MICHELE TESSIER-BAILLARGEON, NICOLAS LEDUC,
PHILIPPE R. RICHARD, MICHEL GAGNON

ETUDE COMPARATIVE DE SYSTEMES TUTORIELS POUR L'EXERCICE DE LA DEMONSTRATION EN GEOMETRIE

Abstract. Comparative study of tutorial systems for geometry proof learning. This article proposes a state of the art of tutorial systems for high school planar geometry proof learning. The chosen approach is part of exploring the research problem that motivates the development, by our research team, of a tutorial system named QED-Tutrix, that we will present in another paper. In the following article, a synthesis and a comparison of existing tutorial systems is carried out on the basis of a set of original indicators highlighting the differences between the systems covered by our analysis. Each indicator aims to describe the functioning of the studied software according to the geometric work made possible at their interface. Eleven tutorial systems are compared according to their integration of a geometric figure, the structure they impose on the student's reasoning and the tutorial intervention they offer.

Résumé. Cet article propose un état de l'art sur les systèmes tutoriels pour l'exercice de la démonstration en géométrie plane à l'école secondaire. La démarche employée s'inscrit dans l'examen d'un problème de recherche motivant le développement par notre équipe de recherche d'un tutoriel, nommé QED-Tutrix, lequel sera l'objet d'un second article. Dans l'article qui suit, une synthèse et une comparaison du fonctionnement de systèmes tutoriels existants est menée à partir d'un ensemble d'indicateurs originaux permettant de mettre en valeur les différences entre les systèmes sur lesquels porte notre analyse. Chaque indicateur vise à décrire le fonctionnement des logiciels étudiés en fonction du travail géométrique rendu possible à l'interface de chacun d'eux. Onze systèmes tutoriels sont comparés en fonction de l'intégration que ceux-ci font d'une figure géométrique, de la structure qu'ils imposent au raisonnement de l'élève et de l'intervention tutorielle qu'ils proposent.

Mots-clés. Système QED-Tutrix, étude comparative, systèmes tutoriels, découverte guidée et démonstration en géométrie.

Introduction

Les EIAH sont venus bouleverser les habitudes didactiques pour l'enseignement et l'apprentissage des mathématiques, et plus particulièrement en géométrie. Bien que la recherche technologique ait évolué rapidement ces dernières années et que les systèmes informatiques semblent prometteurs, aucun EIAH (environnement informatique d'apprentissage humain) pour l'apprentissage de la démonstration, n'a réussi à s'imposer.

ANNALES de DIDACTIQUE et de SCIENCES COGNITIVES, volume 22, p.91 - 117.

Avec QEDX, l'objectif de notre équipe de recherche multidisciplinaire est de développer un système tutoriel par la modélisation didactique et informatique des schèmes d'action instrumentés de l'apprenti géomètre et de son enseignant. En choisissant cette méthode de développement anthropocentrique (Rabardel, 1995, p.^pp.), nous visons un système qui respecte l'exercice géométrique authentique grâce à l'analyse des modalités du contrat didactique observé en classe réelle.

Dans le cadre de l'examen de notre problème de recherche pour le développement de QED-Tutrix (QEDX), nous avons mené une revue des environnements informatiques d'apprentissage humain conçus pour l'exercice de la pensée géométrique afin de prendre connaissance de l'offre actuelle. Dans un contexte d'effort conjoint entre informatique et didactique, cet exercice de recension a mis en lumière l'absence d'articles en didactique des mathématiques qui font l'état de l'art des systèmes existants pour l'apprentissage de la démonstration chez l'apprenti géomètre. En revanche, de tels bilans sont communs dans un cadre informatique. L'étude comparative qui suit répond à une volonté d'abord méthodologique afin de regrouper, de décrire et de retracer l'évolution des deux dernières décennies en ce qui a trait aux logiciels pour le développement des compétences de démonstration en géométrie plane.

Sans recenser les système existants, Balacheff, dans son article *Didactique et intelligence artificielle* (1994), se livre à un exercice de classement des EIAH en mathématiques en trois types : les micromondes, les tuteurs et les environnement de découverte guidée. Prenant appui sur les catégories avancées par Balacheff, qui propose d'ordonner les EIAH en fonction du degré d'initiative laissée à l'élève ou, réciproquement, selon le degré de directivité du système (Balacheff, 1994), nous allons préciser notre échantillon d'EIAH à analyser en se restreignant aux systèmes qui offrent une assistance à l'élève en résolution de problème. Ensuite chaque section fera état des fonctionnalités d'un de ces systèmes en décrivant la manière dont est intégrée la figure géométrique, la façon dont est coordonné le travail de l'élève ainsi que les interventions tutorielles disponibles.

1. La directivité du système : Micromondes, Tuteurs et environnement de découverte guidée

Selon Balacheff (1994), l'initiative laissée à l'élève dans son processus de résolution de problèmes mathématiques constitue un bon critère de classification des environnements informatiques d'apprentissage humain. Balacheff propose un continuum qui s'échelonne d'une totale liberté accordée à l'élève à une prise en charge complète du travail mathématique par l'EIAH.

Les micromondes conçus pour la construction et l'exploration des propriétés de figures dynamiques sont classés par Balacheff comme étant les EIAH les moins directifs : puisque sans régulation enseignante extérieure, ces systèmes ne proposent pas de tâche à résoudre ni d'aide à l'élève. Conséquemment, à l'exception des exigences minimales régissant une communication mathématique, aucune contrainte ou intention didactique ne vient encadrer l'interaction entre l'élève et l'EIAH. Ainsi, dans un contexte d'apprentissage de la démonstration en géométrie, Hanna (2000) suggère que l'exploration des limites d'une construction géométrique pour en déduire ses propriétés intrinsèques mène à un apprentissage grâce à l'organisation de ces découvertes sous forme de démonstration. Selon elle, le défi de l'enseignant est d'exploiter cet engouement qui découle de l'exploration des figures dynamiques pour amener les élèves à produire une démonstration. Cette idée est partagée par De Villiers :

> *It is far more meaningful to INTRODUCE proof within a dynamic geometry context, NOT as a way of making sure, but rather as a means of explanation, understanding, and discovery before dealing with the more formal and abstract functions of verification and systematisation.* (Villiers, 2007, p. 50)

Toutefois, la tâche d'amener l'élève à systématiser ses découvertes et à en vérifier la démarche sous-jacente est traditionnellement assumée par l'enseignant, mais elle peut aussi être la responsabilité d'un agent tutoriel intégré à l'EIAH. La directivité de ces agents tutoriels est variable et dépend du type de soutien envisagé par les concepteurs du système tutoriel et elle vient justifier la distinction des deux autres catégories d'EIAH proposées par Balacheff (1994), soit les tuteurs comme tels et les environnements de découverte guidée.

Les tuteurs sont pour Balacheff (1994) à l'opposé des micromondes sur l'axe continu de directivité imposée par les EIAH. S'apparentant à un enseignement, que Balacheff qualifie de frontal ou directif, les tuteurs, sans réel souci ou moyen d'assurer la compréhension de l'élève et au risque de dénaturer le problème à résoudre, s'assurent, grâce à des indices ou à des instructions, que l'élève produise sans détours la ou les solutions déterminées comme étant admissibles ou idéales. Ainsi, sans accorder de valeur didactique à l'erreur ou aux aléas du raisonnement mathématique, les tuteurs font preuve d'une *impatience pédagogique* en « souhaitant que soit abordé au plus vite l'objet d'apprentissage visé » (Pluvinage & Rigo Lemini, 2008, p. 57). Cette impatience a pour effet d'amener l'élève à suivre passivement le déroulement séquentiel d'une méthode sélectionnée et d'inhiber le processus idiosyncratique de résolution de problème de démonstration, ce qui peut à terme nuire à la compréhension de ce processus mathématique. D'ailleurs comme le dit Tanguay :

> *La compréhension de la structure d'une démonstration tant soit peu complexe nécessite de la part de l'élève un travail – de lecture ou d'écriture – ponctué de pauses, de retours sur les propositions déjà énoncées, de réaménagement et simultanéisations (pour rapprocher des propositions ou blocs de propositions non contigus dans le texte), de recul, d'appréhension globale (pour saisir certains éléments de macro-organisation), de contrôle, de réflexion. Toutes choses que ne permet pas cette [...] pratique orale du texte, faite de fluence, de séquentialité, d'irréversibilité.* (Tanguay, 2005, p. 64)

De ce fait, il serait souhaitable qu'un système d'aide tutorielle puisse accompagner l'élève dans sa recherche heuristique d'une solution et, au final, susciter les apprentissages nécessaires à la rédaction de sa démonstration. Balacheff propose l'appellation *environnements de découverte guidée*[1] pour désigner les EIAH qui parviennent à un équilibre entre respect du raisonnement personnel et imprévisible de l'apprenti géomètre et accompagnement de l'élève lorsqu'il se trouve en difficulté.

