Gestion des erreurs dans une interface pédagogique

Groupe COMBIEN
Duma J.[1], Giroire H.[2], Le Calvez F.[3], Tisseau G.[2], Urtasun M.[3]

Résumé : Une interface pédagogique de résolution de problèmes doit permettre à un élève utilisateur de formuler une solution à un problème qui lui est posé, mais surtout d'exploiter au mieux cette activité pour augmenter ses connaissances et ses capacités. Dans une telle interface la gestion des erreurs commises par l'élève ne concerne pas uniquement l'aspect ergonomique et fonctionnel, comme dans toute interface, mais relève en premier lieu d'une décision pédagogique. Il faut définir quelles erreurs seront rendues impossibles à commettre de par la structure même de l'interface, et comment réagir pédagogiquement aux autres erreurs jugées « instructives ». Nous affirmons que la conception d'une interface pédagogique doit reposer sur une analyse préalable des erreurs possibles et nous proposons une méthode de conception fondée sur ce principe. Nous illustrons cette méthode dans le contexte d'interfaces pour résoudre des exercices de dénombrement. Nous avons répertorié et catégorisé les erreurs possibles, ce qui n'est pratiquement réalisable que si l'on se fixe d'abord un cadre précis : une classe de problèmes, une méthode de résolution et un modèle conceptuel rendant la résolution équivalente à l'édition d'un objet structuré. Cela permet d'identifier les concepts autour desquels doit être organisée l'interface, et qui vont correspondre aux boutons de validation proposés à l'élève. Nous montrons ensuite comment implémenter la gestion des erreurs à l'aide d'un *agent pédagogue*.

Mots clés : interfaces pédagogiques, erreurs, méthode de conception, EIAO, dénombrement.

1. Introduction

Le projet "Combien ?" a pour but de définir une méthodologie de conception de différents composants d'un EIAO (Environnement Interactif d'Apprentissage avec Ordinateur). Pour valider nos réflexions, nous réalisons un EIAO pour l'apprentissage des dénombrements. Les exercices correspondant sont de la forme : "Etant donnés des ensembles servant de référentiels, compter dans un certain univers les éléments vérifiant des contraintes de sélection". Nous avons défini les fondements mathématiques d'une méthode de résolution (la méthode constructive) adaptée aux conceptions usuelles des élèves et permettant d'accéder à la théorie mathématique du domaine [Tisseau et al. 96], [Le Calvez et al.97]. Nous avons défini une classification des problèmes du domaine et les schémas de résolution associés aux différentes classes. Nous avons introduit pour chaque classe une "machine à construire une solution". Chaque machine se présente pour l'élève sous forme d'une interface pédagogique qui conduit l'élève à construire la solution d'un exercice de la classe considérée.
Dans une interface pédagogique la gestion des erreurs est très importante pour optimiser l'efficacité pédagogique de l'interface. Contrairement à une interface usuelle, il faut pouvoir laisser à l'élève la possibilité d'exprimer sa solution et donc de commettre des erreurs. Généralement on essaie dans une interface de canaliser l'activité de l'intervenant pour qu'il

[1] Lycée Jacquard, Paris
[2] LIP6, Pôle IA, SYSDEF, Université Pierre et Marie Curie, Paris VI
[3] CRIP5, SBC, Université René Descartes, ParisV

effectue le moins d'erreurs possibles [Choplin and al. 98]. Il est en effet techniquement possible de concevoir l'interface pour que l'utilisateur ne puisse pas commettre certaines erreurs. En effet, les erreurs sont provoquées par des actions de l'utilisateur (cliquer sur un bouton, choisir un item de menu) et il est parfois possible de prévoir avant même que l'action soit exécutée si elle pourra provoquer une erreur ou non. Dans ce cas, on peut faire en sorte que les possibilités d'action offertes à l'utilisateur ne puissent pas provoquer d'erreur, et cela dynamiquement en fonction de la situation (rendre indisponibles les boutons qui provoqueraient des erreurs ou ne faire figurer dans un menu que les items qui ne provoquent pas d'erreur). Les autres erreurs sont détectées sur-le-champ, signalées et éventuellement corrigées. Dans une interface pédagogique il faut quelquefois laisser l'élève s'exprimer même si l'on prévoit une erreur ou si l'on en a détecté une, soit pour qu'il s'en aperçoive tout seul plus tard, soit pour lui montrer qu'il arrive à une impasse et profiter de l'occasion pour lui apprendre quelque chose.
Cependant toutes les erreurs ne sont pas de la même difficulté, ni du même intérêt pédagogique. Une étude sur les différents types d'erreurs doit permettre de décider quelles sont celles qui seront permises à l'élève et celles qu'il ne pourra commettre. D'autre part, les erreurs peuvent être de type différents (par exemple, provoquées par l'utilisation de l'interface ou correspondant à des concepts mathématiques). Dans certains cas on essaiera de proposer à l'élève des possibilités d'interaction où il ne peut pas faire ces erreurs.
Dans cet article, nous inventorions les différentes catégories d'erreurs dans la machine "ConstructionEnsemble". Nous proposons une méthode de conception des interfaces pédagogiques qui s'appuie sur la gestion des erreurs. Nous indiquons ensuite, comment nous réalisons la gestion des erreurs dans cette machine.

