World Journal on Educational Technology: Current Issues

Vol 8, Issue 2, (2016) 119-131

http://sproc.org/ojs/index.php/wjet

Automated problem generation in learning management systems: a tutorial

Jaime Romero [*], Universidad Autónoma de Madrid – Facultad de Ciencias Económicas y Empresariales. Campus de Cantoblanco, Madrid 28049, Spain.

Mercedes Rozano, Universidad Autónoma de Madrid – Facultad de Ciencias Económicas y Empresariales. Campus de Cantoblanco, Madrid 28049, Spain.

Suggested Citation:

Romero, J. & Rozano, M. (2016). Automated problem generation in learning management systems: a tutorial. *World Journal on Educational Technology*: *Current Issues. 8*(2), 119-131.

Received May 07, 2016; revised June 18, 2016; accepted July 11, 2016;
Selection and peer review under responsibility of Assoc. Prof. Dr. Fezile Ozdamli, Near East University.

Abstract

The benefits of solving problems have been widely acknowledged by literature. Its implementation in e–learning platforms can make easier its management and the learning process itself. However, its implementation can also become a very time–consuming task, particularly when the number of problems to generate is high. In this tutorial we describe a methodology that we have developed aiming to alleviate the workload of producing a great deal of problems in Moodle for an undergraduate business course. This methodology follows a six-step process: exercise design, data generation, exercise computations, design of interpretation rules, wording generation and generation of cloze questions. It allows evaluating student's skills in problem solving, minimizes plagiarism behaviors and provides immediate feedback to students (thus improving their results). Additionally, it also reduces the workload of teachers in large groups and helps evaluating the student's learning more objectively. We have designed this methodology based upon our experience in Economics curricula, where we have applied it at undergraduate and graduate courses. However, we consider that it can be applied with minor modifications in a very wide range of college courses and e–learning platforms. We expect that that this tutorial encourage other educators and educational developers to apply our six steps process, thus benefiting themselves and their students of its advantages.

Keywords: Learning management systems, problem generation, randomization, cloze questions.

[*] ADDRESS FOR CORRESPONDENCE: **Jaime Romero,** Universidad Autónoma de Madrid – Facultad de Ciencias Económicas y Empresariales. Campus de Cantoblanco, Madrid 28049, Spain. *E-mail address*: jaime.romero@uam.es / Tel.: +34-91-497-3544

1. Introduction

Nowadays, it is common to use information technologies in our educational systems (Nagy, 2014; Gutiérrez, Trenas, Ramos, Corbera, & Romero, 2010; Hung, Fen, & Hwang, 2010). In the context of the European Union, the implementation of the European Higher Education Area encourages its usage, seeking to promote students working by their own through blended learning (Marchand & Gutiérrez, 2012; Soo & Bonk, 1998). The employment of web–based courses is increasingly common in different areas, complementing classroom sessions. In this context, Learning Management Systems (LMS) help to assess continuously the progress of students. They are assessed on the skills they acquire and not just for their knowledge (Cordeiro & Helfert, 2013; Dochy, Segers, & Sluijsmans, 1999; Hung et al., 2010; Malmi, Korhonen, & Saikkonen, 2002).

LMS are a powerful platform that supports a number of activities performed by both teachers and students (Gutiérrez et al., 2010; Despotović–Zrakić, Marković, Bogdanović, Barać, & Krčo, 2012). Web–based courses provide substantial advantages to students: convenience, flexibility, opportunities to increase their interactions with teachers and peers, etc. (Hung, Chou, Chen, & Own, 2010). For teachers the adoption of web–based courses allows for the design of a more complete portfolio of activities. It also permits improving feedback given to students, as well as facilitates the implementation of continuous assessment (Whitelock & Warburton, 2011; Verdú et al., 2011). Indeed, this continuous assessment is one of the main advantages of LMS. This tool allows, among others activities, the creation of a pool of questions that can be used to generate self-assessment quizzes, e.g. multiple choice, multiple answer, true/false, matching and ranking (O`Leary & Ramsden, 2002). Teachers need to learn all the possibilities offered by e–learning platforms in terms of students' assessment (González, Jover, Cobo, & Muñoz, 2010).

Traditionally, Science and Engineering subjects have more readily used LMS solutions in teaching (Malmi et al., 2002; Douce, Livingstone, & Orwell, 2005; Verdú et al, 2011), namely, automatic assessment. However, the spread of LMS in education communities extends its usage to other disciplines. For example, Economics and Social Sciences, which also focus on problems detection and solving (Nerguizian, Mhiri, & Saad, 2011), increasingly employ LMS to assess students' performance. Solving problems in these disciplines is an important learning activity (Lazakidou & Retails, 2010) and thus incorporated in web–based courses — students apply their knowledge toward developing solutions (Dennen, 2000). This teaching methodology is adequate for a wide variety of situations, depending of the wideness (micro vs. macro, i.e., providing less or more information to students) and specification level (open vs. closed, i.e., whether students can reach different solutions or not) of the problem' components (Hmelo–Silver, 2004). Solving problems through LMS alleviates one of its main drawbacks: it is a time–demanding activity, particularly in courses where the number of students is high, due to marking and feedback activities. LMS can aid teachers in these tasks. Nevertheless, solving problems when used under LMS can also provoke opportunistic behaviors in some students, such as plagiarism. To avoid such opportunistic behaviors teachers have no option but to generate a high number of problems, thus making plagiarism impossible (Butakov & Scherbinin, 2009). Unfortunately, problem generation at a large extent can be very time consuming. Orientations to perform this task do not exist in previous research. Thus, the main purpose of this tutorial is to present an approach that we use to perform this task in an undergraduate course of Business Administration, which employs Moodle as LMS. This methodology allows generating a great deal of problems in Moodle, while maintaining an adequate problem complexity through the incorporation of numeric, multiple choice and short answer questions based on the same dataset.