Les environnements de découverte guidée ou systèmes *coach* (Balacheff, 1994) imposent une directivité intermédiaire entre celle du micromonde et celle du tuteur en laissant « une liberté apparente, ce qui signifie qu'il n'y a pas de rétroaction systématique ou immédiat suite aux erreurs commises par l'élève, mais certaines règles permettent au système de planifier l'interaction en fonction d'une évaluation du comportement de l'apprenant » (Balacheff, 1994, p. 27). Par conséquent, cette liberté calculée dont jouit l'élève assure que la résolution effective d'un problème découle directement des compétences mathématiques de l'élève. Toutefois, cette autonomie de pensée n'est pas menacée par un soutien qui est prévu si et seulement si le système tutoriel détermine que l'élève ne parviendra pas à une solution sans l'intervention d'un agent aidant, puisque alors cette intervention rend possible un apprentissage autrement inaccessible. Ainsi, les environnements de découverte guidée obéissent à la règle de ne pas intervenir tant que l'élève a une chance de parvenir seul à une solution et, en cas de blocage, de guider l'élève pour maximiser les opportunités d'apprentissage de ce dernier.

Dans ce qui suit, nous allons considérer toute forme de soutien à la résolution de problèmes comme étant accompli par un système tutoriel, puisque bien qu'une assistance agisse de manière plus ou moins directive, elle intervient à titre d'aide

---

[1] La formulation découverte guidée est inspirée du principe de découverte guidée proposé par Elsom-Cook, M. (1990), principe selon lequel l'équilibre entre directivité et non directivité prend appui sur une prise en compte simultanée de l'état de l'apprenant et de la complexité de l'objet d'enseignement (Balacheff, 1994).

tutorielle. Ainsi, les systèmes étudiés présentent tous uns composante tutorielle et sont conçus pour l'exercice de la démonstration en géométrie. Cet article ne constitue pas une description exhaustive, un recensement systématique ou un ordonnancement à proprement parler des systèmes tutoriels sélectionnés en fonction de leur niveau de directivité, mais plutôt une étude comparative des aspects clés de leur fonctionnement. Cet exercice permettra au lecteur d'avoir un portrait d'ensemble des particularités des systèmes tutoriels disponibles pour l'exercice de la géométrie plane au secondaire, mais aussi de préciser et de mettre en contexte le fonctionnement de QEDX, un système tutoriel de type environnement de découverte guidée.

2. Un ensemble novateur de variables pour décrire le fonctionnement des systèmes étudiés

Pour notre comparaison, nous avons d'abord cherché un ensemble d'indicateurs existants pour décrire et comparer des systèmes tutoriels conçus pour l'exercice de la démonstration en géométrie. Nous avons principalement identifié des descriptions d'EIAH données à titre de revue de littérature pour justifier la pertinence d'un système tutoriel donné, sans toutefois pouvoir trouver de précision explicite quant aux indicateurs ad hoc utilisés. Nous avons alors élaboré notre propre ensemble d'indicateurs en fonction des systèmes tutoriels sélectionnés. Pour ce faire, nous avons d'abord identifié un ensemble de systèmes conçus pour la production de démonstrations géométriques avec comme objectif le dégagement de différents marqueurs qui qualifient le fonctionnement et la logique didactique de chacun d'eux tout en faisant ressortir les particularités qui les distinguent. Plus précisément, cette méthode d'analyse qualitative de *questionnement analytique* (Paillé & Mucchielli, 2013) nous a permis de dégager non pas des invariants, mais des indicateurs éloquents qui servent notre ambition de décrire et de comparer les systèmes tutoriels étudiés.

Une première analyse a d'abord mis en lumière trois grands volets qui caractérisent les systèmes tutoriels :

- L'intégration de la figure géométrique au sein du système tutoriel.
- Les interfaces et les structures sous-jacentes prévues pour l'exercice du raisonnement mathématique et pour sa communication.
- Les modes d'intervention des systèmes tutoriels.

Ensuite, les analyses itératives subséquentes ont révélé différentes variables qui précisent ces trois grands volets et permettent par le fait même de mettre en évidence les particularités propres à chacun des systèmes tutoriels analysés. Par exemple, si nous prenons le second volet qui concerne le raisonnement géométrique de l'élève au sein des systèmes tutoriels décrits, au fil des lectures et des relectures du matériel et en fonction de ce qui a été recensé, le questionnement analytique s'est précisé comme suit :

- Comment s'opère la résolution de problème pour chaque système tutoriel ?
- Est-ce que l'élève est contraint de raisonner dans un ordre prédéterminé ?
- Est-ce que le système fonctionne selon un chaînage avant, arrière, mixte ou sans chaînage ?

L'arborescence finale qui résulte de notre analyse et qui fera office de grille d'observation pour notre étude comparative est illustrée au tableau I.

Tableau I
Caractéristiques des systèmes tutoriels pour l'exercice de la démonstration en géométrie

Intégration de la figure géométrique				Structure prévue pour l'exercice du raisonnement de démonstration géométrique						Intervention tutorielle							
Figure statique		Figure dynamique		Solutions admises				Ordres des entrées		Intervention		Accompa-gnement	Rétroaction				
													Validation				
Dessin	Figure interactive	Exploration	Construction	Moteur de déduction automatique	Programmées par un expert ou un enseignant	Exploration simultanée de multiples stratégies / Reconnaissance de plan	Phases séquentielles du raisonnement	Chaînage (avant, arrière)	Chaînage mixte ou exploration libre	À la demande de l'élève	Gérée par le système tutoriel	Indication prochaine étape (chaînage)	Locale : énoncés, inférences	Globale : solution, démonstration	Démonstration en graphe (réseau déductif)	Annotation de la démonstration	Explication des erreurs

Afin d'éviter d'alourdir la description de chaque système tutoriel présenté, nous allons définir d'entrée de jeu chacune des variables (les ramifications des trois catégories qui forment la première ligne du tableau I). Ainsi, un lecteur souhaitant s'informer à propos d'un système tutoriel en particulier peut se référer à ces explications et aux paragraphes dédiés au système tutoriel concerné.

2.1. Intégration de la figure géométrique

« Le raisonnement en géométrie est favorisé par la présence d'une figure qui permet une saisie globale du problème et aide à la création d'images mentales » (Bernat, 1993, p. 27). Cette affirmation fait l'objet d'un consensus au sein de la communauté didactique, mais l'introduction des environnements informatiques pour l'exercice de la géométrie vient redéfinir les effets potentiels de l'intégration d'une figure géométrique sur le raisonnement géométrique figural. S'il est évident que la géométrie dynamique a révolutionné la construction et l'exploration de figures géométriques, avant d'aborder ce type de figure, deux catégories de figures statiques ont été observées au sein des systèmes tutoriels analysés.

D'abord, la figure statique qui accompagne l'énoncé d'un problème peut être un simple dessin semblable à celui que l'élève retrouve conjointement à l'énoncé d'un problème traditionnel. La figure géométrique statique peut aussi être interactive et s'animer en fonction des actions de l'élève ou du tuteur. Par exemple, un segment peut changer de couleur lorsqu'il est nommé dans un énoncé soumis par l'élève, ou la mesure d'un angle peut s'afficher lorsque le curseur le survole. La figure interactive ne doit pas être confondue avec la figure dynamique puisque bien qu'elle soit plus sophistiquée que le dessin et qu'elle puisse attirer l'attention de l'élève sur certains éléments figuraux et potentiellement initier une réflexion chez ce dernier, la figure interactive demeure une figure statique ne pouvant être déformée ou modifiée par l'élève.

Pour sa part, la figure dynamique existe et évolue en fonction des relations géométriques (appartenance, équidistance, parallélisme, perpendicularité, etc.) qui lient les éléments géométriques qui la constitue (Baulac, 1990). Conséquemment, la figure dynamique se « déforme en respectant les propriétés géométriques qui ont servi à son tracé et celles qui en découlent » (Laborde & Capponi, 1994, p. 173). Ce dynamisme particulier, qui maintient la logique interne d'une construction lors d'un déplacement (*dragging*), permet la visualisation d'un nombre de cas de figures autrement inaccessible pour une définition géométrique donnée. De ce fait, le registre des figures dynamiques constitue un registre sémiotique distinct de celui des figures traditionnelles ou statiques (Coutat *et al.*, 2014) puisqu'il permet l'exploration des limites de la construction géométrique et des invariants géométriques qui révèlent les propriétés géométriques sous-jacentes. En ce sens, la figure dynamique peut contribuer davantage que la figure statique au raisonnement de preuve de l'élève puisqu'elle lui permet à la fois de poser et de mettre à l'épreuve des conjectures (Hanna, 2000).

