International Journal of Engineering Education Vol. 27, No. 1, pp. 61–69, 2011
Printed in Great Britain
0949-149X/91 $3.00+0.00

A Web-Based E-Learning Tool for Database Design Courses*

JOSEP SOLER,[1] IMMA BOADA,[1] FERRAN PRADOS,[1] JORDI POCH[1] and RAMON FABREGAT[2]
[1] Department of Computer Science and Applied Mathematics. University of Girona.
E-mail: josep.soler, imma.boada, ferran.prados, jordi.poch@udg.edu
[2] Institute of Informatics and Applications. University of Girona. E-mail: ramon.fabregat@udg.edu

ACME-DB is a web-based e-learning platform designed for giving support to the teaching and learning of main database course topics. The platform is composed of a set of correction modules capable of automatically correcting different types of database problems such as entity-relationship diagrams, relational database schemas, normalization, relational algebra and SQL. The capabilities of ACME-DB provide us with a powerful tool for skills training and automatic assessment of main database course topics. In this paper we describe how ACME-DB has been used in different database courses at our university and how it has influenced academic results in a positive manner.

Keywords: database design; e-learning; web-based tools

1. Introduction

All the computing disciplines specified by the ACM/IEEE/AIS curricula [1] provide the general guidelines of a database course. However, how to meet these guidelines is not addressed. In the last few years, different studies have been carried out to analyze the contents of a database course [2–4]. The majority of authors agree that the main focus of the course has to be database design steps and how to develop a database, i.e. database design, relational model and Structured Query Language (SQL). These topics are part of the core of computing disciplines and, hence, are extensively covered in leading database textbooks [5–8]. Therefore, how to take students through all the steps of the database development process is an important issue to be taken into account when defining the teaching methodology. In general, teachers agree that on the one hand students have to receive a solid background in concepts, which can be introduced in theory classes. On the other hand, they have to put these concepts into practice by developing databases. Such skills can be acquired in practical sessions. In this context, e-learning platforms have become fundamental and a valuable tool to complement and support teaching (*blended learning*).

Currently, there are different e-learning tools able to cover the majority of database course contents. However, it is difficult to find a unique environment able to support all of them. Moreover, the majority of platforms that give advice during the solving process are not capable of automatically correcting the exercises. To overcome these limitations, in 2003 we started to develop the ACME-DB environment, which is part of the ACME platform [9] (the acronym ACME stands for the Catalan for 'Continuous Assessment and Improvement of Skills'). ACME-DB is a web-based e-learning tool designed for skills training and automatic assessment of main database course topics with the aim of improving database design learning process. Its main feature is the capability to automatically correct database exercises. It is composed of a set of correction modules, each one designed for a specific type of problem. Amongst them, entity-relationship diagrams, relational database schemas, normalization, relational algebra and SQL. Moreover, it provides support to different teaching requirements such as continuous assessment or student tracking. In the 2004/2005 course, we started to use it at our university as a complement to traditional classroom database courses in Computer Science degrees. We obtained very promising results and since then the number of activities carried out with the platform has been increased leading to an improvement in academic results.

Our aim is to investigate if the application of ACME-DB improves the learning process and allows us to obtain better academic results. With this purpose in mind, in this paper we describe the experience of applying ACME-DB and how it has influenced academic results. The paper is structured as follows. In Section 2, related work is presented. In Section 3, we present the ACME-DB environment. In Section 4, we describe how ACME-DB has been used in teaching/learning of database design and we present the obtained results. Finally, in Section 5, conclusions and future work are given.

* Accepted 2 September 2010.

2. Related work

To put our work in context, in this section we give a quick overview of database design contents and a description of the teaching methodology applied in our university. Afterwards, we briefly review some of the e-learning tools described in the literature to support teaching and learning contents in database design.

2.1 Teaching database design

Database design is one of the main topics of any introductory database course. The creation of a database requires a design process to define types, structures and constraints for the data to be stored in the computer. This process can be summarized in four main steps:

- First, there is an analysis process to identify the requirements that have to be satisfied to represent data in a real world situation.
- Second, a *conceptual schema has to be designed* to give a high level description of the database and the requirements that data must satisfy. The most popular approach for conceptual schema designing is the Entity-Relationship (ER) model [10]. This model considers the world as a set of entities and the relationships between them.
- Third, the definition of a *logical design*, which gives a high level schema implementable on a database management system. The relational model [11] is the most common data model used for database management. It represents the database as a collection of relations where each relation resembles a table of values. To create relations with no redundant data, with an efficient data organization that can be modified in a consistent and correct manner, a *normalization* process is applied.
- Fourth, the definition of a *physical design* which represents the internal data storage details.

When all these concepts are acquired, we can consider that students have the basic knowledge required to implement a relational database schema using any one of the available Database Management System (DBMS).