2. La machine ConstructionEnsemble

Les machines permettent pour chaque classe de problème de construire un élément de l'ensemble à dénombrer (en suivant la méthode constructive), puis à partir de cette construction de compter le nombre d'éléments de cet ensemble. [Tisseau et al. 00-1] [Tisseau et al. 00-2]
Les exercices de la machine ConstructionEnsemble sont du type : "Soit un ensemble donné (le référentiel), compter le nombre de configurations contenant X éléments de ce référentiel (l'ensemble des configurations est appelé univers), et satisfaisant certaines contraintes. Les contraintes sont de la forme : exactement X1 éléments appartenant au sous ensemble R1 du référentiel, exactement X2 éléments appartenant au sous ensemble R2 du référentiel etc. Les sous ensembles Ri étant tous disjoints deux à deux."
Dans la suite de l'article, l'exercice suivant : "Avec un jeu de 52 cartes, combien y-a-t-il de mains de 8 cartes contenant exactement 1 as, 2 rois et 2 valets ? ", sera utilisé pour illustrer nos propos. Cet énoncé est un énoncé pris dans un livre d'exercices, il signifie : combien y-a-t-il de mains de 8 cartes contenant exactement 1 as, exactement 2 rois et exactement 2 valets et 3 cartes qui ne sont ni as, ni roi, ni valet ?
Le modèle conceptuel que nous avons défini pour représenter le domaine des dénombrements est décrit dans. [Tisseau et al. 00-1] [Tisseau et al. 00-2]. Dans le cas du jeu de 52 cartes, chaque carte est représentée en utilisant deux attributs **couleur** et **hauteur**. La figure 1 nous montre l'aspect de la machine après construction d'un élément de l'univers des mains de 8 cartes.

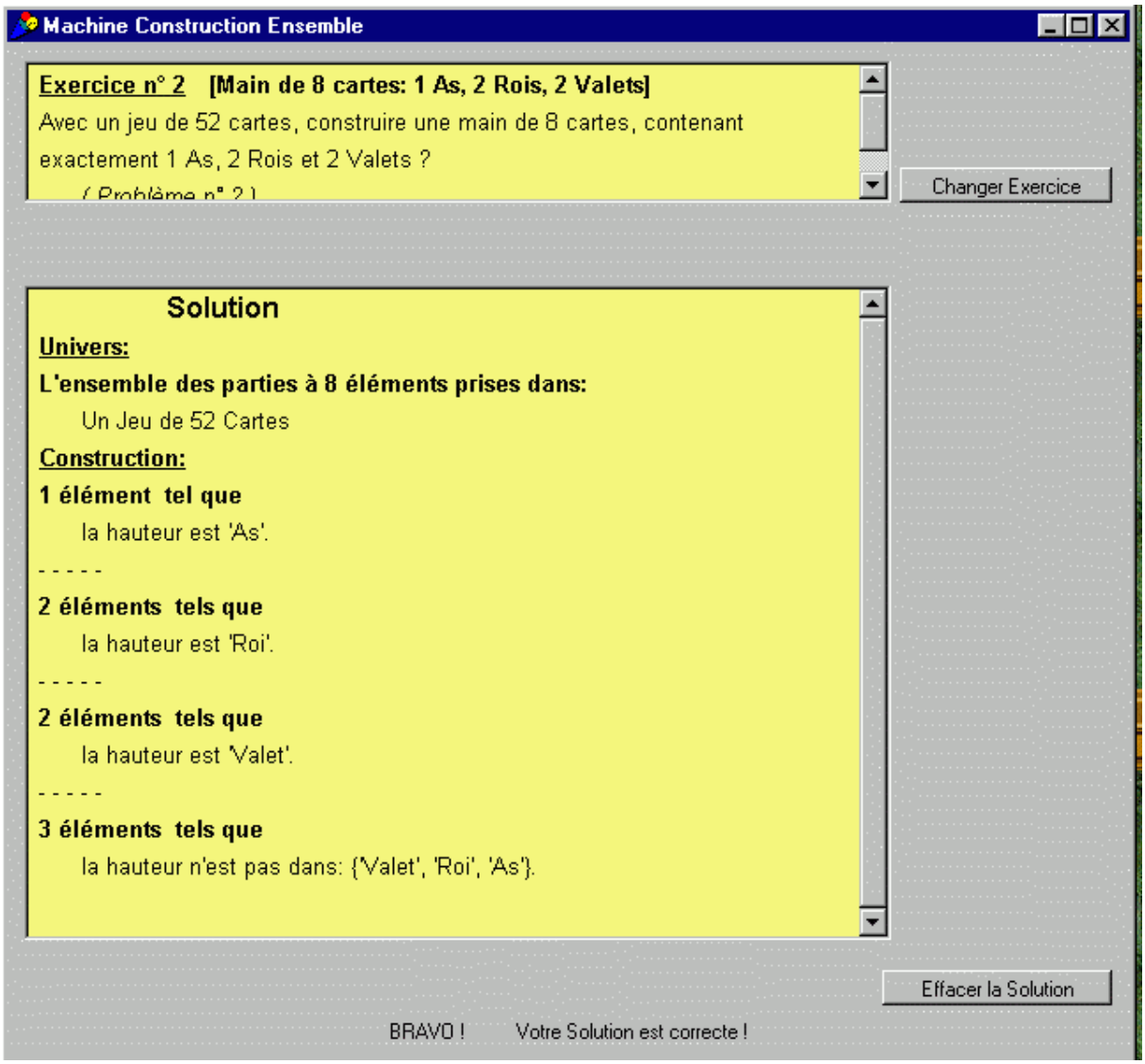

Figure 1

A partir de cette construction et dans cette machine, le nombre de configurations possibles est le produit des possibilités à chaque choix.
Il faut bien voir que cette image n'est que l'aspect final de la réponse et ne montre pas les différents éléments d'interaction qui ont permis la génération de cette réponse (boutons, menus, etc.). Ceux-ci apparaissent et disparaissent dynamiquement en fonction de l'état d'avancement de la réponse.

3. Les erreurs dans la machine ConstructionEnsemble

Dans toute machine, la première chose que fait l'élève est de choisir l'exercice à résoudre. Celui-ci fait partie de la classe de problèmes associée à la machine. Il y a ensuite deux grandes phases dans le travail de construction : la définition de l'univers et ensuite les différentes étapes qui correspondent aux contraintes de sélection. La figure 2 montre un état de la machine où l'exercice a été choisi et validé, l'univers aussi, ainsi que certaines étapes de la construction. Mais la construction elle-même n'a pas encore été validée. Nous avons dressé

un inventaire des erreurs possibles dans ces deux phases pour les catégoriser et décider de leur traitement. Dans la suite de cet article, nous donnons les catégories d'erreurs pour la machine ConstructionEnsemble en les illustrant sur l'exercice de la figure 2.