Aussi, une figure dynamique au sein d'un micromonde permettra la construction de nouveaux éléments figuraux par l'élève. Cet atout supplémentaire n'est pas exigé pour qu'une figure soit qualifiée de dynamique, mais s'avère essentiel dans le cas d'un EIAH qui prend en compte les actions graphiques comme éléments d'une démonstration.

Comme l'apport d'une figure dynamique pour l'exploration de propriétés géométriques et pour l'examen d'un problème de démonstration est abondamment discuté dans la littérature (Coutat *et al.*, 2014), il semble qu'un environnement de découverte guidée en géométrie puisse difficilement renoncer à l'inclusion de la géométrie dynamique pour l'obtention d'un milieu pertinent pour l'exercice de la démonstration (Balacheff & Margolinas, 2005).

2.2. Structure prévue pour l'exercice du raisonnement de démonstration géométrique

Comment est orchestré le raisonnement géométrique de l'utilisateur des différents systèmes tutoriels ? Comment l'élève chemine-t-il au sein de l'espace du problème ? À la lumière de l'analyse des différents systèmes tutoriels étudiés, nous avons divisé ce questionnement en sous-questions ciblées qui permettent de différencier les systèmes tutoriels. D'abord, comment sont élaborées les solutions expertes (ou témoins) à partir desquelles les solutions de l'élève sont évaluées ? Ces solutions étant admises par le système, l'élève peut-il en explorer plusieurs à la fois dans sa résolution d'un problème? L'élève doit-il articuler les phases de son raisonnement selon une séquence prédéterminée ? Enfin, lorsque l'élève soumet les éléments de sa solution, doit-il les soumettre suivant un ordre particulier ?

Dans un premier temps, en ce qui concerne les solutions, c'est-à-dire les démonstrations reconnues comme admissibles par le système tutoriel, celles-ci peuvent être produites de manière autonome par le système grâce à un moteur de déduction automatique ou par des experts didacticiens ou enseignants. Dans le cas d'un traitement par moteur de déduction automatique, cette informatisation du raisonnement déductif implique, du moins actuellement, une modélisation selon un paradigme de géométrie formelle (Houdement & Kuzniak, 2006). Un paradigme de géométrie formelle limite la gestion de l'erreur et l'emploi des raccourcis inférentiels[2]. Qui plus est, les moteurs de déduction automatiques, au lieu d'un

[2] À titre d'exemple de raccourci inférentiel, la propriété qui dit qu'un quadrilatère ayant trois angles droits est un rectangle peut être perçue comme un raccourci inférentiel puisque cette propriété résulte d'une suite d'inférences logico-déductives. Une démonstration qui fait appel à cette propriété est donc plus courte que celle qui inclut l'ensemble des inférences desquelles découle cette propriété.

éventail de solutions possibles, ne génèrent communément qu'une seule démonstration, celle jugée la plus *heuristiquement efficace* (Richard & Fortuny, 2007), solution qui n'est toutefois pas nécessairement la plus compréhensible pour l'élève utilisateur, ni celle qui concorde avec la pratique ou les exigences habituelles de son enseignant. Ainsi, malgré les avantages logistiques évidents associés à l'intégration d'un moteur de déduction automatique à un système tutoriel, certains systèmes tutoriels s'appuient plutôt, et avec le coût que cela représente, sur des solutions expertes préalablement programmées par un expert didacticien ou un enseignante En effet, afin de respecter le contrat didactique (Brousseau, 1998) auquel les élèves et les enseignants utilisateurs sont accoutumés, les solutions admises doivent permettre une flexibilité quant au paradigme géométrique de référence, tandis que les démonstrateurs automatiques obéissent par défaut à une géométrie formelle.

Indépendamment de la manière avec laquelle sont implémentées les différentes solutions idéales auxquelles se mesurent les raisonnements des apprentis géomètres, la plupart des systèmes tutoriels reconnaissent plusieurs solutions à un même problème. Cependant, lorsque l'élève est invité à résoudre le problème, certains systèmes tutoriels cherchent à dégager le plus rapidement possible une solution dominante à partir des actions de l'élève, et le contraignent ensuite à conclure cette solution sans changement possible de stratégie au cours de son raisonnement. Qui plus est, certains systèmes obligent l'élève à résoudre le problème en enchainant séquentiellement en premier lieu les phases de résolution heuristique associées à la recherche d'une solution et en second lieu les phases de rédaction, et ce, sans retours en arrière possibles. Pourtant, la résolution d'un problème de mathématique s'effectue rarement de manière séquentielle, et l'élève choisit rarement une stratégie au début de la résolution pour ensuite rédiger sa solution sans questionner sa stratégie ou la changer complètement : « *we learn by establishing connections and relationships, by building a web of ideas rather than a linear and logical sequence of implications; ideas grow synergetically rather than strictly on top of each other* » (Dreyfus, 1999, p.98). C'est pourquoi il apparaît pertinent de souligner la capacité qu'ont certains systèmes tutoriels de reconnaitre différentes stratégies présentes dans les actions des élèves et de permettre à ceux-ci de faire des allers-retours entre ces solutions distinctes tout au long du processus de résolution. Cette capacité de s'adapter dynamiquement aux changements de stratégie repose sur la *reconnaissance de plan,* qui implique qu'un système soit doté d'un algorithme qui permette de reconnaitre les différents plans mis en œuvre par l'élève en fonction de ses actions et de distinguer le plan dominant qui se dégage de ses actions les plus récentes afin de fournir à ce dernier une aide ou une rétroaction juste et adaptée à son statut cognitif du moment.

En ce qui concerne l'ordre dans lequel les étapes d'une solution donnée doivent être soumises au système tutoriel par l'élève, le concept informatique de chaînage permet au système tutoriel d'ordonner les étapes d'une solution afin de reconnaitre le raisonnement d'un élève. On appelle chaînage avant, le fait de n'utiliser que des hypothèses ou des résultats intermédiaires prouvés comme antécédents de chaque inférence ajoutée à la démonstration jusqu'à atteindre la conclusion. Le chaînage arrière consiste à faire le chemin inverse : on part d'une inférence ayant comme conséquent la conclusion et on ajoute, à rebours, des inférences pour prouver les antécédents qui ne sont pas des hypothèses du problème, jusqu'à ce que tous les antécédents non prouvés soient des hypothèses. Pour sa part, le chaînage mixte, évoqué par Py (2001), consiste en un mélange de chaînages avant et arrière. Comme ce chaînage n'impose pas d'ordre quel qu'il soit, nous allons assimiler le chaînage mixte à ce que nous appellerons une exploration libre des solutions.

Le fait d'imposer une structure en chaînage avant ou arrière a pour effet de contraindre l'élève à raisonner de manière séquentielle, allant à l'encontre d'une pensée éclatée nécessaire lors d'un processus de découverte et aussi d'une réalité fondamentale en éducation : « *students frequently understand and construct their knowledge in ways quite different from what is anticipated or planned, and confirms the basic thesis of constructivism that learning is idiosyncratic* » (De Villiers, 2007, p. 53).

2.3. Intervention tutorielle

Dans un premier temps, l'aide tutorielle peut être fournie à la demande de l'élève au moyen d'un bouton par exemple, ou bien être assurée de manière autonome par le système tutoriel. Dans le premier cas, plusieurs concepteurs soulèvent le risque d'abus des demandes d'aide chez l'élève qui ne souhaite pas s'investir dans la compréhension du problème à résoudre. En ce qui a trait à la gestion autonome de l'aide par le système tutoriel, ceci implique que ce dernier soit en mesure d'identifier avec justesse le moment où l'élève requiert de l'assistance. Dans les deux cas de figure, l'enjeu consiste à offrir de l'aide au bon moment sans dénaturer le problème à résoudre. Ce délicat équilibre entre directivité et liberté concerne l'ensemble des interventions tutorielles, qui se retrouvent sous trois formes dans les systèmes tutoriels analysés.

L'aide tutorielle peut être proactive et accompagner l'élève dans son raisonnement. L'accompagnement s'opère grâce à l'émission d'indices par rapport à la prochaine étape à effectuer. Cette prochaine étape peut être déterminée par le système tutoriel, qui a recours au chaînage, ou selon la reconnaissance de la stratégie actuellement mise en œuvre par l'élève.

