A Computerized Adaptive System under WIMS

Jill-Jênn Vie
Supervisors: Jean-Pierre Boudine & Antoine Bodin

August 30, 2010

Contents

List of Figures

Introduction

I did my internship at LTE: "Le Test Édumétrique" ("édumétrie" means "measurement of learning"). Throughout the summer, I understood the stakes of such a company, which weren't clear for me at first sight.

1 Definitions

1.1 The core knowledge

France is moving towards an educational crash [5].

Studies of PISA (Programme for International Student Assessment) show that France is ranked 13$^{\text{th}}$ among the 29 countries of the OECD class [3]. One of the many reasons is our schedules: they are way too heavy. It would be more efficient to guarantee a real minimum level, a *core knowledge* for all pupils of an age group.

In Mathematics, several experts built a question bank made up of 5 categories: Geometry, Measurement and Quantity, Numbers, Proportionality and Uncertainty. All of this now has to be tested.

1.2 CAT

Most French tests are built linearly: the subject contains questions 1 thru n, and the examinee has to do all of them, no matter what the order.

A *CAT* (Computerized Adaptive Testing) system selects in a bank the questions asked to the examinee, based on his/her previous answers. In other words, the question n depends on the answers to the questions $1, \ldots, n-1$. Besides, some rules decide the first question and others determine when the test can be stopped. The main advantage of such a system is to determine the most accurate score with few questions.

For example, Akinator [2] (or 20Q [1]) is a "genius" (an artificial intelligence) that can guess what you are thinking of with approximately 20 questions. Throughout the game, it asks the question that will bring the most information to it. When Akinator is convinced to have found what was in your mind, it goes on asking, just to enhance its knowledge. Similarly, in SAT (Scholastic Aptitude Test), there are some questions that doesn't count in the final score; they are just there to be tested, in order to value their difficulty [5].

From the results data, we can build an implication graph whose nodes are the questions. There exists an arrow from Q_1 to Q_2 iff more than a certain percent of people that answered correctly Q_1 answered Q_2 too (for example, $p \geqslant 95\%$). Thus, it is not interesting to ask Q_2 to someone who solved Q_1. Furthermore, if Q_1 and

Q_2 are mutually linked (this hardly ever happens in practice), either Q_1 or Q_2 can be removed from the bank.

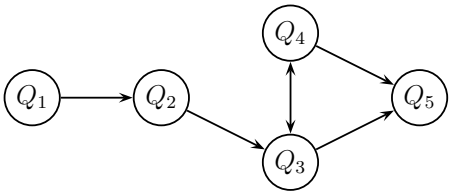

Figure 1: An implication graph. If the examinee answers correctly to Q_1, Q_2 must not be asked to him. Also, Q_3 or Q_4 can be definitely removed from the test.

As the examinee answers questions, there is a confidence interval (about the examinee's level) that decreases. A good stopping rule involves three parameters:

- The confidence interval becomes less than or equal to a certain threshold.
- The covering rate of the questions bank becomes greater than or equal to a certain threshold.
- A time limit is reached.

At the end of the test, scores are calculated for all examinees, then a final score is given to each of them, according to the mean level. In light of those results, the question bank is then recalibrated. Indeed, a test session brings information about the difficulty of the items.

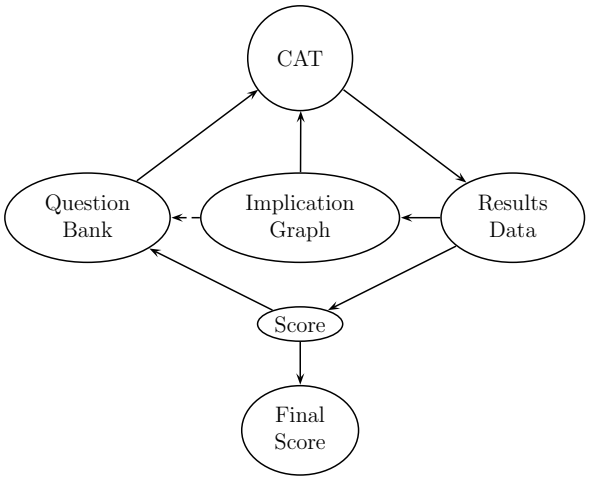

Figure 2: Le Test Édumétrique. The arrow between the implication graph and the question bank is dashed because it is very rare to find two questions mutually linked.

