Rommevaux. Le premier pas dans l'espace. *Annales de Didactiques et de Sciences Cognitives*, 4 :85–123, 1991.

[Schneiderman 82] B. Schneiderman. The future of interactive systems and the emergence of direct manipulation. *Behavior and Information Technology*, 7 :237–256, 1982.

[Self 85] J. Self. *Microcomputers in education : a critical appraisal of educational software.* Harbester Press, 1985.

[Stefik 95] M. Stefik. *Introduction to Knowledge Systems.* Morgan Kaufman, San Francisco, 1995.

[Vivet 91] M. Vivet. Expertise pédagogique et usage des tuteurs intelligents. *Journées Francophones : Formation Intelligemment Assistée par Ordinateur*, Genève, 1991.

[Watt 93] A. Watt. *3D Computer Graphics.* Addison-Wesley, 2e édition, 1993.

[Wenger 87] E. Wenger. *Artificial Intelligence and Tutoring Systems.* Morgan Kaufman, 1987.

Liste des publications

[Desmoulins 96] C. Desmoulins et N. Van Labeke. Towards Student Modelling in Geometry with Inductive Logic Programming. P. Brna, A. Paiva et J. Self, éditeurs, *Euro-AIED - European Conference on Artificial Intelligence in Education*, Lisbon, Portugal, 1996.

[Van Labeke 97a] N. Van Labeke. Calques 3D : Guide de l'utilisateur. Manuel no. 97-R-184, LORIA - UHP/Nancy I, 1997. Version provisoire.

[Van Labeke 97b] N. Van Labeke et P. Bernat. Conception d'un environnement d'apprentissage adaptable : une étude en géométrie dans l'espace. *EIAO'97 - Cinquièmes Journées EAIO de Cachan*, pages 121–132. HERMES, 1997.

[Van Labeke 98a] N. Van Labeke. Calques 3D : a microworld for spatial geometry learning. *ITS'98- 4th International Conference on Intelligent Tutoring Systems - System Demonstrations*, San antonio, Texas, 1998.

[Van Labeke 98b] N. Van Labeke et J. Morinet-Lambert. Influence of Didactic and Computational Constraints on ILE design. Rapport interne no. 98-R-003, LORIA - UHP/Nancy I, 1998.

[Van Labeke 98c] N. Van Labeke, J. Morinet-Lambert et M. Grandbastien. Designing Adaptable Educational Software : A Case-study for Spatial Geometry. *ED MEDIA'98 - World Conference on Educational Multimedia and Hypermedia*, Freiburg, Germany, 1998. AACE.

[Van Labeke 99a] N. Van Labeke. Développement d'un logiciel pour l'enseignement de la géométrie dans l'espace en partenariat Université/Second degré. *actes du Plan National de Formation "Développement de l'utilisation des technologies d'information et de communication dans la mise en œuvre des nouveaux programmes de mathématiques du collège"*, Nancy, 1999.

[Van Labeke 99b] N. Van Labeke, R. Aiken, J. Morinet-Lambert et M. Grandbastien. IF "What is the Core of AI & Education ?" is the Question THEN "Teaching Knowledge" is the Answer. S.P. Lajoie et M. Vivet, éditeurs, *AIED'99 - 9th International Conference on Artificial Intelligence in Education*, pages 241–248, Le Mans, France, 1999. IOS Press.

Cinquième partie

Annexes

Annexe A

Compte-rendus de réunions du projet *Calques 3D*

A.1 Compte-rendu n° 1 : réunion du 16/10/1996

Présents :

Mme Françoise JEAN, Mme Christine MANCIAUX, Mme Marie-Hélène MUNIER, Mme Agnès VOLPI, M. Jean-Claude DEMOLY, M. Philippe LOMBARD, M. Christian GINET, M. Bernard PIERROT, M. Philippe BERNAT, M. Nicolas VAN LABEKE.

A noter

Bonne nouvelle : le Rectorat accorde aux enseignants du secondaire qui participent au projet une décharge de 0,25 HSA.

Résumé de la présentation et des interventions

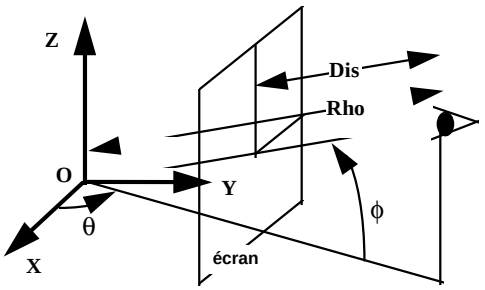
L'objectif de cette séance était surtout de présenter l'état actuel de la maquette et de mettre en place notre collaboration future. Disposer d'une première maquette suffisamment complète a permis de faire ressortir les possibilités de l'ordinateur pour aider à l'enseignement de la géométrie spatiale (pouvoir changer le point de vue de l'observateur, manipulation directe de la construction,...) mais aussi les problèmes posés (du point de vue pédagogique et implémentation). En effet, comme je l'ai signalé au début de la session, nous avons dû faire un certain nombre de choix personnels lors du développement qui se sont avérés en fait peu adaptés à un contexte pédagogique. A mon sens, une collaboration étroite entre utilisateurs potentiels (dans ce cas les enseignants) et les développeurs se justifie donc pleinement, comme nous avons pu le constater au travers de l'échange fructueux de cette première réunion. Ce compte-rendu présente une synthèse de l'état de la maquette et des remarques formulées durant la présentation.

Les principaux points sur lesquels nous avons discutés se classent en trois catégories :

- la représentation de l'espace (notamment les éléments visuels de compréhension)
- la manipulation des objets géométriques
- la définition et la représentation des objets géométriques
- la définition des objectifs pédagogiques de l'application

La représentation de l'espace et des objets géométriques

Le mode de projection de l'espace dans le plan (écran de l'ordinateur) que nous avons choisi est une projection en coordonnées sphériques : le plan de projection est perpendiculaire à un axe reliant un observateur virtuel et le centre du repère. Cette projection est la plus simple pour en appréhender les modifications : pour changer l'orientation de la scène, il suffit soit de déplacer l'observateur par rapport au repère (angles *thêta* et *phi*, distance *rho*), soit de déplacer le plan de projection par rapport à l'observateur (distance *Dis*), soit de changer le type de projection. Deux projections sont implantées : la *projection en perspective*, qui permet d'introduire les notions de points de fuite et de ligne d'horizon, et la *projection parallèle*. Ces deux termes que j'utilise dans la maquette (en référence à des ouvrages de *géométrie computationnelle*) ne sont généralement pas ceux utilisés dans l'enseignement de la géométrie et mettent en évidence le problème du vocabulaire à adopter et à figer une fois pour toute.



Une caractéristique qui découle de ce choix de représentation est que l'axe Oz reste toujours vertical à l'écran.

Le choix de ce mode de représentation de l'espace sur l'écran ne semble a priori pas poser de problème particulier, à la condition toutefois que l'utilisateur dispose d'outils lui permettant de changer l'orientation de n'importe quel objet dans la scène, indépendamment des autres.

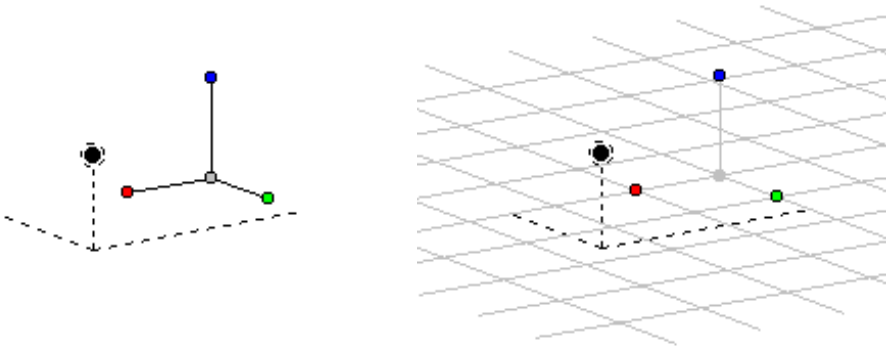
Pour visualiser l'espace dans lequel l'utilisateur va travailler, nous avons choisi deux types de représentation visuelle : le *repère orthonormé* (représentation des axes Ox, Oy et Oz) et le *plancher opaque/transparent* (représentation du repère et du plan xOy sous forme d'un maillage). Le caractère *opaque* de ce dernier permettra de masquer ou d'atténuer (en fils de fer par exemple) les objets se trouvant de l'autre coté du plan xOy par rapport à l'observateur.

La position relative des objets (sous le plan xOy, devant, derrière,...) n'a pas encore été parfaitement réglée. Seuls les points admettent une représentation visuelle de leur position dans l'espace, par projection sur les axes.

Il apparaît nécessaire de compléter ces éléments visuels pour aider davantage à la compréhension. Ces éléments visuels ont cependant l'inconvénient de ne pas vraiment aider à la compréhension de l'espace. D'une part, les projections sur les axes n'apparaissent que lorsque l'objet est sélectionné, ce qui veux dire qu'il n'y a aucune information sur les positions relatives des autres objets de l'univers.

Il semble de plus que la représentation du repère est souvent inutile : par exemple dans la géométrie du solide, la présence du cube, du cylindre ou de tout autre objet sur lequel s'appuie la construction suffit à appréhender de manière satisfaisante l'espace.

Une proposition intéressante a été faite en ce qui concerne l'amélioration de ces éléments visuel de compréhension. A la place du repère orthonormé ou du simple plan horizontal (opaque ou non), la construction et la manipulation des points (et des autres objets géométriques) pourrait



se faire dans un espace *'cloisonné'*, c'est-à-dire avec l'utilisation conjointe de plans horizontaux, gauche et arrière par exemple, délimitant un environnement clos, et sur lesquels se visualiseraient les projections des points. Les positions relatives des objets serait alors plus explicites et cela permettrait en outre *'d'effacer'* les objets situés derrière ces cloisons. De plus, il a été proposé que cette restriction de l'espace de travail puisse s'atténuer petit-à-petit, soit lorsque la construction se complète et ne nécessite plus cet environnement clos, soit lorsque l'enseignant le décide (par exemple en géométrie analytique ou lorsque qu'il juge que l'élève en maîtrise suffisamment les concepts). Le terme *d'étayage de l'apprentissage* a été utilisé pour définir cette idée.

La manipulation directe des objets géométriques

L'appréhension de l'espace étant suffisamment difficile, nous avons choisi, pour le moment, de limiter le déplacement des objets aux seuls points. Tout objet géométrique étant construit à partir de points, cela n'empêche pas la déformation de la construction. La méthode que nous avons choisie d'implanter dans la maquette consiste à décomposer le déplacement dans l'espace en déplacements dans le plan : un point sélectionné se déplacera dans un plan parallèle au plan xOy et selon une droite parallèle à l'axe Oz lors que la touche SHIFT est maintenu pressée. Cette méthode, qui donne d'assez bons résultats en général devient ambigue, voire absurde, lorsque l'observateur fait face au plan xOy ou se trouve dans ce plan. Une autre proposition faite consisterait à associer à chaque point, lors de sa sélection, une série de poignée définissant sans équivoque le sens du déplacement (contraint dans un plan, sur une direction donnée...). De même, avec l'environnement cloisonné présenté ci-dessus, un point pourrait être aussi déplacé en agissant sur chacune des projections sur les cloisons : on obtiendrait ainsi un déplacement dans l'espace moins ambigu.

Aucune de ces méthodes ne doit être préférée à une autre : le choix final doit revenir à l'enseignant.

Définir les objets géométriques

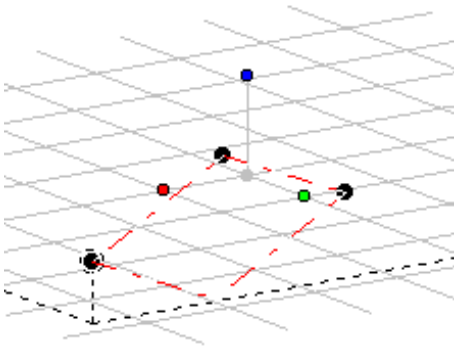
Comme je l'ai dit précédemment, nous avons décidé que seul le point serait un objet de base de la construction, c'est-à-dire sans ancêtres. Nous imposons ainsi qu'une droite soit, par exemple, explicitement construite comme passant par deux points (ou par un point et parallèle à une direction bien sûr). Ce choix s'est fait sur la même base que pour la manipulation directe des objets, la compréhension de l'espace étant par nature suffisamment compliquée pour ne pas rajouter des éléments perturbateurs. Néanmoins, il reste à savoir si on laisse à l'utilisateur la

possibilité de créer les points nécessaires à la construction d'un objet *à la volée*, c'est-à-dire au fur et à mesure des besoins, sans avoir à les construire explicitement. Les difficultés liés à la construction et à la manipulation des points présentés ci-dessus limite cependant cette possibilité.

Plus généralement se pose la question de la gestion des implicites : lorsque l'utilisateur veut construire un point quelconque mais désigne en fait l'intersection (réelle) de deux droites ou un segment par exemple, faut-il lui autoriser à créer dans ce cas le point intersection ou le point sur le segment ? N'ayant dans cette situation aucune information sur l'intention véritable de l'utilisateur, ce problème ne peut être résolu dans l'absolu et doit être considéré comme un *paramètre didactique* de l'apprentissage.

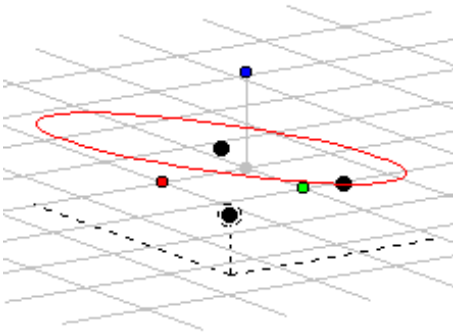
Durant la présentation de la maquette, nous avons relevé différents problèmes liés à la construction et à la représentation de quelques objets géométriques existant dans la maquette.

Le plan se définit par trois points (non alignés) et est représenté par un simple parallélogramme reliant ces trois points. Or le statut du plan est ambigu : selon le niveau de l'enseignement, l'objet plan peut être implicitement défini par l'existence des trois points et ne nécessite pas d'être représenté en tant que tel ou peut être assimilé à la notion de face (dans la géométrie du solide). Cette multitude de représentations d'un même objet *conceptuel* est très importante dans le processus d'adaptation du logiciel à l'enseignant et demande à être approfondie. Pour pallier aux défauts de l'aspect visuel du plan, une idée proposée est de conserver la représentation du plan comme un parallélogramme (ou un rectangle ?) mais de modifier sa taille dynamiquement que manière à ce qu'il englobe non seulement les points qui ont servis à le construire mais aussi tous les objets appartenant à ce plan (par exemple, les points intersection d'une droite et de ce plan).

