

An Educational Software Supported Constructivist Methodology for Data Modeling and Database Design Skills Improvement

Ricardo Jaime

ricardojaime@udi.edu.co

Grupo de Investigación en Tecnologías de la Información Aplicadas
a la Educación GIDSAW

Universitaria de Investigación y Desarrollo UDI
Bucaramanga, Colombia

Abstract

This paper is about an in progress experience, applying a constructivist approach to improve the design skills students obtain in an introductory database course at an undergraduate computer science career. It was necessary to develop educational software and a methodological model to produce a constructivist context in which design skills were more significant than ability to use business database management software. Also it was modified the current curricula in database introductory course.

Keywords: constructivism, data modeling, database design skills, educational software.

1. INVESTIGATION PROBLEM

This investigation addresses the problem of low student's performance in a Computer Science undergraduate career, particularly in Database courses, at Universitaria de Investigación y Desarrollo UDI, a private university in Colombia.

Because of that, students are often frustrated about the skills they develop; frequently they fail courses and some of them drop out, or remain but poorly motivated.

Before this investigation, several teachers claimed the cause was students were accustomed to a low level of effort during the middle school, due to unfortunate educational system reforms.

This situation was giving rise to generational conflicts even. The majority of current teachers, when they were students, made their careers in terms of restricted access to technology, no more than one or two hours a week; on the contrary, current students usually have a computer almost permanently, nevertheless they and their teachers wonder about how is it possible, when students are about to graduate, they are not able enough to transfer their analytical

and problem solving skills to other domains, increasing both students and teachers disappointment.

With the aim of finding ways to overcome this problem, GIDSAW, a research group affiliated to UDI, proposed this project and started it about January 2008; currently it is in progress.

2. INVESTIGATION PROJECT

Preliminary

During the project design, there was an interest to confirm or deny, the often-quoted teachers observation, about a bad attitude they supposed students had, originated in their pre-university education.

A literature review gave the project team indications that the problem occurs also in other contexts. Connolly (2005) reported that companies in Europe considered there were lacking database design skills in IT graduate students and in IT staff in companies. It does not seem to be product of characteristics similar to the Colombian educational system.

At this stage of research, review of literature showed consistent pedagogical, technological and methodological issues, to

promote the adoption of a constructivist approach, and to support this approach with computer tools made for it, not with business software. (Chen, 2001) (Van Gorp & Grissom, 2001) (Chen & Ray, 2004) (Fessakis, Dimitracopoulou, & Komis, 2005) (Connolly & Begg, 2005) (Cheong, 2008) (Wals Zurita, 2005) (Bogdanovic, Stanimirovic, Davidovic, & Stoimenov, 2008).

From previously referred author's ideas, it could be derived that constructivism is an appropriate approach, to get the interest of students from their own experiences, but this required teachers to overcome their prejudices, and to outline strategies for giving students significant learning experiences.

Also it was derived from literature, that students find particularly difficult to learn database analysis and design, not because their bad attitudes, but because they find that curriculum does not match with their expectations.

According with ACM's (2008) reflections on the computer crisis, in some countries like China and India, computer science has expanded in recent years, but students enrollment in programs is declining in other countries like US, Canada, UK and Mexico, because among other things, a wrong public image about career.

Enrollment in computer science careers are yet increasing in Colombia, but it is true that public image of the field is wrong, similar to ACM (2008) raises : in pre-university education, students does not construct right ideas about computer science, and when they begin a career, they really want to learn quickly particular skills related with entertainment, computer games, web development, expertise in software required for employment, but they are unaware about the complexity of curriculum, effort it implies, and time it will elapse.

A typical example of the previous situation is that some students want to learn to use Oracle, or SQL Server, or MySQL, but they are unaware they have to learn about databases before of that.

In this context, student's confidence often decreases quickly, because they realize concrete things they thought they could do relatively soon, require some advanced

and abstract tasks like design which will delay the concrete things.

Research questions

From the mentioned evidence, raised several questions:

- ¿Given students and teachers profiles, how they could be involved in the constructivist approach, to improve data modeling and database design skills in Computer Science Database course?
- ¿What should be the characteristics of educational software for supporting the learning environment proposed?
- ¿How it should be the methodology to have a successful integration between constructivist approach and educational software?
- ¿What could be the appropriate changes in regular curriculum of database introductory courses?