2.2 Database design in our University

The methodology applied in the database courses at our university is based on the fact that theory is as important as practice. Therefore, the course is composed of lectures and practice sessions. In lectures the teacher introduces main theoretical concepts. In laboratory sessions, exercises are proposed to the students to put in practice theoretical concepts. The goal is to take student trough the different steps of the design process considering real situations. In the first phase, the student solves exercises of low complexity, where conceptual models require a small number of entities. The complexity of the exercises is increased gradually. In the second phase, logical design exercises are proposed. The student transforms a conceptual model into a logical model targeted to a relational database implementation. Students learn to map a conceptual model into a logical model that can in turn be readily transformed into a relational database schema. At this point, the central concepts of the relational model (domains, keys, integrity constraints, and so on) are engaged. Finally, the database schema is created. In this step, the student defines the tables, with the corresponding attributes, primary and foreign keys, etc.

The methodology on how to acquire database design skills in the practical sessions was very valuable since students solved exercises similar to real-world problems. However, from the teacher's perspective it was very difficult to carry out. The main limiting factor was the large amount of exercises that have to be corrected. It has to be taken into account that the solution of a problem is not unique. To tackle this problem instead of assigning individual exercises to the student we provided them with a list with all the exercises of the course and we solved some of them in the class. We also assigned at each student two of them as homework and the teacher corrected these. The main drawback of this strategy was that at the end of the course the students only had solved the two assigned problems.

Due to the importance of practice in the context of these courses and considering the requirements of the new European Higher Education Area, we decided to integrate e-learning tools in the applied methodology in order to overcome the detected limitations. Since, we had experience using e-learning platforms; we decided to extend our e-learning platform ACME to support the automatic correction of ER diagrams, relational database schemas and normalization problems. With this extension we will have in the same environment all the functionalities to support the course management, and also the functionalities required to cover the database design needs. This new environment, denoted ACME-DB, is described in Section 3.

2.3 Database design tools

In last few years a large number of e-learning environments that provide and support different teaching and learning topics of database courses have been proposed. In this section, we describe the most representative. To present them we consider three different groups according to the main topics of the database design process: conceptual and logical design and normalization.

ER diagram Correction Tools: As previously mentioned, the ER model is the most popular approach for conceptual schema designing. Such a model considers the world as a set of entities and the relationships between them. Therefore, an ER diagram tool has to provide an environment able to support the definition and correction of ER diagrams. The system has to be able to interpret the diagram giving the corresponding feedback to the student.

Over the last decade several attempts to develop ER problem solving web environments have been made [12–15]. The most representative is KERMIT proposed by [14]. KERMIT is an intelligent tutoring system that contains a set of problems and ideal solutions to them. The system compares the student solutions with the ideal one using domain knowledge represented in the form of constraints, which are classified into syntactic and semantic ones.

Relational Database Schemas Correction Tools: The relational model represents the database as a collection of relations. To define this model the student has to define a set of tables for a given situation of the real world. Therefore, the tool has to provide an environment able to support the definition of tables, with the corresponding attributes, keys and foreign keys and also, the capability to correct and give feedback about correction. It has to be taken into account that the solution of a relational design problem is not unique. Few tools to support and correct logical database design have been proposed. The most representative is ERM-Tutor [16], a constraint based tutor that teaches logical database design (i.e. mapping conceptual to logical database schemas). Students solve the problem step by step and receive feedback on their solutions.

Normalization Tools: The normalization process is applied during the relational model definition to create relations with no redundant data and with an efficient data organization. To teach the normalization process a set of functional, multivalued and join dependencies are given to the student and he has to obtain the corresponding normalized schema. Normalization tools display a relation with the corresponding dependencies, the student proposes a normalized schema entering a set of normalized relations and the system has to correct them. Most of the existing tools for database normalization have not been conceived as web-based environments but as software applications that automate some step of the normalization process. To the best of our knowledge there are few web environments that support database normalization learning [17–19]. The most representative is NORMIT proposed by Mitrovic [18]. NORMIT is a web-enabled tutor for database normalization where the student selects a problem and goes through a number of steps to analyze the quality of the database following a fixed sequence of actions (determine the candidate keys, the closure of a set of attributes and prime attributes; simplify functional dependencies, etc.). There is a specific web page for each one of these tasks. Student solutions are analyzed by the system receiving feedback. The environment can be seen as a question-answer based approach since a student does not propose a solution, he just answers if the displayed solution is correct or not.

In summary, despite the advantages provided by the described tools with respect to more classical teaching methodologies, we consider that they still present several limitations. A first one is that they are specific applications not integrated into a general e-learning platform. Hence, they do not support lecturer tasks, such as continuous assessment, the tracking of students work to detect weak points, capabilities to obtain statistics of common errors, number of attempts required to solve a problem, etc. For a lecturer, all these features are very important since they provide information of student progress and also which topics need further work allowing the lecturer to perform a formative evaluation. On the other hand, another common limitation of the majority of tools is that they do not support automatic correction and only provide a way to show the results of a given exercise for a predefined database. The student has to compare the obtained result with the correct solution. Such an approach is difficult to apply when more than one solution per exercise is possible.