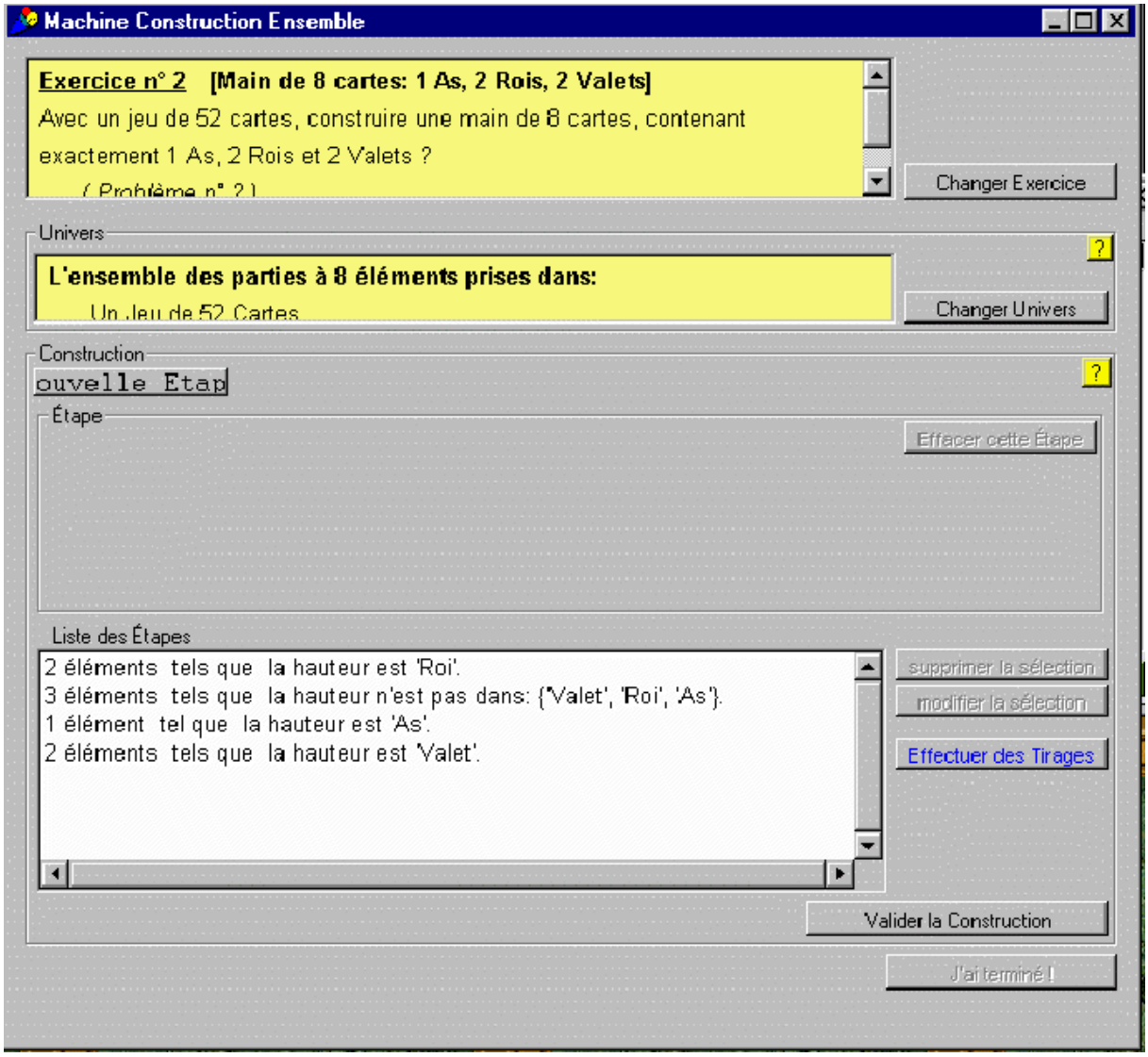

Figure 2

Dans la définition de l'univers, il faut préciser le référentiel et le nombre d'éléments de chaque élément de l'univers (c'est-à-dire de chaque configuration). Les erreurs possibles portent donc sur un mauvais choix du référentiel ou une erreur dans le nombre d'éléments de la configuration. Cette dernière erreur peut-être de deux sortes : nombre impossible (main de 56 cartes prises dans un jeu de 52 cartes) et nombre incorrect c'est-à-dire ne correspondant pas à l'énoncé du problème (main de 9 cartes alors que l'énoncé de l'exercice parle de mains de 8 cartes).

Au niveau de la définition des contraintes dans la solution, il y a à chaque étape définition du nombre d'éléments de la configuration sur lesquels porte la contrainte et la définition du sous ensemble (SE) auquel ils doivent appartenir.

A chaque étape, à propos du nombre des éléments, on retrouve les mêmes sortes d'erreurs que pour le nombre d'éléments dans une configuration : nombre impossible (14 cœurs dans un jeu de 52 cartes) et nombre incorrect (2 as alors que l'énoncé de l'exercice parle de mains contenant exactement 1 as). La définition de SE se fait à l'aide de sélection sur les attributs

("hauteur est roi", ou "hauteur est roi ET couleur n'est pas noire"). A ce niveau les erreurs possibles concernant l'attribut sont : attribut inexistant pour le référentiel de l'exercice (masse pour une carte) ou valeur erronée pour un attribut (hauteur = 2 dans un jeu de 32 cartes). Une première erreurs sur SE est de définir pour SE l'ensemble vide (2 cartes de couleur pique et de couleur rouge). Les autres erreurs sur une seule étape font intervenir des sous-ensembles qui ne font pas partie d'une solution correcte de l'exercice à résoudre. L'élève définit la contrainte "2 cartes de hauteur dame" alors que l'on attend 2 valets ou 2 rois ou bien il définit la contraintes "2 cartes de couleur rouge" alors que l'on attend 2 cœurs.
Les erreurs que l'on vient de décrire correspondent à une étape isolée. Certaines erreurs apparaissent lorsque l'on considère l'ensemble des étapes.
Il peut y avoir d'une part des erreurs sur la somme des nombres définis à chaque étape. Il y a deux types d'erreurs : des incohérences et des incorrections.