The structure of this paper is as follows. In the next section we describe our methodology for problem generation in Moodle. Subsequently we show an application of it to our course in the Degree in Business Administration of our university (Principles of Marketing). Finally, we present an assessment of our proposal and our conclusions.

2. Background

Educational research acknowledges the benefits of solving problems (Boud & Feletti, 1991; Savery & Duffy, 2001; Hmelo–Silver, 2004). Firstly, the learning process gives learners a central role. Students are actively engaged in the task and learn in the context in where knowledge is going to be applied. Secondly, solving problems constitutes a cognitive apprenticeship focused on a specific knowledge domain. Thirdly, teachers are not mere lecturers but adopt a facilitator or coach role.

2.1. Target course

The course “Principles of Marketing” is part of the Degree in Business Administration of our university. In 2010 we were in charge of incorporating Moodle in our teaching system. We had a mixed role of course designers and educators (we redesigned and taught this course; all the remaining educators of this course did not participate in its redesign). Traditionally, we used solving problems in this course, where our students had to solve a set of four problems at home during the course. Additionally, students had to solve one of these problems in an exam at the end of the course. Each problem involved some numerical analysis and the interpretation of results. Hence, we decided to integrate our solving problems routines in our new Moodle courses. The adoption of solving problems in e–learning platforms can dramatically reduce the time devoted to marking problems and giving feedback to students, thus facilitating its management and the learning process itself (Del Canto et al., 2011; Whitelock & Bill, 2011). LMS provide instructors with tools to track and evaluate students’ performance (Sancho, Moreno-Ger, Fuentes-Fernández, &Fernández-Manión, 2009).

Particularly, we followed a typical problems development process (Savery & Duffy, 2001):

1. Specification of learning goals: related to self–directed learning, content knowledge and problem solving. We clearly stated the skills that students would have to develop to pass.

2. Problem generation: we chose between micro vs. macro and open vs. closed problems. Micro problems clearly specify the multiple tasks that students will have to conduct for solving the problem. Macro problems provide students with a context in which several unspecified tasks must be performed to solve the problem. Open problems have more than one valid solution, whereas closed problems not. In our case we decided to use micro, closed problems. In general, once the educator decides what type of problem will be used, this must be set up within a realistic context, raising the main concepts and ideas from the content domain.

3. Problem presentation: incorporation of data in the problem.

Unfortunately, an inconvenience already reported by literature arose in our course: the online context made difficult to control whether students had really solved their problems by their own or committed plagiarism (Rosales et al., 2008; Butakov & Scherbini, 2009). Although the performance of students in problem solving during the course was apparently satisfactory, we found that the scores obtained by students in the same task in the final exam were lower than expected. When asked after the course, some groups of students reported that they had no incentive to work on their own on problem solution during the course. The reason for such behavior was that they all had to solve the same problem, thus reducing individual training for the final exam. To avoid this, we decided to provide students with individual problems. Nevertheless this is only a feasible solution for courses where the number of students is very low, which was not our case: around 250 students were enrolled in this course (distributed in six groups). Clearly, under these circumstances the problem development process stated above could not be performed directly without using a great deal of time. Thus, we looked for available procedures to generate problems, allowing us to take full advantage of automatic marking throughout the e–learning platform and avoiding the manual creation of a huge amount of problems for the course.

As a first solution we explored the possibility of using Moodle's simple calculated and calculated multichoice questions. These tools allowed creating a great deal of exercises through the employment of random parameters to generate data, which students had to use to provide either a numeric or a multiple choice answer through the application of a single formula. Nevertheless, we discarded these exercises because they would have been limited to one single question. As in many college courses, we required an array of questions around the same dataset in order to ensure an adequate level of complexity. For this purpose, Moodle users can employ cloze questions — arrays of questions that are presented all together. Unfortunately, cloze questions did not admit random parameters. Given this, we decided to develop our own procedure to include this randomization in cloze questions. We present it in the next section. Using this problem generation process we generated 2000 problems, 500 for each of the four different types of problems in our course.

3. Problem generation based on random parameters in cloze questions

Our procedure for problem generation has six steps. This process is implementable using standard office and teaching software. Although some basic notions of programming are advisable, teachers who are not skilled in programming can still fully implement the methodology, using the standard software package Microsoft Office, as we show afterwards.

Plainly stated, our generation of sets of problems consists in the creation of wordings, questions, and answers that Moodle can automatically correct. Particularly, the aim of this methodology is to generate a wide set of mathematical problems that involve several answers, to embed them in cloze questions — each student in the course must work on a different problem, that is, use different data. Cloze questions can include multiple choice, short answers, and numerical answers. These types of answers are usually enough in closed problem settings, as it was our case. Additionally, our methodology for problem generation aims to import the cloze questions to Moodle without effort.

Our methodology has the next steps:

1) Problem design: selecting the structure of the problem that student have to solve.

2) Data generation: creating datasets that fit the problem structure of the previous stage.

3) Problem computations: specifying the computations that students have to perform and execute them for the datasets of the previous stage.

4) Design of interpretation rules: specify the logic that students must apply to reach conclusions from their computations and use them to define the answers that students have to provide (numeric, multiple choice or short answers).

5) Wording generation (pdf or rtf format): incorporating the datasets of stage 2

6) Generation of cloze questions in XML format: create a file to be imported to Moodle, containing the set of problems and their answers.

We next explain these steps in detail:

Problem design. The first step to generate a set of problems is to set up problems' structure. To do so we recommend departing from a problem developed by traditional methods. Subsequently, we must identify which data in the problems is going to remain constant across students and which is going to change. For the latter, we need to decide how the problem variables change. For instance, we might need to decide if they must vary within a range or not, whether these variables are integer or have decimal part, if they vary according to a given type of distribution (Normal, Poisson, Uniform, etc), and so on.

Data generation. After deciding problem structure and describing variables' behaviors we must generate such variables. Some of them will be different for each problem in the set. To generate the data educators can use many software tools. Any standard statistical package (e.g., SPSS) will provide educators with several functions to generate data randomly. Worksheets such as MS–Excel or many open source software that incorporates routines for data generation are a good option as well. In any case, we recommend generating data in matrix form, in which each row contains all the variables (in columns) for a problem. This is helpful when manipulating data in subsequent steps.