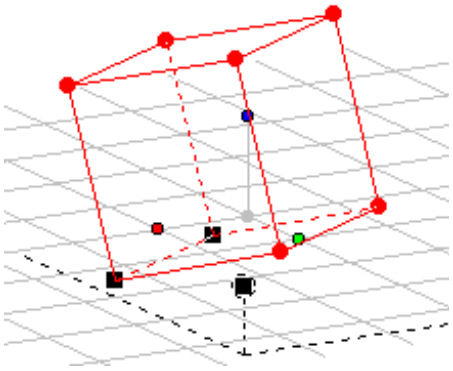


Le cercle se construit par désignation de son centre, d'un point de la circonférence et d'un troisième point définissant le plan dans lequel ce cercle va être construit. Le problème posé par cette construction est que ce troisième point n'appartient pas au cercle mais joue tout de même un rôle important dans sa définition. Mais il ne faudrait pas confondre cette construction avec la construction d'un cercle passant par trois points. Cet exemple nous montre aussi qu'un même objet peut avoir plusieurs modes de construction.

La construction **du cube** pose les mêmes problèmes que pour le cercle : deux points sont désignés pour définir une arête du cube et un troisième pour définir le plan sur lequel il va reposer. Les autres arêtes sont définies de manière non ambigue par ses trois points (par produit vectoriel et report de la distance de la première arête). L'ennui est que le troisième point n'appartient pas



(a priori) au cube. Cette construction par trois points n'est pas la seule possible : on peut aussi imaginer de construire le cube à partir d'une unique arête, l'orientation générale du cube étant définie arbitrairement (*'au dessus et à gauche de l'arête'*), à partir de son centre de gravité et d'un autre point (quel est ce sommet ?) ou d'une diagonale (quelle est cette diagonale ?),...



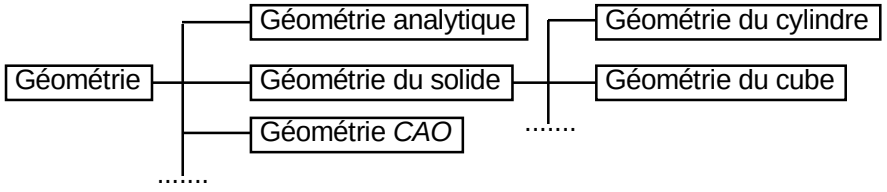
De même, l'aspect visuel du cube est sujet à discussion : élimination des faces cachées ou arêtes cachées en fils de fer (géométrie du solide), représentation sous forme arêtes (géométrie *'du squelette'*),...

Les solutions à toutes ces questions posées (multiples représentations, choix des modes de constructions et de manipulation des objets,...) ne sont pas à figer définitivement dans un logiciel. Il faut aussi éviter de le transformer en véritable *'usine à gaz'* qui deviendrait inutilisable de par sa complexité. La solution consiste à organiser ces différents choix et/ou paramètres selon des objectifs pédagogiques bien définis et laisser la possibilité à l'enseignant, soit de les redéfinir complètement, soit de les adapter à ses propres besoins. Pour cela, un travail important reste à faire : définir précisément ce que l'on entend par objectif pédagogique et ce que cela implique au niveau du logiciel.

Définir les objectifs pédagogiques du système

En initiant ce projet de conception d'un logiciel de géométrie dynamique dans l'espace, nous n'avions ni la prétention, ni la possibilité de couvrir tous les champs de l'enseignement de la géométrie spatiale. L'une des raisons pour laquelle nous avons voulu travailler de concert

avec des enseignants (premiers utilisateurs de logiciels éducatifs avant les élèves) est de pouvoir justement fixer les objectifs d'un tel logiciel. En d'autres termes il nous faut définir quels sous-ensembles de la Géométrie (au sens large du terme) nous voudrions voir intégrés. Nous avons introduit le terme de *géométrie didactique* pour désigner ces sous-parties de la Géométrie, en partant du principe que leur élément discriminant est bien l'objectif d'apprentissage sous-jacent, les éléments visuels de compréhension n'en étant qu'une conséquence ou une illustration. Ces différentes géométries, ou plutôt les différents paramètres didactiques qu'elles impliquent, ne sont bien évidemment pas exclusifs et demandent à être définis et organisés. Par exemple, une première topologie pourrait être la suivante :



Le cadre d'une session d'utilisation du logiciel pourrait être la *Géométrie du cube* (construire des sections du cube, des intersections de faces et de droites,...), ce qui impliquerait un certain nombre de paramètres didactiques mais l'enseignant peut souhaiter disposer de capacités analytiques (calcul de volume, d'aire,...), provenant de la *Géométrie analytique*, et de représentation des vues de face, de côté,... provenant de la *Géométrie CAO*. Ces différentes géométries, leur définition, les éléments visuels de compréhension qu'elles impliquent,... restent à définir.

Préparation de la prochaine réunion

Certaines personnes ayant des difficultés à bloquer régulièrement le Mercredi après-midi pour les réunions du groupe, il a été décidé d'alterner autant que possible les jours des réunions, de manière à ce que tout le monde puisse y assister régulièrement. Il a été envisagé de créer des sous-groupes travaillant en parallèle sur des niveaux d'enseignement particuliers, Mais, n'étant pas si nombreux que ça et pouvant avoir des idées et des remarques à faire sur l'ensemble des réflexions, il est peut-être préférable de *'spécialiser'* chaque réunion de travail. Dans tous les cas, que vous puissiez être présent aux réunions ou non, un compte-rendu (peut-être pas aussi étoffé que celui-ci) vous sera envoyé à tous. Ceci vous permettra d'être au courant de ce qui a été fait, dit ou décidé et de l'ordre du jour de la réunion suivante. N'hésitez donc pas à nous faire parvenir vos réflexions par écrit.

Objectifs de la prochaine réunion

Nous avons pensé qu'il était important que chacun réfléchisse aux différentes *géométries didactiques* qu'il utilise ou qu'il souhaite voir mettre en œuvre dans le système. Chacune d'elles devrait pouvoir se définir en particulier par la donnée des éléments suivants (ou d'autres qui vous semblent pertinents) :

- son nom (exemple : *géométrie du cube*)
- les objectifs pédagogiques associés à cette géométrie (*apprendre à réaliser une construction sur la base d'un objet géométrique complexe comme le cube*)
- les objets géométriques disponibles dans ce contexte, plus précisément
    - les types d'objets (*points, droites,...*),

- leurs modes de construction (*un segment à partir de deux points, le cercle soit par son centre et un rayon, soit passant par trois points, les parallèles à une droite ou à un segment,...*)
- les éventuelles contraintes associées aux constructions (*un point sur un face est limité à cette face, à la différence d'un point dans un plan*)
- les différentes représentation de l'univers et des objets (*cloisons, représentation des collisions et des intersections,...*)

La prochaine réunion est prévue pour

Jeudi 14 Novembre 1996 de 15h à 18h à **l'IREM, Faculté des Sciences, Bâtiment du 1er Cycle**

A.2 Compte-rendu n° 3 : réunion du 11/02/97

Présents :

Mme Françoise JEAN, Mme Christine MANCIAUX, Mme Marie-Hélène MUNIER, Mme Agnès VOLPI, Mme Monique GRANDBASTIEN, Mme Josette MORINET-LAMBERT, Mme Marilyne MACRELLE, M. Jean-Claude DEMOLY, M. Christian GINET, M. Bernard PIER-ROT, M. Nicolas VAN LABEKE.

A noter

- Vous trouverez joint à ce compte-rendu (pour ceux qui n'étaient pas à la réunion) une disquette contenant la version actuelle de ***Calques 3D***, ainsi qu'une fiche décrivant l'installation et l'utilisation du logiciel (l'aide en ligne n'étant pas encore réalisée). Il serait bien que vous mettiez votre nom sur la disquette et que vous me la rendiez à chaque réunion, de manière à ce que je puisse vous fournir les nouvelles versions au fur et à mesure de leur amélioration, tout en sachant qui a eu ou non cette version.
- Nous travaillons actuellement avec des étudiants du DESS *Imagerie Numérique et Interactivité* d'Epinal pour la réalisation de l'habillage du CD-ROM que nous devons fournir à la région (partenaire du projet) : réalisation de la pochette du CD-ROM, de la charte graphique, d'une présentation interactive associée au logiciel et présentant ses caractéristiques, objectifs,... Nous avons donc défini une double page de présentation de ***Calques 3D*** que je joins au compte-rendu de manière à ce que vous la commentiez.
- Enfin, dans l'optique de conformer le logiciel aux exigences des utilisations pédagogiques de ***Calques 3D***, je joins en annexe l'ensemble des textes (menus, explication des commandes, aide textuelle des tâches de créations des objets géométriques, messages d'erreur,...) utilisés dans ***Calques 3D***. Toutes les suggestions sont les bienvenues !

Compte-rendu de la séance

Cette réunion a été entièrement consacrée à une présentation de l'état actuel de la maquette. L'objectif à court terme est de finaliser et de stabiliser le noyau existant de la maquette plutôt que de commencer à développer de nouvelles fonctionnalités. Nous avons passé en revu l'ensemble des possibilités du logiciel, depuis l'interface jusqu'aux messages et rétroactions, en passant , bien évidemment, par les tâches de construction des objets géométriques. Voici l'essentiel des remarques faites durant cette réunion.

L'interface de l'application.

L'interface et les fonctionnalités annexes (c-a-d autres que purement géométriques) ont évolué depuis la dernière fois mais présentent encore un certains nombre de problèmes. Ceci est du principalement au fait que certains outils étaient nécessaires pour le développement (accès aux paramètres notamment) mais s'avèrent être mal adaptés à un contexte utilisateur. L'exemple le plus représentatif de ce problème est l'accès aux paramètres de visualisation de l'univers (position de l'observateur, type de perspective ou du repère,...) par une boîte de dialogue surchargée et donc peu utilisable. Il a donc été proposé, pour simplifier et rendre compréhensible les différents paramètres à l'utilisateur, de répartir l'ensemble de ces paramètres soit directement dans les menus existant (ou en particulier ajouter un menu **Visualisation**), soit dans la barre d'icônes de l'application. Celle-ci s'avère être aussi peu utilisable : d'une part parce qu'elle se trouve

directement sous les menus et le lien avec la fenêtre active est loin d'être évident, d'autre part parce qu'elle se contente de proposer sous forme d'icônes, de simples raccourcis à certaines commandes (ouverture, sauvegarde des documents,...). Elle sera donc déplacée directement dans la fenêtre de ***l'univers géométrique*** (voir documents en annexe), ce qui permettra d'y rajouter des accès directs aux paramètres précédents (par des curseurs, des listes déroulantes,...). Cette barre d'icônes pourra être utilisée comme indicateur d'états (par exemple quelle est la tâche active ou la projection choisie) et permettre d'accéder au paramétrage par défaut des différentes commandes (par exemple, double-cliquer sur l'icône de la tâche *'Création d'un point'* pour modifier l'aspect par défaut des futurs points construits). D'autres modifications du même ordre viendront petit à petit améliorer l'accessibilité de l'application.

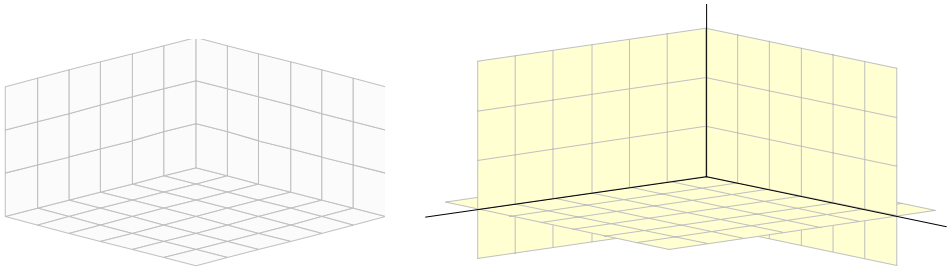
Au niveau des fonctionnalités, il est possible de copier l'intégralité d'une construction géométrique pour la coller sous la forme d'un dessin dans une autre application comme WORD, PAINT,... L'image copié est bien entendu statique, il ne s'agit pas d'un transfert OLE (pour ceux qui ne connaissent pas, ce transfert permet de modifier directement les données d'une application insérées dans une autre, cf. WORD 6 et l'éditeur de dessins). Cette fonctionnalité permet donc, outre l'impression des documents, d'insérer une construction dans un document secondaire (fiche d'exercice pour les élèves par exemple).

La représentation de l'univers géométrique, des objets et des rétroactions.

D'une manière générale, l'accord s'est fait concernant l'importance de la visualisation d'une construction géométrique et plus particulièrement du repérage de la position des objets les uns par rapport aux autres. L'accent doit donc être mis aussi bien sur les outils mis à la disposition de l'utilisateur pour faciliter cette lecture que sur les représentations visuelles des objets géométriques, et rétroactions utilisées (pour le déplacement, les projections des coordonnées, les intersections temporaires,...).

L'univers géométrique.

Au niveau de la représentation de l'univers, un moyen dont on dispose pour aider à ce repérage est l'utilisation d'éléments contextuels. Dans le cas de la géométrie du solide, la présence de ce solide est une aide suffisante à la visualisation des objets. Mais dans d'autres cas (et notamment pour la création et le déplacement des points libres), il faut trouver d'autres solutions, d'autres ***éléments visuels de compréhension***. Le premier est les ***cloisons*** : représenter l'espace (ou du moins une de ses portions) par trois plans perpendiculaires. Cela permet de disposer d'un support visuel pour la création et la visualisation des objets.



Cependant, il y avait ambiguïté dans la perception que l'on avait de l'orientation des cloisons, notamment dans le cadre d'une projection cavalière (cf. la première figure ci-dessus : *"Est-ce qu'on voit l'intérieur ou l'extérieur des cloisons"*). Jouer sur les couleurs ou le style des traits des cloisons (pointillé/plain) n'est pas suffisant. Pour essayer de résoudre ce problème, il a été

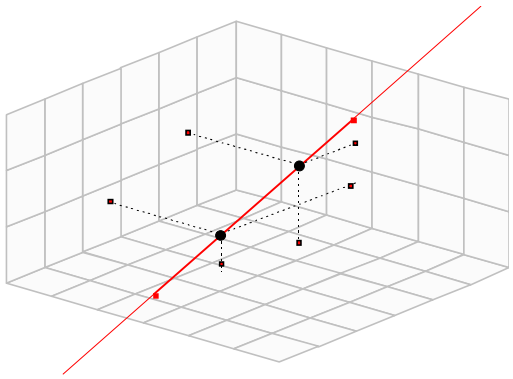
suggéré de ne pas borner les cloisons à celles mitoyennes mais de les poursuivre légèrement sur le derrière. De plus, la matérialisation des trois 'axes' peut avoir son utilité (voir la deuxième figure ci-dessus). A l'expérience, il s'avère que l'ambiguïté n'est toujours pas levée lorsqu'on observe la partie centrale des cloisons. D'autres méthodes doivent être alors envisagées : par exemple placer un objet comme une *'chaise'* dans la partie centrale ou un *'pécheur'* sur le bord de la cloison horizontale permettrait de briser la symétrie, source d'ambiguïté, et d'indiquer clairement si on observe l'intérieur ou l'extérieur des cloisons.