Inversement, l'action tutorielle peut être rétroactive et réagir, après coup, aux actions de l'élève sans que ce dernier questionne directement le milieu. Les rétroactions peuvent simplement consister en une validation des actions de l'élève. Cette validation peut s'opérer localement après que l'élève ait soumis un énoncé (une justification, une conjecture ou une hypothèse) ou encore nécessiter l'entrée d'un triplet inférentiel (antécédent(s), justification, conséquent). Dans les deux cas, la validation est qualifiée de locale, car elle pose un jugement sur la validité d'une action indépendamment des actions posées précédemment ou du plan de résolution de l'élève. Naturellement, la validation peut aussi être globale et retourner à l'élève une appréciation générale de sa solution prise dans son ensemble. La solution peut prendre la forme d'une démonstration ou d'une construction géométrique selon les systèmes tutoriels étudiés. La validation est souvent communiquée à l'élève sous la forme d'un message très concis ou encore d'un symbole évocateur. Conséquemment, la validation ne fournit pas de précisions à l'élève quant à la pertinence, au rôle d'un énoncé ou d'une inférence correcte ou encore quand à la nature d'une erreur commise. La validation à elle seule est donc une forme de rétroaction assez primitive qui offre peu de soutien à l'élève en difficulté. C'est pourquoi la rétroaction peut aussi prendre d'autres formes plus complexes et détaillées. D'abord, en parallèle avec une validation qui souligne la présence d'erreurs, celles-ci peuvent être expliquées par le système tutoriel, donnant à l'élève la chance d'en comprendre la nature. Ce type de rétroaction suppose qu'un système reconnaisse ces déductions erronées, ce qui implique un travail préalable d'identification des erreurs les plus probables. Dans le même ordre d'idée, le système tutoriel peut annoter une solution finale et mettre en évidence les erreurs et les bons coups de l'élève, de manière à lui offrir une appréciation globale de la validité de son raisonnement. Cette façon d'intervenir du système tutoriel peut permettre à l'élève de prendre connaissance du contexte de chacun des commentaires du tuteur et ainsi, de mieux en saisir la portée. Finalement, le système tutoriel peut représenter les étapes de la solution fournies par l'élève sous forme de graphe déductif (Tanguay, 2005). Ce type de représentation facilite la visualisation par l'élève des liens logiques entre les propositions qu'il soumet en mettant en évidence la structure ternaire des inférences (Tanguay, 2006).

Maintenant que les différentes variables émergentes de l'analyse des systèmes tutoriels sélectionnés sont définies et expliquées, nous pouvons procéder à la comparaison des différents systèmes tutoriels entre eux en fonction de ces caractéristiques. Pour ce faire, nous allons décrire chacun des systèmes tutoriels, du plus ancien au plus récent, puisque nous ne souhaitons pas classer les systèmes les uns par rapport aux autres, mais plutôt dresser un portrait objectif de leurs caractéristiques. Les systèmes précédents QEDX sont décrit en fonction de la littérature disponible les concernant et aussi en fonction de notre expérience utilisateur à leur interface, lorsque celle-ci était disponible. En ce qui concerne

QEDX, sa description qui sera abordée dans la seconde partie de cet article, sera plus exhaustive puisque nous avons accès à l'interface mais aussi aux algorithmes et choix didactiques sous-jacents.

3. Synthèse du fonctionnement de systèmes tutoriels pour l'exercice de la démonstration en géométrie

3.1. Geometry Tutor (Anderson *et al.*, 1985, Ritter *et al.*, 2010)

Une première version de Geometry Tutor a été publiée en 1985 et la dernière version analysée date de 2010. Nous allons d'abord décrire la première de ces versions pour ensuite préciser la nature des modifications apportées pour donner lieu à la plus récente.

La fenêtre d'accueil de Geometry Tutor comprend un dessin de figure géométrique qui accompagne l'énoncé du problème ainsi qu'un début de réseau déductif (arbre de démonstration) où sont inscrites les hypothèses et la conclusion du problème. L'élève doit compléter ce réseau d'inférences qui reliera les hypothèses du problème à la conclusion en y ajoutant des nœuds, c'est-à-dire des déductions (conséquents) et des justifications. Le choix des concepteurs d'organiser la preuve sous forme de graphe avait pour objectif de communiquer la structure logique de la preuve et du raisonnement déductif. Geometry Tutor ne permet pas à l'élève d'ajouter des nœuds au fil de ses besoins puisque le système ne permet que le chaînage avant et arrière. Les quelques solutions admises par le système pour chaque problème sont préalablement programmées par un expert didacticien ou un enseignant. Le système tutoriel permet à l'élève d'explorer plusieurs solutions en même temps et effectue une reconnaissance de plan (*model-tracing paradigm for instruction*) en analysant pas à pas la validité locale des actions de l'élève et en les comparant aux solutions témoins. Chaque action de l'élève est analysée, et le système tutoriel détermine si elle est correcte ou erronée. Si elle est erronée, le tuteur intervient et offre un indice pour rediriger l'élève vers une des solutions admises. Si l'entrée appartient à une des solutions, le tuteur détermine si cette action est indicative d'un nouveau plan de solution, mais n'intervient pas tant que l'élève ne commet pas d'erreur. Si l'entrée n'appartient ni à une solution ni à une erreur connue, ou si l'élève requiert de l'aide, le système tutoriel dicte la prochaine étape du dernier plan reconnu.

Dans la version de 2010, au fur et à mesure que l'élève fait référence aux éléments géométriques de la figure dans les inférences du schéma déductif, ils sont surlignés à même la figure, qui obtient donc le statut de figure interactive. Quant aux rétroactions du système tutoriel, les inférences erronées sont dorénavant ornées de rouge, ajoutant à l'aspect visuel de la validation. En ce qui a trait à l'accompagnement et à l'aide à la prochaine étape, au lieu de dicter directement la prochaine étape du dernier plan de solution reconnu, la directivité des indices du système tutoriel évolue graduellement.

3.2. Angle (Koedinger & Anderson, 1990, Koedinger & Anderson, 1993)

Le logiciel Angle s'inspire des compétences d'experts en résolution de problèmes de démonstration pour assister l'apprenti géomètre. Son fonctionnement s'appuie sur l'hypothèse que les experts en résolution de problème de démonstration imaginent un plan de résolution (*implicit planning*, Koedinger et Anderson, 1993), qui est composé de moments clés tirés de la démonstration détaillée. Ces moments remarquables reposent sur l'adéquation *entre raisonnement de démonstration* et *construction géométrique*, et correspondent à des constructions intermédiaires qui découlent d'un ensemble de déductions et qui lient une figure de départ (les hypothèses) et une figure finale (la conclusion). Ces constructions intermédiaires sont appelées *Diagram configuration* (DC) et chaque solution admise pour un problème donné est composée d'un enchainement de ces figures géométriques indicatives d'une étape clé des solutions des experts. En quelque sorte, les DC montrent où l'attention du géomètre était concentrée à un moment précis de la démonstration. Par exemple, si l'utilisateur veut démontrer qu'un segment donné est la médiatrice BH de la base d'un triangle ABC, il peut d'abord identifier la perpendicularité de ce segment BH avec la base AC du triangle avec une construction qui illustre seulement ces deux segments et la relation entre eux. La démonstration détaillée prend donc la forme d'un réseau déductif où *énoncés discursifs* et *figures statiques* (dessins) cohabitent. À chaque fois que l'élève soumet un DC, les déductions (liens entre le DC) dont cette configuration intermédiaire découle sont considérées comme prouvées. L'élève peut compléter cet organigramme (graphe) au fil de son exploration du problème et dans l'ordre qui lui plait, et la figure interactive adjacente à la fenêtre du schéma s'anime lorsque l'élève fait mention des éléments figuraux dans sa solution. L'élève, qui peut explorer plusieurs pistes de solution simultanément, complète une solution lorsqu'il réussit à relier par une suite ininterrompue de DC et de déductions sous-jacentes les hypothèses (figure initiale) à la conclusion (figure finale). À la demande de l'élève, le système tutoriel peut le diriger vers le prochain DC (en chaînage avant) de la solution identifiée comme dominante parmi celles qu'il a travaillées pour ensuite l'aider à compléter le détail des déductions.

3.3. Chypre (Bernat, 1993)

Pour ce qui est de Chypre (acronyme pour Conjecture HYpothèse PREuve), l'élève ne rédige pas de démonstrations comme telles, mais crée plutôt un réseau déductif en soumettant des assertions (hypothèses, conclusions intermédiaires ou finale) découlant de l'examen de l'énoncé du problème et de l'exploration d'une figure dynamique. Ces énoncés sont tirés d'un répertoire limité, et l'élève peut les proposer dans l'ordre qui lui plaît; le système, doté d'un module de déduction automatique, valide de manière locale et autonome le statut logique d'une entrée (prouvée ou non) et ajoute les liens logiques entre les différentes propositions du réseau déductif. Chypre se distingue par sa capacité à identifier la justification à laquelle fait appel chaque inférence et l'ajoute automatiquement au réseau déductif, mais cet atout dépend d'une programmation humaine préalable. De ce fait, l'élève n'a pas à fournir de propriétés ou de définitions pour compléter une inférence ; le conséquent obtient donc le statut de prouvé dès que tous les antécédents nécessaires à l'inférence sont soumis. Le problème est résolu lorsque le réseau déductif est complet et ne contient que des conséquents prouvés. Le module d'aide suggère, sous forme de menu déroulant, les énoncés que l'élève peut sélectionner pour éventuellement générer son réseau déductif. Toutefois, Chypre n'offre pas d'aide du type « prochaine étape » et les affirmations hors contexte ne sont pas identifiées comme telles, ne donnant pas l'opportunité à l'élève de se rendre compte qu'il s'écarte des possibilités de raisonnements valides.

