

Rapport de projet
PING
« **ETUDE DES TRACES D'UTILISATION D'UN
LOGICIEL D'ENSEIGNEMENT** »

Etudiantes :

De Assis Lizete & Zeroug Naïla Ilhem

Encadrantes :

Françoise Le Calvez & Hélène Giroire

Université de Pierre et Marie Curie
2005/2006

Table des matières

INTRODUCTION.....	3
1. PRESENTATION DU CONTEXTE : COMBIEN ?.....	3
1.1 EXEMPLES D'EXERCICES PROPOSES DANS L'EIAH	3
1.2 STRUCTURE D'UNE RESOLUTION.....	23
2. DEROULEMENT DU PROJET	5
3. ANALYSE DES TRACES « .XML ».....	5
4. ETUDE DE L'IHM	10
5. CONCEPTION.....	11
5.1 EXTRACTION DES DONNEES DES SCRIPTS XML EN CLASSES DE DOCUMENT, EXERCICE, ET COMMANDE. .	11
5.2 ANALYSE D'UN EXERCICE	11
5.3 ENREGISTREMENT DE LA BASE DE DONNEES	14
6. NOTRE INTERFACE	16
7. TESTS ET INTERPRETATION DES RESULTATS.....	18
8. PROBLEMES RENCONTRES	20
9. CONCLUSION ET PERSPECTIVES.....	21
10. LEXIQUE.....	22
11. ANNEXE.....	23

Introduction

Ce projet consiste à créer une interface pour permettre aux enseignants d'analyser les résultats de traces d'utilisation d'un logiciel d'enseignement par un ou plusieurs étudiants.

Il s'agit du logiciel « Combien ? », dont le but est d'aider les élèves à résoudre des problèmes de dénombrement de type classe de terminale ou premières années d'université.

Ce logiciel propose quatre interfaces appelées machines qui constituent chacune un schéma de résolution pour un type d'exercice.

L'étudiant est amené à en choisir une, puis à résoudre un ou plusieurs exercices. Le logiciel crée au fur et à mesure une trace structurée en format « DESCRIPT » de toutes les actions de l'étudiant de telle façon que l'on puisse obtenir dans les traces les étapes de résolution, la durée de chaque étape ainsi que les erreurs effectuées lors de celles-ci.

Notre travail est de proposer une interface d'analyse de ces traces, en en extrayant des informations significatives.

1. Présentation du contexte : Combien ?

Le groupe de recherche « Combien? » a construit un système pédagogique pour aider les étudiants à résoudre des problèmes dans le domaine mathématique des dénombrements.

Les interfaces pédagogiques correspondantes permettent aux élèves d'apprendre le raisonnement nécessaire à la construction de la solution. Ces interfaces enregistrent toutes les actions de l'utilisateur et permettent de rejouer la solution. Chacune d'entre elles correspond à une instantiation pour une classe de problèmes de la méthode « constructive » de résolution qui, elle, est générale.

L'étude des traces permet de faire des bilans spécifiques pour les étudiants, pour les professeurs, pour les chercheurs. Ces bilans dépendent du but recherché qui peut être par exemple une évaluation des élèves, une évaluation des exercices pour une meilleure utilisation de l'outil, une évaluation d'un groupe d'étudiants en vue de créer des groupes de niveaux.

1.1 Exemples d'exercices proposés dans l'EIAH

Le logiciel « Combien? » propose quatre machines correspondant à quatre classes de problèmes :

- Exemple d'exercice proposé pour la machine CE (Construction d'Ensembles) :
Ex n°2 : 5 cartes parmi 32 avec 2 Cœurs, 2 Piques, qui a pour énoncé :
« Dans un jeu de 32 cartes, combien peut-on construire de mains de 5 cartes, contenant exactement 2 cœurs et exactement 2 piques? »
- Exemple d'exercice proposé pour la machine LST (construction de Listes) :
Ex n°1 : Mots de 5 lettres avec 2 A, qui a pour énoncé :
« Combien peut-on écrire de mots de 5 lettres contenant exactement 2 fois la lettre A ? »
- Exemple d'exercice proposé pour la machine CEC (construction d'Ensembles par Cas) :
Ex n°2 : Jetons : 3 impairs, 1 rouge pair, 1 vert, dont l'énoncé est :

« On dispose de 9 jetons : 5 jetons rouges numérotés de 1 à 5 et de 4 jetons verts numérotés de 1 à 4. Combien y a-t-il de sous-ensembles possibles de 4 jetons avec au plus 3 jetons impairs, au plus un rouge et pair, exactement un vert »

- Exercice proposé pour la machine LSTC (construction de Listes par Cas) :
Ex n°1 : Mots de 5 lettres avec 3 voyelles et au plus 1 A, dont l'énoncé est :
« Combien peut-on écrire de mots de 5 lettres contenant exactement 3 voyelles et au plus une fois la lettre A ? »

Il faut noter tout de même que d'une machine à l'autre, la méthode de résolution consiste à :

- Définir l'univers qui est l'ensemble de toutes les configurations dont un sous ensemble est l'ensemble à dénombrer. Les éléments de ce sous ensemble doivent vérifier les contraintes.
- Puis à construire une configuration générique de ce sous ensemble, qui est une réunion de sous parties chacune vérifiant une contrainte donnée.

1.2 Structure d'une résolution

Pour la résolution d'un exercice, l'étudiant est amené à :

- Choisir un exercice parmi la liste d'exercices de la machine choisie préalablement
- Définir l'univers de résolution
- Spécifier les étapes de construction de l'élément générique :
Chaque étape consiste à :
 - o Saisir le nombre d'éléments vérifiant la contrainte à spécifier,
 - o Saisir la ou les propriété(s)
 - o Saisir la formule de calcul
- Saisir l'opération à effectuer entre les valeurs calculées à chaque étape

The screenshot shows the 'Machine Construction Ensemble' interface. The main window title is 'Machine Construction Ensemble' and the subtitle is 'Combien ?'. The interface is divided into several sections:

- Evaluation:** Contains the exercise title 'Exercice n° 1 Avec un jeu de 32 cartes, combien peut-on former de mains de 2 cartes comportant une dame et un as ?' and a 'Préférences' button.
- Univers:** Contains the universe description 'L'ensemble des parties à 2 éléments prises dans un jeu de 32 cartes' and a 'Changer Univers' button.
- Construction:** Contains a 'Nouvelle Étape' button, an 'Aide sur Construction' button, and an 'Effacer cette Étape' button.
- Étape:** Contains the step configuration area with fields for 'On veut tirer' (set to 1), 'éléments vérifiant' (set to 'une propriété'), and 'la hauteur' (set to 'est'). Below this are 'Étape Correcte', 'Formule' (set to 'valeur'), and 'n' (set to '4').
- Liste des Étapes:** Contains a list of steps: '1 élément tel que la hauteur est Dame : 4' and '1 élément tel que la hauteur est As : ?'. There are buttons for 'supprimer la sélection' and 'modifier la sélection'.
- Bottom:** Contains a 'Que faire avec?' field (set to '4'), a 'Valider la Construction' button, and a 'J'ai Terminé !!' button.