3. INVESTIGATION MILESTONES

Mentioned questions became milestones for the investigation process. Each one of them is developed in following parts of this section.

Making design skills significant again

Design is a task that has become less significant for students than it was in the past. ¿How students and teachers could be involved in a constructivist approach that let them make design significant again?

Because of the scarcity of computer resources in the past, databases courses were theoretical, design testing was done on desktop, and the practical implementation was limited to one or two sessions at the end of the course, barely enough for learning the tool. Design was meaningful for students, because they had to support the validity of their design in public, and then they did not feel pressured to learn to use software before they understand design issues.

In comparison, today the classes are held in permanent disposal of computers, the tools are available and the student can implement their basic designs and evaluate them by trial and error as many times as it was necessary. But this advantage brings with it a new risk: to focus attention on implementation, students not only avoid the discussion about their designs, but they are forced to allocate their time between learning of

concepts, and learning the characteristics of a particular development tool, turning it the central activity (Abdullat, 2001). Design ceased to be significant for the student; the meaningful thing is now the use of database management software, and students discuss among themselves about their skills with software, from a subjective perspective. For them the challenge is, for example, learning how to create a table, learning how to make a query, without this having anything related with the complexity of the database they design.

Towards the end of the academic period, students have been theoretically guided about how to design databases, but they have failed to discuss their design, and they are finally committed to learn a data declaration language, a data manipulation language, and at the same time to get skills in using database management software, whose characteristics usually exceed those strictly required for teaching purposes.

Sometimes, teachers propose to do the class in a conventional classroom again, which would get the students away from computers and would compel them to carry out a design phase, and also would bring the advantage of avoiding the distractions caused by the use of internet during class sessions.

But there are reasons to conclude that students would misinterpret this as a setback in their educational process.

To make design significant again, it was necessary to think about how it could be possible maintaining the presence of software and internet, but in a way that design was the central element.

Something helpful to achieve this purpose, is giving students contexts in which they see the database design as something useful in modeling real information (Chen, 2000). Real information means information that is meaningful for students, even if it is not meaningful for teachers, at least so that modeling becomes a significant challenge for students.

Students should be encouraged to use Internet to enrich their background about subjects interesting to themselves, and to propose database designs complex enough to contain efficiently all details they

consider important, and to answer all questions they consider interesting.

Using software to give feedback

As important as to achieve students to be interested in design, is to give them feedback, so they can evaluate by themselves the convenience of the designs they propose, and to they can argue about it. ¿How it should be educational software to give better feedback than business database management software?

Since the fundamental of a relational database is the concept of table, and debugging techniques seek to optimize storage of data in tables and data recovery through queries, it was established that one of the first software requirements should be useful in basic problems of storage, such as those relating to redundancy and multi-valued attributes. It was decided to meet this requirement by using a spreadsheet at the beginning of the course (Steinberg, 2007). As spreadsheet is general purpose software, it allow to deal with any subject proposed by students, and normally they do not need additional training to use this kind of tools.

Once the basic concept of table, and the need to deploy multiple related tables were clear to students, they would use a software which should allow to define relational systems through a simple interface, but most important than this, would allow to enter data in tables to see the relational model, specially the foreign keys operational.

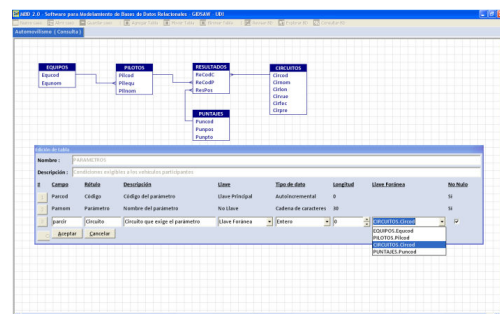


Figure 1. Creation of a relational model

As a part of this investigation, it was developed software named MBD, for database modeling in Spanish. During the second academic period of 2008 it was developed MBD 1.0, and ending the period it was made usability testing of the tool. The results of analysis were formulated as requirements for the new version. During

the first academic period of 2009, project team has been tracking the progress of two groups of students, one of which uses the tool.

With MBD 2.0, students enter the data dictionary and from this, the software plots the relational diagram as shown in Figure 1. It is important to say that design is not drawn, but is determined by the introduction of the data dictionary.

