

Usage de WIMS dans un lycée

Georges Khaznadar <georgesk@offset.org>

lycée Jean Bart – Dunkerque/OFSET

Novembre 2008



Table des matières

- 1 WIMS : Wys In Meer Scholen ?
 - ... ou Wêpe In Mårchure Si ?
 - Originalité de WIMS
- 2 Le comportement des élèves
 - Pompage, ou collaboration ?
 - Les examens
- 3 Crédits
 - Illustrations
 - Ce document



WWW Interactive Multipurpose Server

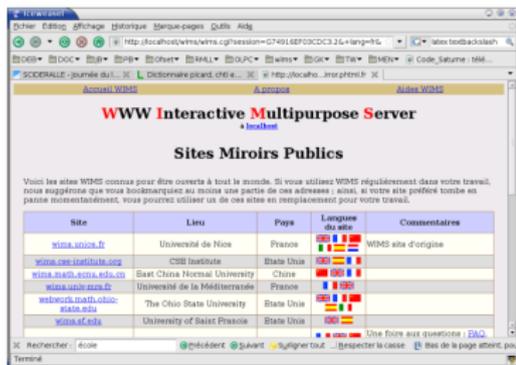
Il y a eu quelques concours d'idées pour donner un sens à l'acronyme WIMS, qui à l'origine signifiait « WWW Interactive Math System ». On peut tenter d'ajouter quelques essais en Picard ou en Flamand ... ?

Ce système a été créé par Gang XIAO il y a une dizaine d'années, et diffusé très tôt sous la licence libre GNU-GPL. C'est un moteur écrit en langage C, donc à la fois compact et très rapide, qui utilise le standard CGI pour collaborer avec un serveur web comme Apache. Ce moteur est extrêmement modulaire et accepte des greffons très divers, pourvu que ceux-ci arrivent à fonctionner sur la même machine et se laissent piloter par des échanges de fichiers ou de messages texte.



WIMS dans le monde, WIMS dans le lycée

Une liste de serveurs *officiels* existe, elle est accessible à partir de chacun des serveurs WIMS existants.



WWW Interactive Multipurpose Server
Sites Miroirs Publics

Voici les sites WIMS connus pour être ouverts à tout le monde. Si vous utilisez WIMS régulièrement dans votre travail, nous suggérons que vous bookmarkiez au moins une partie de ces adresses : ainsi, si votre site préféré tombe en panne momentanément, vous pourrez utiliser un de ces sites en remplacement pour votre travail.

Site	Lieu	Pays	Langue de site	Commentaires
wims.unice.fr	Université de Nice	France	FR FR FR	WIMS site d'origine
wims.cwi.tue.nl	CWI Institute	Etats Unis	FR FR FR	
wims.math.ucla.edu	Sant Clara Normal University	Chine	FR FR FR	
wims.univ-montp.fr	Université de la Méditerranée	France	FR FR FR	
webwork.math.ohio-state.edu	The Ohio State University	Etats Unis	FR FR FR	
wims.usf.edu	University of Saint Francis	Etats Unis	FR FR FR	

FR FR FR Une foire aux questions | FAQ



La génération à la volée d'exercices variants

Dès le départ, WIMS a été fourni équipé d'un langage de haut niveau, bien plus adapté que le langage C pour créer des modules de cours et d'exercices.



La génération à la volée d'exercices variants

Dès le départ, WIMS a été fourni équipé d'un langage de haut niveau, bien plus adapté que le langage C pour créer des modules de cours et d'exercices. Ce langage (**ModTool** pour ne pas le nommer) permet d'écrire directement les pages web, un peu à la façon de PHP, directement dans le code.



La génération à la volée d'exercices variants

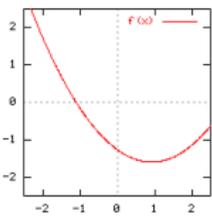
Dès le départ, WIMS a été fourni équipé d'un langage de haut niveau, bien plus adapté que le langage C pour créer des modules de cours et d'exercices. Ce langage (**ModTool** pour ne pas le nommer) permet d'écrire directement les pages web, un peu à la façon de PHP, directement dans le code. Il possède des enrichissements qui permettent de réaliser une petite programmation et surtout des **primitives de haut niveau** facilitant les tirages au sort les plus divers afin de fabriquer des questions variantes. D'autres primitives permettent de lancer des analyses de réponses qui prennent en compte des syntaxes variées.