Annotations with blue arrows point to the following elements:

- Choix et énoncé de l'exercice:** Points to the exercise title.
- Choix de l'univers:** Points to the universe description.
- Nombre d'éléments vérifiant la propriété:** Points to the '1' in the 'On veut tirer' field.
- Une étape:** Points to the entire 'Étape' configuration area.
- Choix de la formule:** Points to the 'valeur' dropdown in the 'Formule' field.
- La propriété validée:** Points to the '4' in the 'Que faire avec?' field.

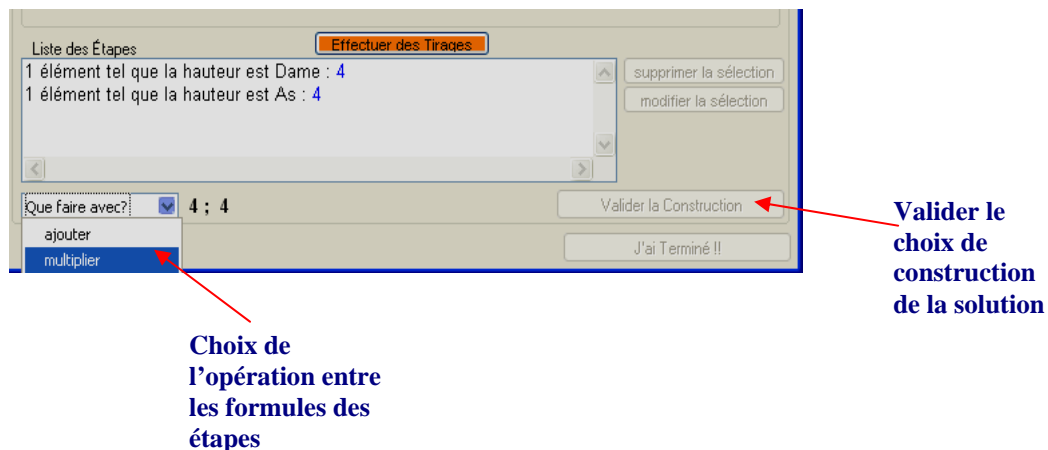


Fig 1 – Interface Combien ? et structure de la résolution

2. Déroulement du projet

Durant ce projet d'ingénierie, notre démarche a été la suivante :

1. Etudes des traces DESCRIPT et XML
 - 1.1 Conception d'un lexique
 - 1.2 Etudes de la structure de résolution (structure des documents xml)
2. Parsage du fichier xml : récupération de toutes les commandes
3. Conception de l'interface
 - 3.1 création des tables de statistiques
 - 3.2 création du texte d'analyse
 - 3.3 traitement des erreurs et des hésitations

3. Analyse des traces « .XML »

Nous nous sommes basées sur les traces d'utilisation de ce logiciel par les étudiants de Licence 1^{ère} année en Sciences et Techniques de l'Université Paris 5. Ces traces sont enregistrées dans un premier temps dans le format DESCRIPT, langage de type XML pour lequel nous disposons d'un traducteur en XML.

Une trace à analyser correspond à un fichier ou session d'apprentissage : il s'agit de l'enregistrement du travail d'un étudiant sur une machine.

Lors de cette session, l'étudiant peut avoir effectué un ou plusieurs exercices avec la même machine, c'est-à-dire de même type.

Les traces XML sont présentées de la sorte :

Fig 2 – Balises « commandes » dans un document xml.

Nous considérons qu'un fichier de trace XML est un ensemble d'actions effectuées par l'étudiant. Ces actions sont décrites dans des balises `<executer concept = «Script»>` ou `<executer concept = «Commande»>` car les traces ont été définies pour pouvoir rejouer l'exercice.

Nous nous intéresserons particulièrement à la deuxième balise qui définit principalement les commandes effectuées par l'étudiant et la réaction du système (erreur).

Une commande possède :

- Le nom de l'action effectuée,
- Le libellé : ou argument de cette action,
- Le temps auquel l'action a été effectuée.

Les balises des commandes rencontrées sont :

```
<executer concept="Commande" time="54">
  <commande as="Symbol">choisir</commande>
  <argument>Ex n° 3 : 5 cartes parmi 32 avec 2 Cœurs, 2 Dames</argument>
</executer>
```

Cette commande désigne le choix d'un exercice. L'étudiant a mis 54 secondes avant de choisir l'exercice.

Le libellé de l'exercice est entre les balises `<argument/>`.

```
<executer concept="Commande" time="55">
  <commande as="Symbol">valider</commande>
  <argument>Valider</argument>
</executer>
```

Commande de validation du choix précédent.

```
<executer concept="Commande" time="64">  
  <commande as="Symbol">saisirCardinal</commande>  
  <argument>5</argument>  
</executer>
```

Commande de saisie du cardinal de l'univers ou de l'étape.

```
<executer concept="Commande" time="66">  
  <commande as="Symbol">saisirReferentiel</commande>  
  <argument>un jeu de 32 cartes</argument>  
</executer>
```

Commande de saisie du référentiel de l'univers choisi.

```
<executer concept="Commande" time="67">  
  <commande as="Symbol">valider</commande>  
  <argument>Ok</argument>  
</executer>
```

Commande de validation pour vérification.

```
<executer concept="Commande" time="1002">  
  <commande as="Symbol">creerInstance</commande>  
  <argument>Nouvelle Ã‰tape</argument>  
</executer>
```

Commande de création/instanciation d'une nouvelle étape de résolution.

```
<executer concept="Commande" time="966">  
  <commande as="Symbol">fixerNbProprietes</commande>  
  <argument>une propriété</argument>  
</executer>
```

Commande pour fixer le nombre de propriétés dans une étape.

```

<executer concept="Commande" time="971">
  <commande as="Symbol">saisirAttribut</commande>
  <argument>la hauteur</argument>
</executer>

<executer concept="Commande" time="972">
  <commande as="Symbol">saisirRelation</commande>
  <argument>est</argument>
</executer>

<executer concept="Commande" time="978">
  <commande as="Symbol">saisirCible</commande>
  <argument>Dame</argument>
</executer>