1.3 WIMS

WIMS (WWW Interactive Multipurpose Server) is a training software created in 1998 by Xiao Gang, which can value individually the behaviour of several pupils and offer various works according to their learning needs.

Teachers can create *virtual classes* that contains exercise sheets or exams. Their pupils then log in those classes to train or sit for an exam.

WIMS exercises can be coded in *OEF* (Open Exercise Format), which is noteworthy by its simplicity and its flexibility. The main point of interest is that you can add random parameters to an exercice.

```
\title{Addition}
\integer{a=random(-10..10)}
\integer{b=random(0..10)}
\statement{\a + \b = \embed{reply 1}?}
\answer{answer}{\a + \b}{type=numeric}
```

This code randomly asks: "$-2+7=?$", "$0+2=?$" and so on, with an input box before the question mark where the pupil can enter his answer. After submission, WIMS tells if it is correct or wrong.

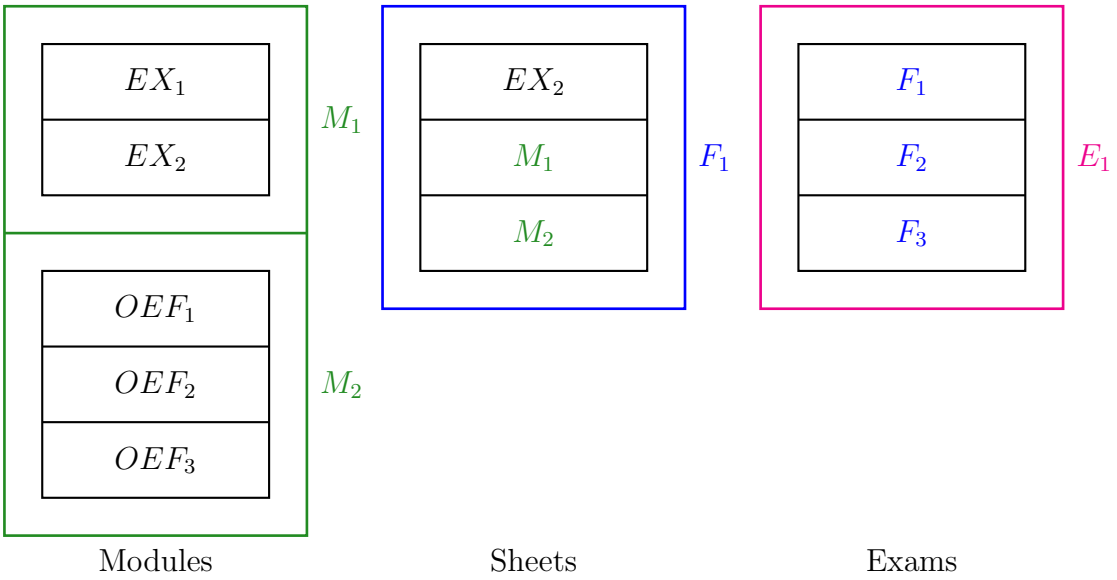

Figure 3: Different WIMS types. A module can contain exercises (notably OEF exercises), a sheet contains modules and exercises, an exam contains exercise sheets.

1.4 LTE's goal

LTE wanted to create a CAT system under WIMS, if possible, in order to test the core knowledge. They had three demands:

- WIMS headers and footers must not appear during the test (this was the easy part).
- Like in any test, when an examinee answers a question, the system must not tell him if he is right or wrong. This feature is currently available in WIMS exams, but not everywhere.
- Obviously, the CAT system. As a first draft, the question bank is supposed to have 9 levels of difficulty. The watered-down rules are as follows:
  - First, a question of level 5 is chosen.
  - If a question of level n is correctly answered, a question of level $n+1$ is then asked. If it is not, a question of level $n-1$ is then asked.
  - The test is stopped after 10 questions.

