



HAL
open science

Génération d'exercices de diagnostic de compétences en algèbre

Dominique Prévité, Elisabeth Delozanne, Brigitte Grugeon

► **To cite this version:**

Dominique Prévité, Elisabeth Delozanne, Brigitte Grugeon. Génération d'exercices de diagnostic de compétences en algèbre. Jun 2007. hal-00161430

HAL Id: hal-00161430

<https://hal.science/hal-00161430v1>

Submitted on 10 Jul 2007

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

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

Génération d'exercices de diagnostic de compétences en algèbre

Dominique Prévité*,, Elisabeth Delozanne***, Brigitte Grugeon******

* LIUM

Avenue Olivier Maessian
72085 LE MANS CEDEX 09
dominique.previt@bretagne.iufm.fr

** IUFM de Bretagne

153, rue de Saint-Malo CS 54310
35043 RENNES CEDEX

*** CRIP5

Université René Descartes
45 rue des Saints Pères
75270 Paris Cedex 6
elisabeth.delozanne@math-info.univ-paris5.fr

**** DIDIREM

Université Paris VII
2 Place Jussieu
75 251 PARIS Cedex 5
brigitte.grugeon@amiens.iufm.fr

RÉSUMÉ. Dans cet article nous présentons un modèle de génération d'exercices de diagnostic de compétences. Ce travail s'inscrit dans le cadre du projet Lingot, projet pluridisciplinaire qui s'attache au diagnostic de la compétence algébrique des élèves en fin de collège. Nous définissons des modèles paramétrés d'exercices qui permettent à un concepteur non informaticien (enseignant, didacticien) de fixer les valeurs des paramètres et de faire générer par notre système PépiGen, d'une part l'interface de l'exercice pour l'élève et, d'autre part, la grille d'analyse des réponses des élèves à cet exercice. Cette analyse ne se limite pas à évaluer la validité des réponses mais, en s'appuyant sur des études de didactique de l'algèbre, permet de caractériser la nature des erreurs et des conceptions idiosyncratiques ou non, sous-jacentes aux réponses.

MOTS-CLÉS : diagnostic de compétences, algèbre élémentaire, génération automatique d'exercices

1. Introduction

Dans cet article nous proposons un modèle pour générer des exercices de diagnostic cognitif qui mettent en œuvre des procédures complexes d'analyse multidimensionnelle des réponses des apprenants. Entre des modélisations très génériques de type QTI [IMS-QTI 2006] qui ne permettent guère d'aller plus loin que de vérifier la validité de réponses simples (e.g. QCM et textes à trous) et des traitements ad-hoc de réponses complexes, nous proposons de définir des classes d'exercices paramétrables.

Notre travail s'inscrit dans le cadre du projet Lingot dont l'objectif est de concevoir un système qui assiste les enseignants dans la régulation des apprentissages en algèbre des élèves de leur classe. Ce projet s'appuie sur un travail en didactique des mathématiques [GRUGEON 1995] qui a produit un modèle de la compétence algébrique dans l'enseignement secondaire prenant en compte trois composantes : la capacité à réaliser du calcul algébrique, l'utilisation de l'outil algébrique pour résoudre les problèmes et la capacité à interpréter et à traduire algébriquement dans différents cadres (géométrique, graphique et langage naturel). Afin de situer les élèves par rapport à ce modèle, [GRUGEON 1995] a proposé un outil de diagnostic papier-crayon permettant de repérer, non seulement des erreurs, mais surtout des cohérences dans le fonctionnement des élèves en algèbre élémentaire.