```

Commandes pour saisir l'attribut, la relation et la cible d'une propriété dans une étape.

```

<executer concept="Commande" time="985">
  <commande as="Symbol">verifier</commande>
  <argument>Vérifier l'Étape</argument>
</executer>

```

Commande pour vérifier l'étape construite précédemment.

```

<executer concept="Commande" time="988">
  <commande as="Symbol">selectType</commande>
  <argument>C(n,p)</argument>
</executer>

```

Commande pour la saisie de la formule.

```

<executer concept="Commande" time="996">
  <commande as="Symbol">saisirN</commande>
  <argument>3</argument>
</executer>

<executer concept="Commande" time="997">
  <commande as="Symbol">saisirP</commande>
  <argument>2</argument>
</executer>

```

Commandes pour saisir la valeur de N et P des formules C(n,p) ou A(n,p) ou des valeurs n, ou n!.


```
<executer concept="Commande" time="999">
  <commande as="Symbol">valider</commande>
  <argument>Ajouter l'Étape</argument>
</executer>
```

Commande de validation d'une étape avant ajout de celle-ci dans la construction.

```
<executer concept="Commande" time="1101">
  <commande as="Symbol">selectOperateur</commande>
  <argument>multiplier</argument>
</executer>
```

Commande de saisie de l'opération les formules des étapes ajoutée à la construction de la solution.

```
<executer concept="Commande" time="1102">
  <commande as="Symbol">valider</commande>
  <argument>Valider la Construction</argument>
</executer>
```

Commande de validation de la construction de la solution.

```
<executer concept="CommandeAvecErreur">
  <message concept="Message">
    <erreur>CombienFinalFaux</erreur>
  </message>
</executer>
```

Commande indiquant un message d'erreur. Elle correspond à un message du système.

```
<executer concept="Commande" time="49">
  <commande as="Symbol">aide</commande>
  <argument>Aide sur Univers</argument>
</executer>
```

Commande indiquant la consultation de l'aide pour le choix de l'univers.

```
<executer concept="Commande" time="1174">
  <commande as="Symbol">selectionner</commande>
  <argument>Cas 2 À l'Échelle(s) vérifiant : la hauteur est Dame</argument>
</executer>
```

```
<executer concept="Commande" time="1175">
  <commande as="Symbol">supprimer</commande>
  <argument>supprimer la sélection</argument>
</executer>
```

Commandes de sélection puis suppression d'une propriété après ajout dans la construction.

```
<executer concept="Commande" time="2089">  
  <commande as="Symbol">valider</commande>  
  <argument>J'ai Terminé !!</argument>  
</executer>
```

Commande indiquant que l'étudiant a réussi l'exercice.

Nous nous sommes intéressées à cette structure car elle offre un grand avantage d'un point de vue conception pour l'analyse des actions dans une session de travail.

4. Etude de l'IHM

Notre interface doit permettre à un professeur ou à un étudiant souhaitant connaître son niveau d'avoir à la fois une vision globale et détaillée des informations contenues dans une trace : une base de données était indispensable.

Les premières idées d'analyse ont été de présenter le nombre d'erreurs, le nombre d'hésitations et de consultations d'aide pour chaque exercice d'une session.

Ces données numériques ne semblaient pas cependant suffisantes, il fallait permettre à l'utilisateur de situer ces informations au cours de la résolution de l'exercice.

Cela nous a conduit à nous pencher sur la manière de présenter le déroulement de l'exercice en graphe ou en texte. Cela dit, la présentation en graphes est intéressante mais le texte est suffisamment explicite.

Cela nous a conduit à opter pour la transcription des traces en langage naturel, qui montrent notamment les actions effectuées.

D'un point de vue ergonomie, nous avons souhaité mettre en valeur les parties inhérentes et importantes afin que l'utilisateur puisse se repérer facilement dans la trace, exemple : le choix de l'univers, la construction avec la création de chaque étape et la fin de celle-ci.

5. Conception

5.1 Extraction des données des scripts XML en classes de Document, Exercice, et Commande.

Les scripts sous le format XML se présentent de la forme suivante :

La racine est `<document concept="[nom de la Machine]">`

Cette racine possède autant de fils qu'il y a d'exercice dans ce document. Comme vu dans la partie analyse, il existe plusieurs types de nœuds, les nœuds

- `<executer concept="Script">`
- `<executer concept="Commande" time="[...]">`
- Et les nœuds de type erreur.

Les nœuds commandes correspondent à des actions faites par l'étudiant ou à des réactions du système. Alors que les nœuds scripts correspondent aux grandes parties de résolution comme : le choix de l'univers, la construction d'un élément générique.

Le parseur analyse le fichier ligne par ligne, et en extrait principalement les commandes et les erreurs et les range dans une structure linéaire.

Le fait de changer des traces à structure arborescente en structure linéaire n'est pas handicapant :

- la résolution d'un exercice se fait bien linéairement au fil du temps (pas d'actions parallèles qui nécessiteraient d'être représentées comme fils d'un même nœud.) Cela reste vrai même dans les cas de retour arrière après une modification (grâce au libellé de l'action « selectType »)
- Et grâce à la structure de la résolution d'un exercice vue dans la partie 1.2. et des différentes commandes exécutées par l'étudiant, il est aisé de retrouver la position d'une commande par rapport à l'ensemble de la résolution de l'exercice.

Chaque instance d'exercice contiendra donc un vecteur de commandes et d'erreur. La classe document, regroupe un nombre variable d'exercices.

5.2 Analyse d'un exercice

Après le parsing du fichier XML, et la création du vecteur de commandes et d'erreur, il est nécessaire d'appliquer un prétraitement à toutes ces données pour avoir les informations que nous avons décidé de montrer à l'utilisateur, à savoir :

- nombre d'erreurs
- nombre d'hésitations
- nombre de consultations d'aide
- nombre d'étapes
- nombre de valeurs pour le dénombrement des cas d'une étape
- nombre de formules : nombre de fois où une formule ou une valeur a été validée.

- la réussite ou la non réussite de l'exercice

Pour le nombre d'erreurs il suffisait juste de compter la taille du vecteur d'erreurs.

La réussite ou non de l'exercice est déterminée par la présence ou non de la commande de cliquage sur le bouton « J'ai Terminé !! ».