3.4. Mentoniezh (Py, 1994, Py, 1996, Py, 2001)

Le projet Mentoniezh, appellation qui signifie *géométrie* en breton (Py, 1996), est un système qui date du début des années 1990. Au sein de Mentoniezh, la résolution d'un problème de démonstration s'opère en quatre phases : « lecture et analyse de l'énoncé (1), exploration du problème et recherche d'un plan (2), élaboration de la démonstration (3) et rédaction de la preuve (4) » (Py, 1996, p. 229).

Durant la phase 1, l'élève dégage les informations données dans l'énoncé du problème, soit les hypothèses et la conclusion à démontrer. Comme Mentoniezh ne fournit pas de figure géométrique, l'élève qui souhaite raisonner à l'aide d'une construction géométrique réelle doit la réaliser lui-même, en dehors de l'environnement de Mentoniezh, à partir des informations contenues dans l'énoncé du problème. Ensuite, l'élève sélectionne un par un les énoncés qu'il souhaite soumettre à l'aide d'un menu déroulant de mots clés, les complète avec les paramètres appropriés et les classe dans un tableau à trois colonnes : une colonne pour les hypothèses, une colonne pour la conclusion à démontrer et une colonne pour les observations supplémentaires. Le système tutoriel procède à une validation locale de la syntaxe de chaque entrée et à une validation globale du tableau. Il est en mesure de retourner des messages quant au statut des affirmations, par exemple indiquer à l'élève qu'il a classé la conclusion dans les hypothèses, diriger l'élève vers un énoncé manquant ou soulever les conjectures mentionnées de manière prématurée. L'élève peut passer à la prochaine étape de la résolution seulement lorsqu'il a correctement complété le tableau pour lequel il n'y a qu'une solution programmée préalablement par un enseignant.

La seconde phase, destinée à l'exploration du problème, vise l'élaboration d'un plan de démonstration. L'exploration prend la forme d'un dialogue entre le système tutoriel et l'élève, où ce dernier doit statuer s'il serait en mesure de prouver des propriétés de la figure géométrique (décrite dans l'énoncé) ciblées par le tuteur. Au cours de cette phase, une fenêtre affiche l'état de la démonstration en précisant si les propriétés invoquées par le système tutoriel sont étudiées, en cours d'étude ou à étudier. Une propriété est considérée comme étudiée lorsque l'élève a correctement identifié un plan (théorèmes, antécédents) pour la démontrer. Si l'élève est bloqué, il peut demander de l'aide au système tutoriel, qui lui proposera des théorèmes en lui demandant s'ils sont applicables. L'élève peut passer à la prochaine phase de résolution lorsqu'il aura étudié toutes les propriétés et donc établi un plan pour chacune des inférences formant la démonstration globale.

Au cours de l'élaboration de la preuve, troisième phase du processus de résolution, l'élève construit une preuve valide en articulant hypothèses, théorèmes et conclusion, et ce, en chaînage avant, arrière ou mixte. Durant cette étape, l'interface est divisée en trois fenêtres : une fenêtre de travail (brouillon), une fenêtre où s'affiche l'état de la démonstration en fonction du statut des énoncés la constituant (données et propriétés démontrées, à démontrer et conjectures) et une troisième fenêtre où s'affiche l'inférence courante sur laquelle l'élève travaille. En effet, dans Mentoniezh, bien que l'élève puisse élaborer sa preuve sans ordre prescrit, il doit soumettre ses pas de déduction sous forme de triplets inférentiels (hypothèse, théorème, conclusion) au tuteur. Au fur et à mesure que les inférences sont soumises par l'élève, elles sont validées par le tuteur qui les compare à une liste d'inférences produites par un moteur de déduction automatique, et l'état de la démonstration est mis à jour. Bien que Mentoniezh sache reconnaitre le plan de résolution de l'élève, la dernière version disponible n'est pas en mesure de proposer d'aide à la prochaine étape et se contente de corriger les inférences proposées par l'élève. Conséquemment, l'intervention tutorielle décrite par Py ne s'appuie pas sur la reconnaissance de plan pour fournir une aide plus personnalisée à l'élève.

Finalement, la quatrième et dernière phase de résolution consiste en la rédaction en langage naturel de la démonstration. L'élève rédige chacune des phrases constitutives de la preuve à partir d'une liste d'éléments susceptibles d'y figurer (hypothèses, théorème, conclusion, mots de liaison, symboles de ponctuation), et le système tutoriel valide chacune de ces phrases en procédant à un contrôle de la validité grammaticale (ordonnancement des éléments) et à une vérification de la logique géométrique (contenu et continuité thématique).

3.5. Cabri-DEFI (Luengo & Balacheff, 1995) et Cabri Euclide (Luengo, 2005)

Cabri-DÉFI (DÉFI : Démonstration et exploration de la figure interactive) et son successeur Cabri-Euclide sont tous deux le résultat d'une fusion de Cabri-Géomètre, un environnement de géométrie dynamique, et d'un module de rédaction de démonstrations, respectivement DÉFI et Euclide. Le développement de ces deux systèmes successifs est fondé sur l'idée d'une distinction entre le processus non contraignant de résolution de problème qui s'articule grâce à la figure dynamique et la rédaction séquentielle d'une démonstration. Comme le dit Luengo, concepteur de Cabri-Euclide :

> *In Cabri-Euclide, we chose to create a figure (graphic) workspace for heuristic analysis and the production of pragmatic proofs, and a separate text workspace for intellectual proofs (processing of linguistic expressions and analysis of their organisation). The system manages the relations between these two workspaces.* (Luengo, 2005, p. 20)

Dans Cabri-DÉFI, l'élève peut construire une figure dynamique en fonction de l'énoncé de problème fourni et l'explorer librement. Le système tutoriel graphique valide la construction de l'élève en vérifiant la présence d'éléments figuraux clés à l'aide des oracles de Cabri-Géomètre. Une fois la construction complétée, le système tutoriel dirige l'attention de l'élève sur des éléments figuraux de la construction qui véhiculent une ou des déductions en commençant par la conclusion du problème et en parcourant, à rebours (en chaînage arrière), les déductions sous-jacentes. Selon les concepteurs de Cabri-DÉFI, cette démarche heuristique d'exploration guidée précédant l'activité de rédaction permet de mettre en lumière des sous-problèmes de démonstration plus élémentaires tout en étant constitutifs du problème original. Pour chaque conjecture ciblée, le tuteur demande à l'élève s'il se croit capable de la démontrer et, dès que l'élève se dit en mesure de se lancer, le système bascule vers le module de rédaction. Même si au début du problème, l'exploration heuristique de la figure précède l'activité de rédaction, en cours de résolution du problème, l'élève peut alterner librement entre les modules d'exploration figurale et de rédaction.

Pour rédiger sa démonstration, l'élève doit d'abord choisir, à l'aide de mots clés, un théorème qui justifie la conjecture retenue parmi une liste d'énoncés de géométrie euclidienne programmée par un expert. Il doit ensuite fournir les antécédents qui complètent l'inférence. Le module de rédaction fonctionne en chaînage avant, et l'élève doit compléter une inférence avant d'en entamer une nouvelle. Chaque inférence est vérifiée, à la demande de l'élève, par le système tutoriel qui s'assure d'une part, à l'aide d'un moteur de déduction automatique, de la validité de la déduction logique et, d'autre part, de la présence des éléments figuraux mentionnés dans l'inférence. Les messages d'erreur retournés portent sur le statut opératoire des énoncés qui forment les inférences et non sur la pertinence de l'inférence dans la résolution du problème. Toutefois, le système tutoriel vérifie aussi automatiquement que chaque inférence ne mène pas à une preuve sans issue et interrompt le travail de l'élève si c'est le cas.

Pour ce qui est de Cabri-Euclide, son fonctionnement ressemble quelque peu à celui de son prédécesseur Cabri-DÉFI, moyennant quelques ajouts. D'abord, une fenêtre entièrement gérée par Cabri-Graph (Carbonneaux *et al.*, 1995) a été ajoutée à l'espace de travail de l'élève. Cabri-Graph réorganise les inférences fournies par l'élève en un graphe déductif. L'élève peut s'y référer à n'importe quel moment pour vérifier la progression de son raisonnement sans toutefois pouvoir modifier directement le graphe.