From our point of view, besides the functionalities of a common e-learning platform we consider that an e-learning environment has to support the automatic correction of problems. The capability to automatically generate and correct different exercises allows personalized attention to be given to each student.

3. The ACME-DB environment

In 1998, we started to develop ACME an e-learning platform to improve both teaching and learning at the technical/engineering degree programs at our university [9]. The platform was conceived to integrate new types of problems with minimal modifications. The ACME-DB environment is an extension of the ACME tool designed to support, in a unique environment, the main topics of a database course. We started to develop ACME-DB in 2003 with the development of the relational database schemas module [20]. Since then, entity-relationship diagrams [21], normalization [22], SQL and relational algebra [23] correctors have been developed and integrated in the environment. In this paper, we only consider entity-relationship diagrams, relational database schemas and normalization correctors. All these modules apply the same methodology that is illustrated in Fig. 1 and briefly described below.

- ACME-DB has a repository of problems with the problems entered by the teachers. The problem consists of a descriptor that gives details of the database that has to be created, and the procedures required for the automatic correction of the exercise. These procedures are required by the correction module to determine if the solution proposed by the student is correct or not. In the case of ER diagrams, relational database schemas and normalization problems, which due to their complexity, the problem also has attached possible solutions, entered by the teacher. To enter the problems in the repository ACME-DB provides a text editor in order to ensure that all parameters are correct.

  The next example illustrates the structure of an exercise stored in the repository. First we indicate the type of problem by entering a code, in this case a relational database exercise. Then, the problem descriptor and finally the correct solutions are entered. Note that for each solution we specify the name of the tables and for each table the primary and foreign keys and the name of the fields that compose it.

  <Code of problem = rel>
  <E> Problem descriptor </E>
  <SOLUTION1> Solution 1

TABLE NAME	*< table name >*
Primary KEY	*< field name,..>*
Foreign KEY	*< field name,..>*
Other fields	*< field name,..>*

  . . .
  </SOLUTION1>
  <SOLUTION2> Solution 2
  . . .
  </SOLUTION2>
  . . .

- The student enters in the ACME-DB environment and once in the system a list with the exercises that compose their workbook appears. This information is presented in a tabular form as illustrated in the example of Fig. 2(a). In this example the workbook contains three topics, labeled in the first column of the displayed table from 1 to 3. For each topic, as can be seen, in the second column of the table, there are a different number of exercises. Each row of the table displays the topic, the number of exercise, if it has been solved or not and the number of sent solutions. The exercises of the workbook have been assigned automatically by following the teacher's specifications and by using the problems stored in the repository of the system.
- The student selects one of the exercises in his workbook and proposes a solution. In Fig. 2(b) an example of a problem descriptor is given. To enter the solution, there is a specific interface which is specifically designed for each type of problem. In Fig. 3 we illustrate the interfaces corresponding to ER diagrams and relational database schemas. The ER interface (Fig. 3(a)) integrates different designing tools to create an ER diagram. The different parts of the interface are: (a) buttons to draw the elements of the ER-diagram, when a button is pressed its corresponding glyph graphic is drawn in the working area, a rectangle for entities, a diamond for relation-

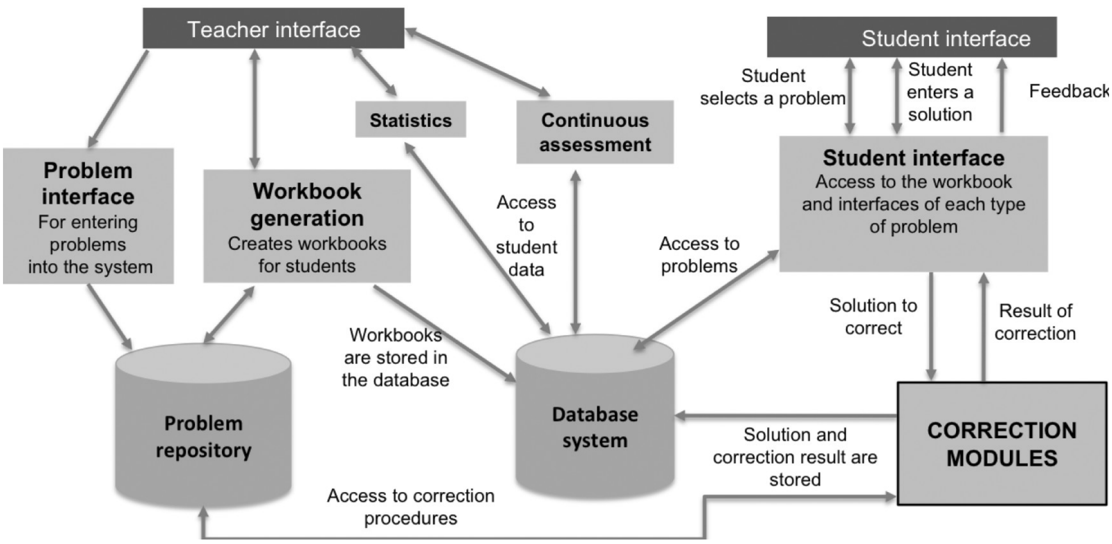

Fig. 1. ACME-DB environment.