Le nombre de consultations d'aide nécessite aussi un parcours de l'ensemble des commandes pour compter celles où l'action était « aide »

Le nombre de consultations d'aide n'est pas associé spécialement à une erreur commise (suite à une action erronée pour la solution, un message d'erreur qui apparaît)

Pour compter le nombre d'étapes, nous nous sommes basées tout d'abord sur les commandes « Ajouter l'Etape », mais nous avons vu qu'il fallait considérer les cas où l'étudiant supprimait l'étape ou alors la modifiait. De même que pour le comptage de « nombre de valeurs » et « nombre de formules ».

Exemple :

ACTION	nbEtapes	nbValeur	nbFormule
[temps = 35 sec] Action : selectType - Libelle : valeur			
<i>Avec l'action ci-dessus, le programme mémorise que l'étudiant a choisi de donner le nombre de possibilités correspondant à cette étape. Il n'incrémentera nbValeur qu'à l'action ci dessous :</i>			
[...]			
Fin de l'Etape 1 (--> Etape ajoutée)	1	1	
[temps = 44 sec] L'étudiant sélectionne la propriété : 1 élément tel que la hauteur est Dame : 4	1	1	
<i>Lorsqu'on tombe sur la sélection d'une étape, le programme analyse si le dernier élément est un nombre ou une formule. Ici, c'est une valeur. Il mémorise ce fait, en prévision de la suite des traces.</i>			
[temps = 46 sec] L'étudiant va modifier la sélection	1	1	
<i>Mais ici, l'étudiant décide de ne rien modifier à cette sélection.</i>			
[temps = 51 sec] L'étudiant sélectionne la propriété : 1 élément tel que la hauteur est Dame : 4	1	1	
[temps = 52 sec] L'étudiant va supprimer la sélection	0	0	
<i>A l'action de suppression d'une étapes, le programme décrémente automatiquement le nombre d'étape, et il choisit de décrémente la valeur de nbValeur car il a mémorisé que l'étape sélectionnée avait une valeur.</i>			

Pour compter le nombre d'hésitations, nous avons :

- testé chaque commande avec les suivantes
- comparé les actions
- si les actions relatives à la résolution mathématique de l'exercice comme choisirCardinal, saisirN, etc., sont les mêmes pour 2 commandes successives, mais que les libellés de ces actions sont différents alors nous avons considéré cela comme une hésitation.
- Par contre, cela ne s'applique pas à des actions de validation, création d'étapes, etc.

Mis à part ces informations numériques, nous avons décidé de transformer cette succession de commandes en une trace plus proche du langage naturel pour une meilleure lisibilité des traces par le professeur. Initialement, les données que l'on extrait des traces XML, sans traitement, ressemblent à cela :

----> Exercice = Ex n° 2 : Mains de 5 cartes parmi 32 avec 2 dames
0 [temps = 10] Action : choisir - libelle : Ex n° 1 : Mains de 5 cartes avec 2 coeurs parmi 32
1 [temps = 11] Action : choisir - libelle : Ex n° 1 : Mains de 5 cartes avec 2 coeurs parmi 32
2 [temps = 16] Action : choisir - libelle : Ex n° 2 : Mains de 5 cartes parmi 32 avec 2 dames

Ci-dessus, le choix de l'exercice importe peu, donc nous n'affichons que l'exercice effectivement choisi.

3 [temps = 23] Action : valider - libelle : Valider
4 [temps = 34] Action : saisirCardinal - libelle : 1
5 [temps = 35] Action : saisirCardinal - libelle : 1
6 [temps = 55] Action : saisirReferentiel - libelle : les chiffres
7 - [erreur] UniversMauvaisReferentiel
8 - [erreur] UniversMauvaisCard
9 [temps = 79] Action : saisirCardinal - libelle : 2
10 [temps = 83] Action : saisirReferentiel - libelle : un jeu de 32 cartes
11 - [erreur] UniversMauvaisCard
12 [temps = 89] Action : saisirCardinal - libelle : 2
13 [temps = 92] Action : saisirCardinal - libelle : 2
14 - [erreur] UniversMauvaisCard
15 [temps = 206] Action : saisirCardinal - libelle : 4
16 - [erreur] UniversMauvaisCard
17 [temps = 215] Action : saisirCardinal - libelle : 5
18 [temps = 216] Action : valider - libelle : Ok

L'action 18 affichée telle quelle n'apporte pas d'information : le « ok » correspond à la validation du choix du référentiel et du commencement de la résolution. Donc dans la trace en langage naturel, nous explicitons cette action en « [temps = 216 sec] L'étudiant commence la construction d'un élément générique de l'ensemble à dénombrer... »

19 [temps = 220] Action : creerInstance - libelle : Nouvelle Étape
20 [temps = 224] Action : saisirCardinal - libelle : 2
21 [temps = 229] Action : fixerNbProprietes - libelle : une propriété
22 [temps = 232] Action : saisirAttribut - libelle : la hauteur
23 [temps = 234] Action : saisirRelation - libelle : est
24 [temps = 237] Action : saisirCible - libelle : Dame

Ces trois dernières actions sont plus parlantes, présentées en une seule ligne : « La propriété choisie est " la hauteur est Dame. " »

Ainsi, chacune des commandes sera réécrite avec un texte plus proche du langage naturel, et sera étiquetée par le nom de son type : hésitation, erreur, aide, début d'étape, fin d'étape, étape ajoutée, fin de l'exercice, annulation, suppression et modification. Lors de l'affichage du texte de la trace, en fonction de chaque type de commande, le texte sera mis en forme, et certaines informations seront mises en avant.

5.3 Enregistrement de la base de données

Nous avons une base de données avec 2 tables : une table des erreurs commises et une table contenant les informations générales des exercices. Ces tables servent à faire des statistiques sur le taux de réussite d'un exercice, les types d'erreurs les plus commises dans un exercice, etc.

Ces données sont stockées dans une base de donnée semi-structurée : au format XML.