Problem computations. To solve the problems, students have to make some calculations that are relatively normalized, i.e., they seldom vary across students. They have to make these computations using the data generated in the previous step. Some of them indeed constitute the answers to the problem or the information that the students have to evaluate in order to provide an answer. Educators have to perform these computations in advance using the data of step 2 and check whether the results that the students will reach are reasonable. In our opinion, the more disaggregated these computations are done the better. This provides teachers with intermediate results that are relevant when providing feedback to students (for instance, for detecting their mistakes).

Design of interpretation rules. In case of incorporating answers that arise from the interpretation of some calculations, educators must develop logical rules to produce answers that Moodle can interpret. Otherwise they can skip this stage. Applied to data these rules indicate what the right answer is. Multiple choice answers are a useful way to control the range of answers that Moodle evaluates.

Wording generation. The wording of problems includes all the information that the student should use to solve problems. Cloze questions admit a direct incorporation of the wordings. However, sometimes educators need to present the information in a particular format (for instance in tables or with graphics). In these situations they can distribute it in an external file (for example in pdf or rtf format). This file can be distributed in the cloze question through a hyperlink (Papasalouros, Kotis & Kanaris, 2011). In any case, standard rules for wording generation apply here.

Generation of cloze questions in XML format. When the amount of problems that are generated is high, importing questions to Moodle can represent a time saving. The XML format is a standard in data manipulation, which allows importing several types of questions to Moodle in a convenient way. An XML file containing cloze questions has the following structure: (i) introduction: general information about the questions that will be imported (course name, destination folder, etc.); (ii) cloze questions; and (iii) closing code: to indicate the end of the file.

Moodle help files, as well as many free access websites on the Internet, provide examples of XML code for importing cloze questions. Moodle users can employ these examples as a base for XML code generation. After importing the XML file in Moodle, the questions are available for inclusion in courses quizzes.

4. Example

In this section we present an example of our problem generation process. For brevity and clarity sake, we describe a portion of one of the problems that we generated for our course.

Problem design. In this example the problem that students must solve consists in the measurement of attitudes toward brands in a marketing environment. Brand attitudes arise after the evaluation of product attributes by consumers. Consumers consider that some product attributes are more important than others are. This point is key when analysing brand attitudes. Typically, students receive information about product attributes as shown in figure 1. The students must measure brand attitudes and indicate which brand is preferred by customers (questions 1 to 3). They have to calculate brand attitudes through a weighted sum of brand attributes as presented in the following equation:

$$A_i = \sum_{k=1}^{K} w_k \times b_{ik}$$

where A_i is the attitude toward brand i, w_k is the importance of attribute k and b_{ik} is the evaluation of attribute k in brand i. The brand with the higher score (A_i) is the most preferred one. In this problem, we set that information about all brand attributes and its importance vary across problems. Particularly, they vary in a range of [1–10] following a uniform distribution.

Data generation. For our problem, data generation just requires using any software capable of generating random numbers from a uniform distribution within a range. Particularly we use MS–Excel, as shown in figure 2. We generate data for 500 problems.

Problem computations and design of interpretation rules. In this problem students must calculate weighted sums of brand attributes using Equation 1 in order to measure brand attitudes and thus answering questions 1 and 2. Afterwards they must apply a logical rule (“the brand with the higher score is the most preferred one”) in order to answer question 3. These computations and the application of the logical rule must be performed for each of the 500 problems generated in the previous step. In figure 3 we show how we make the computations and implement the interpretation rule in our Excel dataset.

Wording generation. In case of using MS–Excel for data generation, MS–Word’s mail merge feature facilitates creating the wording of the problems that students solve. Mail merge allows the incorporation of data from an external source (for instance a worksheet) into a document. Instructions and tutorials to use successfully this feature are available on the Internet. We conduct wording generation for this problem using MS–Word, though other software can serve the same function. To do so we create a Word document containing the wording.

The variables in this Word document (the ones that we want to change across problems) come from our Excel worksheet (figure 4). Subsequently, we complete the merge between the wording and the data. The result is a new document containing 500 problems (figure 5). This document can be distributed as is to students or split in as many documents as problems. In this example we assume that we distribute it in independent documents. For these cases, we suggest naming these files as the combination of a text string and the problem number. For instance, in this example about brands attitudes, we set file names as “BA” plus the problem number, that is, BA1, BA2,…, and BA500.

Generation of cloze questions in XML format. The last step of problem generation is the creation of a XML file that contains all the problems generated in previous stages. As stated above, this XML file has three parts. The first one includes code that specifies in which course and destination folder (if any) the problems will be located. The second one consists of several lines that define the answers in the cloze question: answer type, answer name, external links if necessary (useful when the wording file has been split in one-page documents), the answers themselves, whether the answers must be shuffled, etc. The third part includes lines to indicate the end of the file. Thus, the first and the last parts of the XML file do not depend on the dataset generated in previous step, whereas the second part takes into account the calculations implemented in previous stages. Consequently, we can create the first and last parts as a standard text files and generate the second one, again, using Word's mail merge or a similar tool.

The XML code can incorporate directly numeric and short answers. Unfortunately, this does not happen for multiple choice answers. In the latter type, we need to code the options as "Option 1~ Option 2~ Option 3~ Option 4", putting a "=" character before the text of the right option (this might vary across Moodle versions). We can perform this task using logical rules in the software employed for data generation and manipulation. In figure 6 (column O) we show how we do it in Excel, together with the names of the files containing the wording of each problem (column P).