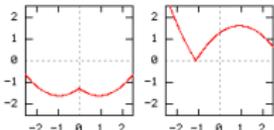
La génération à la volée d'exercices variants

Un exemple : un exercice avec choix entre deux clics (mot-clé : *absolute*)

Voici le graphe d'une fonction $f(x)$.



Question. Parmi les dessins suivants, lequel représente la fonction $f(|x|)$? Cliquez dessus.



... appelez ce même exercice sur deux ordinateurs différents, ou renouvelez l'exercice plus d'une fois.



OEF : Open Exercise Format

Très tôt, le langage ModTool a été complété par un langage de niveau supérieur, nommé **OEF** (pour Open Exercise Format), qui est beaucoup plus accessible, et qui formalise des objets pédagogiquement pertinents. Il offre des primitives pour :



OEF : Open Exercise Format

Très tôt, le langage ModTool a été complété par un langage de niveau supérieur, nommé **OEF** (pour Open Exercise Format), qui est beaucoup plus accessible, et qui formalise des objets pédagogiquement pertinents. Il offre des primitives pour :

- un énoncé (`\statement{ ... }`)



OEF : Open Exercise Format

Très tôt, le langage ModTool a été complété par un langage de niveau supérieur, nommé **OEF** (pour Open Exercise Format), qui est beaucoup plus accessible, et qui formalise des objets pédagogiquement pertinents. Il offre des primitives pour :

- un énoncé (`\statement{ ... }`)
- un champ de réponse (`\embed{référence de réponse}`)



OEF : Open Exercise Format

Très tôt, le langage ModTool a été complété par un langage de niveau supérieur, nommé **OEF** (pour Open Exercise Format), qui est beaucoup plus accessible, et qui formalise des objets pédagogiquement pertinents. Il offre des primitives pour :

- un énoncé (`\statement{ ... }`)
- un champ de réponse (`\embed{référence de réponse}`)
- une analyse de réponse (`\reply{ ... }{modèle}{type}`)



OEF : Open Exercise Format

Très tôt, le langage ModTool a été complété par un langage de niveau supérieur, nommé **OEF** (pour Open Exercise Format), qui est beaucoup plus accessible, et qui formalise des objets pédagogiquement pertinents. Il offre des primitives pour :

- un énoncé (`\statement{ ... }`)
- un champ de réponse (`\embed{référence de réponse}`)
- une analyse de réponse (`\reply{ ... }{modèle}{type}`)
- une rétroaction (`\feedback{condition}{ ... }`)



OEF : Open Exercise Format

Très tôt, le langage ModTool a été complété par un langage de niveau supérieur, nommé **OEF** (pour Open Exercise Format), qui est beaucoup plus accessible, et qui formalise des objets pédagogiquement pertinents. Il offre des primitives pour :

- un énoncé (`\statement{ ... }`)
- un champ de réponse (`\embed{référence de réponse}`)
- une analyse de réponse (`\reply{ ... }{modèle}{type}`)
- une rétroaction (`\feedback{condition}{ ... }`)
- une suggestion (`\hint{ ... }`)



OEF : Open Exercise Format

Très tôt, le langage ModTool a été complété par un langage de niveau supérieur, nommé **OEF** (pour Open Exercise Format), qui est beaucoup plus accessible, et qui formalise des objets pédagogiquement pertinents. Il offre des primitives pour :

- un énoncé (`\statement{ ... }`)
- un champ de réponse (`\embed{référence de réponse}`)
- une analyse de réponse (`\reply{ ... }{modèle}{type}`)
- une rétroaction (`\feedback{condition}{ ... }`)
- une suggestion (`\hint{ ... }`)
- une solution rédigée (`\solution{ ... }`). Notez bien qu'en son absence une correction automatique est émise.