Lors de l'analyse d'un exercice, les informations générales suivantes sont conservées dans la table des informations générales, trouvées par le prétraitement :

- identification de l'étudiant (nom du fichier)
- nom de la machine
- nom de l'exercice
- nombre d'erreurs
- nombre d'hésitations
- nombre de consultations d'aide
- nombre d'étapes
- nombre de valeurs
- nombre de formules
- la réussite ou la non réussite de l'exercice

Les informations sur l'exercice seront stockées dans un élément de la table sous cette forme :

```
<element>
  <id>numéro de l'élément dans la base</id>
  <session> identification de l'étudiant (nom du fichier)</session>
  <machine>nom de la machine</machine>
  <exercice>nom de l'exercice</exercice>
  <nbErreur>nombre d'erreurs</nbErreur>
  <nbHesitation>nombre d'hésitations</nbHesitation>
  <nbAide>nombre de consultations de l'aide</nbAide>
  <nbEtape>nombre d'étapes</nbEtape>
  <nbValeur>nombre de valeurs</nbValeur>
  <nbFormule>nombre de formules</nbFormule>
  <reussite>la valeur « true » ou « false » selon la réussite de
l'exercice</reussite>
</element>
```

Chaque élément correspond donc à un exercice et une table est composée de plusieurs éléments. Ce qui nous donne, pour la table des informations générales, le fichier XML suivant :

```

<table>
  <element>...</element>
  <element>...</element>
  ...
</table>

```

La table des erreurs commises est créée avec la même structure, sauf que les informations des éléments sont :

- le nom de la session
- le nom de la machine
- le nom de l'exercice
- le nom du type d'erreur
- le nombre de fois où cette erreur a été commise dans cet exercice.

Ce qui donne un élément de cette forme :

```

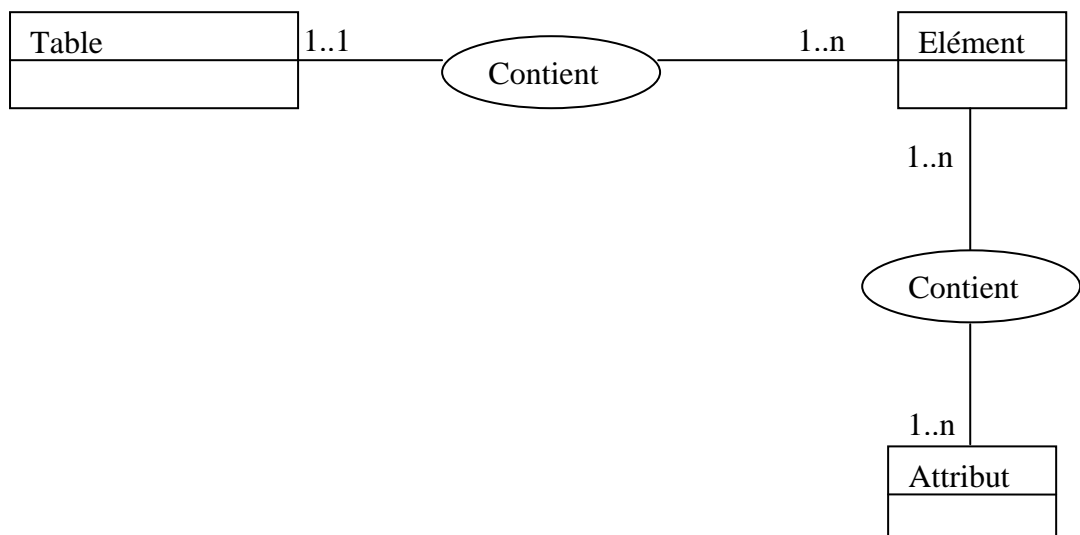
<element>
  <id>numéro de l'élément dans la base</id>
  <session> identification de l'étudiant (nom du fichier)</session>
  <machine>nom de la machine</machine>
  <exercice>nom de l'exercice</exercice>
  <typeErreur>type de l'erreur</typeErreur>
  <nbErreur>nombre de fois où l'erreur a été commise dans cet
exercice</nbErreur>
</element>

```

L'utilisation de la base de données se fait de la manière suivante :

- on parse le fichier XML de la table (erreur ou informations générales)
- on crée une instance de la classe TableStat ou TableErreur
- et on ne manipule que des vecteurs, ce qui est d'accès et de manipulation rapide.
- A partir de ces classes, il est possible de réenregistrer le nouvel état de la table dans le fichier XML.

Voici le schéma d'une table de notre base de données :



6. Notre interface

A l'ouverture d'un fichier de trace XML, nous présentons à l'utilisateur un premier onglet « Tous les exercices » où il peut visionner un tableau récapitulatif des exercices contenus dans le fichier de trace. Chaque ligne correspond à un exercice, et grâce au petit marqueur de couleur à gauche, les exercices réussis sont rapidement repérés (marqueur vert). Toutes les informations affichées pour ces exercices permettent au professeur de choisir les exercices dont il souhaite analyser la résolution de l'étudiant en détail.

Le champ « clé » dans le tableau permet de connaître le numéro de l'onglet qui présente le détail de l'exercice correspondant.

de	session	machine	exercice	temps	nbErreur	nbHesitation	nbAide	nbEtape	nbFermule	nbValeur
0	CE-hv08231.xml	CE	Ex n° 2 : Mains de 5 cartes parmi 32 avec 2 dames	229	2	5	0	1	0	0
1	CE-hv08231.xml	CE	Ex n° 3 : Mains de 5 cartes parmi 32	123	0	0	0	1	1	0
2	CE-hv08231.xml	CE	Ex n° 2 : Mains de 5 cartes parmi 32 avec 2 dames	349	6	4	0	2	4	0
3	CE-hv08231.xml	CE	Ex n° 1 : Mains de 5 cartes avec 2 coeurs parmi 32	119	0	1	0	3	3	0
4	CE-hv08231.xml	CE	Ex n° 5 : 5 cartes parmi 32 avec 2 Coeurs, 2 Piques	99	0	1	0	3	3	0
5	CE-hv08231.xml	CE	Ex n° 9 : 12 cartes parmi 32 avec 7 Coeurs, 4 As, 2 Piques	122	2	0	0	2	0	0
6	CE-hv08231.xml	CE	Ex n° 9 : 12 cartes parmi 32 avec 7 Coeurs, 4 As, 2 Piques	511	17	5	0	4	3	0

Fig. 3 - Tableau statistiques sur une même session

Il est aussi possible d'ouvrir plusieurs fichiers de traces en même temps, dans ce cas, cet onglet aura l'apparence suivante (Figure 4). La couleur des lignes permet de reconnaître les exercices d'un même fichier de traces. De même que, les onglets d'exercices ont une petite icône de couleur différente à chaque groupe d'exercices d'un même fichier de traces.