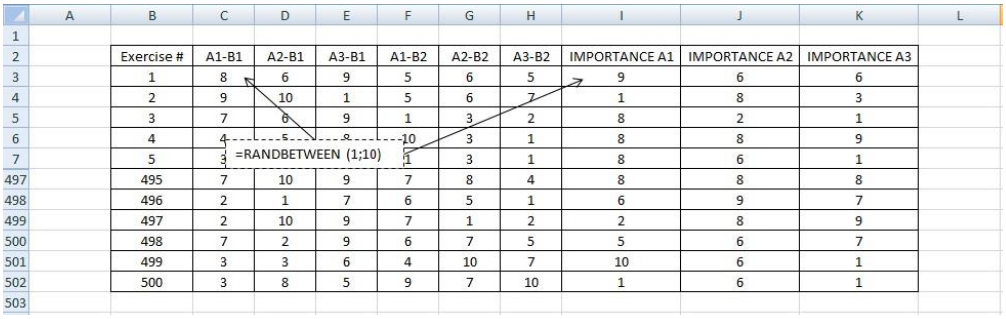
In order to evaluate brands in category X, consumer usually consider attributes A1, A2, and A3. Company B1 recently conducted a survey asking consumers to evaluate these attributes in its brand and its main competitor (company B2), as well as the importance of each attribute when they make a purchase in category X. The scale for evaluating attributes as well as their importance ranges from 1 to 10. The survey results are:

	Brand B1	Brand B2	Importance
Attribute A1	5	7	4
Attribute A2	8	4	5
Attribute A3	2	1	2

Questions:

1) Which is the attitude toward brand B1?
2) Which is the attitude toward brand B2?
3) Which brand is the most preferred by customers?

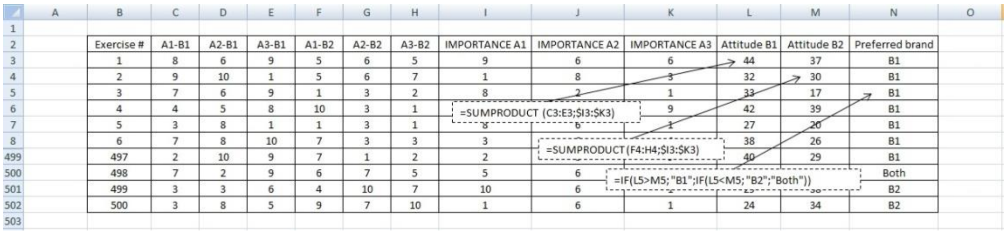
Figure 1. Problem structure



	A	B	C	D	E	F	G	H	I	J	K	L
1												
2		Exercise #	A1-B1	A2-B1	A3-B1	A1-B2	A2-B2	A3-B2	IMPORTANCE A1	IMPORTANCE A2	IMPORTANCE A3	
3		1	8	6	9	5	6	5	9	6	6	
4		2	9	10	1	5	6	7	1	8	3	
5		3	7	6	9	1	3	2	8	2	1	
6		4	4	5	8	10	3	1	8	8	9	
7		5	3	[illegible]	[illegible]	1	3	1	8	6	1	
497		495	7	10	9	7	8	4	8	8	8	
498		496	2	1	7	6	5	1	6	9	7	
499		497	2	10	9	7	1	2	2	8	9	
500		498	7	2	9	6	7	5	5	6	7	
501		499	3	3	6	4	10	7	10	6	1	
502		500	3	8	5	9	7	10	1	6	1	
503												

=RANDBETWEEN (1;10)

Figure 2. Data generation



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2		Exercise #	A1-B1	A2-B1	A3-B1	A1-B2	A2-B2	A3-B2	IMPORTANCE A1	IMPORTANCE A2	IMPORTANCE A3	Attitude B1	Attitude B2	Preferred brand	
3		1	8	6	9	5	6	5	9	6	6	44	37	B1	
4		2	9	10	1	5	6	7	1	8	3	32	30	B1	
5		3	7	6	9	1	3	2	8	2	1	33	17	B1	
6		4	4	5	8	10	3	1	[illegible]	[illegible]	9	42	39	B1	
7		5	3	8	1	1	3	1	8	6	1	27	20	B1	
8		6	7	8	10	7	3	3	3	[illegible]	[illegible]	38	26	B1	
499		497	2	10	9	7	1	2	2	[illegible]	[illegible]	40	29	B1	
500		498	7	2	9	6	7	5	5	6	[illegible]	[illegible]	[illegible]	Both	
501		499	3	3	6	4	10	7	10	6	[illegible]	[illegible]	[illegible]	B2	
502		500	3	8	5	9	7	10	1	6	1	24	34	B2	
503															

=SUMPRODUCT (C3:E3;$I3:$K3)

=SUMPRODUCT(F4:H4;$I3:$K3)

=IF(L5>M5;"B1";IF(L5<M5;"B2";"Both"))

Figure 3. Computations and interpretation rules

PROBLEM: BRAND ATTITUDES

In order to evaluate brands in category X, consumer usually consider attributes A1, A2, and B3. Company B1 recently conducted a survey asking consumers to evaluate these attributes in its brand and its main competitor, as well as the importance of each attribute when they make their purchase in category X. The scale for evaluating attributes as well as their importance ranges from 1 to 10. The survey results are:

	Brand B1	Brand B2	Importance
Attribute A1	«A1B1»	«A1B2»	«IMPORTANCE_A1»
Attribute A2	«A2B1»	«A2B2»	«IMPORTANCE_A2»
Attribute A3	«A3B1»	«A3B2»	«IMPORTANCE_A3»

Questions:

1) Which is the attitude toward brand B1?
2) Which is the attitude toward brand B2?
3) Which brand is the most preferred by customers?