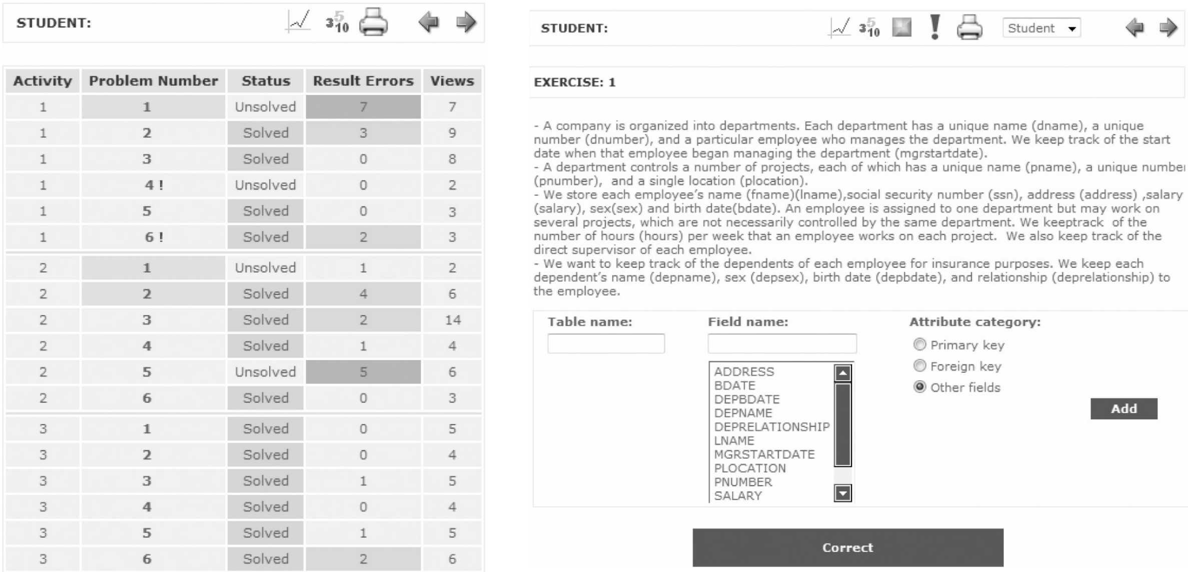

Fig. 2. (a) Workbook interface and (b) Problem descriptor.

ships, with two lines connecting the two entities previously selected; (b) working area; (c) menus for input attributes; (d) zoom and (e) correction button. The interface for relational database schemas problems is represented in Fig. 3(b). To design a relational database schema the student defines the tables and the different attributes that compose them. These tables are represented in the screen. Each time the student press the Add button of this interface the information of the items appears in the corresponding table. The normalization interface is similar to the one of Fig. 3 (b).

- The solution entered by the student is sent to the corresponding corrector and the correction process is done automatically. In our case, the correction strategy is based on a matching process that compares the student solution with each one of the solutions entered by the teacher and stored in the repository. If no matching is found, the most similar solution is selected and is used to return feedback messages. These feedback messages are