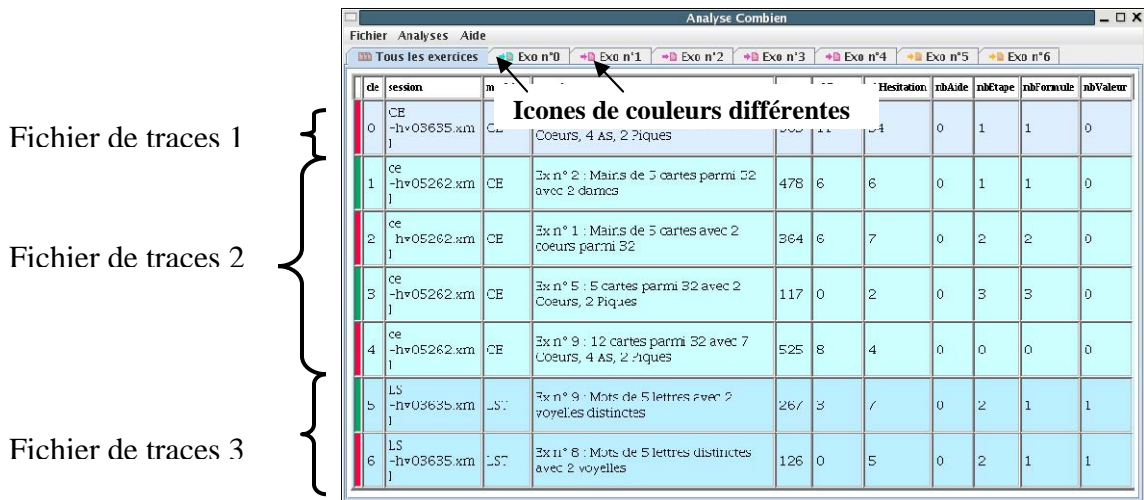


Fig. 4 - Tableau statistiques sur plusieurs sessions

En cliquant sur l'onglet dans exercice pour en avoir les détails de résolution, l'utilisateur a à sa disposition plusieurs pages qui présentent les différents aspects de résolution de cet exercice.

La première page montre les statistiques sur la résolution par l'élève courant (session courante) et en parallèle les statistiques sur cet exercice enregistré dans la base de données (statistiques résultantes des résolutions du même exercice par d'autres élèves).

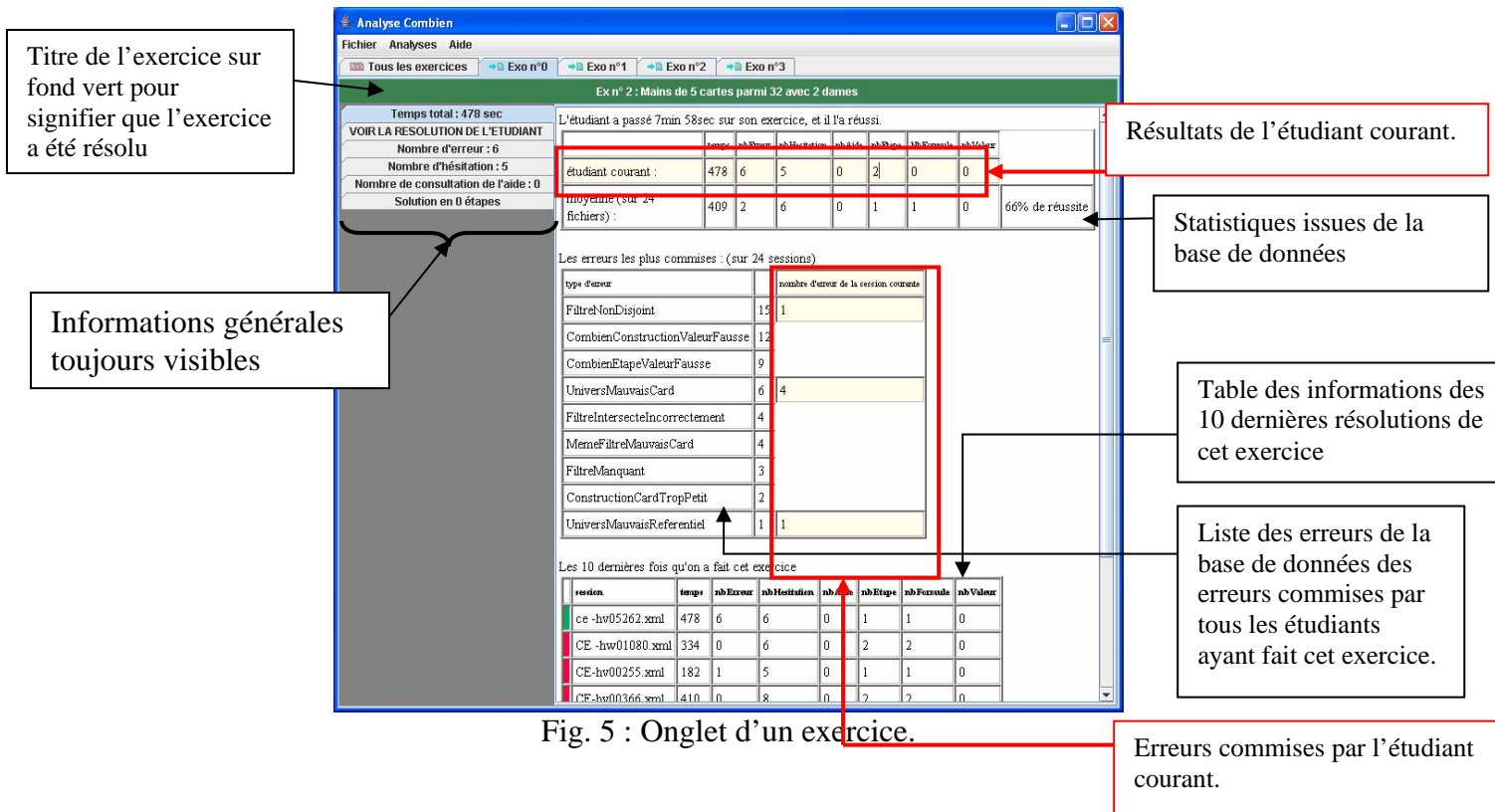


Fig. 5 : Onglet d'un exercice.

Mise en évidence des étapes pour aider l'utilisateur pour se repérer dans la trace.

Nous avons gardé l'affichage du temps auquel l'action a été lancée pour que l'utilisateur puisse avoir une vision chronologique de la résolution : si l'étudiant met du temps cela peut signifier une difficulté à ce niveau de l'exercice.

Les actions les plus courantes sont écrites en langage naturel de telle façon que cela permette une meilleure compréhension pour l'utilisateur.