Figure 4. Wording generation

PROBLEM: BRAND ATTITUDES

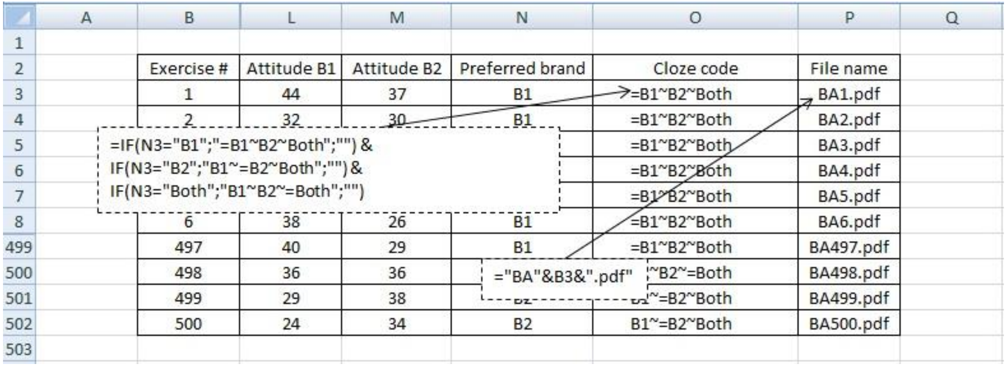
In order to evaluate brands in category X, consumer usually consider attributes A1, A2, and B3. Company B1 recently conducted a survey asking consumers to evaluate these attributes in its brand and its main competitor, as well as the importance of each attribute when they make their purchase in category X. The scale for evaluating attributes as well as their importance ranges from 1 to 10. The survey results are:

	Brand B1	Brand B2	Importance
Attribute A1	8	5	9
Attribute A2	6	6	6
Attribute A3	9	5	6

Questions:

1) Which is the attitude toward brand B1?
2) Which is the attitude toward brand B2?
3) Which brand is the most preferred by customers?

Figure 5. Wording for problem 1

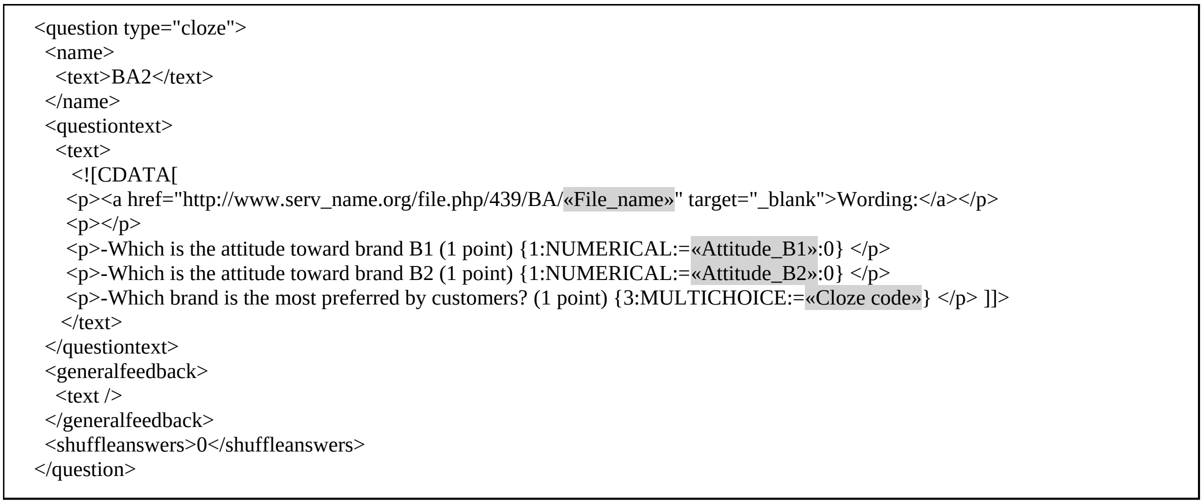


	A	B	L	M	N	O	P	Q
1								
2		Exercise #	Attitude B1	Attitude B2	Preferred brand	Cloze code	File name	
3		1	44	37	B1	=B1~B2~Both	BA1.pdf	
4		2	32	30	B1	=B1~B2~Both	BA2.pdf	
5						=B1~B2~Both	BA3.pdf	
6						=B1~B2~Both	BA4.pdf	
7						=B1~B2~Both	BA5.pdf	
8		6	38	26	B1	=B1~B2~Both	BA6.pdf	
499		497	40	29	B1	=B1~B2~Both	BA497.pdf	
500		498	36	36		~B2~=Both	BA498.pdf	
501		499	29	38		~=B2~Both	BA499.pdf	
502		500	24	34	B2	B1~=B2~Both	BA500.pdf	
503								

=IF(N3="B1";"=B1~B2~Both";"") &
IF(N3="B2";"B1~=B2~Both";"") &
IF(N3="Both";"B1~B2~=Both";"")

="BA"&B3&".pdf"

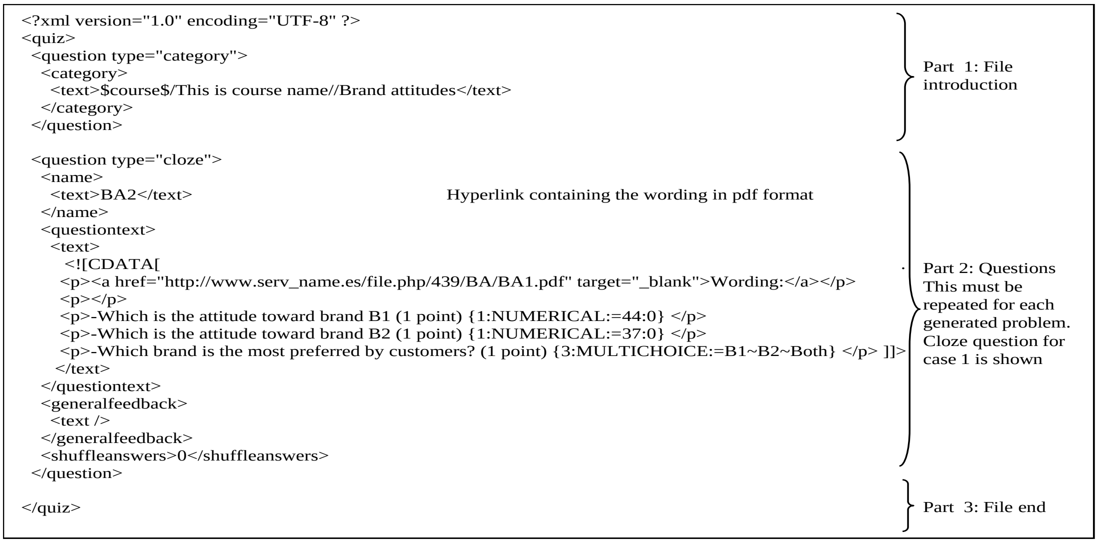
Figure 6. Multiple choice code and file name



```
<question type="cloze">
 <name>
  <text>BA2</text>
 </name>
 <questiontext>
  <text>
    <![CDATA[
   <p><a href="http://www.serv_name.org/file.php/439/BA/«File_name»" target="_blank">Wording:</a></p>
   <p></p>
   <p>-Which is the attitude toward brand B1 (1 point) {1:NUMERICAL:=«Attitude_B1»:0} </p>
   <p>-Which is the attitude toward brand B2 (1 point) {1:NUMERICAL:=«Attitude_B2»:0} </p>
   <p>-Which brand is the most preferred by customers? (1 point) {3:MULTICHOICE:=«Cloze code»} </p> ]]>
  </text>
 </questiontext>
 <generalfeedback>
  <text />
 </generalfeedback>
 <shuffleanswers>0</shuffleanswers>
</question>
```