Cette analyse didactique a permis de définir une grille d'analyse des réponses usuelles des élèves à ces exercices. Chaque réponse d'un élève est codée selon plusieurs dimensions : la validité de la réponse, l'utilisation des lettres (e.g. inconnue, variable, nombre généralisé, abréviation), le calcul algébrique (e.g. utilisation correcte ou non des parenthèses, identification des opérations, règles de calcul correctes ou non), la traduction entre différents registres sémiotiques (graphique, géométrique, algébrique, langue naturelle), les types de justification (e.g. par l'algèbre, par l'exemple, par l'utilisation de règles fausses). Une analyse transversale des différents codes obtenus permet de détecter des cohérences dans les réponses et de construire le profil cognitif de l'élève en algèbre.

[JEAN 2000] a conçu et réalisé le logiciel Pépite qui opérationnalise ce travail de didactique. Un enseignant peut ainsi faire passer un test à ses élèves sur le logiciel Pépite. Le logiciel code les réponses des élèves et construit automatiquement un profil cognitif de chaque élève. L'enseignant peut ensuite consulter, compléter ou corriger le diagnostic établi par le logiciel. Ce premier prototype avait pour principal objectif de tester la faisabilité de l'automatisation du diagnostic. Les exercices y sont figés et les procédures de diagnostic sont définies de façon ad hoc. Or les expérimentations et les retours d'usage de ce logiciel [DELOZANNE et al, 2002] ont fait apparaître qu'une des conditions pour qu'un tel système s'intègre dans l'activité des enseignants est qu'il puisse être utilisé à différentes étapes de la construction de la compétence en algèbre et à différents niveaux de classe.

Dans cet article, nous étudions dans quelle mesure nous pouvons mettre en évidence dans les exercices de Pépite des invariants qui nous permettent de définir un modèle conceptuel plus générique permettant d'envisager la génération automatique ou assistée d'exercices de diagnostic. Notre objectif est de concevoir un système qui génère des exercices de diagnostic à partir de modèles paramétrés d'exercices. Pour certains exercices les paramètres prennent des valeurs aléatoires sous certaines contraintes comme dans [Auzende et al. 2007]. Pour d'autres, certains enseignants souhaitent spécifier eux-mêmes les paramètres. Nous nous limitons dans ce texte à ce dernier type d'exercices. Le scénario d'utilisation est le suivant : un modèle paramétré d'exercices est mis au point par un informaticien à partir de spécifications établies par un spécialiste en didactique. Puis un concepteur non informaticien (enseignant ou didacticien) fixe les paramètres et fait générer par le système PépiGen un exercice particulier, c'est-à-dire d'abord l'interface de l'exercice pour les élèves puis la grille d'analyse des réponses des élèves à cet exercice. Lorsqu'un élève répond à l'exercice pour produire le codage automatique des réponses, une procédure de diagnostic unifie la réponse des élèves à la grille d'analyse de l'exercice.

Dans la section 2, nous caractérisons les différentes classes d'exercices du logiciel Pépite. Dans les sections suivantes, en nous appuyant sur une classe particulière, celle du « Prestidigitateur », nous décrivons le modèle de classe d'exercices, puis nous présentons PépiGen le module de génération d'un exercice à partir d'un modèle de classe et des paramètres rentrés par un concepteur.

2. Caractérisation des différentes classes d'exercices

Nous avons d'abord regroupé les exercices en classes d'exercices, puis, pour chaque classe d'exercices, nous avons mis en évidence les paramètres et les contraintes sur les paramètres qui permettent de générer des exercices équivalents du point de vue du diagnostic cognitif. Les exercices sont caractérisés selon trois axes : la *catégorie de l'exercice*, les *compétences mises en jeu* et les *interactions élève logiciel*. Dans Pépite nous distinguons trois catégories d'exercices associées aux trois composantes de la compétence algébrique. A chaque catégorie d'exercices correspond un certain nombre de compétences, telles qu'elles sont exprimées dans les programmes de l'enseignement secondaire français [BO 2004], Ce sont par exemple :

- pour la catégorie calcul algébrique : réduire une expression littérale à une variable, développer une expression littérale à une variable, connaître et utiliser les identités remarquables, résoudre une équation du premier degré à une inconnue ;
- pour la catégorie utilisation de l'algèbre pour résoudre des problèmes : produire une formule, mettre en équation, démontrer des propriétés, prouver des résultats, généraliser ;

- pour la catégorie traduction dans différents cadres : traduire algébriquement, reconnaître différentes écritures d'une même expression, associer différentes représentations d'une expression.