- Les incohérences sont internes à la solution de l'élève, il a défini les mains de 8 cartes pour l'univers et de 10 cartes (ou de 3) à la fin des étapes.
- Les incorrections sont externes par rapport à la solution de l'élève. Sa solution est cohérente mais ne correspond pas à une solution de l'exercice à résoudre. Il a défini les mains de 5 cartes pour l'univers et a bien défini les étapes pour 5 cartes alors que dans l'exercice on demande des mains de 8 cartes.

Il peut y avoir d'autre part des erreurs sur la définition des différents SE. La méthode constructive implique pour cette machine que les SE soient disjoints. L'erreur consiste à avoir des SE non disjoints (le SE défini par "hauteur est as" et celui défini par "couleur est rouge").
Même pour un exercice aussi simple, on constate qu'il existe un grand nombre d'erreurs possibles, mais aussi qu'il apparaît une catégorisation à visée pédagogique possible.

En résumé, nous avons distingué trois catégories d'erreurs :
Des erreurs purement internes, qui sont détectables sans prendre en compte une solution de l'exercice, et ne font pas intervenir la méthode constructive. Nous avons utilisé pour elles les termes "impossible" ou "incohérent". Nous pouvons les considérer comme des erreurs d'inattention.
Des erreurs qui font intervenir la méthode constructive et qui sont internes. C'est l'utilisation de SE non disjoints pour la machine ConstructionEnsemble. Ces erreurs sont délicates à expliquer, car elles correspondent à des conditions uniquement imposées par la méthode, et dont l'élève ne pourra voir la raison d'être que plus tard, au moment de compter. Pour lui, elles peuvent présenter un caractère arbitraire.
Des erreurs dites externes, elles proviennent de la comparaison des solutions correctes et de la solution proposée par l'élève ne comportant pas d'erreurs internes.

Ces catégories (internes, faisant intervenir une méthode de résolution, externes) nous semblent générales pour un EIAO.

4. La définition et la gestion des erreurs

Pour spécifier l'interface une fois que les erreurs ont été répertoriées et catégorisées, il faut décider quand et comment les détecter, les signaler et les expliquer.

4.1. Cadre conceptuel : structure à éditer

Nous nous plaçons dans le cadre suivant : les interfaces pédagogiques que nous considérons sont des outils offerts à un élève pour exprimer la solution de problèmes d'une classe particulière dans un domaine particulier, cette solution pouvant être formalisée comme instance d'une certaine *structure* arborescente prédéfinie. Cette structure n'est pas entièrement figée : certains nœuds peuvent être polymorphes (leur classe peut être choisie parmi plusieurs classes différentes) ou contenir une valeur multiple de cardinal non spécifié a priori (une liste d'éléments). D'autre part, cette structure est « à dimension humaine » : le nombre de nœuds et de niveaux qu'elle comporte permet l'édition de ses instances par un élève, interactivement, à travers une interface qui permet d'en remplir les champs.
Il faut noter que la théorie d'un domaine d'étude donné ne fournit pas automatiquement une telle structure. Celle-ci est à inventer à chaque fois qu'on a l'intention de créer une interface pédagogique destinée à un certain type d'élèves possédant certaines connaissances et en fonction d'un certain niveau d'approfondissement et d'expertise visé. C'est ce que nous avons dû faire pour les dénombrements.

4.2. Validité d'une solution, erreurs

La validité d'une solution est définie par une conjonction de *conditions de validité* portant sur la structure. Une *erreur* est définie comme étant une violation d'une de ces conditions de validité. On peut ainsi distinguer des *types d'erreur* correspondant aux conditions de validité : une erreur est de tel type si elle viole telle condition.
On peut assigner à chaque type d'erreur un intérêt pédagogique : lorsque l'élève commet une erreur et qu'on la lui signale, quel bénéfice peut-il en retirer du point de vue de la compréhension des concepts du domaine, de l'identification des difficultés et de l'apprentissage de méthodes de résolution ? Suivant les buts pédagogiques visés, une erreur pourra être jugée *instructive* ou non.
Les erreurs non instructives sont considérées comme des diversions hors sujet qui encombrent le chemin vers l'apprentissage visé. Cela ne signifie pas qu'elles sont sans importance théorique, mais simplement qu'elles éloignent trop du but particulier choisi. Il est bon alors que l'interface soit conçue de telle façon que ces erreurs ne puissent pas apparaître. Cela revient à dire que c'est le système lui-même qui doit assurer automatiquement le respect des conditions de validité associées.
Par exemple, si une condition de validité non instructive spécifie qu'un nombre x doit être un nombre entier compris entre 1 et 5, l'interface pourra offrir à l'élève un menu avec les cinq valeurs 1, 2, 3, 4, 5 pour la saisie de x (plutôt qu'un champ de texte où l'élève édite librement un texte quelconque). Notons que cette configuration de l'interface pour satisfaire les conditions de validité peut être dynamique : si une condition spécifie que x doit être un entier compris entre 1 et y et que l'élève a saisi y, le menu pour x sera mis à jour pour comporter la liste des entiers de 1 à y.