Figure 7. XML generation for cloze questions

The answers for the problem questions and the file name containing the wording (or alternatively the problem wording itself) is all the information that we need to build part 2 of the XML file using Word's mail merge (figure 7). Once that we create it, we combine it with parts 1 and 3 in a single file — just by copying and pasting the three parts in a text file. Note that the resulting file starts with the information about the course (part 1 in figure 8). Next, we include the XML code for all the questions that have been generated (part 2 in figure 8). Finally, we incorporate the line to indicate the end of the file (part 3 in figure 8). We import this file to Moodle following standard procedures (to do this, users must open the questions module in the e–learning platform, go to "Import questions from file", specify that the file format is "Moodle XML format" and upload the file). Once we perform this task, the questions are available in course site. Users just need to create a quiz in Moodle and include one of these questions (randomly selected). In figure 9 we provide a preview of the cloze questions that we generate for the problem we are presenting here. By clicking in the hyperlink "Wording" students obtain a pdf file containing the problem wording.



```
<?xml version="1.0" encoding="UTF-8" ?>
<quiz>
 <question type="category">
  <category>
   <text>$course$/This is course name//Brand attitudes</text>
  </category>
 </question>

 <question type="cloze">
  <name>
   <text>BA2</text>
  </name>
  <questiontext>
   <text>
     <![CDATA[
    <p><a href="http://www.serv_name.es/file.php/439/BA/BA1.pdf" target="_blank">Wording:</a></p>
    <p></p>
    <p>-Which is the attitude toward brand B1 (1 point) {1:NUMERICAL:=44:0} </p>
    <p>-Which is the attitude toward brand B2 (1 point) {1:NUMERICAL:=37:0} </p>
    <p>-Which brand is the most preferred by customers? (1 point) {3:MULTICHOICE:=B1~B2~Both} </p> ]]>
   </text>
  </questiontext>
  <generalfeedback>
   <text />
  </generalfeedback>
  <shuffleanswers>0</shuffleanswers>
 </question>

</quiz>
```

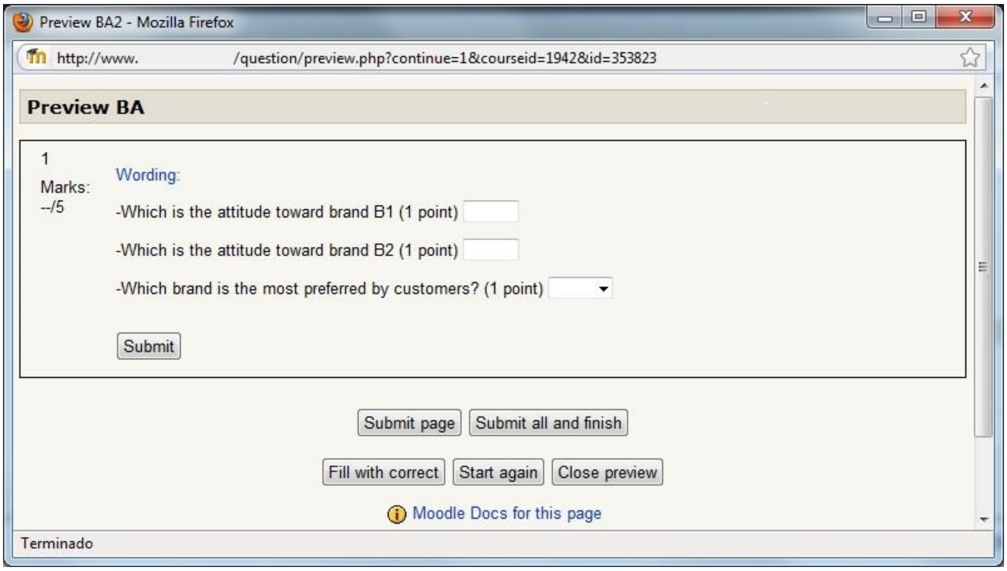
Hyperlink containing the wording in pdf format

Part 1: File introduction

Part 2: Questions This must be repeated for each generated problem. Cloze question for case 1 is shown

Part 3: File end

Figure 8. XML file


Preview BA2 - Mozilla Firefox

http://www. /question/preview.php?continue=1&courseid=1942&id=353823

Preview BA

1
Marks:
--/5

Wording:

-Which is the attitude toward brand B1 (1 point)

-Which is the attitude toward brand B2 (1 point)

-Which brand is the most preferred by customers? (1 point)

Figure 9. Cloze question

5. Assessment and further implementations

As explained before, we applied this problem generation procedure in a Business Administration course. In this section we present an assessment of this methodology. This evaluation consists of two parts. Firstly, we have collected the opinions of the educators that have used it (but did not participate in its development) after its implementation (May-June 2012). They indicate as its main advantages:

- The possibility of really assigning a different problem to each student. Once that the methodology is implemented it is easy to create an unlimited number of problems with the same structure.
- The chance of easily incorporating modifications in the problem once that the structure has been created
- The availability of intermediate calculations that facilitate providing feedback to students in case they request it.