A l'interface, les exercices mettent en jeu différents types d'interaction entre les élèves et le logiciel : clic sur des cases ou des zones, déplacement d'objets (e.g. étiquettes, curseurs), saisie de texte en langage mathématique, naturel ou mathurel (mélange de langage mathématique et naturel), le texte à saisir pouvant être libre ou contraint (menu, formulaire, palettes de termes).

Du point de vue de la génération d'exercices, nous disons que deux exercices sont équivalents s'ils appartiennent à la même catégorie, permettent de tester les mêmes compétences et proposent le même type d'interaction. Une classe d'exercices est un ensemble d'exercices équivalents. Deux exercices équivalents se distinguent par les valeurs d'un ou plusieurs paramètres auxquels sont associées des variables didactiques. Nous illustrons ce modèle dans la section suivante.

3. Modélisation d'une classe d'exercices

Le modèle d'une classe d'exercices est composé de la description de l'interface élève, des compétences visées par l'exercice, des paramètres associés aux variables didactiques mises en jeu, et de la grille d'analyse des réponses.

3.1. L'interface élève

Du point de vue élève, elle est constituée d'un énoncé, d'une zone de saisie pour les justifications, d'un bouton radio pour le choix d'une réponse. Les possibilités d'actions de l'élève sur le système sont la lecture de l'énoncé, la saisie des justifications et la sélection d'une option (Cf. Figure 1). Par exemple deux instances de la classe du « Prestidigitateur » peuvent être associées aux énoncés suivants.

Énoncé 1 : Un prestidigitateur est sûr de lui en réalisant le tour suivant. Il dit au joueur : « Tu penses un nombre, tu ajoutes 8, tu multiplies par 3, tu retranches 4, tu ajoutes ton nombre, tu divises par 4, tu ajoutes 2, tu soustrais ton nombre : tu trouves 7. Prouve-le ». Indique si cette affirmation est vraie ou fausse. Justifie ta réponse.

Énoncé 2 : « Tu prends un nombre, tu ajoutes 6 à ce nombre, tu multiples le résultat par 3, tu soustrais 3 fois le nombre de départ : tu trouves 18. Prouve-le ». Indique si cette affirmation est vraie ou fausse. Justifie ta réponse.

Un prestidigitateur est sûr de lui en réalisant le tour suivant. Il dit au joueur :
 « Tu penses un nombre, tu ajoutes 8, tu multiplies par 3, tu retranches 4, tu ajoutes ton nombre, tu divises par 4, tu ajoutes 2, tu soustrais ton nombre. Tu as trouvé 7. Prouve-le. »

Indique si cette affirmation est vraie ou fausse. Justifie ta réponse.

Justification

Réponse

L'affirmation est : vraie fausse

Figure 1. Interface élève pour un exercice de la classe « Prestidigitateur »

3.2. Les paramètres

Les deux instances présentées en 3.1. sont caractérisées par l'expression algébrique représentant le programme de calcul donné dans l'énoncé. La classe d'exercices « Prestidigitateur » est donc paramétrée par cette expression. A ce paramètre sont associées des variables didactiques qui caractérisent la complexité de l'exercice (niveau de parenthèses, nombre et type d'opérations) et le niveau scolaire adapté à cet exercice (5°, 4°, 3° ou 2°).