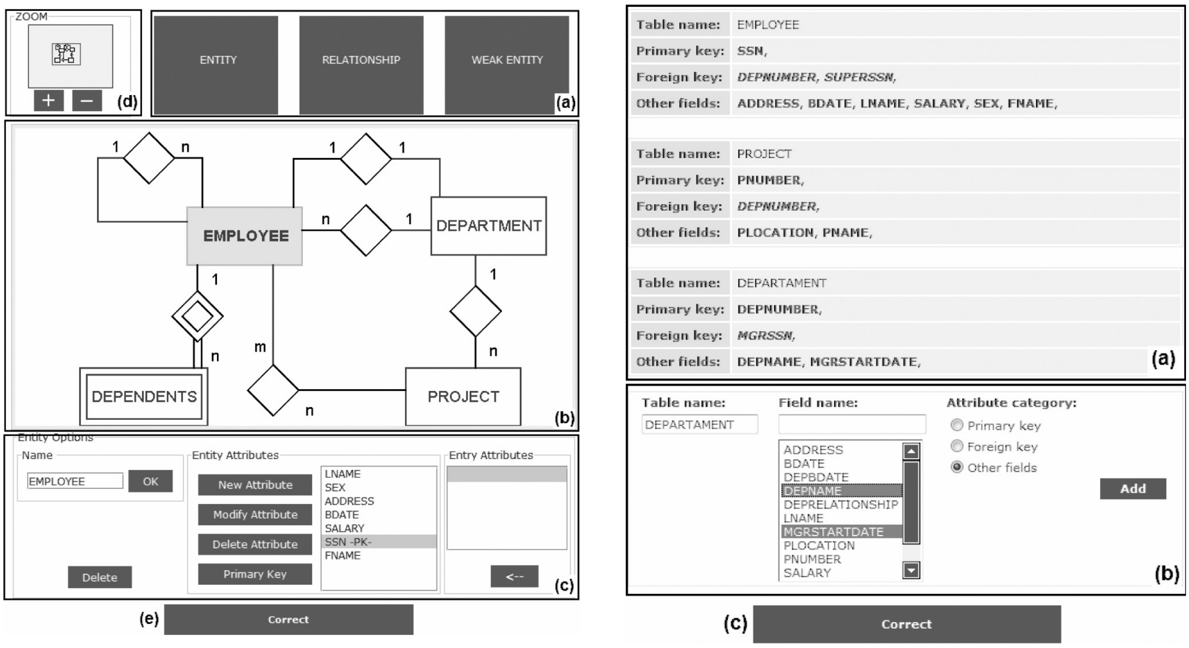

Fig. 3. Interfaces for entering solutions (a) ER diagrams and (b) relational database schemas.

designed according to the type of problem, and in the case of errors, some help to obtain the correct solution is given. Some examples of these messages are: 'More entities are required', 'There are incorrect relationships', etc.

- For each student the system records all the entered solutions. This information is used to grade the exercises. The grading process uses a function specifically designed for each type of problem where different parameters are taken into account. For instance, in the ER diagrams we consider the number of correct/incorrect entities/relationships/attributes, the number of attempts before a correct solution is obtained, etc. The marks for all the problems are used for continuous assessment.

ACME-DB has been implemented using PHP, Java applets for implementing the graphical interfaces and JavaScript for supporting relational database schemas. Correction modules are also implemented in PHP.

4. Results

The different modules were designed, implemented and then tested on an experimental database course during 2004/05. We evaluated the modules independently, obtaining very promising results. In this section, we describe how the ACME-DB environment has been used in database design and the obtained results.

Due to the bad results achieved with the previous methodology (see Section 2.2) we decided to modify it introducing the ACME-DB platform. The idea was to introduce this tool to improve database design skills acquisition. We applied a new teaching/learning methodology where ACME-DB had a main role. The goal was to exploit all the capabilities of the platform to assign, solve and automatically correct the activities assigned to the students. The system creates a different workbook for each student containing exercises of the main topics: ER-diagrams, relational database schemas and normalization. For each topic we define exercises of different complexity according to the number of entities or relations. The easier exercises ranged from 5 to 10 entities/relations while the more difficult from 15 to 20. These exercises were used for continuous assessment contributing to 50% of the final mark and the other 50% is obtained from an exam done at the end of each topic. The continuous assessment mark was obtained automatically taking into account student work: the number and type of errors, the number of solutions before the correct one is obtained, etc. The capability to automatically obtain this mark reduces the time required to correct exercises allowing the teacher to spend time on other tasks. To control students work the teacher tracks their solutions and according to detected errors, he can propose additional tasks in order to improve the learning process.

In Table 1, we collect the results obtained using the platform in the last four years. For each of the three topics (ER-diagrams, relational database schemas and normalization), we give the number of students, the number of assigned problems, the % of students that: have read the problems, have sent solutions, and have sent correct solutions. We also collect the % of students that have solved the problems at the first, the second, the third and with more than three attempts. Note that more than 80% of students read the exercises and more than 70% solve them correctly. The number of students that obtain the correct solution at the first or second attempt is between 63% and 80%, and the rest requires three or more attempts.

To evaluate the impact on the academic results, we compare the results obtained before the application of ACME-DB with the results obtained in these last four years. This data is collected in Table 2. For each course we give the number of students and also