Secondly, we have compared students' ability to solve problems during the exam that takes place at the end of the course. We have collected information about alumni performance in this exam for two groups of students: the first one includes students who took the course before the implementation of individual problems (before academic year 2010-2011). The second one is formed by alumni who had to solve individual problems during the course (academic year 2011-2012). The final exam not only includes the problem (25% of the total score) but also some theoretical questions (75%). Our experience reveals that some students might decide not preparing one of the parts of the exam and leaving it in blank. To ensure comparability between groups, we have selected students who have provided some answer in each part.

Table 1 shows students' scores during the final exam. We have found that on average the scores based on theoretical questions are lower after incorporating individual problems, whereas the problem scores are higher (7.97 vs. 8.50). The distribution of scores in the groups is relatively similar (figure 10), with a slight predominance of higher scores in the group that solved individual problems. We have checked whether there are significant differences between these scores of the two groups (alternative hypothesis: the problem scores with individual problems are higher than without them). Our results indicate that the students that had to solve individual problems during the course obtained higher scores for this task in their final exams (t=--1.81, p-value=0.0367). The increase in students' scores is statistically significant at a 95% level.

Table 1. Group scores in the final exam

		Minimum	1st Quartile	Median	3rd Quartile	Maximum	Mean	Standard error
Before individual problems (N=204)	Problem	2.50	6.50	8.50	10.0	10.0	7.97	1.98
	Theory	2.33	5.00	6.33	7.80	10.00	6.30	2.12
	Total	2.62	5.48	6.89	8.00	9.88	6.72	1.70
After individual problems (N=66)	Problem	3.00	6.60	10.0	10.0	10.0	8.50	1.79
	Theory	2.08	3.48	4.60	6.30	10.0	4.92	2.07
	Total	2.98	4.93	5.76	6.82	9.75	5.82	1.36

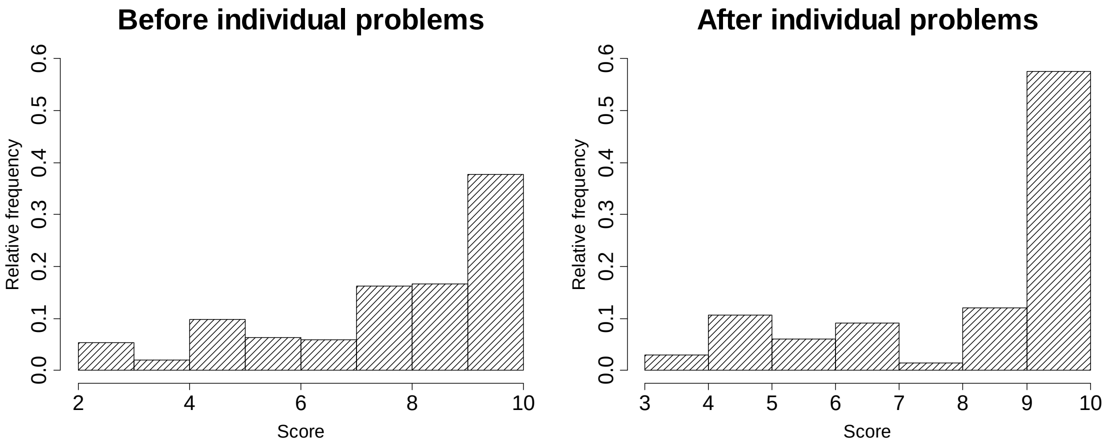


Figure 10. Problem scores

Extensions. Apart from applying it in an undergraduate course, the methodology has been implemented successfully in a Master course taught by one of the authors (Marketing modelling; academic year 2012-2013). In this course, numerical and multiple choice questions were combined with open questions in a business case (40 students). The numerical questions involved mathematical optimizations and the multiple choice questions involved the interpretation of their results. Each business case had its own dataset. Although this added some complexity in terms of data generation, our methodology is flexible enough to accommodate this. Unfortunately, we lack comparable data about student performance prior to and after the incorporation of our methodology in this master course. Nevertheless, this latter successful implementation suggests that our approach for problem generation is fully applicable to a wide variety of learning contexts. More specifically, our methodology can be applied to micro problems where some answers are open.

6. Conclusions

The usage of e–learning platforms in institutions of higher education is increasingly more popular during the last years (Nagy, 2014; Marchand & Gutiérrez, 2012), providing many benefits for both students and teachers. Nevertheless, its adoption by teachers can be quite challenging because it might require an adaptation in terms of implementing some teaching methodologies. In the case of solving problems, employing LMS generally requires increasing the number of problems distributed to students. When the number of students in a course is high, performing this task can be very time consuming if done by standard procedures. Similarly, renewing these problems in the future can be also quite cumbersome.

In this manuscript we describe a methodology that we develop to generate a high number of problems in LMS. This methodology is adapted to the open source e–learning platform Moodle. Educators can apply it employing standard software, such as Microsoft Office — thus facilitating its adoption among interested users. It allows for the generation of any number of wordings, thus covering as many scenarios as adequate for learning objectives. Particularly we use cloze questions, which permit presenting sets of questions to students and automatically correct their answers. Students can immediately receive feedback about their answers, without a heavy marking workload for teachers. More importantly, plagiarism opportunities disappear.

We originally develop this methodology for an undergraduate Business Administration course, where practicing with business situations that involve mathematical and statistical computations is very common. We find that applying this methodology in this course leads to a significant improvement in student scores. Although very preliminary, these results suggest that our approach for problem generation might force students to avoid opportunistic behaviors and therefore to reach a better training for their final exams.