L'expression algébrique obtenue pour l'énoncé 1, $((x+8)^3-4+x)/4+2-x$ correspond au niveau de la classe de troisième : elle comporte deux niveaux de parenthèses, sept opérateurs dont une division et l'élève a la possibilité de réduire au même dénominateur ou de simplifier l'expression. L'expression obtenue pour l'énoncé 2, soit $(x+6)^3-3x$ comporte 4 opérateurs, ne comporte pas de division et correspond à un niveau de quatrième. Ces deux exercices sont équivalents du point de vue de la génération d'exercices. L'interface est la même, ils appartiennent à la même catégorie d'exercices et mettent en œuvre les mêmes compétences. Par contre ils sont associés à des paramètres différents dont les variables didactiques ont des valeurs différentes. La grille d'analyse des réponses est générée pour chacun d'eux en fonction des valeurs du paramètre et des variables didactiques qui lui sont associées, ce qui est illustré en reprenant notre exemple dans la section qui suit.

3.3. La grille d'analyse des réponses et le diagnostic

Selon l'analyse didactique [Grugeon 1995], sur cette classe d'exercices, les élèves utilisent deux types de preuve : une preuve algébrique ou une preuve pragmatique (démarche arithmétique et justification par l'exemple). A chaque type de preuve sont associés un commentaire et un codage différent.

Par exemple dans le cas de l'énoncé 1, l'expression algébrique attendue est : $((x+8)3-4+x)/4+2-x$.

Parmi les élèves qui utilisent une preuve algébrique, certains utilisent une démarche globale. Ils écrivent l'expression $((x+8)3-4+x)/4+2-x$. Ils développent l'expression $(x+8)3$, puis réduisent au même dénominateur ou simplifient. Cette démarche correspond à une preuve algébrique qui est codée T1 car la solution est correcte, L1 pour une utilisation correcte des lettres, C1 car la conversion du français à l'algèbre est correcte et R1 car l'élève propose une justification par l'algèbre qui est aussi correcte. Réduire au même dénominateur est codée M1, simplifier M2.

<p>Lecture de la première ligne de la réponse</p> <p>Tant que la ligne n'est pas vide (exercice non terminé)</p> <ul style="list-style-type: none"> - Tant que la ligne est une phrase (3 lettres consécutives) lire la ligne suivante Fin tant que - Si l'expression est algébrique ou numérique (existence d'une lettre dans l'expression) <ul style="list-style-type: none"> alors démarche = numérique (ou algébrique) - Si l'expression est globale ou partielle (pas à pas) <ul style="list-style-type: none"> alors expression = globale (ou partielle) - Construire un arbre représentant chaque expression algébrique ou numérique écrite sur la ligne - Comparer avec les arbres représentant les expressions prévues par la grille d'analyse des réponses - Attribuer le codage correspondant - Mettre à jour la liste des règles fausses identifiées éventuellement <p>Fin Tant que</p>

Tableau 1. Algorithme de la procédure de diagnostic de la réponse de l'élève

D'autres élèves adoptent une démarche qui utilise des expressions algébriques partielles : $(x+8)3=3x+24$; $3x+24-4 = 3x+20$; $3x+20+x= 4x+20$; $(4x+20)/4 = x+5$; $x+5+2=x+7$; $x+7-x=7$. Cette démarche est codée T2, L1, C2, M1, R1. De plus des erreurs peuvent apparaître dans l'application des règles de calcul algébrique. Une erreur fréquemment rencontrée est une utilisation inadaptée des parenthèses qui conduit à un résultat incorrect : $(4x+20)/4 = 4x+20/4 = 4x+5$ (codage M32) ou à un résultat correct $(4x+20)/4 = 4x+20/4 = x+5$ car l'élève a gardé en mémoire l'opération initiale $(4x+20)/4$ (codage M31).

Enfin, il se peut que l'élève utilise une démarche non algébrique (preuve par des exemples). Par exemple, il peut choisir une valeur pour x et proposer une solution qui met en jeu des expressions partielles. Pour 1, $(1+8)3 = 27$; $27-4 = 23$; $23+1=24$; $24/4 = 6$; $6+2 = 8$; $8-1= 7$ réponse qui sera codée T3, L5, R2, C2, M1.