4.3. Aspects dynamiques et interactifs

La saisie de la solution par l'élève est vue comme une tâche d'édition d'une structure arborescente. Avec les structures que nous considérons, cette tâche n'est pas atomique ni immédiate. Elle passe par différentes étapes qui permettent d'engendrer la structure complète petit à petit, incrémentalement. Pour tenir compte de cet aspect, nous avons construit des interfaces qui prennent en compte les erreurs de façon incrémentale. Cela est adapté au type

de tâche demandé, mais aussi et surtout, cela permet de profiter le mieux possible des instants où l'élève est dans une situation d'apprentissage optimale, c'est-à-dire au moment où il commet des erreurs.
L'incrémentalité est prise en compte en décomposant le traitement des erreurs en différents sous-traitements, qui eux-mêmes sont exécutés en plusieurs occurrences à différents moments. Ces sous-traitements sont : la *détection* d'une erreur (qui est une procédure interne au système, « silencieuse » car elle n'est pas visible par l'élève), son *signalement* (par exemple l'affichage d'un message pour l'élève : « il y a une erreur ») et son *explication*. Dans notre EIAO, nous voulons pouvoir choisir pour chaque erreur le moment où on la détecte, celui où on la signale et celui où on l'explique. Cela contraste avec d'autres EIAO où le travail de l'élève n'est évalué que lorsqu'il valide la solution finale de l'exercice.
Pour les erreurs "intéressantes", il faut décider **à quel moment elles sont détectées**. Le fait de les détecter n'implique pas d'en faire part à l'élève, c'est un choix pédagogique que de déterminer **le moment du signalement** de cette erreur et donc si l'élève peut ou non continuer, bien qu'ayant commis une erreur. Il faut ensuite décider comment se manifeste le signalement : on peut refuser de valider sa proposition (soit en ayant un bouton qui reste grisé, soit en indiquant que la proposition n'est pas valide), ou bien on peut indiquer qu'il y a une erreur sans autre explication et le laisser continuer s'il le désire pour qu'il découvre quelle est cette erreur. Enfin, il faut décider à quel **moment on lui donne une explication** sur ses erreurs et quel type d'explication donner.

A partir de ces réflexions nous avons défini une méthode de conception de nos interfaces pédagogiques, qui nous semble générale.

5. Une méthode conception "dirigée par les erreurs"

La méthode de conception "dirigée par les erreurs" d'une interface pédagogique que nous proposons comporte un certain nombre d'étapes décrites ci-dessous:

Définir la structure à éditer
Choisir les éléments validables
Choisir les possibilités de modification données à l'élève
Choisir le moment de signalement des erreurs détectées
Définir les explications et les aides

Figure 3

5.1. Définir la structure à éditer

Nous avons vu qu'une interface pédagogique est un éditeur de structure arborescente. Le travail que nous avons fait précédemment [], nous a permis de définir un modèle du domaine du dénombrement y compris des problèmes et des solutions. A partir de ce modèle nous définissons les structures à éditer pour les différentes machines correspondant aux différentes classes de problèmes. Dans le cas de la machine ConstructionEnsemble, la structure à éditer est celle de la figure 4. L'élève choisit son exercice et construit sa solution.

```
Exercice : un Exercice
Solution : une Solution
        Univers : un UniversParties
                Référentiel : unRéférentiel
                Cardinal : unEntier
        Construction : une ConstructionDePartie
                SousParties : * ConstructionSousPartie
                        Cardinal : unEntier
                        Filtre : unFiltre
                                Propriétés : 0 , 1 ou 2 Propriétés
                                        IdentificateurAttribut
                                        Appartenance/Comparateur/estVrai
                                        Domaine/ValeurCible/IdentificateurPredicat
```

Figure 4

Essayons de voir comment, à partir de la déclaration ci-dessus, on peut faire des choix pédagogiques et ergonomiques qui conduisent à l'interface finale. Le premier choix concerne les éléments à valider.

5.2. Choisir les éléments validables

Nous appelons *élément validable* une partie de la structure arborescente dont l'édition se manifeste à l'écran dans une zone identifiable comportant un bouton « valider » s'appliquant à toute la zone. A l'intérieur de la zone, en cours de saisie, toutes les informations sont modifiables et annulables à volonté. Mais lorsque l'élève clique sur le bouton « valider », il signifie par là qu'il assume tous les choix qu'il a faits dans la zone et celle-ci est alors figée avec les informations saisies (il apparaît cependant éventuellement un bouton « annuler » qui permettra de revenir sur la saisie plus tard).
Le choix des éléments validables relève de la pédagogie choisie, en fonction du domaine et de la méthode de résolution que l'on veut faire utiliser. A partir du travail sur nos interfaces, nous avons relevé quelques critères de choix.

- L'élément à valider correspond à un concept qu'on veut **rendre visible à l'élève**. On rencontre souvent ce problème en pédagogie : tous les concepts de la théorie ne sont pas à mettre au même niveau. Présenter explicitement certains concepts à l'élève trop tôt lui compliquerait les choses et l'empêcherait de se concentrer sur l'essentiel. Ici, par exemple, on souhaite que l'élève retienne qu'une solution est formée d'un univers et d'une construction, celle-ci étant une liste d'étapes. Pour chacun de ces concepts, on doit introduire un mot qui le désigne (pour l'élève, le mot étape correspond au concept de SousPartie de la structure), une définition et des exemples.

Les concepts non validables ne sont pas forcément explicités à l'élève. Il faut alors que l'interface les rende " évidents ". Pour cela, on utilisera le langage naturel, une métaphore, des structures graphiques etc.. Ainsi, dans la structure ci-dessus, l'élève ne valide pas chaque élément de la SousPartie. L'interface lui fait remplir une phrase à trous à l'aide de menus.