En ce qui a trait à l'aide tutorielle à la démonstration, d'entrée de jeu, lorsque l'élève soumet un énoncé qui se révèle faux, le système tutoriel retourne un contre-exemple sous forme de construction géométrique permettant à l'élève de percevoir concrètement son erreur. Cabri-Euclide se démarque aussi de Cabri-DÉFI par sa capacité de réfutation et de négociation avec l'élève (Luengo, 2005). Les messages retournés par l'agent tuteur ressemblent davantage au propos que tiendrait un tuteur humain, et le système tutoriel s'adapte selon les arguments avancés par l'élève. Les concepteurs de cet environnement souhaitent même qu'une version ultérieure du logiciel accepte des théorèmes applicables qui n'étaient pas préalablement implémentés sur les bases d'une argumentation de l'élève validée par un moteur de déduction automatique. Cet aspect très prometteur révolutionnerait le potentiel des systèmes tutoriels pour l'apprentissage de la démonstration en géométrie. Toutefois, pour le moment, Cabri-Eulcide se limite à la validation locale des inférences et n'offre pas une aide adaptée ou des indices quand l'élève est incapable d'avancer, à cause du choix des concepteurs de ne pas donner de « réponses » à l'élève.

3.6. Baghera (Webber *et al.*, 2001)

Baghera est un environnement dont l'interface est divisée en trois fenêtres. Dans la première, on peut prendre connaissance de l'énoncé du problème. La seconde fenêtre est consacrée à une figure dynamique. Celle-ci a été préalablement conçue grâce au logiciel Cabri-Java par l'expert qui a créé le problème, et elle peut être manipulée librement par l'élève. Toutefois, Cabri-Java ne permet pas l'ajout d'éléments figuraux à la construction d'origine. La dernière fenêtre est dédiée à la rédaction d'une démonstration formelle en chaînage avant.

Les concepteurs de Baghera lui donnent l'appellation de système multi-agents puisque l'intervention tutorielle de Baghera intervient à plusieurs niveaux. En fait, le système comporte trois agents aidants : le compagnon, le tuteur et le médiateur. Le compagnon guide l'élève qui ne maîtrise pas les différents aspects techniques de l'interface du logiciel. Quant à lui, le tuteur gère l'itinéraire de problèmes proposés à l'utilisateur en fonction de son parcours personnalisé. Enfin, le médiateur fournit une aide à l'élève en processus de rédaction de démonstration en vérifiant systématiquement la validité locale de chacune des justifications et des conjectures les unes par rapport aux autres à mesure qu'elles sont soumises par l'élève. Une fois que l'élève indique avoir terminé sa démonstration, un moteur de déduction automatique procède à l'évaluation formelle de la démonstration et, en guise de rétroaction, propose une annotation de celle-ci pour souligner les erreurs, proposer des contre-exemples ainsi que des alternatives au raisonnement soumis.

Baghera est aussi doté d'une interface expert où l'enseignant peut composer de nouveaux problèmes à résoudre et où il peut dialoguer librement avec ses élèves, à distance, dans des classes virtuelles.

3.7. Turing (El-Khoury *et al.*, 2005)

Turing est le premier système développé par le laboratoire de recherche du même nom. Son évolution a menée au développement de GeogebraTUTOR (GGBT) et, plus récemment, de QEDX. Turing comprend deux interfaces : une pour l'enseignant et une pour l'élève.

Turing est conçu comme une plateforme où aucun contenu ou problème n'est initialement implémenté. L'interface enseignante demande à l'enseignant de concevoir ses propres problèmes et d'implémenter des solutions qui concordent avec le paradigme géométrique qu'il exige habituellement. Dans Turing, l'aide tutorielle n'est ni générée ni gérée par un système tutoriel autonome, mais doit être programmée préalablement par l'enseignant. Les messages d'aide associés à des énoncés déductifs prédéterminés (hypothèses, justifications, conjectures) sont donc formulés et divulgués en fonction des choix ad hoc de l'enseignant. Comme l'enseignant peut également prévoir des « solutions » sans issue découlant d'erreurs communément observées dans ses classes, il peut aussi enregistrer des messages d'aide pour rediriger l'élève lorsque ce dernier s'écarte des solutions admises. Bien que Turing permette une grande souplesse à l'enseignant, son utilisation exige un investissement non négligeable de sa part.

Pour sa part, l'élève a accès à une figure dynamique qui peut être manipulée, modifiée et complétée dans l'environnement Cabri-Géomètre, qui est intégré à l'interface de Turing. L'espace de travail de l'élève comprend aussi une fenêtre pour la construction d'une démonstration. L'élève y complète sa solution en soumettant des hypothèses, des conjectures et des justifications sans être contraint au chaînage avant ou arrière, et le système valide chacune des entrées en vérifiant sa présence dans les solutions expertes. L'apprenti géomètre peut explorer simultanément plusieurs chemins de solution, mais Turing ne fait pas de reconnaissance de plan puisque l'aide tutorielle ne dépend pas de l'identification de la solution travaillée par l'élève. Pour valider le raisonnement global de l'élève, le système tutoriel ordonne les pas de déduction soumis pour recréer les différentes avenues de démonstration explorées par l'élève et les comparer à celles implémentées préalablement par l'enseignant. L'élève est réputé avoir résolu le problème lorsqu'il a réussi à reproduire une des solutions expertes.

3.8. Advanced Geometry Tutor (Matsuda & VanLehn, 2003)

L'Advanced Geometry Tutor s'inspire de son ancêtre Geometry Tutor, mais la genèse des solutions aux problèmes, au lieu d'être assurée par un enseignant expert, est prise en charge par un moteur de déduction automatique. Advanced Geometry Tutor établit un parallèle entre la construction d'une figure géométrique et le processus déductif de démonstration. Le moteur de déduction automatique GRAMY (Matsuda & VanLehn, 2004) génère, de manière autonome, les démonstrations admissibles à partir des hypothèses initiales du problème, d'une figure initiale associée à l'énoncé du problème et de la conclusion à démontrer. Le moteur de déduction automatique connaissant la conclusion à prouver procède en chaînage avant à partir des hypothèses pour construire les solutions admissibles. Chemin faisant, à chaque fois qu'une solution fait appel à un élément figural absent de la figure courante, le moteur de déduction ajoute l'élément et redémarre en chaînage avant. Ainsi, le moteur de déduction génère une suite de propositions discursives formant une démonstration ainsi que l'ensemble des éléments figuraux manquant à la figure initiale pour faire correspondre le raisonnement déductif et la complétion de la figure géométrique.

L'élève doit rédiger une démonstration en deux colonnes (propositions, justifications), soit en chaînage avant soit en chaînage arrière, tout en complétant la figure au fur et à mesure que de nouveaux éléments figuraux sont mentionnés, faute de quoi le système refuse le pas de déduction proposé. La plupart de ces éléments à construire consistent en des segments reliant deux points de la figure initiale. Toutefois, comme la figure n'est pas dynamique mais interactive, l'ajout d'éléments figuraux se fait grâce à des commandes en mode texte.

Comme l'élève ne peut changer de stratégie une fois sa solution entamée, le système tutoriel identifie, dès les premières actions de l'élève, la stratégie de ce dernier. L'intervention tutorielle est de directivité variable et dépend du niveau de compétence de l'élève. Concrètement, le système valide automatiquement chaque action de l'élève et, ce faisant, mesure le niveau de compétence de ce dernier (il augmente si l'action est reconnue ou diminue s'il y a erreur). En s'appuyant sur ce diagnostic, le système tutoriel accompagne l'élève pour l'amener à compléter la prochaine étape de sa démonstration, soit en expliquant la prochaine étape et en l'exécutant pour l'élève, soit en laissant l'élève l'exécuter par lui-même ou encore, en encourageant simplement l'élève à procéder à la prochaine étape.

3.9. Agent Geom (Cobo *et al.*, 2007)

Agent Geom permet, d'une part, la construction et l'exploration de figures dynamiques et, d'autre part, la rédaction de preuves discursives. L'élève peut alterner librement entre le module figural et le module de démonstration tout au long de sa résolution du problème. Ce système permet l'exploration en parallèle de plusieurs solutions et ne contraint pas l'apprenant au chaînage avant ou arrière.

Agent Geom est aussi doté d'une interface destinée à l'enseignant, qui peut concevoir des problèmes ou sélectionner les énoncés préalablement implémentés qu'il souhaite soumettre à ses élèves en identifiant les chemins de solution qu'il juge admissibles en fonction de sa pratique enseignante.