Le diagnostic est la procédure qui prend en entrée la grille d'analyse des réponses d'une part, les productions de l'élève d'autre part et détermine le codage de la réponse et la liste des règles fausses identifiées éventuellement dans cette réponse. Le tableau 1 donne l'algorithme de la procédure d'analyse des réponses.

4. Module de génération d'un exercice d'une classe

A partir du modèle et des paramètres définis par le concepteur de l'exercice (un enseignant ou un didacticien), le logiciel PépiGen génère l'énoncé et la grille d'analyse des réponses correspondant.

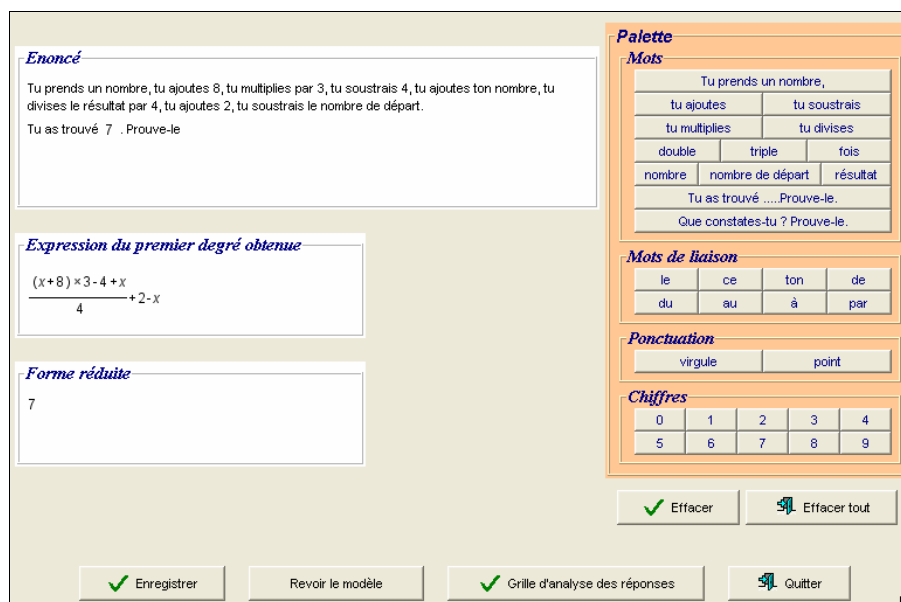


Figure 2. Interface de création de l'énoncé pour la classe « Prestidigitateur »

4.1. Interface de saisie des paramètres

Pour les exercices de la classe du « Prestidigitateur » le concepteur saisit l'énoncé à l'aide d'une palette comportant des mots, des mots de liaison, des chiffres et des signes de ponctuation (Figure 2). PépiGen construit et affiche pas à pas l'énoncé et l'expression algébrique traduisant l'énoncé saisi. Dès que le concepteur a

saisi la phrase finale, PépiGen calcule et complète selon le cas la phrase « *Tu as trouvé ... Prouve-le* » par la forme réduite de l'expression et l'affiche. Le concepteur peut alors modifier l'énoncé, afficher ou non la grille d'analyse des réponses, enregistrer ou non l'exercice.

4.2. Génération de la grille d'analyse et du diagnostic

Des analyses théoriques et empiriques sur des cohortes d'élèves nombreuses et diverses ont permis de spécifier les catégories de réponses aux exercices. Ces études, si elles ne permettent pas de garantir le codage de toutes les réponses possibles permettent d'analyser les réponses observées dans notre corpus.