Les objets géométriques.

L'aspect des objets a amené une discussion sur leur sémantique. Actuellement, un certain nombre de formes disponibles est proposé au choix de l'utilisateur, sans aucune restriction (point carré, rond, en forme de croix ; droite d'épaisseur différente,...). Une remarque a été faite quant à l'intérêt d'associer certaines de ces formes à une sémantique particulière des objets. Par exemple, associer le trait pointillé pour les arêtes cachées des solides, les traits pleins pour les arêtes visibles (cf. le cube ou cette représentation est implantée) ; éclaircir la couleur d'une droite pour représenter ses portions se trouvant derrière une cloison ; associer la forme carrée aux points libres, la forme ronde aux points intersections,...

Les rétroactions.

Enfin, quelques remarques ont été faites sur les rétroactions. J'entend par ***rétroactions*** toutes les informations visuelles temporaires permettant de comprendre une action en cours. Cela désigne évidemment la forme du curseur, les messages d'aide,.. mais aussi, et surtout, tous ce qui aide à la visualisation d'une construction géométrique mais qui n'a pas forcement d'existence à un niveau conceptuel (par exemple, les projections des points manipulés sur les cloisons). On a pu noter certaines améliorations de ce côté, comme par exemple la matérialisation des intersections des droites avec les cloisons et des parties se trouvant derrière, cf. figure ci-contre.



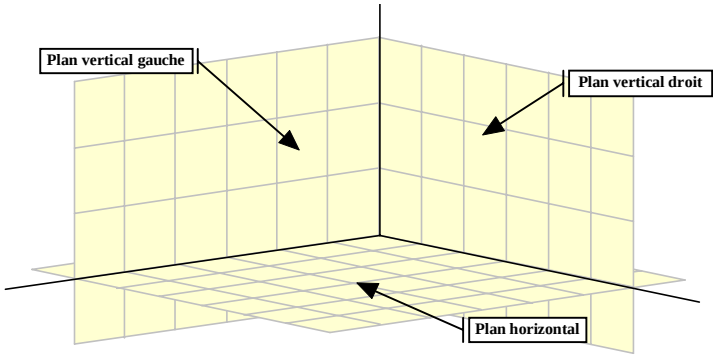
D'autres par contre manquent cruellement. C'est la cas par exemple de la matérialisation des intersections (non explicitement construites) des plans entre eux, des plans avec des droites, des plans et des cloisons,...

L'aspect provisoire (projection des points sur les cloisons lors de leur construction/déplacement) ou définitif (intersection des cloisons et des droites) de certains de ces éléments a été mis en question. Il serait en effet intéressant sinon indispensable, avant la construction d'un point par exemple, de pouvoir matérialiser les projections ou intersection d'un certains nombres d'objets (préalablement désignés) de manière à disposer ce point relativement à ceux-ci. Cette matérialisation pourrait durer le temps de la tâche de création ou jusqu'à ce que l'utilisateur le

décide autrement en désactivant la commande.

Création et manipulation des points libres.

La construction et le déplacement des points libres continue à poser problème, du fait de la perte d'une dimension lors de la projection sur l'écran de l'ordinateur. La première méthode implantée consistait à cliquer dans la fenêtre pour désigner l'emplacement du point supposé par défaut dans le plan horizontal (z=0). Cela introduisait un implicite qui n'était pas forcément aisément compréhensible. De plus, on ne pouvait pas le placer directement à une position précise, il fallait activer la tâche de déplacement et le bouger. La méthode que l'on propose maintenant permet une création plus intuitive et plus maniable. Dans un premier temps, on désigne à la souris le plan de base sur lequel va être posé le point. Cela nécessite alors l'utilisation des cloisons comme support visuel. Les trois axes symbolisent une division de l'espace de l'écran (en plans horizontal, vertical gauche et vertical droit, voir figure ci-contre) permettant la désignation non ambiguë du plan de base. On peut alors déplacer le point, de la même manière que dans la tâche de déplacement : soit dans un plan parallèle au plan de base, soit sur une droite perpendiculaire à ce plan. Actuellement, seul la désignation du plancher horizontal est possible, le déplacement correspondant à modifier la position horizontal du point et sa hauteur.



Nous nous sommes posé alors la question du comportement des cloisons vis-à-vis d'un déplacement vers l'arrière de l'une d'entre elles : la cloisons doit-elle bloquer le déplacement ? Doit-elle se modifier dynamiquement de manière à englober la nouvelle position du point ou ne pas réagir du tout ? Actuellement, il n'y a pas de limitation du déplacement, cette question reste à l'étude.

Les tâches de constructions

Le choix qui a été fait pour le mode de construction des objets est ce qu'on désigne par **action/sélection** : on choisit d'abord l'opération à effectuer puis les objets nécessaires à cette opération. Cette méthode est, à notre sens, la plus efficace car elle permet facilement de fournir une aide sur la tâche en cours (texte explicatif, curseur adapté à l'étape actuelle de la tâche,...) et respecte plus une démarche d'apprentissage ("Qu'est-ce que je peux construire ? Quels sont les objets dont j'ai besoin pour construire..."). Dans la mesure du possible, on a essayé de respecter ce choix : toutes les constructions se font donc par choix de la commande de construction (par les menus ou les icônes) puis par désignation successive des objets cibles de la construction (par exemple, pour construire un segment, on désigne l'un après l'autre les deux points qui constitueront ses extrémités). A tout moment, il est possible d'annuler la tâche en cours en utilisant la touche ESC. Cela détruit tous les éléments intermédiaires éventuellement construits ou dessinés (notamment les rétroactions visuelles symbolisant l'objet en cours de construction)

et retourne à la tâche de base sans modification de l'état de la figure. La ***tâche de base*** (au sens de CALQUES) permet la modification de l'univers géométrique (choix des éléments visuels, de la perspective,...), des propriétés des objets (aspects, nom,...) ainsi que le déplacements des points libres.

Il est important de fournir à l'utilisateur à la fois une explication succincte, claire et immédiate sur la tâche en cours (définie comme l'***aide contextuelle***) et une description plus complète à la demande (***aide en ligne***). Le rôle de la barre de statut (la zone de texte dynamique située en bas de la fenêtre de l'application) est justement de fournir contextuellement ces explications réduites sur les différentes phases d'une tâche de construction. Quand à l'aide en ligne (c-a-d dire l'aide hypertexte de Windows), elle n'est pas encore réalisée.

Le type des objets pouvant être désignés durant la tâche et le rôle des objets désignés dans la construction doit clairement être expliqué à l'utilisateur pour éviter les ambiguïtés. Par exemple, le cube de ***Calques 3D***, se défini par trois points : les deux premiers désignent une arête, le troisième sert à définir le plan sur lequelva être construit le cube et son orientation dessus/dessous par rapport à ce plan. Ce choix, imposé par le besoin d'avoir suffisamment d'informations pour permettre la création du cube, n'est pas si naturelle que cela, d'autant plus qu'il introduit certains implicites non évidents (cf. l'orientation du cube par produit vectoriel!!!). Il faudra conduire une discussion sur ce point pour définir précisément les différentes tâches de constructions (c-a-d l'objet résultat de la construction, le type et le rôle des objets cibles, ainsi que leur ordre de désignation,...)

Le vocabulaire.

La question des aides nous a amené au problème des termes utilisés dans l'application. Par exemple, j'utilise les termes de projection parallèle et perspective qui proviennent du monde de l'informatique graphique pour désigner les termes de perspectives cavalière et fuyante utilisés dans l'enseignement. Tous les textes utilisés dans l'application proviennent essentiellement de mes propres choix et doivent donc être éventuellement modifiés. A plus long terme, cela signifie aussi que cette partie de l'application doit d'une façon ou d'une autre être accessible à la paramétrisation par l'enseignant utilisateur s'il veut pouvoir correctement s'approprier le logiciel.

C'est pour cette raison que je fournis en annexe au compte-rendu la liste de tous ces textes (messages d'aide, d'erreur, nom des menus,...) de manière à ce que vous les commentiez et les modifiiez au besoin.

De la même façon, un glossaire explicitant tous les termes utilisés dans le cadre de ***Calques 3D*** (***tâche de base***, ***univers géométriques***,...) devrait voir bientôt le jour.

Prochaine réunion

Nous consacrerons quelques minutes lors de la prochaine réunion au recueil de vos remarques sur la maquette (mais il est toujours possible de me joindre par téléphone ou par mail). Mais cette réunion sera surtout prétexte à l'étude du formalisme des ***contextes d'utilisation*** présentés lors du dernier compte-rendu. Nous essayerons de définir ensemble un certain nombre de ces contextes pour des activités bien précises (l'exemple fourni concernait la construction d'un plan intersection d'un cube) et d'en évaluer le formalisme en tant qu'utilisateur du logiciel et son utilisation en tant que spécification pour le développement. Des documents préparatoires vous parviendront sous peu.

En ce qui concerne les aides fournies à l'utilisateur, il serait intéressant que, sur la base de votre utilisation de ***Calques 3D*** et des textes joints en annexe, vous puissiez vous-même rédiger l'aide en ligne de chacune des tâches de l'application (modifier les propriétés de l'univers, construire une droite,...).

La prochaine réunion est prévue pour le

Mardi 25 Mars 1997 de 15h à 18h au CRIN, Bâtiment LORIA, Salle B011 (Rez-de-chaussée)

A.3 Compte-rendu n° 4 : réunion du 25/03/97

Présents

Mme Françoise JEAN, Mme Christine MANCIAUX, Mme Marie-Hélène MUNIER, Mme Agnès VOLPI, Mme Monique GRANDBASTIEN, Mme Josette MORINET-LAMBERT, Mme Marilyne MACRELLE, M. Bernard PARZYSZ, M. Bernard PIERROT, M. Nicolas VAN LABEKE.

Pièces Jointes

- les nouvelles versions des fiches *contextes d'utilisation*
- la contribution de Bernard Pierrot à la rédaction de nouveaux contextes.

Compte-rendu de la séance

Etat de la maquette : problèmes abordés et modifications à apporter

Vocabulaire dans CALQUES 3D (en cours)

Certains termes utilisés doivent être fixés. Par exemple :

- définir le cube *fil de fer* par rapport au *cube opaque*,
- utiliser *projetantes des points* plutôt que *visualisation de ses coordonnées*,
- utiliser *perspective parallèle* plutôt que *perspective cavalière*,...

Les commentaires sur les documents fournis avec le dernier compte-rendu ont été pris en compte. Un glossaire reprenant tous les termes employés est en cours de rédaction.

Modalité de construction du cube (à examiner)

Le cube se construit en désignant trois points : les deux premiers désignent une arête , le troisième permet de définir le plan sur lequel sera posé le cube et son orientation dessus/dessous par rapport à ce plan (par produit vectoriel avec l'arête). Or, en déplaçant l'un des points, il se peut que le cube change d'orientation sans raison apparente (en fait lorsque l'on modifie la position relative des points, c'est-à-dire le produit vectoriel). Il faut donc revoir la modalité de construction du cube, notamment le statut du troisième point. Une solution (cf. figure ci-dessous) serait de construire le cube en commençant par désigner deux points constituant une arête du cube. A ce moment apparaîtraient deux cercles, centrés sur chacun des points, dans les plans perpendiculaires au segment et de rayon la longueur du segment (figure A.1). Ces cercles, ainsi que le segment matérialisant l'arête, ne sont que des éléments de rétroactions, et non des objets géométriques. Il suffirait alors de désigner un des cercles pour créer le troisième point à l'endroit pointé (figure A.2). Ce troisième point serait alors contraint par son appartenance au cercle utilisé. La face du cube ainsi définie pourrait être matérialisée et le cercle non utilisé disparaîtrait. L'orientation finale du cube pourrait être définie soit arbitrairement, soit par l'utilisateur en désignant un quatrième point sur le cercle, à angle droite par rapport à son centre.

Ce mode de construction permet d'éliminer l'ambiguïté d'orientation du cube et le troisième point appartiendrait effectivement au cube. Le déplacement du cube se ferait alors, soit en modifiant l'arête initiale, soit en déplaçant le troisième point sur le cercle que l'on ferait réapparaître pour l'occasion (figure A.3).

Gestion de l'espace d'affichage (mise à jour)

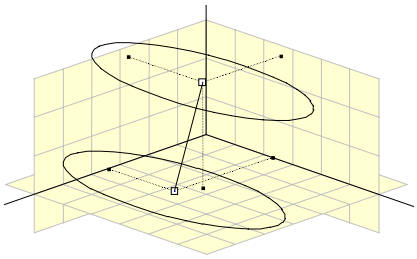


Figure A.1

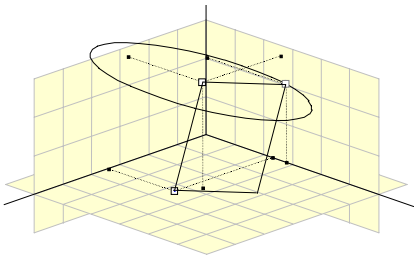


Figure A.2

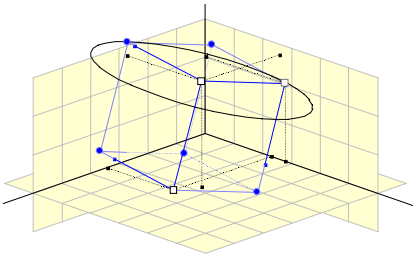


Figure A.3

Le *référentiel* apparaissait toujours centré dans la fenêtre d'affichage, quelle que soit la disposition du dessin. Afin de mieux gérer la visibilité de l'espace d'affichage, nous avons doté le logiciel d'une fonction de translation de l'ensemble *référentiel* et dessin.

Continuité visuelle entre référentiel cloison et plancher (mise à jour)

Dans l'univers cloisonné, le centre du *référentiel* est placé logiquement à l'intersection des trois cloisons. Dans le cas du plancher, le *référentiel* était centré sur l'objet graphique représentant ce plancher. Or le changement de *référentiel* se traduisait visuellement par une impression de déplacement des objets géométriques dans l'univers lors du passage entre cloisons et plancher. Nous avons modifié l'affichage du plancher de manière à conserver une continuité visuelle : en éliminant simplement les deux cloisons verticales, dessin et *référentiel* restent à la même place.

Construction d'objets géométriques temporairement inexistants (mise à jour)

Ce problème, soulevé par les logiciels de géométrie dynamique dans le plan, se retrouve dans l'espace : quelle stratégie faut-il adopter pour la construction d'objets géométriques n'ayant pas d'existence à un instant donné du fait des particularités du dessin (objets *inexistants*) ? Un exemple : l'intersection de deux segments qui ne se coupent pas. La stratégie actuelle de CALQUES 3D est qu'un tel objet doit exister effectivement au moment de sa création pour qu'il puisse être *matérialisé*[27]. En d'autres termes, l'intersection des deux segments précédents ne peut être matérialisée que s'ils se coupent lorsque l'utilisateur les désigne. Par contre, une fois créé, l'objet peut être *inexistant*[28] si ses conditions d'existence ne sont pas respectées : il disparaît momentanément à l'interface (ainsi que ses descendants) sans être détruit et peut réapparaître lors de modifications ultérieures de la figure. Le problème porte alors sur les messages d'erreur.