Table 1. Information obtained from the ACME-DB tool with respect to its use

							% attempts			
Topic	**Course**	**Students**	**Problems**	**% read**	**% send solution**	**% solved**	**1**	**2**	**3**	**>3**
Entity relationship diagrams	2005/06	61	6	92.5%	90.5%	81.5%	45.3%	26.4%	16.0%	12.3%
	2006/07	50	8	94.0%	90.0%	76.4%	39.8%	25.1%	15.7%	19.4%
	2007/08	46	8	96.2%	94.0%	81.0%	57.4%	23.8%	9.8%	9.0%
	2008/09	36	8	91.0%	90.5%	80.0%	42.3%	36.9%	10.1%	10.7%
Relational Database schemas	2005/06	61	6	92.5%	84.6%	70.5%	33.0%	30.2%	16.2%	20.7%
	2006/07	50	8	92.8%	84.4%	74.4%	46.8%	27.4%	12.9%	12.9%
	2007/08	46	8	96.0%	90.9%	75.4%	36.5%	33.2%	13.9%	16.3%
	2008/09	36	8	86.2%	82.4%	78.1%	43.3%	28.7%	13.4%	14.6%
Normalization	2005/06	61	6	89.6%	86.9%	82.0%	48.0%	26.8%	11.4%	13.8%
	2006/07	50	8	91.5%	88.5%	82.5%	50.9%	23.6%	17.6%	7.9%
	2007/08	46	8	84.2%	77.8%	72.3%	57.4%	23.8%	9.8%	9.0%
	2008/09	36	8	81.4%	78.6%	75.0%	51.4%	28.6%	11.4%	8.6%

Table 2. Academic results no using and using (grey background rows) the ACME-DB environment

Course	Students	A	B	C	D	NP	No pass	Pass
2002-03	87	2.3%	14.9%	26.4%	39.1%	17.2%	56.3%	43.7%
2003-04	111	1.8%	9.0%	49.5%	25.2%	14.4%	39.6%	60.4%
2004/05	104	2.9%	13.5%	43.3%	16.3%	24.0%	40.4%	59.6%
2005/06	61	3.3%	37.7%	23.0%	16.4%	19.7%	36.1%	63.9%
2006/07	50	4.0%	24.0%	36.0%	20.0%	16.0%	36.0%	64.0%
2007/08	46	2.2%	21.7%	47.8%	15.2%	13.0%	28.3%	71.7%
2008/09	36	2.8%	52.8%	22.2%	5.6%	16.7%	22.2%	77.8%

their final marks graded from A to D, where A corresponds to the best mark and D to the worst. The NP column represents the students that have not assisted in the course. Finally, in the two last columns we give the % of students that have passed or not passed the course, respectively. Note that the results obtained when using ACME-DB (2005/06 to 2008/09) are better than when not using the platform. Moreover, if we analyze the number of students that have passed the course, there is a considerable increment compared with the results obtained when no using the platform. We want to remark that although the number of students in previous courses has decreased, the ratio student per teacher is almost the same, therefore this reduction does not affect in the results. We consider that these improved results are due to the fact that the student feels supported by the environment when solving exercises. He knows that when he has a solution he can obtain the automatic correction immediately and at any moment of the day.

In Table 3 we evaluate the relationship between the number of students that have solved the ACME-DB exercises and the number of students that have passed the course. Our hypothesis is that ACME-DB has a positive effect on the students and hence the students that solve more exercises obtain better academic results. Instead of considering using ACME or not, we considered it more relevant to take into account the degree of participation of the students. For this reason, we group students into four different groups according to the exercises that they have solved correctly. Each group corresponds to one row of the table. Group 1 are students that have solved less than 25%, Group 2 between 25% and 50%, Group 3 between 50% and 75% and Group 4 more than 75%. We compute for each group the mean and the standard deviation considering each one independently and all together. A graphical representation of this data is given in Fig. 4. Observe that the mark of the majority of students of Group 1 is 0 and it has low variability. In the case of Group 2 students are divided almost equally between those who passed (note > 0.5) and those who do not pass (note < 0.5), indeed the median is 0.45. The last two groups, 3 and 4, present the minimum at 0.45 although the median and quartiles of Group 4 (0.69, 0.6 and 0.76, respectively) are higher than Group 3 (0.57, 0.45 and 0.65). From these results we can conclude that ACME-DB has a positive effect and hence the students that solve more exercises obtain better academic results.

To corroborate this fact, we carried out a test for comparing means in order to guarantee that the observed differences between groups are significant. To verify that the ANOVA test can be applied first we performed a Levene's test using the 'vartestn' function of MATLAB to assess the equality of variance in different samples. The obtained results are shown in Table 4. Observe that all the p-values are greater than a significance level of 0.05, so we accept that the variances are equal. Therefore, we can apply the ANOVA test to compare means. An analysis of variance showed that the effect of group was significant, *F (3,190) = 130, 2, p = 0.000*.

In a multi-group study, tests indicated that Group 1 *(M = 0.059, Sd = 0.13)* was significantly lower than Group 2 *(M = 0.46, Sd = 0.12), t(35) = –9.26, p = 6.07E-11,* Group 3 *(M = 0.56, Sd = 0.11), t(70) = –15.78, p = 9.14E-25* and Group 4 *(M = 0.67, Sd = 0.13), t(117) = –18.30, p = 3.91E-36*. We also observed that Group 2 *(M = 0.46, Sd = 0.12)*

Table 3. Means and Standard deviation of 4 courses independently and all together