OEF : Open Exercise Format

Voici une copie d'écran où certaines primitives ont été mises en évidence :

Modifier le fichier [src/airerectangle.oef](#) dans le module *Périmètres et Aires*:

```
\statement{
Un rectangle a une longueur de \tlongueur \unitel et une largeur de \t
<center>
<i>Attention aux unités !</i><br>
<br>
Quelle est son aire ? &nbsp; \embed {reply 1.6} \uniter^2
</center>
}
\latex{
Un rectangle a une longueur de \tlongueur \unitel et une largeur de \t
Quelle est son aire ,en $\uniter^2$? \dots
\emph{Attention aux unités}
}
\reply{}{\rep}
\text{rep=wims(translate , to . in \rep)}
\real{resultat=\rep}
\condition{Bonne Reponse}{\resultat=\aire}
\solution {La réponse est \treponse \uniter^2}
```

Enregistrer les changements

Abandonner



les roulettes d'OEF

OEF fournit des primitives diverses et variées pour fabriquer des exercices variants, en voici quelques-unes :



les roulettes d'OEF

OEF fournit des primitives diverses et variées pour fabriquer des exercices variants, en voici quelques-unes :

- un nombre aléatoire (`random{déb ... fin}`)



les roulettes d'OEF

OEF fournit des primitives diverses et variées pour fabriquer des exercices variants, en voici quelques-unes :

- un nombre aléatoire (`random{déb ... fin}`)
- un entier aléatoire (`randint{déb ... fin}`)



les roulettes d'OEF

OEF fournit des primitives diverses et variées pour fabriquer des exercices variants, en voici quelques-unes :

- un nombre aléatoire (`random{déb ... fin}`)
- un entier aléatoire (`randint{déb ... fin}`)
- un enregistrement aléatoire (`wims(randfile nom_fichier)`)



les roulettes d'OEF

OEF fournit des primitives diverses et variées pour fabriquer des exercices variants, en voici quelques-unes :

- un nombre aléatoire (`random{déb ... fin}`)
- un entier aléatoire (`randint{déb ... fin}`)
- un enregistrement aléatoire (`wims(randfile nom_fichier)`)
- une liste permutée (`shuffle{liste}`)



les roulettes d'OEF

OEF fournit des primitives diverses et variées pour fabriquer des exercices variants, en voici quelques-unes :

- un nombre aléatoire (`random{déb ... fin}`)
- un entier aléatoire (`randint{déb ... fin}`)
- un enregistrement aléatoire (`wims(randfile nom_fichier)`)
- une liste permutée (`shuffle{liste}`)

En combinant ces roulettes et quelques autres, on peut générer des exercices à données variantes dont la structure reste identique.



les roulettes d'OEF

La primitive **randint** qui permet d'obtenir un entier aléatoire est souvent utilisée ci-dessous :

Modifier le fichier *src/airerectangle.oef* dans le module *Périmètres et Aires*:

```
\title{Aire d'un rectangle}  
\language{fr}  
\range{-5..5}  
\author{Christophe CAILLET}  
\email{c.caillet@erevia.com}  
\computeanswer{no}  
\format{html}  
\precision{100000}  
  
\text{unites=mm,cm,dm,m,dam,hm,km}  
  
\integer{ilargeur=randint(11..999)}  
\integer{ilongueur=randint((\ilargeur+1)..1000)}  
\real{largeur=\ilargeur/10}  
\real{longueur=\ilongueur/10}  
\real{aire=\longueur*\largeur}  
\integer{ur=randint(3..5)}  
\integer{u1=randint(\ur-2..\ur+2)}  
\integer{u2=randint(\ur-2..\ur+2)}  
\real{longueur=\longueur*10^(\ur-u1)}
```

Enregistrer les changements

Abandonner



les roulettes d'OEF

Essayez ...

Classes virtuelles

--- Authentification de participant ---

Pour entrer dans **Quelques essais** de **Tests pour le MASTEL**, veuillez donner votre nom de login et votre mot de passe personnel.