27. Il y a un ici problème de vocabulaire : peut-on parler de construire un objet qui n'existe pas ? Je propose plutôt le terme *matérialiser*.

28. Même remarque pour *existant* : je suggère le terme de *virtuel*.

Lorsqu'un objet devient *inexistant*, l'utilisateur peut avoir des explications par l'*historique* de la construction (double-clic sur la ligne correspondant à l'objet inexistant dans la fenêtre de l'historique). Dans le cas de l'intersection des deux segments, le message fourni est celui-ci :

Les deux segments ne se coupent pas, le point est donc invisible. Déplacer les segments pour le faire réapparaître.

Or, lors de la matérialisation d'une intersection, si elle est inexistante, le même message apparaît alors que le point d'intersection n'est pas créé. Nous avons modifié en conséquence le message d'erreur :

Les deux segments ne se coupent pas, le point est donc invisible. Déplacer les segments pour que l'intersection puisse être matérialisée.

Nous maintenons la règle générale suivante : *un objet inexistant dans un dessin ne permet pas la matérialisation (construction) de cet objet dans la figure correspondante.*

Visualisation du plan (mise à jour)

Suite à des remarques, nous proposons une nouvelle représentation visuelle du plan : le plan est représenté par le plus petit rectangle englobant l'ensemble des points déclarés comme appartenant au plan (avec une certaine marge, de manière à ce que l'on ne confonde pas le plan avec un polygone) et tel que l'un de ses côtés soit parallèle au plan horizontal. Cette propriété permet en effet d'avoir les deux autres côtés qui représentent la *ligne de plus grande pente* du plan, concept souvent utilisé en enseignement.

Visualisation des intersections de plans

Avec les cloisons (en cours)

A l'instar des droites, le plan doit avoir le même comportement visuel que les droites vis-à-vis des cloisons. Il faut donc matérialiser les intersections du plan avec les cloisons et estomper les parties du plan se trouvant derrière les cloisons (cf. figure ci-contre).

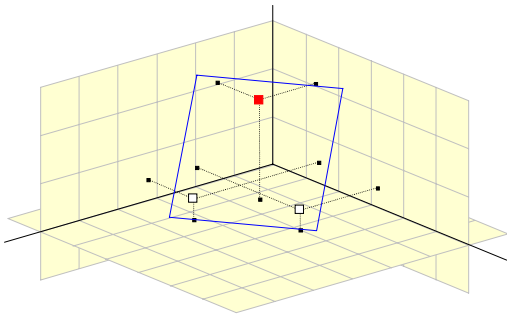


Figure A.4

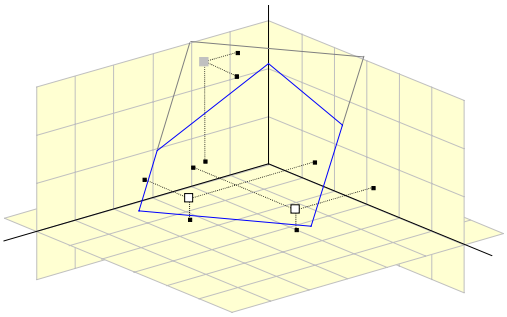


Figure A.5

Entre deux plans (à examiner)

Traditionnellement, les deux plans sont représentés de manière à ce que leurs contours respectifs se coupent effectivement. Avec notre représentation, la taille des rectangles englobants ne dépend que des points appartenant au plan, et il se peut que les bordures ne se coupent pas effectivement.

Problèmes de lisibilité des constructions géométriques

Les principales difficultés dans l'espace sont liées à la compréhension des positions relative des objets, c'est-à-dire principalement à la mise en évidence des relations *devant/derrière*, *des-*

sus/dessous, *dedans/dehors* et *appartient/n'appartient pas* existant entre eux.

Les projetantes des points sur les cloisons permettent de situer un point. Cependant, elles n'apparaissent que lors du déplacement d'un point. Il a été donc demandé de pouvoir les visualiser temporairement à l'initiative de l'utilisateur. Nous avons ajouté une commande (ci-contre l'icône correspondante dans la fenêtre d'affichage) qui permet la désignation des points dont on souhaite la visualisation des projetantes. Celles-ci restent visible jusqu'à désactivation de la commande. *(mis à jour)*

La forme d'un objet peut aider à son identification. Nous cherchons donc à identifier des catégories de points : par exemple *point créé* (par l'utilisateur), *point sur objet* (droite, plan,...), *point intersection d'objets* (droite/droite, droite/plan,...), *point intersection avec cloisons*,... Cela nous permettrait d'associer à chaque catégorie une représentation visuelle cohérente. Mais cela pose un certain nombre de questions auxquelles il faudra proposer des solutions, en particulier au niveau de l'interface : liberté totale laissée à l'utilisateur ? Paramétrisation globale par l'utilisateur ou l'enseignant ? Valeurs par défaut ? *(à examiner)*

Des incohérences dans la traduction visuelle des propriétés de certains objets ont été mises en évidence. En effet, les parties de droites se trouvant derrière les cloisons sont représentée dans le même style de trait mais avec une atténuation de la couleur (ou en grisé si la résolution de l'écran ne permet pas ce dégradé). Les arêtes cachées d'un cube sont représentées en trait pointillé, avec la couleur attribuée au cube. Or la conjonction des deux propriétés (lorsqu'une partie du cube se trouve derrière une cloison) ne produit pas le résultat attendu. De même, l'atténuation de la couleur d'un point situé derrière les cloisons n'est pas effective. *(mis à jour)*

Enfin, le plus important reste le problème de la visualisation de l'éloignement des objets. Une première tentative dans ce sens, fréquemment utilisée dans l'enseignement, a consisté à *gommer* légèrement, au niveau de leur intersection visuelle, la partie d'une droite se trouvant sous une autre (voir figure ci-dessous).

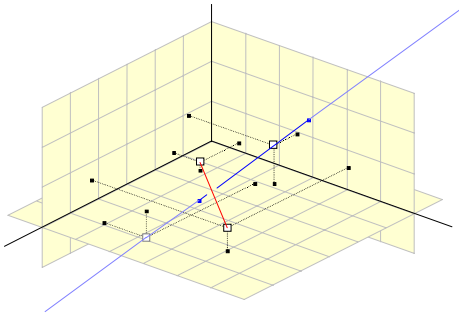


Figure A.6

Beaucoup de ces méthodes indispensables à la lisibilité d'une figure géométrique sont cependant coûteuses en temps de calcul : il y a un compromis à faire entre une visualisation suffisamment explicite et une réaction aux actions de l'utilisateur suffisamment rapide.

Modifications des fiches contextes d'utilisation.

Objectif de ces fiches : ce sont des documents de recherche, internes au groupe de travail, considérés comme un cadre d'expression commun entre différentes personnes impliquées dans la réalisation d'un logiciel pédagogique. Nous avons étudié les trois exemples de *contextes* que nous

vous avions fait parvenir dans le dernier courrier et les avons rédigés à nouveau en commun, de manière à expliquer clairement ce que nous attendons de vous au travers de ces documents. Les remarques suivantes sont ressorties de ce travail :

- Il faut ajouter le *niveau initial* requis pour la réalisation de ces activités, le champ *"objectifs"* n'étant pas suffisant pour cibler le public. Cela nous amène à supposer qu'il faudrait certainement rajouter des informations sur les *acquis* attendus (*les attentes*), les moyens disponibles pour l'*évaluation*,...
- Le rôle des *objectifs* n'est pas assez précis : il faut pouvoir faire la distinction entre ceux qui relèvent de la pédagogie et ceux qui relèvent plus de l'utilisation ou de la prise en main du logiciel.
- Les exemples d'activités du contexte "Création et manipulation d'un point libre" nous amènent à dire qu'il y a une différence entre *placer un point* ("soit P1 un point libre, le placer à tel endroit") et *construire un point* ("Construire le point P1 tel que...").
- L'activité 2 demande à déplacer un point en se référant aux projetantes des 3 points préalablement positionnés dans l'espace et FIXÉS, c'est-à-dire non déplaçables par l'utilisateur. Il faudrait donc rajouter le statut *punaisé* aux points. De même, la référence aux cloisons par les termes gauche, droite et horizontale n'est pas adéquate. Il faut leur donner des noms de telle manière qu'on puisse facilement les identifier et s'y référer. Ces noms ne doivent pas prêter à confusion avec des noms pouvant être donnés à des objets géométriques puisque les cloisons n'en sont pas. Je propose pour le moment les noms **Cg**, **Cd** et **Ch** pour les cloisons respectivement gauche, droite et horizontale.

Prochaine réunion

L'objectif des prochaines réunions (il ne nous reste plus beaucoup de temps d'ici les vacances d'été) est de clarifier la notion de fiche de contextes d'utilisation et d'en rédiger ensemble un nombre suffisant pour en tirer des renseignements utiles à la poursuite de notre travail. Je vous demanderais donc, en tenant compte des remarques faites lors de la réunion et de vos propres besoins, d'en rédiger un ou deux concernant des activités de votre choix. Si vous avez le temps de les faire avant la prochaine réunion, vous pouvez me les faire parvenir et je les diffuserai à l'ensemble des personnes du groupe, ce qui nous permettra d'en discuter à la réunion.

La prochaine réunion est prévue pour le

Mardi 29 Avril 1997 de 14h à 17h au CRIN,
Bâtiment LORIA, Salle B011 (Rez-de-chaussée)

A.4 Compte-rendu n° 5 : réunion du 29/04/97

Présents

Mme Françoise JEAN, Mme Christine MANCIAUX, Mme Agnès VOLPI, Mme Monique GRANDBASTIEN, Mlle Marilyne MACRELLE, Mme Josette MORINET-LAMBERT, M. Jean-Claude DEMOLY, M. Nicolas VAN LABEKE.

Pièces Jointes

- la contribution de Françoise JEAN et Marie-Hélène MUNIER à la rédaction de nouveaux contextes.
- la contribution de Jean-Claude DEMOLY.

A noter

Nous arrivons bientôt à la fin de nos réunions car, pour cause de baccalauréat et de fin d'année scolaire, il va être difficile de nous réunir (prochaine réunion en Mai et peut-être encore une en Juin). Par contre, je suis à votre disposition en dehors des réunions prévues si vous désirez faire une séance de travail en tête-à-tête (revoir certains aspects de la maquette, travailler sur les *contextes d'utilisation*, ...).

Compte-rendu de la séance

Etat de la maquette : problèmes abordés et modifications à apporter

Modalité de construction du cube (mise à jour)

La proposition que nous avions faite dans le dernier compte-rendu permettait de réduire la charge cognitive (élimination du troisième point qui n'appartenait pas effectivement au cube) mais présentait encore quelques lourdeurs au niveau ergonomique. Lors de la construction, on ne propose plus le choix d'un des deux cercles pour placer le troisième point mais uniquement celui centré sur le deuxième point de l'arête initiale.

La procédure de construction actuelle se décompose donc comme suit :

- désignation de l'arête initiale (par deux point existants)
- apparition temporaire d'un cercle dans un plan orthogonal à l'arête, centré sur sa deuxième extrémité et de rayon la longueur de l'arête
- désignation d'un point appartenant à ce cercle (construction automatique de la première face du cube)
- détermination de l'orientation du cube (en désignant un point se trouvant à 90° de part ou d'autre du 3ème point)

Il est à noter que ce cercle temporaire, utilisé aussi bien dans la construction que dans la modification du cube, n'est pas un objet géométrique qui existe réellement mais une *rétroaction* (un *élément visuel de compréhension*) et ne peut donc pas être utilisé ultérieurement. Cette procédure induit certaines propriétés peut-être non désirables. En particulier, la manipulation des différents points mobiles ne rend pas la même présentation du cube : déplacer l'une des deux extrémités de l'arête initiale introduit une déformation du cube (modification de la taille de l'arête) tandis que celle du troisième point (sur le cercle) introduit une rotation du cube autour de l'arête initiale. Nous nous sommes aussi posé la question de la complexité de cette

tâche de construction et de son adéquation. Il nous a finalement semblé que celle-ci avait un intérêt pédagogique non négligeable en présentant, étape par étape, les propriétés intrinsèques d'un cube (construction de l'arête, évolution dans le plan - la face, évolution dans l'espace - le cube). Elle doit cependant être testée pour en évaluer l'intérêt et sera donc mise en œuvre dans la prochaine version du logiciel que je vous transmettrais.

Sémantique visuelle des objets (à examiner)

Nous avions déjà suggéré d'associé à la sémantique des objets une forme ou une couleur particulière, notamment en ce qui concerne les points et les droites. Cette association demande donc une catégorisation des objets, par exemple :

- *point mobile* (déplacement libre dans l'espace)
- *point sur* (déplacement contraint sur un lieu : droite, plan , ...)
- *point calculé* (pas de déplacement possible : cf. les points d'intersection)
- *point de rétroaction* (éléments visuels pouvant disparaître à la demande : intersection droites/cloisons, projetés des points sur les cloisons, ...)

J'avais suggéré d'associer une forme à la sémantique (rond pour les points mobiles, croix pour les points intersections, ...) et une couleur pour l'instanciation de la propriété (par exemple un point sur un plan à la même couleur que le plan). Mais cela pose quelques problèmes : quelle couleur pour un point intersection d'une droite et d'un plan ? Cette catégorisation ne peux pas être faite uniquement par l'informaticien : les propositions doivent venir des utilisateurs (enseignants). Nous en rediscuterons donc à la prochaine réunion mais vous pouvez déjà y réfléchir. Dans tous les cas, il faut toute liberté à l'utilisateur pour définir le codage visuel avec des valeurs par défaut.

Visualisation du plan (à examiner)

Il a été remarqué que la forme parallélogramme donné par défaut au plan ne se justifie pas toujours : dans un référentiel de type cloison, elle prête même à confusion en ne donnant pas une restitution correcte des intersections visuelles du plan avec les cloisons. Or, on ne peut pas non plus se limiter à une visualisation par ses intersections sur les cloisons ou à une réduction sur les objets le permettant (section d'un cube par exemple) car celles-ci peuvent ne pas exister. Il faudra donc revoir ensemble ce problème de visualisation, ainsi que d'autres impliquant le plan : l'intersection visuelle de deux plans (visualisation de la droite commune et intersection effective des bordures des plans ?), l'appartenance d'un point à un plan (projetante du point sur les bordures du plan ou ligne de plus grande pente passant par ce point ?), ...

Organisation des tâches de construction (en cours)

Lorsque l'on annule ou quitte une tâche de construction quelconque, on revient sous la tâche dite *par défaut*. Dans cet état, on propose le déplacement des points mobiles (déformation de la construction), la modification des propriétés visuelles des objets (forme, couleur, ...) et de l'univers (référentiel, point de vue, ...). Cela sous-entend deux catégories de fonctions : celles pour la visualisation et celles pour la modification. Si on cible comme objectif principal du logiciel l'observation, on va donc favoriser les fonctions d'observation dans la tâche par défaut et les fonctions de modification seront appelées de manière explicite et feront l'objet d'une tâche à part entière.