	Total		2006		2007		2008		2009	
Group	**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Mean**	**Std Dev**
1	0.059	0.13	0.14	0.20	0	0	0.15	0.089	0	0
2	0.46	0.12	0.46	0.12	0.35	0.18	0.47	0.089	0.55	0.11
3	0.56	0.11	0.58	0.13	0.56	0.10	0.54	0.073	0.60	0.085
4	0.67	0.13	0.71	0.12	0.64	0.13	0.64	0.12	0.72	0.13
Total	0.57	0.12	0.57	0.13	0.55	0.12	0.56	0.11	0.61	0.11

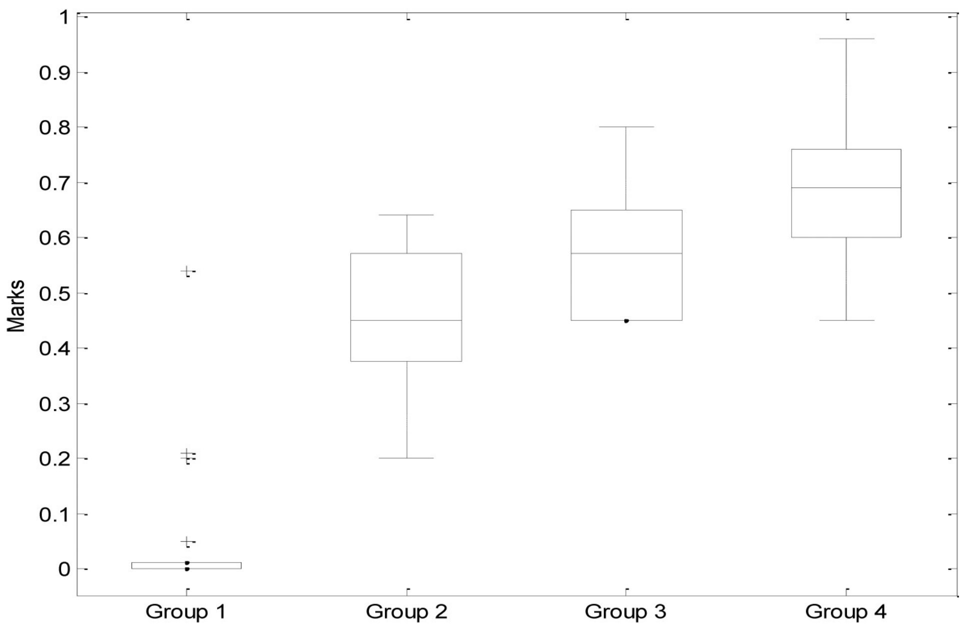

Fig. 4. Box-plot by groups of the marks for all years.

Table 4. Levene's test and p-values

	Total	2006	2007	2008	2009
Levene's statistic	0.73 0.53	1.12 0.35	1.91 0.14	0.94 0.43	0.71 0.55

was significantly lower than Group 3 *(M = 0.56, Sd = 0.11), t (73) = –3.45, p = 9.30E-04* and Group 4 *(M = 0.67, Sd = 0.13), t (120) = –6.94, p = 2.07E-10.* Finally, the pairwise comparison of Group 3 and 4, indicated that Group 3 *(M = 0.56, Sd = 0.11)* was lower than Group 4 *(M = 0.67, Sd = 0.13), t (155) –5.71, p = 5.35E-08*. Therefore, we can conclude that there are significant differences between groups, and confirm that students who do more exercises with ACME-DB obtain better academic results.

5. Conclusions

We have presented ACME-DB, a web-based environment developed to give support for the teaching and learning of database courses. We have described the main modules that integrate this environment and how it has been applied in database courses at our university. We have evaluated the data collected during four years of application and we have compared it with the results of previous courses. From our study we have concluded that ACME-DB has a positive influence on academic results. It has been demonstrated that students that use ACME-DB obtain better results.

Our future work will be centered on a more exhaustive analysis of collected data. We also plan to extend the ACME-DB platform integrating a module that supports the automatic correction of conceptual object modeling using UML class diagrams.

References

1. ACM curricula recommendations, http://www.acm.org/education/curricula-recommendations, Accessed 27 March 2010.
2. M. A. Robbert, M. Wang, M. Guimaraes and M. E. Myers, The Database Course: What must be taught, *ACM SIGCSE Bulletin*, **32**(1), 2000, pp. 403–404.
3. M. A. Robbert and C. M. Ricardo, Trends in the evolution of the Database Curriculum, *ACM SIGCSE Bulletin*, **35**(3), 2003, pp. 139–143.
4. F. Springsteel, M. A. Robbert and C. M. Ricardo, The next Decade of the Database Course: Three Decades speak to the next, *ACM SIGCSE Bulletin*, **32**(1), 2000, pp. 41–45.
5. T. Connoly and C. Begg, *Database Systems: A Practical Approach to Design, Implementation and Management*, Addisson Wesley, 4th edition, 2005.
6. C. J. Date, *An Introduction to Database Systems*, Addisson Wesley, 8th edition, 2004.
7. R. Elmasri and B. Navathe, *Fundamentals of Database Systems*, Addison Wesley, 5th edition, 2007.
8. A. Silberschatz, H. Korth and S. Sudarshan, *Database System Concepts*, McGraw Hill, 5th edition, 2005.
9. J. Soler, J. Poch, E. Barrabes, D. Juher and J. Ripoll, A tool for the continuous assessment and improvement of the student's skills in a mathematics course, *Proceedings of International Symposium of Technology of Information and Communication in Education for engineering and industry TICE 2002*, Troyes (France), October 18–20, 2002, pp. 105–110.