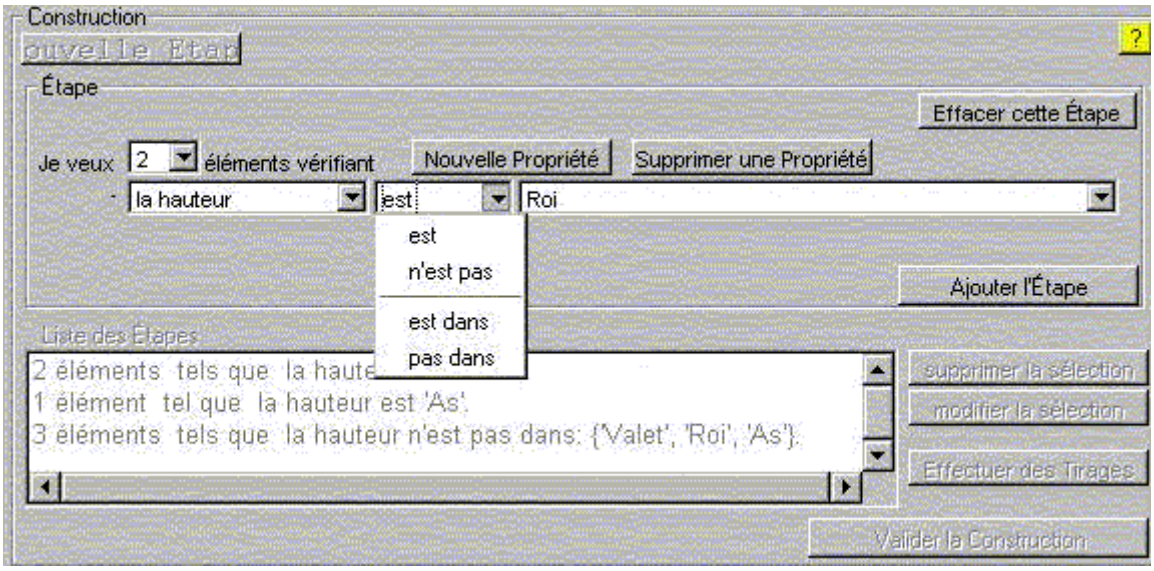

Figure 5

- La validation de l'élément peut donner lieu à des **signalements d'erreurs instructives**. Ce sont les erreurs qui permettent à l'élève d'avancer dans son apprentissage. Pour chaque validation, on établit la liste des erreurs possibles et pour chacune d'elles les informations à apporter à l'élève. On a choisi ici de ne pas signaler d'erreur pendant la saisie dans la zone. On attend que l'élève ait validé sa réponse. Cela comporte un aspect sécurisant pour lui : cela signifie qu'on ne l'observe pas pendant son activité de saisie et qu'on reconnaît qu'il est normal que celle-ci puisse comporter des essais au brouillon, des fautes de frappe et d'inattention. Par contraste, cela souligne l'importance accordée au concept validé.
- Le **nombre d'éléments auxiliaires** à obtenir pour autoriser la **validation ne doit pas être " trop grand ".** Le découpage en parties validables est un garde-fou qui permet à l'élève de ne pas s'enferrer dans une succession d'erreurs qui rendraient le diagnostic et les corrections trop compliqués ou même infaisables. Par exemple, nous avons choisi de faire valider l'univers (ensemble des mains de 8 cartes) qui est composé d'un référentiel et d'un cardinal de façon à ce que l'élève ne puisse pas continuer s'il n'a pas choisi les bons ensembles de départ. Le parcours est balisé par les validations. Dans notre machine, par rapport à la structure décrite figure 4, on a choisi de ne faire valider à l'élève que : le choix de l'exercice, celui de l'univers, de la construction et de chaque sous-partie.

5.3. Choisir les possibilités de modification données à l'élève

Il y a plusieurs façons de permettre à l'élève de revenir sur ce qu'il a fait :
- Avant validation, simplement en saisissant une nouvelle valeur (l'annulation d'un seul item) ou en annulant toutes les informations en cours de saisie dans le contexte à valider.
- Après validation l'élève peut vouloir modifier un élément qui a été validé. Dans ce cas-là, se pose le problème du contexte à restituer à l'élève. Suivant les cas, on peut pré-remplir l'éditeur avec les informations venant de l'élément annulé ou on peut présenter un éditeur vide. C'est un choix d'ergonomie. Cela peut aussi être un choix pédagogique fondé sur le coût d'une annulation. Si annuler demande un gros travail pour saisir à nouveau un élément, l'élève y réfléchira à deux fois la prochaine fois qu'il devra valider. L'annulation d'un élément annule tout le sous-arbre issu de cet élément.

Actuellement, voici les choix de correction explicite qui ont été faits :

Changer exercice (après validation, efface l'exercice)
Changer Univers (après validation, efface l'univers)
Effacer étape (avant validation)
Modifier une étape de la liste des étapes (après validation, permet de l'éditer)
Modifier construction (permet de l'éditer)

Figure 6

5.4. Choisir le moment de signalement des erreurs détectées

Contrairement à ce qui se passe dans d'autres interfaces, n'attendons pas que la solution de l'élève soit complète pour l'analyser et pour signaler des erreurs. Les erreurs sont détectées lors des demandes de validation au cours de la construction de la solution et selon les concepts à valider retenus. Pour des raisons pédagogiques, le moment de signalement des erreurs peut être repoussé après le moment de détection. Par exemple, on pourrait attendre la validation finale pour signaler les erreurs externes, c'est-à-dire celles qui correspondent à une solution d'un exercice de la classe de la machine mais qui ne correspondent pas à une solution de l'exercice à résoudre.
Pour mettre en œuvre ce retard dans le signalement de l'erreur, il faut mémoriser les erreurs en tenant compte des annulations et modifications. Cela nécessite un mécanisme de mise à jour (du type historique, ou pile).
Certaines erreurs peuvent être signalées mais pas expliquées sur le moment. Leur explication peut être reportée à la validation finale. Lors de la détection, on voit juste un message : " Attention, il y a une erreur. Vous pouvez essayer de la corriger, sinon vous pouvez continuer mais vous devrez attendre la validation finale pour avoir l'explication détaillée".
Ce problème de détection et de signalement des erreurs est très différent des principes d'ergonomie qui disent : il faut tout contrôler le plus possible pour éviter l'apparition d'erreurs et s'il y en a elles doivent être détectées et signalées le plus tôt possible, avec des explications. Ici on peut introduire volontairement la possibilité pour l'élève de commettre des erreurs, on peut décider de ne pas les signaler tout de suite ou de ne pas les expliquer complètement.
Actuellement, dans la machine ConstructionEnsemble, les erreurs sont signalées dès qu'elles sont détectées.