Nouvelles fonctionnalités

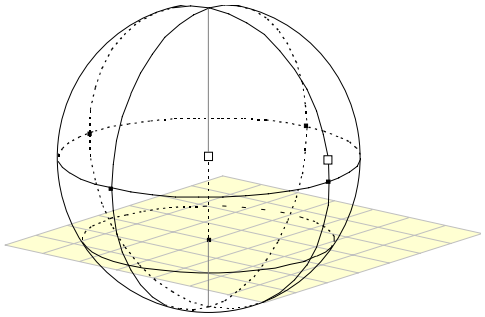
Des demandes ont été faites pour l'ajout de nouvelles fonctionnalités dans le logiciel. Il faut savoir que nous devons rendre le logiciel à la Région à la fin Juin. Ce qui signifie que nous

préférons stabiliser et renforcer le noyau existant plutôt que de développer de nouveaux outils. Cependant, certaines demandent nous semblent justifiées et indispensables et devront mises en œuvre dans les prochaines versions du logiciel, avec quelques réserves sur leur état d'avancement.

Les calques (à examiner)

La notion de calques dans ***Calques 3D*** est une extension à l'espace des calques de `Calques 2`. Il s'agit d'un *filtre* permettant l'extraction d'un sous ensemble d'objets géométriques appartenant à une construction et son affichage dans une fenêtre séparée. L'intérêt de cet outil est de fournir différents points de vue sur une même construction. Par exemple, dans l'activité décrite dans les documents joints en annexe (*"des triangles dans un cube"*), on pourrait extraire de la construction le triangle B'IC', le mettre dans un calque et l'observer sous des angles différents, voire même se mettre dans le plan de manière à l'observer en vraie grandeur. Lors de la prochaine réunion, il nous faudra définir précisément les modalités de ces filtres. Je vous présenterais une première ébauche des calques tels que nous les voyons.

La sphère (à examiner)



Il est vrai que le cube, seul objet volumique, n'est pas suffisant pour montrer les possibilité du logiciel. Il a été suggéré d'y rajouter *la sphère*, à partir de laquelle de nombreux exercices sont possibles. Sa réalisation n'est pas coûteuse mais rajouter les fonctions s'appliquant sur cet objet l'est (intersections sphère/plan ou sphère/droite, ...). Pour le moment, nous proposons de ne se prendre en compte que le cas de figure présenté par Jean-Claude DEMOLY : la visualisation de l'intersection d'une sphère et d'un plan perpendiculaire à l'axe. Cette activité peut-être simulée en choisissant le référentiel *plancher* et en matérialisant l'intersection visuelle d'une sphère et du plancher. Lors de la réunion, nous avons mis en évidence les éléments visuels importants à présenter : l'axe polaire, l'équateur, un ou plusieurs méridiens, ... Il nous restera à préciser la définition exacte de ces éléments (l'équateur est-il parallèle au plan horizontal ou correspond-il au grand cercle passant par le deuxième point définissant la sphère ? Les méridiens sont-ils les grands cercles parallèles aux plans gauche et frontal ou l'un passe par le point et l'autre y est perpendiculaire ? ...).

Modifications des fiches contextes d'utilisation.

Le cadre proposé pour la description de *contextes d'utilisation* semble vous convenir : il y a cependant quelques problèmes pour leur rédaction. Voici les remarques émises lors de la réunion.

Processus de rédaction.

L'ordre des champs présentés des *contextes* induit en erreur car il résulte plus d'une logique informatique que pédagogique. Leur rôle est en effet de donner une idée des informations représentatives d'un *contexte* à extraire et la rédaction d'un champ doit découler de la rédaction

des précédents. Par exemple, il est logique de déterminer les *difficultés* d'une activité avant de proposer des *aides* adaptées. Or, les *paramètres d'environnement* (choix du référentiel, des rétroactions, ...) peuvent être aussi des aides à proposer pour pallier les difficultés. La démarche que nous vous proposons d'essayer est la suivante : à partir d'une *activité* donnée (*"des triangles dans un cube"*, cf. document joint), en extraire l'*objectif* pédagogique (*"extraire une figure plane d'un solide [...]"*). La description de cette activité doit permettre de déterminer les *classes d'objets* initiales de l'activité (*cube, segment, ...*) et d'en préciser, le cas échéant, les *facettes* nécessaires (c'est-à-dire les présentations des objets), ou plutôt inadéquates pour l'activité (l'extraction de figure plane exclue la possibilité de présenter le cube comme plein). Les *instances* des objets peuvent provenir des noms des objets donnés dans l'énoncé (le cube ABCDA'B'C'D', le segment [CC'], ...). En réalisant l'activité, il est alors possible de définir quelles sont les *fonctions* nécessaires à la création des objets intermédiaires (*construction d'un segment, construction du milieu d'un segment, ...*). Pour finir, préciser les *difficultés* introduites par l'activités et les *aides* à apporter.

Difficultés de l'élèves et aides à apporter

Comme nous l'avons dit précédemment, les difficultés et les aides à apportées sont liées. Au même titre que pour *l'objectif*, la description de ces difficultés doit être la plus précise possible : éviter les difficultés 'génériques' pouvant être relevées dans chaque activité. Cette précision doit permettre de proposer immédiatement des aides adaptées à ces difficultés, quelles soient de nature 'classique' (intervention de l'enseignant, retour au cours, ...) ou informatique (modification du point de vue, changement du référentiel, rétroactions, ...). Par exemple, dans l'activité *"des triangles dans un cube"*, la difficulté *"argumentation géométrique (par aller et retour figure extraite - solide)"* est trop vague. Si elle est précisée sous la forme *"visualisation des angles et des longueurs"*, on peut immédiatement réfléchir aux outils à mettre en œuvre pour aider l'élève à voir : extraction automatique d'une figure dans un calque (*"rétroaction : possibilité de ne voir que le triangle"*), modification du point de vue de manière à se mettre dans le plan correspondant (*"point de vue : se placer dans [...]"*).

Prochaine réunion

Je profiterais de cette réunion pour faire le point (final ?) sur la maquette : contenu, tâches de création, rétroactions, vocabulaire ... mais surtout pour faire le point sur les deux points épineux : les calques et la sphère. Un tableau synthétique présentant ma proposition de catégorisation des objets pour le codage visuel de leur sémantique vous sera diffusé. Si vous avez des remarques non encore formulées ou prises en compte concernant la maquette, n'hésitez pas à me les communiquer. Nous mettrons aussi l'accent sur la rédaction des *contextes d'utilisation*. Comme la dernière fois, je vous demanderais donc de venir avec un ou plusieurs contextes rédigés selon les indications fournies dans ce compte-rendu et les précédents.

La prochaine réunion est prévue pour le

Mardi 27 Mai 1997 de 14h à 17h au CRIN,
Bâtiment LORIA, Salle B011 (Rez-de-chaussée)

A.5 Compte-rendu n° 6 : réunion du 27/05/97

Présents

Mme Françoise JEAN, Mme Marie-Hélène MUNIER, Mme Josette MORINET-LAMBERT, M. Jean-Claude DEMOLY, M. Bernard PIERROT, M. Nicolas VAN LABEKE.

Pièces Jointes

– les contributions de Françoise JEAN et Marie-Hélène MUNIER à la rédaction de nouveaux contextes.

A noter

La prochaine réunion sera certainement la dernière. Je reste cependant disponible si vous désirez faire une séance de travail en tête-à-tête.

Compte-rendu de la séance

Etat de la maquette : problèmes abordés et modifications à apporter

Le plan (en cours)

Les problèmes de visualisation du plan ont été plus ou moins réglés. Toujours à partir d'une base rectangulaire dont un des cotés correspond à la ligne de plus grande pente du plan, la forme du plan s'adapte au référentiel utilisé : en présence du plancher par exemple, le plan est toujours en contact avec le plancher, soit (figure A.7) en matérialisant son intersection, soit en étendant le rectangle de manière à ce que le plan repose sur le plancher (figure A.8).

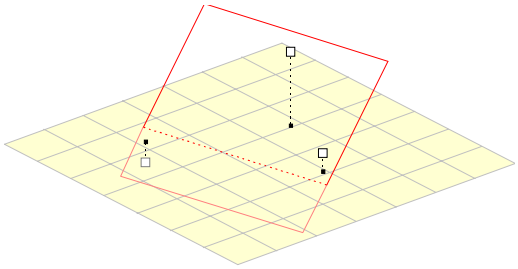


Figure A.7

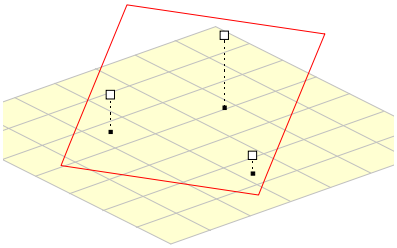


Figure A.8

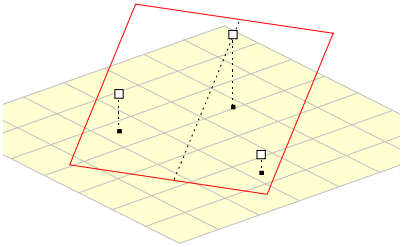


Figure A.9

De manière à permettre une visualisation des points explicitement créés comme appartenant au plan (point sur plan, point intersection plan/droite ; ...), il a été suggéré de matérialiser la ligne de plus grande pente passant par ce point (figure A.9). Comme pour l'affichage des projetantes des points ; cette rétroaction visuelle n'apparaîtrait que lorsque le point serait sélectionné ou lorsque la commande de visualisation temporaire des rétroactions est activée (cf. compte-rendu n° 4 du 24/03). L'adaptation de la représentation du plan au contexte n'est cependant pas encore satisfaisante et demande à être remaniée. Lors de la création explicite de l'intersection de deux plans (cf. compte-rendu précédent et figure ci-dessous), les rectangles englobants devront être coïncidants avec la droite d'intersection.

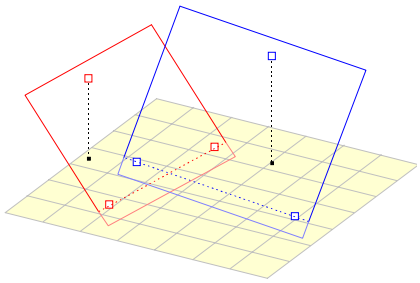


Figure A.10

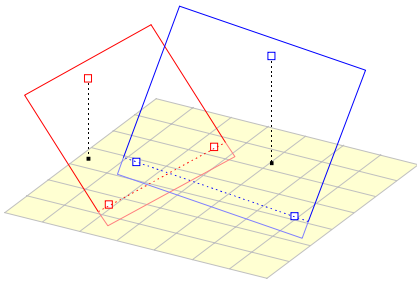


Figure A.11

Dans le contexte du référentiel *'cloison'*, le plan sera matérialisé par ses intersections avec les cloisons lorsqu'elles existent, c'est-à-dire lorsque le plan n'est pas parallèle à l'une des cloisons, et lorsqu'elles sont visibles, c'est-à-dire se trouvent dans les limites soit de la fenêtre, soit des cloisons (*à définir*).

La sphère (en cours)

Nous avons adopté les règles suivantes pour la visualisation de la sphère :

- dessin du contour apparent, de l'équateur de la sphère, du grand cercle parallèle au plan gauche et celui passant par le point définissant la sphère (figure A.12). Sont dessiné aussi les points d'intersection des différents grands cercles.
- en présence du plancher, l'intersection de la sphère et du plancher peut être aussi matérialisée.
- de la même manière que pour le plan, les points construits explicitement comme appartenant à la sphère peuvent être situés par l'affichage, soit du grand cercle passant par ce point, soit des deux rayons passant par ce point et son projeté sur l'équateur (figure A.13).

Il y a cependant risque de surcharge visuelle (même la figure A.13 ci-contre, pourtant simple, n'est pas facile à lire), en particulier avec les éléments de rétroaction. Leur affichage ou non doit donc être paramétrable ou dépendant du contexte (pendant la sélection d'un objet par exemple).

Ces éléments de représentation de la sphère et l'accès aux choix seront implantés dans la prochaine version du logiciel. Par contre, nous ne pouvons développer toutes les fonctions de constructions basées sur la sphère, la priorité étant donné aux deux les plus couramment utilisées : l'*intersection d'une sphère et d'un plan*, l'*intersection d'une sphère et d'une droite* et le *point sur la sphère*. Enfin , il a été signalé que le plancher pouvait ne pas être suffisamment grand pour

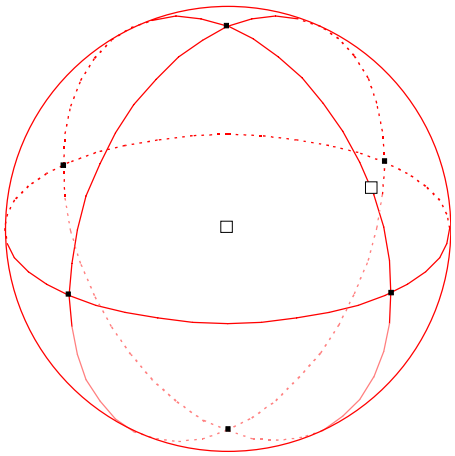


Figure A.12

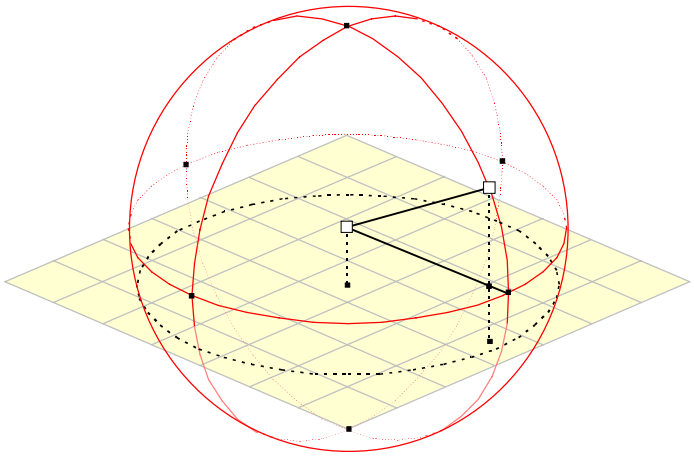


Figure A.13

'supporter' l'intégralité de la sphère. Dans un premier temps, nous y remédierons en permettant la modification à la demande de la taille du plancher (et des cloisons). A terme, cette taille pourrait être automatiquement adaptée de manière à englober un maximum d'objets construits (dans la limite de la lisibilité de l'univers bien entendu).