Solutions correctes	Solutions correctes non attendues	Solutions partielles	Solutions incorrectes
			codage
Preuve algébrique avec des expressions partielles traduisant pas à pas les résultats intermédiaires de l'enchaînement opératoire. C'est une démarche proche d'une démarche arithmétique et la réduction des expressions sont correctes, x a le statut de nombre généralisé			T2,L1,M1,C2,R1
$(x+8) \times 3 = x \times 3 + 24$ $3x + 24 - 4 = 3x + 20$ $3x + 20 + x = 4x + 20$ $\frac{4x + 20}{4} = x + 5$ $x + 5 + 2 = x + 7$ $x + 7 - x = 7$			
Preuve algébrique avec l'interprétation de l'énoncé comme une équation admettant une infinité de solutions.			T2,L1,M1,C1,R1

Figure 3. La grille d'analyse des réponses générée pour le cas d'une solution correcte non attendue

La génération automatique de la grille d'analyse des réponses constitue le point central de notre travail. Dans la classe du « Prestidigitateur », à partir du paramètre constitué par l'expression algébrique nous générons les différentes expressions associées aux démarches prévues par l'analyse didactique (Cf. 3.3) en appliquant des règles de réécriture correctes ou erronées. Nous mémorisons dans un fichier xml chacune de ces expressions ainsi qu'un texte et le code qualifiant la démarche. Le Tableau 3 donne un extrait d'un tel fichier généré par PépiGen et la Figure 3 présente l'affichage de ce fichier par le logiciel. L'ensemble des réponses (correctes ou incorrectes) est classé en 4 catégories repérées par des onglets. L'onglet « *Solutions correctes non attendues* » affiche deux réponses possibles dont la démarche qui utilise des expressions algébriques partielles (Cf. 3.3). A chaque réponse est associée le texte décrivant la démarche, la suite des expressions algébriques associées à la démarche ainsi que le code correspondant.

4.3. Génération automatique des expressions algébriques ou numériques

La génération de la grille d'analyse repose sur l'application d'un algorithme d'unification de règles de réécritures à une représentation des expressions algébriques par des arbres.

4.3.1. Les règles de réécriture

Elles se présentent sous la forme $A \rightarrow B$ où A et B sont deux expressions algébriques telles que A se récrit en B. Par exemple $a \times (b+c) \rightarrow a \times b + a \times c$ est une règle de réécriture. Pour le prestidigitateur nous avons retenu 12 règles correctes en algèbre élémentaire. Nous les avons regroupées en catégories : règles de développement, règles de transformation des fractions, règles de transformation des puissances, règles de factorisation. En ce qui concerne les règles erronées, des travaux menés en didactique [GRUGEON 1995] ont mis en évidence les erreurs les plus fréquentes repérées dans les productions d'élèves. Pour l'exercice du prestidigitateur, 20 règles nous permettent de diagnostiquer les 320 réponses d'élèves du corpus dont nous disposons. Ces règles sont fournies en entrée à PépiGen sous la forme d'un fichier XML qui peut être complété ou modifié si le besoin s'en fait sentir. Le Tableau 2 en présente un extrait. Le type désigne la catégorie des règles et le genre désigne une règle erronée (F) ou correcte (V).

```

- <regles>
  - <regle>
    <genre>V</genre>
    <type>1</type>
    <numero>1</numero>
    <partiegauche>a*(b+c)</partiegauche>
    <partiedroite>a*b+a*c</partiedroite>
  </regle>.....

```

Tableau 2. Extrait du fichier XML des règles de réécriture

4.3.2. L'algorithme d'unification

Ces règles sont applicables si la partie gauche s'unifie à une sous-expression de l'expression algébrique qui doit être transformée. Par exemple, pour l'expression $E = 3(x+8)-4+x$, la règle $a \times (b+c) \rightarrow a \times b + a \times c$ est applicable car $3(x+8)$ s'unifie avec $a \times (b+c)$. Les expressions algébriques sont représentées sous forme d'arbres. La construction issue d'une analyse syntaxique de l'expression algébrique s'appuie sur une grammaire des expressions algébriques. La représentation en arbre et l'ordre de parcours préfixé est bien adaptée pour l'unification. Les arbres représentant l'expression E et la partie gauche de la règle sont parcourus en parallèle simultanément en partant de la racine de l'arbre. Si la partie gauche de la règle est reconnue elle est remplacée par la partie droite de la règle. La Figure 4 représente les

arbres obtenus et l'unification. L'ordre de visite dans l'arbre est repéré par le nombre dans un cercle.