Votre nom de login :

Votre mot de passe :

Entrer

[Cliquez ici](#) pour voir l'exercice en fonctionnement.

(login/pass = beaucamps/beaucamps)



Des outils pour un ENT

Le noyau de WIMS offre une gestion complète de groupes-classes, nommés classes virtuelles, dans les quelles on peut maintenir un cahier de textes, des exercices variants (le carnet de notes se remplit tout seul), des documents pour le cours, des ressources diverses, des forums de discussion supportant les formules mathématiques, des sessions d'examen à sujet prévisible.



WIMS force à une collaboration vertueuse

Quand les exercices sont variants, il ne sert à rien à un élève X de chercher à copier ou à extorquer la bonne réponse d'un élève Y, en effet chacun d'entre eux a le même exercice mais avec des données différentes, et donc la réponse attendue n'est pas la même pour chacun.

Donc quand deux élèves communiquent, je constate que leur dialogue porte toujours sur des éléments de haut niveau cognitif. Ce n'est jamais « faut cliquer à gauche ou à droite ? », c'est plutôt « explique-moi comment il faut faire ». Seule la communication sur la méthode de résolution a une valeur d'échange. Ça profite autant à l'élève qui aide qu'à celui qui est demandeur.



Score, noscore

Dans la situation dite de « feuille d'exercices », les élèves peuvent à tout moment passer en statut **noscore**, ce qui neutralise l'enregistrement des notes. Ils peuvent alors répéter autant de fois qu'ils le veulent un exercice, sans risque qu'une mauvaise note ne leur reste (ni une bonne note d'ailleurs).



Score, noscore

Dans la situation dite de « feuille d'exercices », les élèves peuvent à tout moment passer en statut **noscore**, ce qui neutralise l'enregistrement des notes. Ils peuvent alors répéter autant de fois qu'ils le veulent un exercice, sans risque qu'une mauvais note ne leur reste (ni une bonne note d'ailleurs).

Quand l'élève estime qu'il va réussir, quelle que soit la question suivante, il repasse en statut **score**, et dès lors sa note est enregistrée. On constate donc que la plupart des élèves obtiennent d'excellents résultats. En examinant les journaux, on peut constater que certains utilisent très peu *noscore* et que d'autres vont jusqu'à répéter les exercices quelques dizaines de fois.



Score, noscore

Dans la situation dite de « feuille d'exercices », les élèves peuvent à tout moment passer en statut **noscore**, ce qui neutralise l'enregistrement des notes. Ils peuvent alors répéter autant de fois qu'ils le veulent un exercice, sans risque qu'une mauvaise note ne leur reste (ni une bonne note d'ailleurs).

Quand l'élève estime qu'il va réussir, quelle que soit la question suivante, il repasse en statut **score**, et dès lors sa note est enregistrée. On constate donc que la plupart des élèves obtiennent d'excellents résultats. En examinant les journaux, on peut constater que certains utilisent très peu *noscore* et que d'autres vont jusqu'à répéter les exercices quelques dizaines de fois. Les élèves doués vont plus vite bien sûr, mais *toutes* les bonnes notes sont toujours méritées. Et il y a peu de dispositifs qui arrivent à inciter autant les élèves à répéter longtemps un exercice.



Examen à contenu prévisible

Par construction, WIMS autorise à créer un examen en puisant les contenus dans des feuilles d'exercices déjà publiées dans la classe virtuelle.



Examen à contenu prévisible

Par construction, WIMS autorise à créer un examen en puisant les contenus dans des feuilles d'exercices déjà publiées dans la classe virtuelle.

On pourrait bien sûr cacher les feuilles d'exercices et par là créer une surprise pour les étudiants, mais il est plus rentable de les publier, et de faire savoir que les questions de l'examen portent sur des exercices qu'on peut faire autant de fois que désiré par avance.



Examen à contenu prévisible

Par construction, WIMS autorise à créer un examen en puisant les contenus dans des feuilles d'exercices déjà publiées dans la classe virtuelle.