5.5. Explications et aides

Lorsqu'une erreur est signalée, on peut décider de ne pas expliquer tout de suite à l'élève pourquoi il y a une erreur. Il peut être plus pédagogique de lui laisser chercher la cause de l'erreur. Cependant il ne faut pas qu'un élève se sente bloqué ou qu'il ne comprenne pas. C'est pourquoi nous avons décidé de donner à l'élève la possibilité d'avoir des explications sur son erreur en interrogeant le système (par un bouton).
D'autre part, l'élève ne doit jamais être bloqué dans l'utilisation d'une machine. Nous avons donc prévu une aide contextuelle afin de lui expliquer à tout instant ce que le système attend de lui.
Il nous a semblé alors logique de rassembler aide contextuelle dans le maniement de la machine et explication sur l'erreur qui vient d'être signalée dans une même boîte de dialogue comportant plusieurs onglets. De la même façon, l'élève a besoin d'aide contextuelle sur le

dénombrement. Cette boîte de dialogue comporte aussi un onglet dénombrement qui lui présente la partie du cours dont il a besoin (le cours correspondant à la classe de la machine). Ainsi à tout moment de l'utilisation de notre interface, l'élève peut interroger au moyen du bouton  le système et obtenir soit le cours, soit l'explication d'utilisation de la machine, soit l'explication de l'erreur qui vient de lui être signalée. Dans nos machines, à chaque contexte de validation est associé un bouton  .
Associé à chaque machine, il y aura donc un cours "distribué", "réparti" et présenté par nécessité et opportunité. De cette façon, une même notion pourra être présentée de façon différente selon le contexte.
Les figures 7 et 8 montrent au moment de la validation de l'univers, les aides proposées à l'élève respectivement du point de vue du cours et du point de vue de l'interface.