Les calques (en cours)

La discussion la plus importante de la réunion a portée sur les *calques* : leur définition, leur rôle, leur utilisation. Nous définissons un calque comme un filtre visuel, c'est-à-dire un mécanisme permettant d'extraire un sous-ensemble d'une construction et de le visualiser séparément de la figure initiale. Par sous-ensemble d'une figure, nous entendons un ensemble d'objets géométriques définis (droites, plans, points, ...). Cela veut dire qu'il est impossible d'extraire un bout de droite, un demi-plan, un arc de cercle, l'intersection d'un plan et d'un cube ... si ce lieu géométrique n'a pas était explicitement construit. Ce choix de fonctionnement pose un problème au niveau des objets dit composés comme le cube. Le cube est en effet constitué d'objets géométriques de base (points, segments, faces) avec des relations entre eux. Il est dans ce cas possible d'extraire l'un ou plusieurs de ces composants mais les propriétés qui les lient ne sont pas pour autant supprimées. C'est pour cette raison que les calques sont bien des filtres opérants au niveau visuel. Cela veut dire aussi qu'il n'y a pas de modification des propriétés visuelles des objets, ce qui peut sembler bizarre : les arêtes masquées du cube, représentées en pointillées, gardent cet aspect lorsqu'elles sont extraites dans un calque, même si la face du cube qui les masquait n'est pas extraite. D'une manière générale, la définition et l'utilisation des calques pose un certain nombre de problèmes conceptuels auxquels nous ne pouvant pas répondre immédiatement. Une étude plus approfondie sera menée pour définir correctement cet outil et en cerner les problèmes.

En résumé, voici le fonctionnement des calques tels qu'ils seront implantés dans la prochaine version du logiciel.

- Chaque calque est une fenêtre de visualisation d'une même figure possédant référentiel et point de vue indépendant. Le nombre de calques disponibles simultanément sera à fixer par la suite. L'un de ces calques, noté **C0**, sert d'univers de référence et contient tous les objets géométriques de la figure.
- Ne peuvent être extrais que les objets géométriques explicitement définis ou les éléments

d'un objet composé comme le cube.
- Les objets géométriques à extraire sont désignés les uns après les autres et copiés dans un calque désigné au préalable ou à chaque désignation d'objet *(à définir)*. De la même manière, il est possible de gommer ou de supprimer un objet dans un calque : *gommer* enlève l'objet du calque actif uniquement alors que *supprimer* détruit l'objet (et ses descendants) de tous les calques, même le calque de référence **C0**.
- Chaque calque étant une fenêtre de visualisation standard, il est possible de créer de nouveaux objets à partir de ceux présents dans ce calque. L'objet résultant de cette tâche est construit dans le calque de référence **C0** et est extrait dans le calque actif. Pour le visualiser dans d'autres calques, il faut l'extraire.

Autres fonctionnalités (à examiner)

En discutant sur les calques et leur utilisation pour aider à la visualisation d'une construction, deux demandes de nouvelles fonctionnalités sont apparues.

Projection frontale ou 'transversale'

Le problème est de permettre à l'élève de percevoir certaines propriétés en déplaçant le point de vue dans des positions particulières : par exemple en position frontale pour visualiser une construction (ou un sous-ensemble de cette construction) en vraie grandeur, ou dans un plan pour visualiser l'appartenance ou non d'un point à ce plan. Dans cette optique, il nous semble indispensable de dissocier ce qui relève de l'extraction d'une sous-figure (par l'intermédiaire des calques) et de la visualisation (modifier le point de vue). Nous avons recensé 2 possibilités de permettre cette modification du point de vue :

- par utilisation d'une commande de l'application (*positionnement automatique*) : on désigne l'objet cible de la visualisation et l'application se charge du déplacement du point de vue. Cela permet de palier les limites d'un pas de rotation fixe mais décharge complètement l'élève de cette activité de visualisation.
- par manipulation directe du point de vue (*positionnement manuel*) mais son inconvénient est que le pas de rotation du point de vue fixe ne permet pas toujours de se positionner selon un angle favorable.

Il est possible d'affiner cette méthode en introduisant une *"aimantation"* des objets (*positionnement semi-manuel*). Progressivement, l'élève modifie manuellement le point de vue de manière à s'approcher de la situation idéale et, lorsque les conditions d'aimantation sont remplies (proximité de l'objet cible par exemple), le pas de rotation est adapté de manière à ce qu'il y ait concordance de position. Le problème de cette méthode est qu'il peut être difficile de désigner ou de déterminer la proximité d'un objet sur lequel s'appliquera l'aimantation.

Cette dernière méthode est de loin la plus intéressante d'un point de vue pédagogique ... mais la plus difficile à mettre en œuvre. Pour le moment, seule la commande automatique sera implantée.

Le marquage des propriétés

Nous avons discuté de la possibilité que l'utilisateur puisse annoter certaines propriétés d'une construction (angles droits, égalité de longueur, ...). Ces annotations permettent à l'élève de visualiser le résultat d'une étape de réflexion sur une construction avant de passer à la suivante. Il s'agit d'informations visuelles durables, c'est-à-dire qui restent visibles même après déformation de la figure. Nous avons fait la différence entre le *crayonnage*, annotation à la liberté

de l'élève, et le *marquage*, annotation possible que si la propriété à mettre en évidence existe. Nous avons décidé que seul le marquage était intéressant pédagogiquement. Cela implique la mise en œuvre des tests élémentaires opérés par le logiciel mais introduit le risque de son utilisation abusive par l'élève (marquage systématique sans réflexion et observation du résultat). Par ordre d'importance, nous avons proposé que les vérifications automatiques soient disponibles pour les propriétés suivantes :

- angles droits
- égalité de longueur
- parallélisme
- perpendicularité

Seuls les deux premiers ont été retenus comme important pour le marquage. Il restera cependant à définir l'aspect de ce marquage. Dans la prochaine version du logiciel, nous implanterons les vérifications automatiques des propriétés. Leur éventuel marquage dépendra de la complexité de leur mise en œuvre. Il est à noter que la mise en œuvre d'une vérification de propriétés pose un problème : une résolution purement analytique n'est pas suffisante car la figure peut être un cas particulier (droites orthogonales par coïncidence). Une méthode utilisant des déductions logiques peut être coûteuse à réaliser. Un bon intermédiaire consiste à effectuer un test analytique sur un certain nombre de déformations de la figure et à vérifier la véracité de la propriété dans tous les cas. Cette méthode qui n'est pas encore parfaite (toutes les déformations essayées peuvent être des cas particuliers) présente cependant l'intérêt de fournir un contre exemple lorsque la propriété n'est pas vérifiée.

Rédaction des fiches contextes d'utilisation.

Vous trouverez joint à ce compte-rendu deux nouvelles fiches. En particulier, j'attire votre attention sur celle de Françoise JEAN (pages 6 à 9) qui nous permet de mettre en évidence l'un des objectifs que nous attendions de ce travail. A partir d'une analyse fine de l'exercice réalisé par ses élèves, Françoise JEAN a mis en évidence les erreurs commises par l'élève et a défini les aides que le logiciel pourrait apporter pour pallier ces difficultés. Il apparaît que deux types de difficultés (et donc d'aides) reviennent tout le temps : l'élève se fie au cas particulier de la figure telle qu'il la voit (déplacer le point de vue pour voir la construction sous un autre angle) et voit coplanaires des droites qui ne le sont pas (se placer dans un plan donné). Sans mener cette analyse aussi en profondeur, Françoise JEAN aurait cité spontanément ces difficultés D'un autre côté, cette construction incrémentale des contextes d'utilisation amène à une stabilisation de ceux-ci, ce qui tout naturellement conduit à une stabilisation de la maquette. Nous ne vous demandons pas d'effectuer le même travail (coûteux en temps !) mais, plus précis et nombreux seront les *contextes* dont nous disposerons, meilleure pourra être votre adhésion au logiciel.

Prochaine réunion

La prochaine réunion est prévue pour le

Jeudi 19 Juin 1997 de 14h à 17h au CRIN,
Bâtiment LORIA, Salle B011 (Rez-de-chaussée)

A.6 Compte-rendu n° 7 : réunion du 19/06/97

Présents

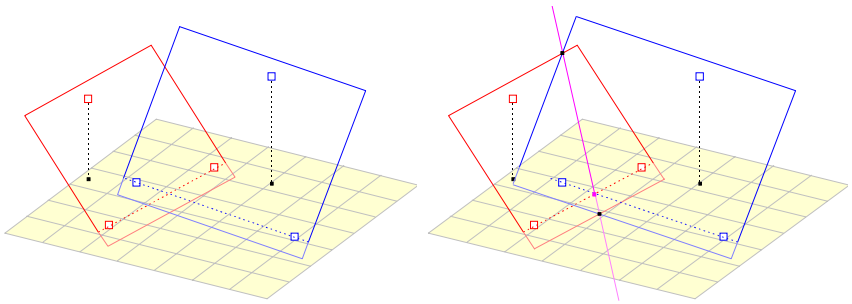
Mme Françoise JEAN, Mme Agnès VOLPI, Mme Josette MORINET-LAMBERT, Mlle Marilyne MACRELLE, M. Christian GINET, M. Alain GIORGETTI, M. Bernard PIERROT, M. Nicolas VAN LABEKE.

Compte-rendu de la séance

Etat de la maquette : problèmes abordés et modifications à apporter

Le plan (mis à jour)

Une ambiguïté s'est glissée dans le compte-rendu précédent. La matérialisation de la droite d'intersection de deux plans, ainsi que la mise en coïncidence des bordures des plans (cf. Figure 1), fait partie des rétroactions visuelles qui sont lourdes à mettre en œuvre et coûteuses en temps de calcul : il faudrait en effet déterminer les intersection de tous les plans deux à deux, de tous les couples plans/droites, droites/droites, ... C'est pourquoi je ne fais coïncider les bordures de deux plans sécants que lorsque la droite d'intersection a été explicitement créée (c'est-à-dire sur demande de l'utilisateur) et non pas dans le cas général (intersection purement visuelle).



Il a été suggéré, afin de limiter les surcharges d'information visuelle, de contraindre l'apparition des rétroactions par certains critères. Par exemple, ne matérialiser l'intersection visuelle d'un plan et du plancher que si l'angle entre les deux dépasse une valeur critique prédéfinie.

Enfin, une attention particulière est portée sur le traitement des exceptions et cas particuliers. Pour le point sur un plan par exemple, la ligne de plus grande pente est matérialisée, même dans le cas où le plan est horizontal , ce qui est un non-sens. Cette erreur, et d'autres du même ordre, ont été corrigées.

La sphère (mise à jour)

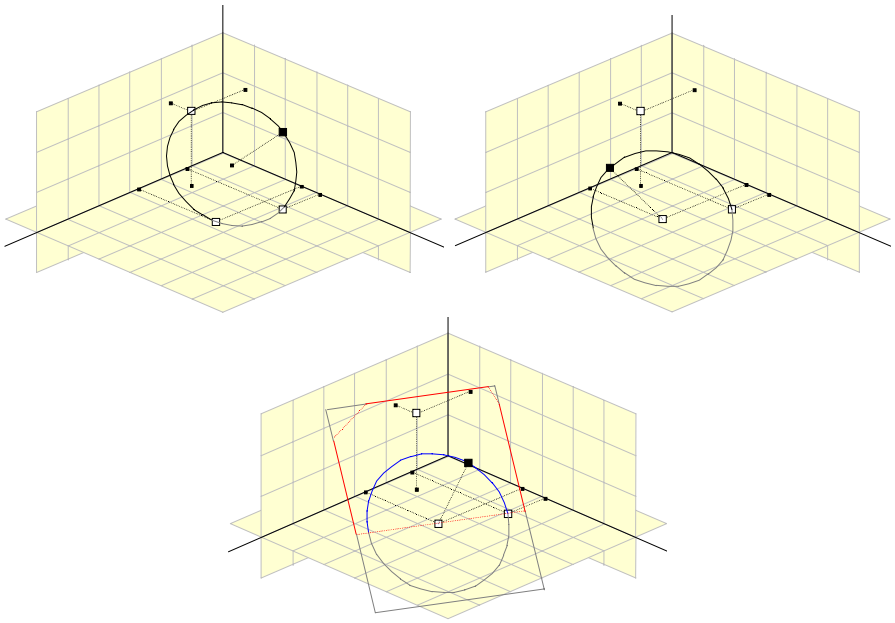
Suite à une remarque concernant le statut visuelle du cercle intersection de la sphère et du plancher horizontal, je précise qu'il s'agit bien d'une rétroaction visuelle et non d'un objet géométrique. Cela justifie donc son dessin en pointillé, de la même façon que pour toutes les rétroactions visuelles des autres objets. Les rayons et méridiens définis par les points appartenants à la sphère sont donc eux aussi dessinés en pointillé, contrairement à ce qui est représenté sur la figure 6 du compte-rendu précédent.

Il a été par ailleurs suggéré que les deux rétroactions (méridien et rayon) des points sur sphère apparaissent à la fois de manière à mettre en évidence les propriétés induites (rayon, latitude et longitude). En ce qui concerne les constructions ayant pour base la sphère, seule la création de points sur la sphère sera implantée dans la **version 1** de ***Calques 3D***. En effet, à partir de

ces points, il est possible de réaliser un grand nombre d'activités en simulant les constructions manquantes comme les intersections sphère/plan ou sphère/droite.

Le cercle (à examiner)

Nous en avons modifié la modalité de création en permettant la construction d'un cercle passant par trois points (cf. Figure 2, à gauche). Ce changement permet par exemple de simuler l'intersection d'un plan et d'une sphère en construisant le cercle à partir de points appartenant à la sphère. Il remet cependant en question la procédure initiale de création : désignation du centre, d'un point appartenant à sa circonférence et d'un troisième point désignant le plan auquel appartient ce cercle (cf. Figure 2, au centre). Le problème de cette construction, identique à celui de la première version du cube, est que le troisième point n'appartient pas effectivement au cercle. Cette incohérence peut être supprimée en créant le cercle explicitement dans un plan donné (désignation d'un plan, du centre du cercle et d'un point de sa circonférence, tout deux appartenant à ce plan, cf. Figure 2, à droite) mais on peut se demander si la construction d'un cercle par son centre et son rayon n'est pas incompatible avec la géométrie dans l'espace. Dans la **version 1** de ***Calques 3D***, cette modalité de construction du cercle ne sera pas disponible.



Pour garder une homogénéité avec les autres objets, nous avons mis à jour la rétroaction visuelle des point appartenant au cercle : matérialisation du centre du cercle (qui ne correspond plus à l'un des points de construction du cercle) et du rayon passant par ce point.

Les calques (en cours)

Les règles de fonctionnement des calques sur lesquelles nous nous sommes mis d'accord sont les suivantes :

- *L'univers* (la fenêtre graphique principale) et *les calques* (extraction de sous-figures de l'univers) sont des vues particulières d'une même figure géométrique. A chaque vue est associée une fenêtre graphique dont les matérialisations de l'espace (référentiel et point de vue) sont indépendantes.
- Les constructions sont possibles uniquement dans la fenêtre de *l'univers*, *les calques* étant destinés uniquement à l'observation.