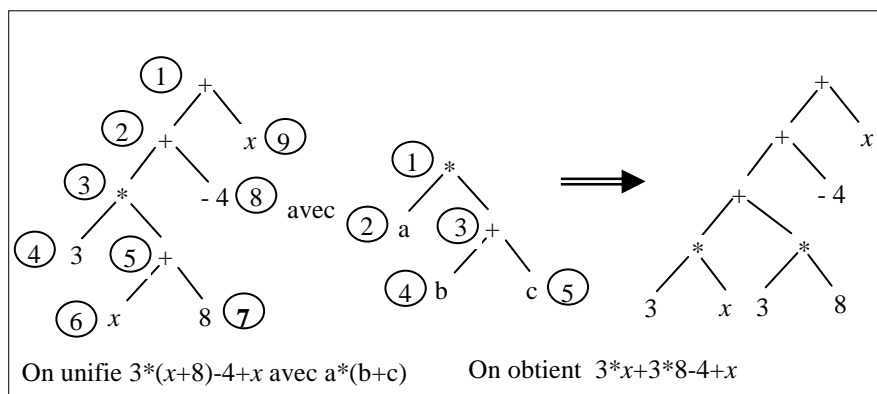


Figure 4. Unification

4.4. Le diagnostic

Le Tableau 3 présente un extrait du fichier XML qui contient les descripteurs de l'exercice, l'énoncé, les textes, les différentes expressions de la grille d'analyse des réponses. La procédure de diagnostic dont l'algorithme est donné dans la section 3.3 prend en compte les données de ce fichier XML d'une part, les productions de l'élève d'autre part et restitue le codage et la liste des règles fausses identifiées éventuellement. Certaines réponses qui n'auraient pas été anticipées par l'analyse didactique seront alors notées comme n'étant pas diagnostiquées.

```
<solutions>
  <solution_correcte>
    <Ou>
      <texte>Preuve algébrique avec expression globale (parenthésée) traduisant le résultat
      de l'enchaînement opératoire; le développement et la réduction de l'expression sont
      correctes, x a le statut de nombre généralisé - Réduction au même dénominateur</texte>
      <codage>T1,L1,C1,R1,M2</codage>
      <expression>((x+8)*3-4+x)/4+2-x = (x*3+24-4+x)/4+2-x = (4*x+20)/4+2-x =
      (4*x+20+8)/4-x = (4*x+28)/4-x = (4*x+28+4*(-x))/4 = 7</expression>
    </Ou> .....
```

Tableau 3. Extrait du fichier XML de la grille d'analyse

5. Conclusion

Dans cet article nous avons caractérisé différentes classes d'exercices et mis en évidence les paramètres et les contraintes sur ces paramètres pour générer des exercices équivalents. Nous avons présenté un exemple de génération d'exercices d'une classe pour laquelle le concepteur spécifie lui-même les paramètres et fait générer par PepiGen les exercices correspondants. Pour d'autres classes d'exercices, PepiGen génère automatiquement les paramètres à partir des contraintes didactiques.

Sur ce point notre démarche se rapproche de la démarche de la version 2 de la spécification QTI et de son extension proposée par [Auzende et al. 2007]. Mais à notre sens cette spécification souffre actuellement d'insuffisances importantes soulignées par les praticiens :

1. Les productions des apprenants analysées sont très simples (clics, déplacement d'icônes, expressions numériques ou littérales simples).
2. Les analyses se limitent à l'étude de la validité de la réponse et à la production éventuelle d'un feedback.
3. La génération automatique se fait à partir de variables sur lesquelles l'enseignant n'a pas la possibilité de définir des contraintes.
4. Les modèles sont instanciés automatiquement et il n'est pas possible de superviser les instances créées.
5. Il n'est pas possible de contrôler le choix des instances posées à l'apprenant.
6. Seuls les scores sont mémorisés ; il n'est pas possible de mémoriser les instances soumises à l'apprenant ni ses réponses pour une étude ultérieure.