Supporté par les oracles de l'environnement de géométrie dynamique, le système tutoriel procède à un diagnostic quantitatif (pourcentage) de complétion de la figure géométrique à construire. La validation de la démonstration s'opère grâce à une comparaison de la solution de l'élève à celle créée ou choisie par l'enseignant. Le système tutoriel renvoie à l'élève des messages préprogrammés par l'enseignant en fonction des lacunes observées dans la démonstration. La structure de l'intervention tutorielle ne prévoit pas la gestion autonome par le système tutoriel d'un accompagnement personnalisé. Néanmoins, l'enseignant volontaire peut programmer des messages et des instructions en fonction de difficultés qu'il aurait anticipées, et l'élève peut solliciter cette aide quand il juge qu'elle s'avère nécessaire.

3.10. Geometrix (Gressier, 2011)

Geometrix est divisé en deux interfaces : une pour l'enseignant et une autre pour l'élève. La conception de problèmes de démonstration et l'élaboration des solutions s'opèrent grâce à la collaboration entre l'enseignant et un moteur de déduction automatique. Pour générer un problème de preuve, l'enseignant construit une figure géométrique (finale) et cache ensuite une partie des traces de cette construction pour générer une figure initiale. Cette figure initiale accompagnera l'énoncé du problème proposé à l'élève. À partir de la figure finale, le moteur de déduction automatique génère une liste de propriétés démontrables et demande à l'enseignant de choisir la conclusion qu'il souhaite voir l'élève démontrer en fonction du problème qu'il souhaite concevoir.

L'interface de l'élève est divisée en deux modules : un pour la reproduction de la construction finale de l'enseignant et un pour la construction de la démonstration discursive associée. Dans le module de construction, l'élève doit dupliquer la figure finale de l'enseignant en reproduisant exactement la démarche de l'enseignant faute de quoi le système ne reconnaitra pas le processus de construction. L'enseignant peut lui-même prévoir des messages à l'intention de l'élève, mais les messages divulgués automatiquement par le système tutoriel en fonction des constructions intermédiaires de l'enseignant sont souvent dépourvus de sens pour l'élève puisqu'ils font référence aux mêmes étapes de la construction avec lesquelles l'élève éprouve des difficultés.

Pour ce qui est de la rédaction de la démonstration, l'élève doit d'abord avoir résolu le problème de construction, qui lui donne accès à la figure finale ainsi qu'à l'énoncé du problème de démonstration. L'élève qui souhaite explorer les propriétés de la figure dynamique doit le faire dans le module de construction puisqu'une fois dans le module de rédaction, la figure fournie est statique. L'énoncé est accompagné de la liste des hypothèses et des justifications (avec explications) à employer et de la conclusion à démontrer. L'élève doit former des triplets inférentiels (hypothèses, justification, conséquent) et lorsqu'un conséquent est prouvé, il obtient le statut de donnée et peut être employé comme antécédent pour un prochain triplet inférentiel. Autrement dit, l'élève construit graduellement la preuve en chaînage avant jusqu'à ce qu'il complète le dernier triplet inférentiel qui démontre la conclusion au problème. La figure interactive met en surlignage les éléments figuraux au fur et à mesure que ceux-ci sont choisis par l'élève. Chaque triplet inférentiel est validé par le système, et ce dernier attire l'attention de l'utilisateur sur les erreurs commises le cas échéant (exemple, une hypothèse manquante pour démontrer une conjecture). Une fois la démonstration complétée, l'élève peut la visualiser.

4. Synthèse

Voici la grille d'observation complétée (Tableau II) en fonction des caractéristiques propres à chaque système tel que décrit dans la section précédente. Le crochet (✔) signifie que le système présente cette caractéristique et le rond (◦) signifie que cette particularité est envisagée pour le développement futur du système concerné.

Tableau II
Grille synthèse du fonctionnement des systèmes tutoriels analysés

	Intégration de la figure géométrique				Structure prévue pour l'exercice du raisonnement de démonstration géométrique						Intervention tutorielle							
	Figure statique		Figure dynamique		Solutions admises				Ordres des entrées		Intervention		Accompa-gnement	Rétroaction				
														Validation				
	Dessin	Figure interactive	Exploration	Construction	Moteur de déduction automatique	Programmées par un expert ou un enseignant	Exploration simultanée de multiples stratégies / Reconnaissance de plan	Phases séquentielles du raisonnement	Chaînage (avant, arrière)	Chaînage mixte ou exploration libre	À la demande de l'élève	Gérée par le système tutoriel	Indication prochaine étape (chaînage)	Locale : énoncés, inférences	Globale : solution, démonstration	Démonstration en graphe (réseau déductif)	Annotation de la démonstration	Explication des erreurs
Geometry Tutor		✔				✔	✔		✔		✔	✔	✔	✔		✔		✔
Angle	✔	✔				✔	✔			✔	✔	✔	✔	✔		✔		
Chypre			✔	✔	✔	✔				✔	✔	✔		✔		✔		
Mentoniezh					✔	✔	✔	✔		✔	✔			✔	✔			
Cabri-DÉFI			✔	✔	✔	✔		✔	✔		✔	✔		✔				
Cabri-EUCLIDE			✔	✔	✔	✔		✔	✔		✔	✔		✔		✔		✔
Baghera			✔		✔				✔			✔		✔	✔		✔	✔
Turing			✔	✔		✔	✔			✔		✔		✔	✔			
Advanced Geometry Tutor		✔			✔				✔			✔	✔	✔				
Agent-Geom			✔	✔		✔	✔			✔		✔		✔	✔			
Géometrix		✔	✔	✔	✔	✔		✔	✔			✔	✔	✔				

Si nous nous concentrons sur le premier volet de notre analyse, soit *l'intégration de la figure géométrique au sein du système tutoriel*, l'analyse comparative révèle de la section 4 révèle que les systèmes tutoriels les plus récents, Chypre et les suivants utilisent principalement la géométrie dynamique comme moyen pour l'élève d'explorer les propriétés de la figure géométrique, à l'exception de Mentoniezh qui ne fournit pas de figure géométrique et de l'Advanced Geometry Tutor qui intègre une figure statique interactive. De plus, la plupart de ces figures géométriques évoluent dans un authentique micromonde où l'élève peut, d'une part, explorer les limites de celles-ci et d'autre part, construire de nouveaux éléments figuraux. Seul Baghera ne permet pas à l'élève de modifier la figure initiale par un processus de construction géométrique.

En ce qui concerne la seconde catégorie qui articule notre analyse, c'est-à-dire la *structure prévue pour l'exercice du raisonnement de démonstration géométriqu*e, on remarque quant aux solutions admises que la plupart des systèmes exigent une programmation humaine à un moment ou à un autre pour assurer leur fonctionnement. De plus, les quelques systèmes tutoriels qui reposent entièrement sur un moteur de déduction automatique, soit Baghera et Advanced Geometry Tutor, forcent l'élève à travailler en chaînage. Nous avons aussi remarqué une relation entre le déroulement imposé au raisonnement global et à l'ordonnancement exigé pour la rédaction de la démonstration (le chaînage). En effet, une minorité des systèmes tutoriels imposent un déroulement séquentiel au raisonnement de l'élève, mais ces systèmes, soit Cabri-DÉFI, Cabri-EUCLIDE et Géométrix, obligent aussi l'élève à opérer en chaînage avant. Finalement, le peu de systèmes qui accomplissent la reconnaissance de plan, la fonction qui permet à l'élève d'explorer plusieurs stratégies simultanément (Geometry Tutor, Angle, Montoniezh, Turing et Agent-Geom), reposent aussi sur la programmation des solutions par un expert et non par un moteur de déduction automatique. En somme, on peut déduire qu'une approche qui admet les aléas du raisonnement de l'élève en cours de processus de découverte ne semble pas compatible avec le paradigme de géométrie formelle imposé par un moteur de déduction automatique.