On pourrait bien sûr cacher les feuilles d'exercices et par là créer une surprise pour les étudiants, mais il est plus rentable de les publier, et de faire savoir que les questions de l'examen portent sur des exercices qu'on peut faire autant de fois que désiré par avance.

Au moment de l'examen, les élèves trouveront des difficultés qu'ils ont déjà affrontées, mais il ne leur est pas possible d'apporter une liste de réponses, ils le savent d'avance et ne le font jamais. Au lieu de ça, ils s'arment de leur savoir-faire.



Règles du jeu pour une session d'examen

Un examen ne peut se dérouler que dans un créneau horaire précis, à partir d'ordinateurs précisément désignés. Toute tentative de passer l'examen autrement échoue. Un élève qui tenterait d'ouvrir deux sessions d'examens différentes à partir de deux postes autorisés est aussi refoulé.



Règles du jeu pour une session d'examen

L'examen peut être tenté *trois fois* durant la session. On constate schématiquement deux stratégies de la part des élèves :



Règles du jeu pour une session d'examen

L'examen peut être tenté *trois fois* durant la session. On constate schématiquement deux stratégies de la part des élèves :

- 1 Un bon élève fait rapidement (en utilisant un peu moins que la moitié de la session), une première tentative, complète. Après ce repérage, il refait son « coup de maître » lors de la deuxième tentative, et il dispose cependant d'une possibilité de rattrapage en cas d'erreur d'étourderie.



Règles du jeu pour une session d'examen

L'examen peut être tenté *trois fois* durant la session. On constate schématiquement deux stratégies de la part des élèves :

- 1 Un bon élève fait rapidement (en utilisant un peu moins que la moitié de la session), une première tentative, complète. Après ce repérage, il refait son « coup de maître » lors de la deuxième tentative, et il dispose cependant d'une possibilité de rattrapage en cas d'erreur d'étourderie.
- 2 Un « moins bon » élève fera une première tentative, avortera cette tentative parce que l'ordinateur est méchant avec lui et lui demande des questions trop dures, avortera une deuxième tentative, puis fera jusqu'au bout une troisième (on ne peut pas l'avorter), cependant que l'ordinateur restera méchant.



Règles du jeu pour une session d'examen

L'examen peut être tenté *trois fois* durant la session. On constate schématiquement deux stratégies de la part des élèves :

- 1 Un bon élève fait rapidement (en utilisant un peu moins que la moitié de la session), une première tentative, complète. Après ce repérage, il refait son « coup de maître » lors de la deuxième tentative, et il dispose cependant d'une possibilité de rattrapage en cas d'erreur d'étourderie.
- 2 Un « moins bon » élève fera une première tentative, avortera cette tentative parce que l'ordinateur est méchant avec lui et lui demande des questions trop dures, avortera une deuxième tentative, puis fera jusqu'au bout une troisième (on ne peut pas l'avorter), cependant que l'ordinateur restera méchant.

Cette méthode des trois sessions fait que les notes d'examen présentent beaucoup de contraste. Les plaintes d'élèves sont rarissimes.

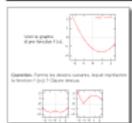


sources des illustrations, licences



copie d'écran de l'auteur, licence :

[Creative Commons Attribution ShareAlike](#)



copie d'écran de l'auteur, licence :

[Creative Commons Attribution ShareAlike](#)



copie d'écran de l'auteur, licence :

[Creative Commons Attribution ShareAlike](#)



copie d'écran de l'auteur, licence :

[Creative Commons Attribution ShareAlike](#)



sources des illustrations, licences



copie d'écran de l'auteur, licence :

[Creative Commons Attribution ShareAlike](#)



(C) Kyle Winkler licence : Free Art 



Licence de ce document

Ce document est librement copiable, dans le respect de la licence GNU-FDL (sans parties invariantes).



Autrement dit, vous pouvez le recopier tel quel, le distribuer, ou encore bâtir un document dérivé et le distribuer, à condition que le document dérivé soit régi par la même licence que celui-ci, et que les auteurs dont le copyright s'applique soient cités.