- Pour extraire un objet depuis la fenêtre de *l'univers* et le copier dans un calque, il suffit de choisir un calque donné parmi ceux disponibles puis le désigner successivement les objets. Ceux-ci doivent être explicitement construits (on ne peux pas copier des rétro-actions visuelles par exemples) et être des éléments constitutifs d'objets plus complexes (les arêtes d'un cube par exemple).
- La copie dans un calque est une copie *dynamique* : toute déformation de la figure initiale dans la fenêtre de l'univers entraîne une modification des objets copiés dans les calques.
- La première copie dans un calque entraîne une ouverture de la fenêtre de celui-ci. A l'ouverture, une organisation particulière des fenêtres est opérée de manière à optimiser la visualisation de l'ensemble. Pour cette raison, il nous a semblé opportun de limiter le nombre de calque à trois.

Le fait d'avoir des représentations de l'espace autonomes d'une fenêtre à l'autre interdit le feuilletage, c'est-à-dire le passage d'une vue à l'autre dans une fenêtre donnée (passer d'un calque à l'autre par exemple). En effet, cette fonctionnalité qui, dans le cadre de la géométrie plane (**Calques 2**), permettait un effet d'animation, entraîne un clash visuel trop important et donc incompréhensible pour l'utilisateur.

Vue frontale ou de profil (en cours)

Cette fonctionnalité s'exécute en désignant les objets que l'utilisateur désire observer selon un point de vue particulier puis à choisir le type de vue : *vue frontale* (pour une observation en vraie grandeur), *vue de profil* ou *vue dans l'axe* (pour une observation de l'appartenance, du parallélisme,...). L'application de la commande se fait par déplacement automatique du point de vue de manière à coïncider avec la position demandée par l'utilisateur. Pour réduire la distorsion visuelle résultante du changement de point de vue, celui-ci s'opère de manière à minimiser le déplacement (en terme d'angle de rotation). Il peut donc arriver que l'on se trouve *'du mauvais côté'* de l'objet cible. Il a été suggéré de rajouter une commande *vue miroir* qui permet de passer d'un côté ou de l'autre, tout en restant dans la position requise. En liaison avec les calques, certaines propriétés seront mises en œuvre dans la **version 1** du logiciel :

- maintient de la vue frontale après une déformation de la figure. En effet, lorsque la figure initiale est déformée dans la fenêtre de l'univers, les objets extraits le sont aussi dans les calques. Cela signifie que le point de vue particulier demandé par l'utilisateur lorsqu'il se place en vue frontal peut être perdu lors de cette manipulation.
- exportation du point de vue d'une fenêtre à une autre. Lorsque l'utilisateur demande a se mettre en vue frontale dans un calque, il peut en effet souhaiter appliquer ce même point de vue dans la fenêtre de l'univers. La réciproque peut être vraie aussi.

Dans la **version 1** de ***Calques 3D***, ne seront implantées que la *vue frontale* et la *vue miroir*, et uniquement par désignation d'un plan ou de trois points.

Nouvelles fonctionnalités

Le zoom (en cours)

Avec la possibilité de disposer plusieurs fenêtres en parallèle dans l'application (voir le paragraphe sur les calques), il devient nécessaire de recadrer la figure à l'intérieur de chacune des fenêtres ouvertes : centrer le dessin mais aussi l'adapter à la taille disponible de la fenêtre, d'où la nécessité de disposer d'un zoom. Cette fonctionnalité sera disponible à la demande de l'utilisateur. Son fonctionnement reste à définir mais nous tenons autant que possible à éviter la saisie

d'une valeur numérique : choix parmi un éventail de valeur prédéfinies (25%, 50%, 150%, ... de la taille réelle) ou par la modification d'un ascenseur

Indicateur graphique du point de vue (en cours)

Nous mettons en place dans la **version 1** de ***Calques 3D*** un outil permettant de se représenter visuellement la position courante du point de vue. Il s'agit d'une palette flottante représentant la position de l'observateur par un curseur sur deux cadrans (cf. Figure 3) :

- l'un pour la représentation de la *longitude* (projection de l'observateur sur le plancher horizontal),
- l'autre pour la *latitude* (élévation de l'observateur par rapport au plancher).

Figure 3. *Indication graphique du point de vue : deux cadrans présentant longitude et latitude.*

Pour le moment, cette palette n'est que visuelle : il n'est pas possible de manipuler directement les cadrans pour modifier le point de vue. Par ailleurs , on pourra y adjoindre un indicateur visuel du zoom.

Poursuite du travail

Nos réunion de travail s'arrêtent donc pour cette année. Notre contrat avec la Région pour le développement de la maquette prend aussi fin en Septembre, date à laquelle nous devons leur livrer une version exempte de 'bogues'. C'est pour cette raison qu'il est préférable de stabiliser l'existant dans une **version 1** de ***Calques 3D*** plutôt que de se lancer dans de nouvelles fonctionnalités. Cette échéance cependant n'implique pas la fin du travail : nous souhaitons continuer à améliorer le logiciel, en particulier prévoir des expérimentation du logiciel avec des élèves en situation réelle et en analyser les résultats. La **version 1** du logiciel vous sera fournie en Septembre, par nous ou par la Région puisqu'il était demandé à ce que le produit puisse être diffusé dans les établissements lorrains.

Annexe B

Exemples de contextes d'utilisation

B.1 Introduction

Les *contextes d'utilisation* qui suivent ont été rédigés par des membres des deux groupes de travail. Ils sont retranscrits tels qu'ils nous ont été soumis au cours des différentes réunions, c'est-à-dire avec les manques et les problèmes que nous avons soulignés en conclusion du chapitre 4.

Nous présentons cinq exemples de *contextes* :

- une séquence autour d'un *polyèdre régulier* (construction et identification),
- une séquence sur l'extraction de *volumes élémentaires dans un solide complexe*,
- un ensemble constitué de trois séquences successives sur la construction et l'extraction de figures planes dans un solide. Les deux premières séquences concernent l'*intersection de plans dans un tétraèdre* (respectivement construction et identification), la dernière concerne l'*identification d'un polygone régulier construit dans un cube*.

B.2 Construction et identification d'un polyèdre régulier

Contexte

Nom de la séquence : Polyèdre régulier : octaèdre

Mots-clefs :

Description : construction d'un octaèdre régulier en partant d'un cube ; identification d'un triangle équilatéral, d'un carré ; orthogonalité de deux plans ; utilisation de la relation de Pythagore et calculs de volumes

Auteur : Michel Samotij

Date de création :

Public : BEP et BAC PRO industriels

Cadre de la formation : formation initiale technique ; formation continue

Durée :

Situation :

Prérequis

Concept : figures géométriques usuelles (triangle équilatéral, carré,...) ; isométrie de polygones .

Savoir-faire. : construction d'un cube ; constructions des diagonales d'un carré

Articulation dans la séquence : nature d'un triangle ; nature d'un quadrilatère ; orthogonalité de deux plans ; relation de Pythagore

Objectif *(selon les 4 catégories suivantes)*

1. **Information :** prise en main du logiciel ; interpréter l'énoncé du problème ; observation des figures en 2D et 3D
2. **Opération :** construction d'un cube ; construction du centre d'un carré en utilisant les diagonales ; construction d'un octaèdre ; isoler des figures usuelles, triangle, carré (2D et 3D) ; orthogonalité de deux plans
3. **Maîtrise :** calcul de longueur (relation de Pythagore) ; calcul de volumes
4. **Expertise :** conclure sur les propriétés d'un polyèdre régulier

Documents fournis :

- On donne la figure de l'exercice 1 sur un tirage papier
-

Activités caractéristiques de la séquence

Activités préparatoires : rappels des notions élémentaires en géométrie (polygones réguliers, diagonales, orthogonalité de deux plans, relation de Pythagore ; calculs de volumes

Pour l'enseignement : rappels concernant l'utilisation du logiciel

Pour l'apprentissage : construction d'un cube ; visualisation des figures intéressantes ; calculs de volume ; *Polyèdre régulier - exercice 1*

Pour la remédiation : rappels sur les polygones réguliers (triangles, quadrilatères...) ; construction de figures simples (carré, triangles) ; rappels et utilisation de la relation de Pythagore ; calculs de volumes simples

Activités de synthèse : construction d'autres polyèdres ; réflexion sur la manière de les construire.

Figure B.1 – *Polyèdre régulier : description de la séquence pédagogique*

Caractéristiques de l'activité ***(niveau d'objectif 1 et 2)***

Nom de l'activité : Polyèdre régulier - exercice 1 : sa construction, son observation et quelques calculs .....

Nature : Apprentissage et approfondissement .....

Énoncé :

A - Construction.
1) Construire un cube ABCDEFGH.
2) Construire les centres des six faces du cube : · I pour ADHE ; J pour ABFE ; · K pour BCFG ; L pour DCGH ; · M pour ABCD ; N pour FGHE.
3) Gommer les constructions permettant d'obtenir les centres des faces.
4) Joindre à l'aide de segment les centres des faces non opposées.
B - Observation du polyèdre.
1) Donner le nombre de faces.
2) Nature des faces ; en copier deux dans les calques 1 et 2, les observer et les comparer (utiliser la commande projection frontale).
3) Nature des quadrilatères INKM, MJNL et IJKL ; les copier dans les calques 1, 2 et 3 et les comparer (utiliser la commande projection frontale).
4) Déterminer la position des plans contenant les quadrilatères INKM, MJNL et IJKL.
5) Nommer le polyèdre.
C - Quelques calculs.
1) En fonction de l'arête a du cube ABCDEFGH, calculer le côté du carré IJKL et exprimer l'aire de ce carré.
2) Déterminer en fonction de a le volume de polyèdre.
3) Déterminer V'/V, rapport du volume du polyèdre au volume du cube.

Liste des objets géométriques	**Présentation de l'objet**
Cube, carré, droite, plan, point, segment	.....
triangle, pyramide, octaèdre	.....
Fonctions de construction	**Autorisée/Non autorisée**
Construction d'un cube	.....
Construction d'un segment	.....
Construction d'un point sur un segment	.....
Construction du milieu d'un segment	Non autorisée.....
Construction d'un plan	.....
Intersection de deux plans	.....
Nommer un objet.....	.....
Fonctions de manipulation	**Autorisée/Non autorisée**
Gommer.....	.....
Copier dans un calque	.....
Fonctions de visualisation	**Autorisée/Non autorisée**
Choix du référentiel.....	.....
Modification du point de vue	.....
Projection frontale.....	.....
Visualisation des calques.....	.....

Erreurs classiques identifiées dans l'activité :

- .....

Difficultés de l'activité et aides à apporter pour réduire la difficulté		
Type de la difficulté	**Nature de l'aide**	**Libellé, contenu, texte, ...**
.....	.....	.....

Figure B.2 – *Polyèdre régulier : description de l'activité*

B.3 Volume d'un solide complexe

Contexte

Nom de la séquence : Volume d'un solide complexe composé de solides identifiables et connus (cône, prisme).

Mots-clefs : identification d'un cône, d'un prisme droit ; volume d'un cône, d'un prisme droit. .

Description : Décomposition d'un solide en éléments faciles à identifier dans le but d'en calculer le volume.

Auteur : Michel Samotij

Date de création :

Public : BEP et BAC PRO industriels

Cadre de la formation : formation initiale technique ; formation continue

Durée :

Situation :

Prérequis

Concept : solides usuels, cône, prisme ; centre d'un arc de cercle

Savoir-faire. : calcul de volumes

Articulation dans la séquence : nature d'un solide régulier

Objectif ***(selon les 4 catégories suivantes)***

1. **Information :** prise en main du logiciel ; interpréter l'énoncé du problème ; observation des figures en 2D et 3D.
2. **Opération :** construction de centres d'arcs de cercle ; isoler et identifier des solides réguliers usuels(cône, prisme droit) ;
3. **Maîtrise :** calcul de volumes
4. **Expertise :** exprimer un volume a l'aide d'une expression littérale.

Documents fournis :

- On donne la figure de l'exercice 1 sur un tirage papier
-

Activités caractéristiques de la séquence

Activités préparatoires : rappels sur les solides réguliers ; calculs de volumes

Pour l'enseignement : rappels concernant l'utilisation du logiciel

Pour l'apprentissage : construction de points, de segments ; visualisation des solides intéressants ; calculs de volume

Pour la remédiation : rappels sur les solides réguliers (cônes, prisme, pyramides,...) ; construction de figures simples (carré, triangles) ; rappels et utilisation de la relation de Pythagore ; calculs de volumes simples

Activités de synthèse : construction d'autres polyèdres ; réflexion sur la manière de les construire

Figure B.3 – *Volume d'un solide complexe : description de la séquence pédagogique*

Caractéristiques de l'activité ***(niveau d'objectif 1 et 2)***

Nom de l'activité : Visualisation d'un solide complexe .....

Nature : Apprentissage et approfondissement .....

Énoncé :

A - Construction.
1) Construire un cube ABCDEFGH.
2) Construire les centres des six faces du cube : · I pour ADHE ; J pour ABFE ; · K pour BCFG ; L pour DCGH ; · M pour ABCD ; N pour FGHE.
3) Gommer les constructions permettant d'obtenir les centres des faces.
4) Joindre à l'aide de segment les centres des faces non opposées.
A partir de la figure que l'on vous propose :
1) Construire les centres des quatre arcs de cercle (O1, O2, O3 et O4).
2) Tracer les segments [O1A], [O2B], [O3C] et [O4D].
3) Tracer les rayons [O1E], [O1F], [O2G], [O2H], [O3I], [O3J], [O4K] et [O4L].
4) Copier dans le calque 1 le solide O1EFA. Observer ce solide.Donner la nature de ce solide. Combien de solides identiques à O1EFA dénombre-t-on dans la figure ?
5) Même question pour le solide O1AFGBO2.
6) Même question pour le solide O2BHICO3.
7) Comment peut-on procéder pour trouver le volume de ce solide ? Donner l'expression du volume de ce solide en fonction de r (rayon des arcs de cercle), h(h = O1A) , a (a = O1O2) et b (b = O2O3) :

Liste des objets géométriques	**Présentation de l'objet**
Point, segment, cône, prisme, arc de cercle	.....
Fonctions de construction	**Autorisée/Non autorisée**
Construction du centre d'un cercle	.....
Construction d'un segment	.....
Nommer un objet	.....
Fonctions de manipulation	**Autorisée/Non autorisée**
Copier dans un calque	.....
Fonctions de visualisation	**Autorisée/Non autorisée**
Choix du référentiel	.....
Modification du point de vue	.....
Visualisation des calques	.....

Erreurs classiques identifiées dans l'activité :

- .....

Difficultés de l'activité et aides à apporter pour réduire la difficulté		
Type de la difficulté	**Nature de l'aide**	**Libellé, contenu, texte, ...**
.....	.....	.....

Figure B.4 – *Volume d'un solide complexe : description de l'activité*

B.4 Intersection de plans dans un tétraèdre

Contexte

Nom de la séquence : Séquence 1 : intersection de plans dans un tétraèdre .....
Mots-clefs : segment, milieu d'un segment, intersection plan-plan .....
Description : .....
Auteur : Jean-Pierre Giorgi .....
Date de création : .....