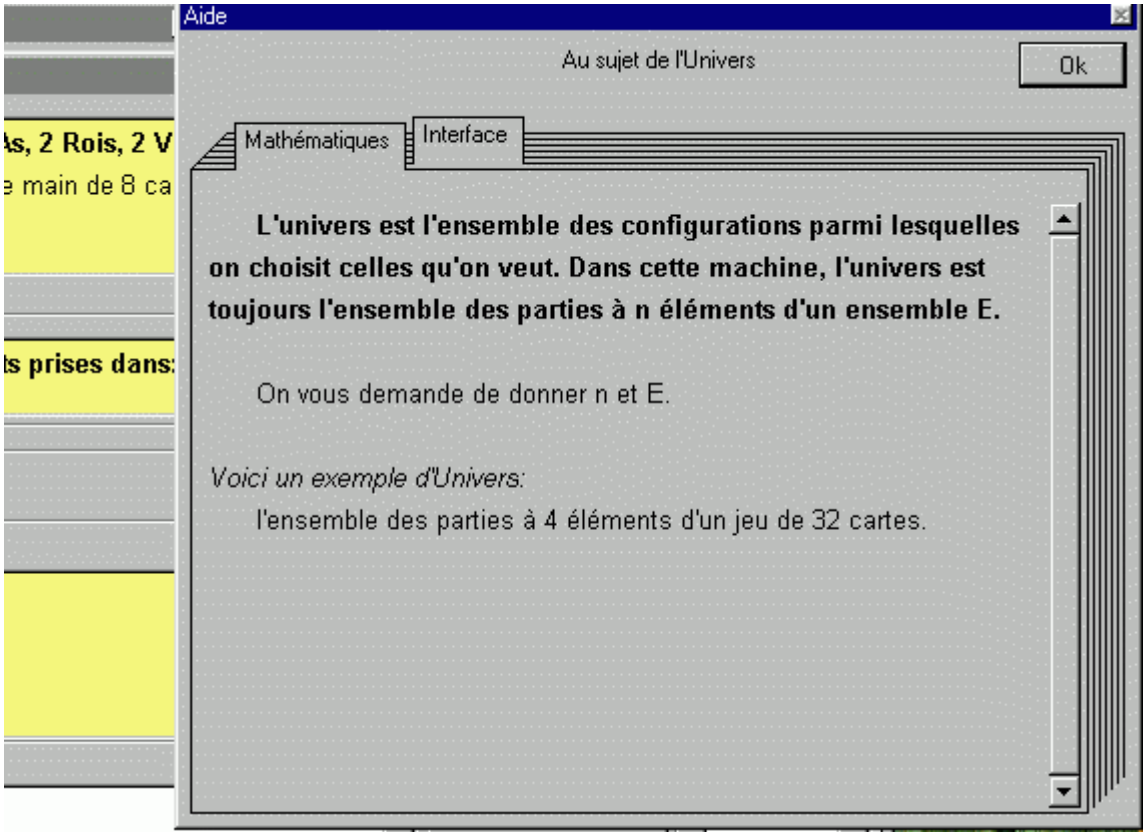

Figure7

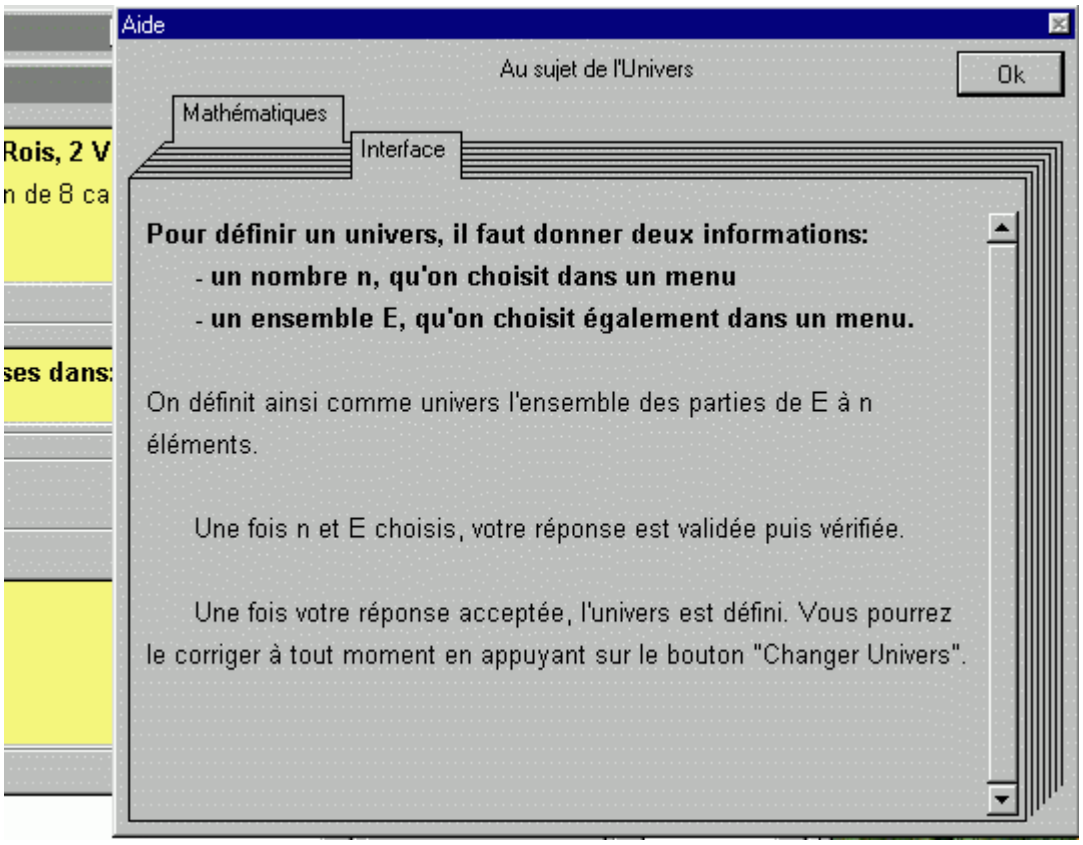

Figure 8

6. Vers une conception-implémentation

Nous abordons la phase de réalisation de la prise en compte des erreurs. Nos avons choisi les concepts à valider. A chacun d'entre eux est associée une condition de validité. La condition de validité s'exprime par une conjonction de conditions. Quand l'élève clique sur le bouton "valider" associé au concept, la condition de validité correspondante est examinée. Si elle n'est pas vérifiée, cela veut dire qu'il y a une ou plusieurs erreurs. Celles-ci sont alors mémorisées sous la forme d'une liste d'objets. Chacune des erreurs contient les informations nécessaires à son explication.
Pour réaliser ceci, nous avons associé à chaque machine un agent pédagogue. Son rôle est de gérer les erreurs. Le pédagogue demande à la partie « experte » du système le test des différents éléments de la condition de validité. Il détecte si il y a une erreur ou pas, et dans le cas où il y en a il va les analyser, puis prendre les décisions de réaction vis à vis de ces erreurs. Ces décisions concernent le moment du signalement, les explications et la réponse à la demande de validation de l'élève. Le pédagogue transmet alors ses décisions à la machine qui va les matérialiser pour l'élève.
Tout pédagogue est représenté par une structure qui contient la machine à laquelle il est associé, la session et les diagnostics. La session comporte toutes les informations liées à l'exercice à résoudre (en particulier la solution du système) et s'enrichit de la solution de l'élève au fur et à mesure de sa construction au moyen de la machine. Il y a autant de diagnostics que de conditions de validité. Chacun d'entre eux est représenté par un booléen (valeur de la condition de validité) et une liste d'erreurs éventuellement vide. La figure 9

montre la structure du pédagogue ConstructionEnsemble associée à la machine ConstructionEnsemble (cf. § 2).

```
Machine : une MachineConstructionEnsemble
Session : une SessionCE
Diagnostics : *Diagnostic
      Valeur : Boolean
      Erreurs : *Erreur
```

Figure 9

Dans cette architecture, le pédagogue gère les erreurs en répondant aux demandes de validation de l'élève que la machine lui transmet. Pour cela, il interroge le domaine et prend des décisions qu'il communique à la machine. La gestion des erreurs est donc bien programmée à part ce qui nous donne une grande souplesse qui permettra de personnaliser cette gestion en fonction de l'élève.

7. Conclusion

Dans le projet Combien?, nous cherchons à formaliser au maximum nos réflexions pour qu'elles soient ensuite utilisables dans d'autres contextes. Nous avons ainsi introduit une méthode de conception d'interfaces pédagogiques à partir des erreurs. L'utilisation de la conception objet nous permet d'implémenter nos réflexions de façon très déclarative. Nous avons ainsi un cadre général de construction des systèmes "machines-pédagogues" qui nous permet de faire des expérimentations de nos différentes réflexions. Nous définissons actuellement à partir de cette méthode la machine ConstructionListe.

Bibliographie

[Choplin and al. 98] Hugues CHOPLIN, Arnaud GALISSON, Sarah LEMARCHAND : *Hypermédias et pédagogie : comment promouvoir l'activité de l'élève ?* - Actes du quatrième colloque Hypermédias et apprentissages, pp 87-98, J.F.Rouet et B. de la Passardière eds, 1998.

[Le Calvez et al. 97] Le Calvez F., Urtasun M. , Tisseau G., Giroire H., Duma J. *Les machines à construire : des interfaces pour apprendre une méthode constructive de dénombrement*. - Actes des 5èmes Journées francophones EIAO, pp 49-60, M.Baron, P. Mendelsohn, J.F. Nicaud eds, Hermès, 1997.

[Tisseau et al. 00-1] Tisseau G., Giroire H., Le Calvez F., Urtasun M., Duma J*., Principes de conception d'un système pour enseigner la résolution des problèmes par la modélisation*. - RFIA'2000, pp.121-130, Paris, 2000.

[Tisseau et al. 00-2] Tisseau G., Giroire H., LE Calvez F., Urtasun M., and Duma J*., Design principles for a system to teach problem solving by modelling.* Lecture Notes in Computer

Science N° 1839, ITS'2000, Springer-Verlag, Montréal, 2000.