Pour ce qui est des interventions tutorielles, dernier volet de notre étude comparative, la majorité des systèmes tutoriels, à l'exception de Mentoniezh, gèrent en totalité ou en partie le déclenchement des rétroactions ou des messages d'aide à la prochaine étape. Par ailleurs, cette dernière forme d'accompagnement, qui implique que le système tutoriel puisse identifier la solution travaillée par l'élève pour le guider vers la prochaine action à poser, est rare. Seuls Geometry Tutor, Angle, Advanced Geometry Tutor et Géométrix sont en mesure d'offrir des indices quant au prochain pas de raisonnement attendu. En outre, l'aptitude à fournir une aide à la prochaine étape semble aller de pair avec l'imposition à l'élève d'une structure plus rigide pour l'exercice de son raisonnement. En effet, parmi les 11 systèmes tutoriels étudiés, Angle est le seuls à accomplir la reconnaissance de plan et à fournir une aide à la prochaine étape tout en laissant l'élève travailler des solutions en exploration libre (sans chaînage). En ce qui a trait à la rétroaction, tous les systèmes tutoriels analysés offrent une rétroaction sous forme d'une validation locale des inférences ou des énoncés soumis par l'élève, et on recense peu de validation globale de démonstrations (Mentoniezh, Baguera, Turing et Agent-Geom), et encore moins d'annotation des solutions (Baguera) ou d'explication des erreurs (Geometry Tutor, Cabri-EUCLIDE et Baguera). De plus, seulement quatre systèmes tutoriels, soit Geomtry Tutor, Angle, Chypre et Cabri-EUCLIDE, ont recours au réseau déductif pour permettre à l'élève de visualiser sa démonstration dans son ensemble. Ainsi, la rétroaction fournie par les systèmes étudiés semble privilégier une validation locale d'actions isolées (première ramification de la colonne rétroaction) au détriment d'une appréciation de la démonstration et de la structure logique sous-jacente dans leur globalité, soit les modes de rétroaction décrits par les quatre dernières ramifications de la colonne rétroaction.

Conclusion

Le présent bilan permet une vue d'ensemble des technologies disponibles pour l'accompagnement d'élèves en résolution de problème de démonstration en géométrie plane. Cette analyse a aussi permis de dégager des ilôts conceptuels autour desquels, grâce à une recherche didactique poussée, nous avons construit un ensemble de concepts et de positions théoriques sur lesquel s'est appuyé la conception du tutoriel QEDX qui sera l'objet de la deuxième partie de notre article. Éventuellement, les indicateurs retenus pour comparer les systèmes tutoriels pourront aussi servir de base théorique pour une analyse plus fine du fonctionnement de ces systèmes. Cette analyse s'appuyant sur les fondements de modèles et de théories didactiques établis permettra un regard sur le travail mathématique assisté par des système tutoriels et sur la contribution de ceux-ci aux apprentissages géométriques.

Bibliographie

ANDERSON J. R., BOYLE F. & YOST G. (1985), The goemetry tutor, *Advanced Computer Tutoring Project*.

BALACHEFF N. (1994), Didactique et intelligence artificielle, *Recherche en didactiques des mathématiques* **14(1/2)**, 9-42.

BALACHEFF N. & MARGOLINAS C. (2005), *Modèle de connaissances pour le calcul de situations didactiques,* La Pensée Sauvage, Grenoble.

BAULAC Y. (1990), *Un micromonde de géométrie, cabri-géomètre*, Thèse de Doctorat, Université Joseph Fourier.

BERNAT P. (1993), Chypre : Un logiciel d'aide au raisonnement, *Repères-IREM* **10**, 25-46.

BROUSSEAU G. (1998), *Théorie des situations didactiques,* La Pensée Sauvage, Grenoble.

CARBONNEAUX Y., LABORDE J.-M. & MADANI R. M. (1995), Cabri-graph : A tool for research and teaching in graph theory, dans *Graph drawing* (Eds. F. J. Brandenburg), **1027**, 123-126, Springer, Berlin Heidelberg.

COBO P., FORTUNY J. M., PUERTAS E. & RICHARD P. R. (2007), Agentgeom: A multiagent system for pedagogical support in geometric proof problems, *International Journal of Computers for Mathematical Learning,* **12**, 57-79.

COUTAT S., LABORDE C. & RICHARD P. R. (2016), L'apprentissage instrumenté de propriétés en géométrie : Propédeutique à l'acquisition d'une compétence de démonstration, *Educational Studies in Mathematics,* **93(2),** 195–221.

DREYFUS T. (1999), Why johnny can't prove, *Educational Studies in Mathematics,* **3**, 85-109.

EL-KHOURY S., RICHARD P. R., AIMEUR E. & FORTUNY J. M. (2005), Development of an intelligent tutorial system to enhance students' mathematical competence in problem solving, Dans *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (Eds. G. Richards), Vancouver.

ELSOM-COOK M. (1990), *Guided discovery tutoring,* Paul Chapman Publishing, Londre.

GRESSIER J. (2011), *Http ://geometrix.Free.Fr/jgressier/*, en ligne, 2011.

HANNA G. (2000), Proof, explanation and exploration: An overview, *Educational Studies in Mathematics,* **44(1/2),** 5-23.

HOUDEMENT C. & KUZNIAK A. (2006), Paradigmes géométriques et enseignement de la géométrie, *Annales de didactique et de sciences cognitives,* **11**, 175-193.

KOEDINGER K. R. & ANDERSON J. R. (1990), Theoretical en empirical motivations for the design of angle : A new geometry learning environment, *Knowledge-Based Environments for Learning and Teaching, AAAI Spring Symposium Series*, Standford.

KOEDINGER K. R. & ANDERSON J. R. (Eds.) (1993), *Reifying implicit planning in geometry: Guidelines for model-based intelligent tutoring system design,* Hillsdale.

LABORDE C. & CAPPONI B. (1994), Cabri-géomètre constituant d'un milieu pour l'apprentissage de la notion de figure géométrique, *Recherche en didactiques des mathématiques,* **14(1.2),** 165-210.

LUENGO V. (2005), Some didactical and epistemological considerations in the design of educational software : The cabri-euclide example, *International Journal of Computers for Mathematical Learning,* **10**, 1-29.

LUENGO V. & BALACHEFF N. (1995), Contraintes informatiques et environnements d'apprentissage de la démonstration en géométrie.

MATSUDA N. & VANLEHN K. (2003), Modeling hinting strategies for geometry theorem proving, *9th international conference on User modeling.* Springer-Verlag, Johnstown, Pensylvania.

MATSUDA N. & VANLEHN K. (2004), Gramy: A geometry theorem prover capable of construction, *Journal of Automated Reasoning,* **32**, 3-33.

PAILLE P. & MUCCHIELLI A. (2013), *L'analyse qualitative en sciences humaines et sociales (3e édition),* Armand Colin, Paris.

PLUVINAGE F. & RIGO LEMINI M. (2008), Mais non, marina!, *Annales de didactique et de sciences cognitives,* **13**, 40-61.

PY D. (1994), Reconnaissance de plan pour la modélisation de l'élève, le projet mentoniezh, *Recherche en didactiques des mathématiques,* **14(1.2),** 113-138.

PY D. (1996), Aide à la démonstration en géométrie : Le projet mentoniezh, *Sciences et techniques éducatives,* **3(2),** 227-256.

PY D. (2001), *Environnements interactifs d'apprentissage et démonstration en géométrie*, Thèse de doctorat, L'Université de Rennes 1.

RABARDEL P. (1995), *Les hommes et les technologies : Approche cognitive des instruments comtemporains,* Armand Colin, Paris.

RICHARD P. R. & FORTUNY J. M. (2007), Amélioration des compétences argumentatives à l'aide d'un système tutoriel en classe de mathématique au secondaire, *Annales de didactique et de sciences cognitives,* **12**, 83-116.

RITTER S., TOWLE B., MURRAY R. C., HAUSMANN R. G. M. & CONNELLY J. (2010), A cognitive tutor for geometric proof, dans *Lecture notes in computer science* (Eds. J. K. V. Aleven, And J. Mostow), **6095**, Springer, Berlin.

TANGUAY D. (2005), Apprentissage de la démonstration et graphes orientés, *Annales de didactique et de sciences cognitives,* **10**, 55-93.

TANGUAY D. (2006), Comprendre la structure déductive en démonstration, *Envol*, **134**, Groupe des responsables en mathématique au secondaire inc, Boucherville, Québec.

TESSIER-BAILLARGEON M., RICHARD P. R., LEDUC N. & GAGNON M. (2014), Conception et analyse de geogebratutor, un système tutoriel intelligent : Genèse d'un espace de travail géométrique idoine, *Revista latinoamericana de investigación en matemática educativa,* **17(4-I),** 191-210.

VILLIERS M. D. (2007), Some pitfalls of dynamic geometry software, *Teaching Mathematics,* **4**, 46-52.

WEBBER C., BERGIA L., PESTY S. & BALACHEFF N. (2001), Baghera project: A multi-agent architecture for human learning, *Workshop Multi-Agent Architectures for Distributed Learning Environments, AIED2001,* San Antonio, Texas.

MICHELE TESSIER-BAILLARGEON
PHILIPPE R. RICHARD
Université de Montréal
Faculté des sciences de l'éducation
Département de didactique
Case postale 6128, succursale Centre-Ville
Montréal, QC H3C 3J7
Tb_michele@hotmail.com
Philippe.r.richard@umontreal.ca

NICOLAS LEDUC
MICHEL GAGNON
École Polytechnique Montréal
2900 Boulevard Edouard-Montpetit
Montréal, QC H3T 1J4
nicolas.leduc@polymtl.ca
michel.gagnon@polymtl.ca