Let N_i (resp. n_i) be the number of questions of difficulty i asked to the examinee (resp. correctly answered by him). The score S is defined by:

$$S = \frac{\sum_{i=1}^{9} \frac{n_i}{N_i} i}{\sum_{i=1}^{9} \frac{n_i}{N_i}}$$

For example, according to the following diagram:

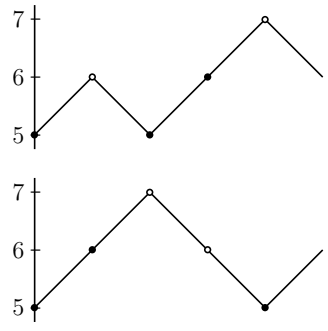

Figure 4: Two results from the test. Black dots mean the question was answered correctly, white ones mean it wasn't.

Both examinees get the score of:

$$S = \frac{\frac{2}{2}5 + \frac{1}{2}6 + \frac{0}{1}9}{\frac{2}{2} + \frac{1}{2} + \frac{0}{1}} \simeq 5.33...$$

LTE could buy a ready-made CAT system, but the most handy of them are extremely expensive and completely hermetic.

I was wondering why they wanted to make it under WIMS instead of coding it from scratch. It is because WIMS allows non-programmers to make exercises (in OEF, for example). All has already been written and works. Problems occur when you try to modify it.

2 The WIMS structure

The main folder of WIMS freshly installed contains more than 14,000 elements. Most of the time, you have to dig until the 4th level to find what you need, for example `public_html/modules/adm/sheet/` for the module of exercise sheets. There is no documentation, just a community of WIMS developers.

2.1 The core

WIMS was coded in C and CGI. Unfortunately, the code is very little commented and contains a lot of 2-letter variables, so it wasn't workable. It is full of goto instructions, I even found the following excerpt:

```
/* Should provide a method to stop infinite loop. */
void exec_goto(char *p)
{
  char lbuf[MAX_NAMELEN+17];
  char *label_start, *line_start;
  int old_line;
  int i;
  executed_gotos++;
  if(executed_gotos>=GOTO_LIMIT) module_error("too_many_gotos");
```

I asked why nobody had made a "WIMS 2" with a human-readable code. Someone answered that Xiao Gang, who is more a mathematician than an computer scientist, had thought about WIMS during a year to the smallest detail without any computer before he coded it, which explains why there are no bugs nor security flaws. Whatever works.

A former WIMS developer suggested that I should make the CAT system in a module.

2.2 Modtool

In the Modtool menu (that you can access if you have a developer account), modules are coded in WIMS, a language which seems to be a preprocessor for C [4]. Most commands are documented but not all of them.

I managed to hide the answer of the server from the examinee by injecting code that redirected violently him to another question after he had answered one. But there were two issues with Modtool:

- I didn't have access to all WIMS commands, notably the I/O ones (which is understandable, else anyone having a developer account could create modules and write files on the server without any limit).
- It only worked with OEF exercises (while there are other formats) and only the ones contained in the module, which is too restrictive.

As the CAT system had to work with any exercise sheet, I had to focus on the system module that handles them: adm/sheet.

2.3 adm/sheet

OEF exercises of the same field were gathered in 5 modules (Geometry, Measurement and Quantity, Numbers, Proportionality, Uncertainty) which were sent to me.

I grouped all the exercises together into one sheet.

In the variable processing code of each module, `var.proc`, I injected the following:

```
!if $cmd=reply and $status=waiting
  skip=1
!endif
```

When a question is answered, $skip is set to 1.

Then, in the display page of each module, `main.phtml`:

```
!if $skip=1
  !filewrite getfile/reponse :1
  <script>location.href='$wims_ref_name?session=$session&+lang=$lang
&+module=adm/sheet&+sh=$worksheet'</script>;
!else
  […]
!endif
```

Now, the code handling the sheet has to be modified.