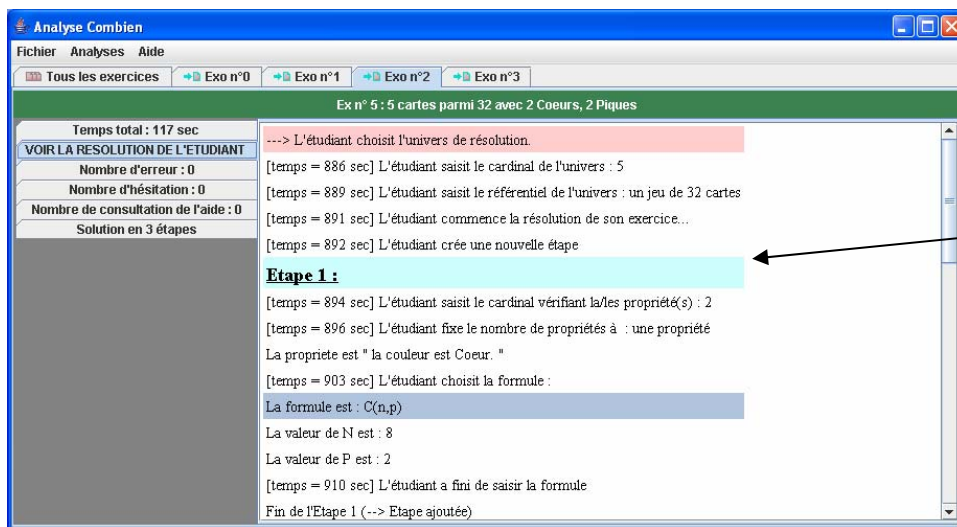


Fig.6 – Trace analysée par texte

L'utilisateur, peut consulter la base de données des exercices, il peut choisir d'afficher les exercices d'une machine en particulier.

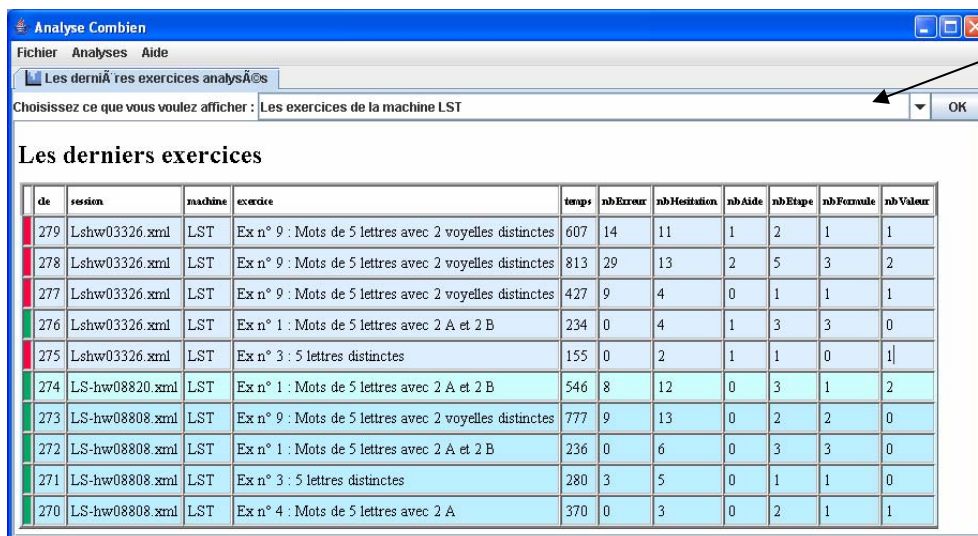


Fig.7 La base de données (toutes les dernières traces)

7. Tests et interprétation des résultats

Voici les informations sur un exercice :

L'étudiant a passé 5min 49sec sur son exercice, et il l'a réussi.

	temps	nbErreur	nbHesitation	nbAide	nbEtape	NbFormule	nbValeur
--	-------	----------	--------------	--------	---------	-----------	----------

étudiant courant :	349	6	4	0	2	2	0	
moyenne (sur 24 fichiers) :	409	2	6	0	1	1	0	66% de réussite

Dans ce premier tableau, l'utilisateur peut comparer le niveau de l'étudiant par rapport aux autres étudiants qui ont effectué le même exercice.

Sur 24 sessions/étudiants, le temps de résolution de l'exercice est en moyenne de 409 secondes. On peut voir que l'étudiant courant a passé moins de temps, néanmoins, le nombre d'erreurs effectuées (6) est supérieur à la moyenne (2) : on en déduit que l'étudiant s'est précipité dans la résolution.

Ce tableau permet aussi de montrer le nombre de saisie de formules (compter le nombre de possibilités vérifiant une propriété), ainsi que le nombre de saisie de valeurs : dans un cours de dénombrement il est approprié d'utiliser des formules du type $C(n,p)$, $A(n,p)$, $n!$, n^p .

Le professeur saura ainsi, selon l'exercice, si l'étudiant a acquis une facilité à les utiliser plutôt que de calculer la valeur brute du nombre de possibilités.

Les erreurs les plus commises : (sur 24 sessions)

type d'erreur		nombre d'erreur de la session courante
FiltreNonDisjoint	15	1
CombienConstructionValeurFausse	12	3
CombienEtapeValeurFausse	9	2
UniversMauvaisCard	6	
FiltreIntersecteIncorrectement	4	
MemeFiltreMauvaisCard	4	
FiltreManquant	3	
ConstructionCardTropPetit	2	
UniversMauvaisReferentiel	1	

Ce tableau met en évidence les types d'erreurs rencontrées pour cet exercice sur un nombre de sessions (ici 24). Les erreurs sont rangées par ordre de fréquence d'apparition, l'utilisateur de l'interface pourra ainsi pister les difficultés des étudiants.

Ainsi, en disposant de la fréquence de chaque type d'erreur dans la session courante, le professeur pourra juger du cas particulier ou non de l'étudiant courant :

Si l'étudiant effectue des erreurs que peu d'étudiants commettent, cela voudrait signifier qu'il ait tenté une autre manière de résoudre l'exercice ou bien qu'il ait mal assimilé le cours par rapport aux autres.