10. P. Chen, The entity-Relationship model, Towards a Unified View of Data, *ACM Transactions on Database Systems,* **1**(1), 1976, pp. 9–36.
11. E. F. Codd, A relational Model of Data for Large Shared Data Banks, *Communications of the ACM,* **13**(6), 1970, pp. 377–387.
12. F. Batmaz and C. J. Hinde, A web-based Semi-automatic Assessment of Conceptual Database Diagrams, *Proceedings of the 6th Web-Based Education WBE 07*, Chamonix (France), March 14–16, 2007, pp. 427–432.
13. M. Constantino-Gonzalez and D. D. Suthers, A coached collaborative learning environment for entity relationship modeling, *Proceedings of 5th International Conference Intelligent Tutoring System ITS 2000*, Montreal (Canada), June 19–23, 2000, pp. 324–333.
14. P. Suraweera and A. Mitrovic, KERMIT: A Constraint-Based Tutor for Database Modeling, *Lecture Notes in Computer Science*, *LNCS 2363*, 2002, pp. 377–387.
15. P. Thomas, K. Waugh and N. Smith, Tools for supporting the teaching and learning of data modeling, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications EDMEDIA 2007*, Vancouver (Canada), June 25–27, 2007, pp 4014–4023.
16. N. Milik, M. Marshall and A. Mitrovic, Responding to Free-Form Student Questions in ERM-Tutor, *Lecture Notes in Computer Science*, *LNCS 4053*, 2006, pp. 707-709.
17. H. Kung and H. Tung, A Web-based Tool to enhance Teaching/Learning Database Normalization, *Proceedings of 2006 Southern Association for Information System Conference*, Jacksonville(USA), March 11–12, 2006, pp. 251–258.
18. A. Mitrovic, Normit: a web-enabled tutor for database normalization, *Proceedings of International Conference on Computers in Education ICCE 2002*, Auckland (New Zealand), December 3–6, 2002, pp. 1276–1280.
19. L. Zhang, R. Kaschek and Kinshuk, Developing a Knowledge Management Support Systems for Teaching Database Normalization, *Proceedings of 5th IEEE International Conference on Advanced Learning Technologies ICALT 2005*, Kaohsiung (Taiwan), July 5–8, 2005, pp. 344–348.
20. F. Prados, I. Boada, J. Soler and J. Poch, An Automatic Correction Tool for relational Database Schemas, *Proceedings of 6th International Conference on Information Technology based on Higher Education and Training ITHET 2005*, Santo Domingo (Dominican Republic), July 7–9, 2005, S3C pp. 9–14.
21. F. Prados, I. Boada, J. Soler and J. Poch, A web based-tool for Entity-Relationship Modeling, *Lecture Notes in Computer Science, LNCS 3980*, 2006, pp. 364–372.
22. J. Soler, I. Boada, F. Prados and J. Poch, A web-based Problem-Solving Environment for database Normalization, *Proceedings of 8th International Symposium on Computers in Education SIIE 2006,* León (Spain), October 24–26, 2006.
23. J. Soler, I.Boada, F. Prados and J. Poch, An automatic correction tool for relational Algebra Queries, *Lecture Notes in Computer Science LNCS 4706*, 2007, pp. 861–872.

Josep Soler received his M.Sc. degree in physics from the Universitat Autònoma de Barcelona. He is an Associate Professor of computer science at the University of Girona, Spain. His research is focused on the application of web-based tools in computer science courses.

Imma Boada received her Ph.D. degree in Computer Science from the Technical University of Catalonia, Spain in 2001. She is Associate Professor of Computer Science at the University of Girona.

Ferran Prados received his MSc. degree in Computer Science from the Universitat de Girona (Spain). He is an Assistant Professor of the Computer Science Department at the same university. His research interest is in the field of e-learning.

Jordi Poch received his Ph.D in Mathematics from the Technical University of Catalonia in 1996. He is an Associate Professor of the Department of Applied Mathematics at the University of Girona. His research interest is in the field of e-learning.

Ramon Fabregat received his Ph.D. in Information Technology from the Universitat de Girona (Spain) in 1998. He is an Associate Professor in the Department of Architecture and Technology of Computers at the same university. His research interests are in the fields of e-learning and adaptive hypermedia systems.