When $skip is set to 1, ":1" is written into `sessions/[sessiondir]/getfile/reponse` (to indicate that the question has been answered) and the examinee is redirected to the sheet.

```
!let sessiondir=!text copy $session mask 111111111100
```

$session contains for instance "247A9A9DF3.1" while "247A9A9DF3" is the directory, so we only keep the first 10 characters.

```
!set data=!record 1 of wimshome/sessions/$sessiondir/getfile/
avancement
!set reponse=!record 1 of wimshome/sessions/$sessiondir/getfile/
reponse
!if $data=$empty
  !set data=1,6 5,7 4,8 2,9 3,10 1,6 5,7 4,8 2,9 5 0
  !filewrite getfile/avancement :$data
!endif
```

Data is read in the session directory. If the file "avancement" doesn't exist, it is created. The pattern of this file is as follows: l_1 l_2 ... l_9 d q where:

- For each i from 1 to 9, l_i is the list of the IDs of the questions of difficulty i.
- d is the current level of difficulty.
- q is the ID of the last question answered.

```
!for i=1 to 9
  !set idiff_$i=!word $i of $data
!next i
<br />
!set diff=!word 10 of $data
!set iancien=!word 11 of $data
!if $reponse=1
  !set nb=!itemcnt $(idiff_$diff)
  !reset liste
  !for i=1 to $nb
    !set itmp=!item $i of $(idiff_$diff)
    !if $itmp!=$iancien
      !set liste=!append item $itmp to $liste
    !endif
  !next i
```

$liste then contains $l_d \setminus q$: we remove the answered question from the list.

```
  !if $liste!=$empty
    !set idiff_$diff=$liste
  !else
    !set idiff_$diff=0
  !endif
```

We can't let an empty list, else it will shift the subscripts...

```
  !set score=!word $iancien of $scores
  !if $score=10
    !if $diff<9
      !set diff=!eval $diff+1
    !endif
  !else
    !if $diff>1
      !set diff=!eval $diff-1
    !endif
  !endif
!endif
```

The CAT rules.

```
!if $(idiff_$diff)=0
 On ne peut plus vous poser de question. Le test est termin&eacute;.
!else
  !set iactuel=!randitem $(idiff_$diff)
  !filewrite getfile/avancement :$idiff_1 $idiff_2 $idiff_3
$idiff_4 $idiff_5 $idiff_6 $idiff_7 $idiff_8 $idiff_9 $diff $iactuel
  !filewrite getfile/reponse :0
  !set ex=!record $iactuel of wimshome/log/classes/$wims_class/
sheets/.sheet$sh
  !distribute lines $ex into di,pa,require,we,ti,de
  !if $iancien=0
    !href module=$di&cmd=new&worksheet=$sh&$pa Cliquez ici pour
commencer le test. ($diff-$iactuel).
  !else
    Nous avons bien enregistr&eacute; votre r&eacute;ponse.
    !href module=$di&cmd=new&worksheet=$sh&$pa Cliquez ici pour
passer &agrave; la question suivante ($diff-$iactuel).
  !endif
!endif
```

If there is not any exercise of the difficulty requested anymore, the test ends. Else, the file "avancement" is updated and a link is provided towards the next question.

Possible improvements

- There might be a way to avoid the JavaScript redirection. I haven't studied the exam mode yet.
- As WIMS can communicate with other software, it may interact with a IRT (Item Response Theory) program so that the stopping rule will be given directly from the software to WIMS, as well as the examinee's score.

Conclusion

It is therefore possible to make a CAT system under WIMS, which is a brilliant tool that deserves more recognition. But WIMS was coded more than ten years ago, so there must be many new librairies that would simplify the code a lot, that is why I think a project should be started to program it again from scratch, instead of trying to understand how the current WIMS work.

References

[1] 20Q. http://www.20q.net.

[2] Akinator. http://www.akinator.com.

[3] Programme for International Student Assessment. http://en.wikipedia.org/wiki/Programme_for_International_Student_Assessment.

[4] WIMS Technical Documentation. http://wims.u-psud.fr/wims/wims.cgi?module=help/wimsdoc.

[5] Jean-Pierre Boudine, en collaboration avec Antoine Bodin. *Le Krach Éducatif*. L'Harmattan, 2010.