Les 10 dernières fois qu'on a fait cet exercice

	session	temps	nbErreur	nbHesitation	nbAide	nbEtape	nbFormule	nbValeur
	ce -hv05262.xml	478	6	6	0	1	1	0
	CE -hw01080.xml	334	0	6	0	2	2	0

CE-hv00255.xml	182	1	5	0	1	1	0
CE-hv00366.xml	410	0	8	0	2	2	0
CE-hv00366.xml	79	0	5	0	2	2	0
ce-hv04127.xml	390	0	8	0	2	2	0
CE-hv08231.xml	229	2	5	0	0	0	0
CE-hv08231.xml	349	6	9	0	2	2	0
CE-hw00037.xml	1337	5	9	0	2	2	0
CE-hw00381.xml	495	6	8	0	2	2	0

Nous proposons également les résultats issus des 10 dernières sessions où cet exercice a été résolu (réussi ou pas).

Ce tableau est affiché en complément du premier tableau, car la notion de « moyenne » a tendance à généraliser les informations déductibles alors qu'il serait plus intéressant d'avoir accès aux données détaillées.

Nous avons constitué notre base de données d'exercice et des erreurs à partir de toutes les traces qui nous ont été fournies (71 fichiers, 281 exercices en tout).

Pour notre jeu de test, nous avons sélectionné une dizaine de traces parmi ces dernières, et nous l'avons complété par quelques traces créées par nous-mêmes.

En analysant quelques traces fournies, nous avons pu détecter différents profils d'étudiants :

- l'étudiant qui s'amuse à cliquer et à résoudre l'exercice sans réfléchir (un grand nombre d'erreur par rapport au temps passé sur l'exercice)
- l'étudiant qui consulte souvent l'aide et qui ne commet pas les mêmes erreurs un grand nombre de fois.

Nous avons noté alors que ces bilans numériques ne peuvent être totalement fiables.

Exemple de cas d'utilisation :

Prenons un étudiant qui a l'habitude de se précipiter dans la résolution d'un exercice, mais à une étape donnée, il doit s'absenter un long moment. Si l'utilisateur se contentait d'observer les statistiques numériques, il ne réaliserait pas que l'étudiant a cette mauvaise habitude.

C'est pourquoi les traces sont utiles à ce moment là : en lisant la trace, l'utilisateur pourrait voir la longue pause de l'étudiant durant la résolution de l'exercice. Ainsi, il portera une appréciation plus juste.

8. Problèmes rencontrés

Au cours de notre projet, nous avons rencontrés quelques difficultés comme :

- Les problèmes liés à l'analyse des traces :
Le listing des différentes balises (notamment celles des différentes commandes) a été laborieux. En effet, il existe certaines commandes propres à un type de machine.
Le jeu de test nous a permis de mieux les transcrire en langage naturel de façon à préserver l'homogénéité du texte.

- La gestion des fichiers résultats :
Le fait qu'il n'y ait pas de détection immédiate de l'existence du fichier résultat (contenant les statistiques) que l'utilisateur souhaite enregistrer, peut fausser les statistiques sur un exercice entre autres.
- L'un des problèmes majeurs est le codage des accents, cela peut être un inconvénient pour la portabilité du programme. Il faudra pour cela revoir les fonctions de comparaison de chaînes de caractères.

9. Conclusion et perspectives

Notre travail a consisté à proposer une interface dédiée à des professeurs d'universités de l'équipe Combien?, à travers laquelle nous proposons différents modes d'analyse des traces de résolution d'exercices de dénombrements issues du logiciel d'enseignement « Combien ? »

Ce projet nous a permis de voir concrètement en quoi consistait l'analyse de traces d'un environnement interactif d'apprentissage humain.

De plus, la base de données existante offre une quantité d'informations importante, il serait enrichissant pour l'interface de mettre en place un système plus complet de gestion et d'affichage des informations issues de la base de données.

En effet, nous avons pu afficher les 10 premiers exercices en fonction de la machine, il serait intéressant de donner la possibilité à l'utilisateur de fixer lui même le nombre d'exercices à afficher, ou bien de saisir le nom de l'exercice pour la sélection dans la base de données.

L'une des fonctionnalités qui nous tenait à cœur mais que l'on n'a pas pu implémenter faute de temps est l'utilisation d'histogrammes et de graphes pour représenter au mieux les données statistiques.

Outre l'affichage, il sera intéressant de séparer dans les statistiques les exercices réussis de ceux non réussis.

Nous pensons que toutes ces fonctionnalités peuvent faire partie des perspectives en vue d'une reprise de notre contribution.

10. Lexique

Action : toutes les actions de l'étudiant enregistrées dans le script.

Univers : l'ensemble de toutes les configurations dont un sous ensemble est l'ensemble à dénombrer. Les éléments de ce sous ensemble doivent vérifier les contraintes.

Résolution d'un exercice : Correspondrait à toute la série d'actions pour résoudre l'exercice. Elle est schématisée de la sorte :

- un univers : ensemble des éléments duquel on extrait les éléments à compter.
- une construction : définition générique d'un élément quelconque de l'ensemble à dénombrer.

Le temps total de résolution de l'exercice : On considère le temps à partir de la validation de l'exercice ou l'abandon (changement d'exercice, ou fermeture de la machine). C'est donc le temps effectif que l'étudiant a passé sur l'exercice.

La méthode de résolution : correspond à la manière employée par l'étudiant pour résoudre son exercice. Elle a un nombre variable d'étapes, exemple :

Pour l'exercice «Ex n°10 : Combien peut-on former d'ensemble de 5 chiffres contenant exactement 2 chiffres impairs ? », l'étudiant a plusieurs possibilités pour résoudre cet exercice.

1^{ère} solution :

2 éléments tels que le chiffre est impair

→ Nombre de possibilités : $C(5,2) = 10$

3 éléments tels que le chiffre n'est pas impair

→ Nombre de possibilités : $C(5,3) = 10$

2^{ème} solution :

2 éléments tels que le chiffre n'est pas pair

→ Nombre de possibilités : $C(5,2) = 10$

3 éléments tels que le chiffre n'est pas impair

→ Nombre de possibilités : $C(5,3) = 10$

Les erreurs : Nous listons les erreurs selon leur type exemple :

- CombienEtapeValeurFausse
- CombienConstructionValeurFausse
- ConstructionCardTropPetit
- ConfigurationFinaleFausse
- ...

Les hésitations : Hésitation sur la valeur d'une action, exemple : « saisirCardinal : 7 » puis « saisirCardinal : 5 ».

Les consultations d'aide : Le nombre de consultations d'aide peu être significatif : sur l'efficacité du message d'aide, ou pour savoir le caractère de l'étudiant (s'il est capable de résoudre l'exercice ou la série d'exercices sans regarder l'aide).

11. [Annexe](#)

- Manuel du programmeur
- Manuel de l'utilisateur