Public : B.E.P.et Bac Professionnel industriels .....
Cadre de la formation : formation initiale .....
Durée : .....
Situation : .....

Prérequis

Concept : notion de tétraèdre, intersection de deux plans non parallèles .....
Savoir-faire. : construire un plan passant par trois points donnés, construire l'intersection de deux plans .....
Articulation dans la séquence : .....

Objectif ***(selon les 4 catégories suivantes)***

1. **Information :** un triangle ABC appartenant au plancher et un point D extérieur au plancher sont donnés. .....
2. **Opération :** tracer l'intersection de deux plans non parallèles .....
3. **Maîtrise :** nature de l'intersection de deux plans. .....
4. **Expertise :** .....

Documents fournis :

- .....
- .....

Activités caractéristiques de la séquence

Activités préparatoires : rappels sur les positions relatives de deux plans dans l'espace. .....
Pour l'enseignement : .....
Pour l'apprentissage : .....
Pour la remédiation : .....
Activités de synthèse : .....

Figure B.5 – *Polyèdre régulier : description de la séquence pédagogique*

Caractéristiques de l'activité ***(niveau d'objectif 1 et 2)***

Nom de l'activité : intersection de deux plans dans un tétraèdre. .....

Nature : apprentissage .....

Énoncé :

Placer trois points non alignés dans le plancher. Les nommer A, B, C. Soit un point extérieur au plancher. Le nommer D. Construire les segments [AB], [AC], [AD] et [BC]. Construire les milieux des segments [CD] et [BC]. Les nommer respectivement I et J. Construire les plans ABI et ADJ. Construire leur intersection.

Liste des objets géométriques	**Présentation de l'objet**
point, segment, plan	.....
Fonctions de construction	**Autorisée/Non autorisée**
point, segment, milieu d'un segment	.....
Construction d'un segment	.....
intersection plan-plan.	.....
Fonctions de manipulation	**Autorisée/Non autorisée**
.....	.....
Fonctions de visualisation	**Autorisée/Non autorisée**
.....	.....

Erreurs classiques identifiées dans l'activité :

- .....

Difficultés de l'activité et aides à apporter pour réduire la difficulté		
Type de la difficulté	**Nature de l'aide**	**Libellé, contenu, texte, ...**
.....	.....	.....

Figure B.6 – *Polyèdre régulier : description de l'activité*

B.5 Intersection de plans dans un tétraèdre

Contexte

Nom de la séquence : Séquence 2 : intersection de plans dans un tétraèdre .....

Mots-clefs : intersection plan-plan, médianes d'un triangle et centre de gravité. .....

Description : .....

Auteur : .....

Date de création : .....

Public : B.E.P. et Bac Professionnel .....

Cadre de la formation : formation initiale .....

Durée : .....

Situation : .....

Prérequis

Concept : positions relatives de deux plans, médianes d'un triangle et centre de gravité .....

Savoir-faire. : construire l'intersection de deux plans, construire les médianes d'un triangle et le centre de gravité, savoir extraire une sous figure, réaliser une projection frontale.. .....

Articulation dans la séquence : séquence 1 (apprentissage de la construction de l'intersection de deux plans) .....

Objectif *(selon les 4 catégories suivantes)*

1. Information : le tétraèdre ABCD est donné .....

2. Opération : tracer l'intersection de deux plans, tracer les médianes d'un triangle et son centre de gravité, extraction d'une sous figure, projection frontale. .....

3. Maîtrise : intersection de deux plans, notion de médianes et de centre de gravité .....

4. Expertise : déterminer l'intersection de deux plans .....

Documents fournis :

- .....
- .....

Activités caractéristiques de la séquence

Activités préparatoires : construire l'intersection de deux plans (séquence 1) .....

Pour l'enseignement : .....

Pour l'apprentissage : .....

Pour la remédiation : .....

Activités de synthèse : droites remarquables d'un triangle (médianes...), intersection des médianes, centre de gravité .....

Figure B.7 – *Intersection de plans : description de la séquence pédagogique*

Caractéristiques de l'activité ***(niveau d'objectif 1 et 2)***

Nom de l'activité : détermination de l'intersection de plans dans un tétraèdre .....

Nature : .....

Énoncé :

Soit un tétraèdre ABCD donné. Construire les milieux des arêtes [DC], [BC], [BD]. Les nommer respectivement I, J et K. Soit O l'intersection des segments [BI], [DJ] et [CK]. 1) Copier dans un calque tous les éléments géométriques contenues dans la face BCD. 2) Construire l'intersection des plans ADJ et ABI. La nommer (D'). 3) Que représente la droite (D')?

Liste des objets géométriques	**Présentation de l'objet**
points, segments, plans, tétraèdre	.....
Fonctions de construction	**Autorisée/Non autorisée**
segment	.....
milieu d'un segment	.....
intersection droite-droite	.....
intersection plan-plan	.....
Fonctions de manipulation	**Autorisée/Non autorisée**
extraction d'une figure	.....
Fonctions de visualisation	**Autorisée/Non autorisée**
Projection frontale	.....
Visualisation des calques	.....

Erreurs classiques identifiées dans l'activité :

- .....

Difficultés de l'activité et aides à apporter pour réduire la difficulté		
Type de la difficulté	**Nature de l'aide**	**Libellé, contenu, texte, ...**
.....	.....	.....

Figure B.8 – *Intersection de plans : description de l'activité*

B.6 Hexagone régulier dans un cube

Contexte
Nom de la séquence : Séquence 3 : hexagone régulier dans un cube .....
Mots-clefs : cube, milieu des arêtes, hexagone régulier. .....
Description : .....
Auteur : Jean-Pierre Giorgi .....
Date de création : .....

Public : B.E.P, Bac Professionnel .....
Cadre de la formation : formation initiale .....
Durée : .....
Situation : .....

Prérequis
Concept : notion de polygone régulier .....
Savoir-faire. : construire le milieu d'un segment .....
Articulation dans la séquence : séquence 2 (manipulation d'un calque) .....

Objectif ***(selon les 4 catégories suivantes)***
1. Information : un cube ABCDEFGH est donné. .....
2. Opération : construire le milieu d'un segment, extraire une sous-figure, tracé d'un cercle passant par trois points, détermination de son centre. .....
3. Maîtrise : utilisation du théorème de Pythagore. .....
4. Expertise : déterminer la nature d'un polygone. .....

Documents fournis :
- .....
- .....

Activités caractéristiques de la séquence
Activités préparatoires : rappels sur la relation de Pythagore .....
Pour l'enseignement : relation de Pythagore dans l'espace .....
Pour l'apprentissage : utilisation des calques .....
Pour la remédiation : rappels et soutien sur la relation de Pythagore et des propriétés sur les polygones réguliers .....
Activités de synthèse : reconnaître la nature d'un polygone régulier (hexagone) .....

Figure B.9 – *Hexagone régulier : description de la séquence pédagogique*

Caractéristiques de l'activité ***(niveau d'objectif 1 et 2)***

Nom de l'activité : hexagone régulier dans un cube .....

Nature : apprentissage, approfondissement, remédiation et évaluation .....

Énoncé :

Soit un cube ABCDEFGH. Construire les milieux respectifs I, J, K, L, M, N, des arêtes [AD], [DC], [CG], [GF], [FE], [EA]. Construire le polygone IJKLMN. Tracer le cercle passant par les points I, J, K et déterminer son centre O. Copier dans calque 1 les points I, J, K, L, M, N, le polygone IJKLMN, le cercle et son centre O. En obtenir une projection frontale. Copier dans calque 2 la face ABFE, les points M et N et le segment [MN]. Sachant que les arêtes du cube ont pour mesure a, exprimer MN en fonction de a. Copier dans calque 3 la face EFGH, les points M et L et le segment [ML]. Exprimer ML en fonction de a. Montrer qu'il en est de même pour les 6 faces du cube. Copier dans calque 4 le triangle IPL, P étant le milieu de [BC]. Quelle est la nature du triangle IPL, calculer IL en fonction de a en remarquant que IP = PL = a. Vérifier que le point O est le milieu du segment [IL]. En déduire la nature du polygone IJKLMN.

Liste des objets géométriques	**Présentation de l'objet**
cube, points, segments, ...	.....
triangles rectangles isocèles	.....
cercle, hexagone régulier	.....
Fonctions de construction	**Autorisée/Non autorisée**
milieu d'un segment	.....
cercle passant par 3 points	.....
centre d'un cercle	.....
Fonctions de manipulation	**Autorisée/Non autorisée**
extraction de figures dans un calque	.....
Fonctions de visualisation	**Autorisée/Non autorisée**
Projection frontale	.....

Erreurs classiques identifiées dans l'activité :

- .....

Difficultés de l'activité et aides à apporter pour réduire la difficulté		
Type de la difficulté	**Nature de l'aide**	**Libellé, contenu, texte, ...**
.....	.....	.....

Figure B.10 – *Hexagone régulier : description de l'activité*

Annexe C

Présentation graphique des objets géométriques de *Calques 3D*

Objet	**Représentation graphique**	**Éléments visuels de compréhension**
Représentation des points		
Point	La forme de la représentation graphique du point peut être modifiée (rond vide ou plein, carré vide ou plein, croix + ou X, ...).	Projetantes du point en fonction du référentiel choisi (projection sur les axes du repère orthonormé, distance au plancher, distances aux diverses cloisons).
intersection sur droite		
sur plan		Ligne de plus grande pente du plan passant par ce point
sur cercle		Centre du cercle et rayon passant par ce point
sur sphère		Demi-grand cercle de la sphère passant par ce point
Représentation des courbes		
Droite	Par une ligne infinie passant par les deux points, d'épaisseur et d'aspect variable (pointillé, traits, alternance traits - pointillés, ligne continue, ...).	Point d'intersections de la ligne avec les cloisons ou le plancher.
Segment	Par un segment de droite reliant les points extrémités, d'épaisseur et d'aspect variable (pointillés, traits, alternance traits - pointillés, ligne continue, ...).	Point d'intersections du segment avec les cloisons ou le plancher.
Cercle		

Objet	**Représentation graphique**	**Éléments visuels de compréhension**
Arc	Tracé de la circonférence du cercle. L'aspect et l'épaisseur du tracé du cercle peuvent être modifiés comme pour les droites.	Points intersection du cercle avec les cloisons ou le plancher. Les parties du cercle situées derrière une cloison sont représentées par un trait plus fin et une couleur atténuée.
Représentation des surfaces		
Plan	Par le plus petit rectangle englobant, avec une certaine marge, TOUS les points appartenant EXPLICITEMENT au plan (les points le définissant, les points construits sur ce plan ou résultats de l'intersection d'une droite et de ce plan, ...). L'aspect et l'épaisseur du contour du plan peuvent être modifiés comme pour les droites.	Droites d'intersection du plan avec les cloisons ou le plancher..
Sphère	La sphère est représentée par son contour apparent, le grand-cercle parallèle au plan horizontal, le grand-cercle passant par le deuxième point définissant la sphère, le grand-cercle perpendiculaire au précédent. L'aspect et l'épaisseur du tracé de la sphère peuvent être modifiés comme pour les droites.	Cercle intersection de la sphère avec le plancher horizontal. Les parties de la sphère situées derrière celle-ci par rapport à l'observateur sont représentées en pointillés.
Cube	Le cube étant composé de points et de segments, chacun peut être modifié selon les attributs de sa catégorie (la forme des points, l'épaisseur des segments, ...). Tout changement de couleur directement sur le cube s'applique à TOUS les objets le constituant. Le cube peut avoir l'attribut transparent (on ne prend pas en considération les faces) ou translucide (les faces 'masquent' légèrement les arêtes situées à l'arrière le cube par rapport à l'observateur : celles-ci sont alors représentées en pointillés).	Points d'intersections des arêtes du cube avec le plancher ou les cloisons.

Table C.1 – Synthèse des représentations visuelles des objets

Résumé

Cette thèse a pour thème la conception d'*Environnements Interactifs d'Apprentissage avec Ordinateur* (EIAO). L'une des difficultés de ce domaine est que les solutions proposées sont rarement acceptées par les enseignants car non conformes à leur propre conception pédagogique de l'enseignement à dispenser. L'objectif du travail est donc de prendre en compte les souhaits des enseignants dès la conception du produit, en identifiant et catégorisant les différents besoins pédagogiques, pour pouvoir offrir à l'*enseignant "prescripteur"* (celui qui utilise le logiciel) un environnement ouvert où il pourra choisir la configuration adaptée à sa pédagogie. Le champ d'application que nous avons choisi est celui de l'enseignement de la géométrie dans l'espace, via la réalisation d'un logiciel de type micromonde, ***Calques 3D***. Afin d'atteindre les objectifs fixés, nous avons collaboré, lors de la conception du logiciel, avec deux groupes d'*enseignants "auteurs"* qui ont enrichi la maquette par leurs approches variées. Pour en favoriser le développement, nous avons mis en place un formalisme, les *"contextes d'utilisation de logiciels pédagogiques"*, permettant aux enseignants auteurs d'exprimer leurs choix de présentation des connaissances et les activités qu'ils souhaitent mettre en place autour de ces connaissances. Ce formalisme nous a permis en retour d'extraire les informations pertinentes à implanter et de choisir les représentations internes adéquates. Ces documents ont ainsi servi de support à la négociation entre enseignants auteurs et informaticiens pour l'obtention d'un accord sur les connaissances géométriques embarquées dans le logiciels et leurs présentations externes à l'interface.

Mots-clés: EIAO, micromonde, génie éducatif, acquisition de l'expertise pédagogique, géométrie dynamique dans l'espace

Abstract

This PhD thesis takes place in the context of designing Interactive Learning Environment (ILE) where one of the difficulties is that the proposed solutions are hardly accepted by teachers because they are often not adapted to their own approach of teaching. Thus, our objective is to take in account the teachers needs directly in the design process, by identifying and categorising the different pedagogical needs, in order to provide the *'user-teacher'* (intended to use the software) with an open environment on which he will be able to choose or define an adapted configuration. The application field we chose is spatial geometry learning, with the design of a microworld, ***Calques 3D***. In order to reach our objectives, we collaborated, during the design process, with two groups of *'author-teachers'* who enriched the prototype with their wide-ranging approaches of the same learning field. In order to ease and conduct the design of the software, we proposed a framework, named *"utilisation contexts of educational software"*, that author-teachers are intended to use for expressing their choices on knowledge presentation and for describing activities they plan to organise around them. This framework allowed us in turn to extract the implicit knowledge that teachers have about "how and why they teach concepts in a particular way" in order to implement it and choose an adapted internal representation. So, it served as a negotiation support between teachers and software engineers in order to come to an agreement on the pedagogical content, in particular on the domain concepts and their external presentations.

Keywords: Interactive Learning Environment, Microworld, Instructional Software Design, Teaching Knowledge Acquisition, 3D Dynamic Geometry