Additionally, we successfully validate our methodology with more advanced students (graduate), thus showing its flexibility in terms of adaptation to various complexity levels. Our approach is also enough versatile to accommodate other content areas, as long as they use closed or semi-closed problems. We consider that extra research efforts are needed to study how solving problems in LMS can be extended to more open problems without losing the advantages of employing LMS. We hope that this tutorial will encourage not only educators and educational developers to apply our proposal in their courses but also to perform such future research.

Acknowledgements

Project ECO2015-69103-R. Ministerio de Economía y Competitividad.

References

Boud, D., & Feletti, G. (1991). *The challenge of problem based learning*. New York, NY: St. Martin's Press.

Butakov, S., & Scherbinin, V. (2009). The toolbox for local and global plagiarism detection. *Computer and Education, 52*(4), 781–788.

Cordeiro, J., & Helfert, M. (2013). Innovative learning techniques and educational technologies. *Education and Information Technologies, 18*(2), 271-274.

Del Canto, P., Gallego, M. I., Santamaría, E., López, J. M., Medina, E., Mochón, F.,…Valero, M. (2011). Follow–up and feedback processes in the EHEA. *Journal of Technology and Science Education, 1*(1), 12-23.

Dennen, V.P. (2000). Task Structuring for On–line Problem Based Learning: A Case Study. *Educational Technology & Society, 3*(3), 329–336.

Despotović–Zrakić, M., Marković, A., Bogdanović, Z., Barać, D., & Krčo, S. (2012). Providing Adaptivity in Moodle LMS Courses. *Educational Technology & Society, 15*(1), 326-338.

Dochy, F., Segers, M., & Sluijsmans, D. (1999). The use of self-, peer and Co-assessment in higher education: a review. *Studies in Higher Education, 24*(3), 331-350.

Douce, C., Livingstone, D.,& Orwell, J. (2005). Automatic Test-based assessment of programming: a review. *Journal of Educational Resources in Computing, 15*(3), 1-13.

González, J.A., Jover, L., Cobo, E. & Muñoz, P. (2010). A web–based learning tool improves student performance in statistics. A randomized masked trial. *Computers & Education, 55*(2), 704-713.

Gutiérrez, E., Trenas, M.A., Ramos, J., Corbera, F., & Romero, S. (2010). A New Moodle module supporting automatic verification of VHDL-based assignments. *Computers & Education, 54*, 562-577.

Hmelo–Silver, C. E. (2004). Problem–based learning: What and how do students learn? *Educational Psychology Review, 16*(3), 235–266.

Hung, P., Fen, Y., & Hwang, G. (2010). Formative assessment design for PDA integrated ecology observation. *Educational Technology & Society, 13*(3), 33-42.

Hung, M., Chou, C., Chen, C. & Own, Z. (2010). Learner readiness for online learning: scale development and student perceptions. *Computers & Education, 55*(3), 1080-1090.

Lazakidou, G. & Retails, S. (2010). Using computer supported collaborative learning strategies for helping students acquire self–regulated problem–solving skills in mathematics. *Computers & Education, 54*(1), 3-13.

Malmi, L., Korhonen, A., & Saikkonen, R. (2002). Experiences in Automatic Assessment on Mass courses and Issues for designing virtual courses. In *Proceedings of The 7th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, ITiCSE'02 Aarhus, Denmark.24-26 June 2002* (pp.55-59). Retrieved from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi =10.1.1.205.6679.

Marchand, G.C., & Gutiérrez, A.P. (2012). The role of emotion in the learning process: comparisons between online and face to face learning settings. *Internet and Higher Education, 15*, 150-160.

Nagy, J. T. (2014). Using learning management systems in business and economics studies in Hungarian higher education. *Education and Information Technologies,* 1-21. doi: 10.1007/s10639-014-9360-6.

Nerguizian, V., Mhiri, R.,& Saad, M. (2011). Interactive e-Learning approach for e-Business. *International Journal of e-Business Management, 5*(1), 48-60.

Rosales, F, García, A., Rodríguez, S., Pedraza, J.L., Méndez, R., & Nieto, M.M. (2008). Detection of plagiarism in programming assignments. *IEEE Transaction on Education, 51*(2), 174-183.

O'Leary, R., & Ramsden, A. (2002). Virtual learning environments. In P. Davies (Ed.), *The Handbook for Economics Lecturers: Teaching*, Bristol: Economics LTSN. Retrieved from: http://www.economicsnetwork.ac.uk/handbook/printable/vle_v5.pdf

Papasalouros, A., Kotis, K., & Kanaris, K. (2011). Automatic generation of tests from domain and multimedia ontologies. *Interactive Learning Environments, 19*(1), 5-23.

Sancho, P., Moreno–Ger, P., Fuentes–Fernández, R., & Fernández–Manjón, B. (2009). Adaptive Role Playing Games: An Immersive Approach for Problem Based Learning. *Educational Technology & Society, 12*(4), 110–124.

Savery, J.R., & Duffy, T.M. (2001). Problem based learning: An instructional model and its constructivist framework. *CRLT Technical Report. Indiana University, 16–01*, 1–19.

Soo, K., & Bonk, C.J. (1998). Interaction: What does it mean in online distance education? In: *ED-MEDIA/ED-TELECOM 98 World Conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecommunications. Proceedings 10th, Freigburg, Germany, 20-25 June 1998* (pp. 1365-1372). Charlottesville, VA: Association for the Advancement of Computing in Education (AACE).

Verdú, E., Regueras, L.M., Verdú, M.J., Leal, J.P., De Castro, J.P., & Queirós, R. (2011). A distributed system for learning programming on-line. *Computers & Education, 58*(1), 1-10.

Whitelock, D., & Bill, W. (2011). Preface CAA 2010: Computer assisted assessment: supporting student learning. *Journal of e–Assessment, 1*(1), 1-16.