Nous avons donc préféré l'approche qui consiste à créer des bases d'exercices contrôlées par des praticiens à celle de la génération automatique ce qui est aussi l'approche retenue par l'équipe de Cognitive Tutor Authoring Tool (CTAT) et de Assisment [Koedinger et al. 2003] [FENG & HEFFERNAN 2005]. Il est peu pertinent de comparer PépiGen à ce dernier projet qui est très différent par l'ampleur des financements et le nombre de personnes impliquées et par le degré d'achèvement des logiciels mis en œuvre ; cependant, ce qui fait l'originalité de notre travail, c'est la modélisation d'une expertise didactique qui permet d'évaluer, non seulement des savoir-faire isolés, mais de détecter des cohérences dans les savoir-faire et dans les conceptions inadaptées des élèves. De ce point de vue, les exercices de preuve comme ceux de la classe du prestidigitateur que nous avons étudiés dans cet article, sont très importants pour situer la compétence des élèves par rapport à la compétence attendue et mettre en place des parcours d'apprentissage adaptés [GRUGEON 1995].

Bibliographie

- [AUZENDE et al. 2007] Auzende O., Giroire H., Le Calvez F., Propositions d'extensions à IMS6QTI 2.1 pour l'expression de contraintes sur les variables d'exercices mathématiques, EIAH'07, (ce volume).
- [DELOZANNE et al 2002 b] Delozanne E., Grugeon B, Jacoboni P., Analyses de l'activité et IHM pour l'éducation, *In Proceedings of IHM'2002, International Conference Proceedings Series*, ACM, 2002, Poitiers, France p. 25-32
- [GRUGEON 1995] Grugeon B., Etude des rapports institutionnels et des rapports personnels des élèves à l'algèbre élémentaire dans la transition entre deux cycles d'enseignement : BEP et Première G, thèse de doctorat, Université Paris 7, 1995.
- [JEAN 2000] Jean S., PEPITE : un système d'assistance au diagnostic de compétences, thèse de doctorat, Université du Maine, 2000.
- [PREVIT et al 2004] Prévité D., Delozanne E., Grugeon B., Modélisation cognitive en algèbre élémentaire : une conception itérative, *Colloque TICE 2004*, Compiègne 20-22 octobre 2004, p 138 – 145
- [FENG & HEFFERNAN 2005] Feng M., Heffernan N., Informing Teachers Live about Student Learning: Reporting in the Assistent System. *The 12th International Conference on Artificial Intelligence in Education 2005, Workshop on Usage Analysis in Learning Systems*, 2005, Amsterdam, p. 25-32.
<http://hcs.science.uva.nl/AIED2005/download.html> (consulté en 2007)
- [KOEDINGER et al 2003] Koedinger K., Alevén V., Heffernan N., Informing Teachers Toward a Rapid Development Environment for Cognitive Tutors : Interactive Event during AIED-03, *The 11th International Conference on Artificial Intelligence in Education 2003, InteractiveEvent*, 2003, Sydney, p. 24-26.

Références sur le WEB

- [BO 2004] B.O. hors série N° 4 du 9 septembre 2004, p 4-6
<http://www.education.gouv.fr/bo/2004/hs4/MENE0401470A.htm> (consulté en 2007)
- [IMS-QTI 2006] IMS Question and Test Interoperability Integration Guide Version 2.0 Final Specification.
http://www.imsglobal.org/question/qti_v2p0/imsqti_intgv2p0.html (consulté en 2007)