MBD 2.0 has been developed as a teaching tool for evaluating relational databases, through a generic interface that allows, by clicking on tables, the inclusion of data once the case has been created in the software, keeping visible and related both the design of the database and its content, which is expected to give better feedback for anyone who is learning how to model databases, as shown in Figure 2. Feedback for students is to realize if their models are able to contain all relevant data, or on the contrary, if some important data cannot be included because design mistakes.

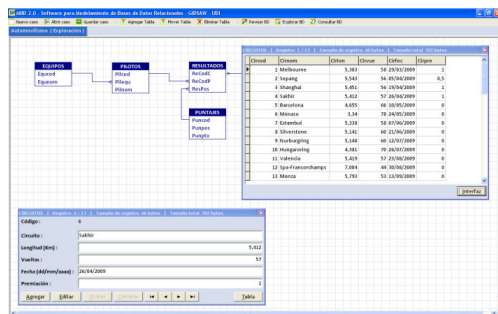


Figure 2. Records inclusion and review

The query mode in MBD 2.0 allow to execute SQL statements, which is a new opportunity for student feedback, as it shows whether the design is actually able to answer questions made for potential users.

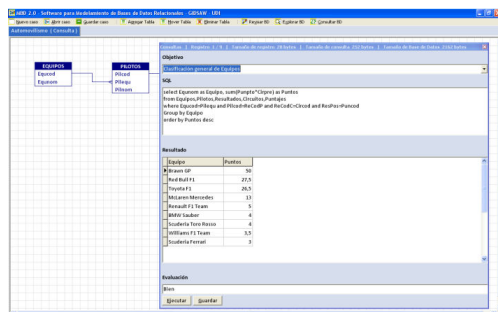


Figure 3. Queries in MBD 2.0

Each query is defined by a goal or question, an SQL section in which the sentence is constructed, a table of resulting data and a field to record qualitative evaluation concepts. The query definitions are stored and run on the data contained in the tables every time they are loaded, helping student to understand independence of queries from data.

MBD 2.0 generated a Paradox database, but both the structure, such as records and queries are finally contained in a single .mbd file, to allow greater portability of the project.

Defining a methodology for problems, projects and cases

The next question is how it should be the methodology to have a successful integration between constructivist approach and educational software?

The methodological aspect was initially based on previous experience with the courses in the field of computers and organizations, that traditionally focus on learning of theoretical concepts and simulation by quantitative modeling, complemented during the past semesters with a qualitative modeling approach, which stresses that in computer science, using language that provides the system dynamics, can be deployed learning process with a positive impact on the innovative use of information technology.

Case Based Reasoning (Case-based reasoning) CBR is an approach to problem solving based on the recovery and adaptation of cases, or descriptions of episodes associated problems and solutions (Allen, 1994 p.40). From this perspective, it is assumed that human knowledge is stored and can be captured in the form of stories (Schank, 1990). When asked by the solution of a problem, it searches the memory of past experiences that can be reapplied in new situations and described telling a story. The reason for reusing these stories is to create new solutions using the knowledge and the solutions used previously (Kolodner, 1993).

But it would be unproductive to a process of case-based reasoning, to compare cases not originated in the same context in which the student is developing its process. For this reason, constructivist models were required to consider the

formulation and development of the first cases, so that in later stages constitute a basis on which comparisons and conclusions were possible. Anticipating that within the group it could be found heterogeneous levels of ability to find information and presentation of hypotheses, some cases arise in a problem-based learning, and others are raised under a model of project-based learning. The end products are in both cases, design proposals and their implementation in software developed for that purpose, thereby establishing the basis for the subsequent case-based reasoning.

Following recommendations of documented experiences, it was proposed a methodology to incorporate the MBD 2.0 tool that involves two types of sessions in cycles.

First session of each cycle is devoted to propose the case. If it is presented as a problem, the approach can be open, without involving students in explicit questions their database design should answer. Anticipating that the target population of the methodology is likely to make use of the internet in search of information, there are web sites recommended in which the student can find reliable and complete information.

It is suggested teacher do not let student transfer responsibility for the inquiry to the teacher, that is, does not expect the teacher to share the information he has collected for the formulation of the case. But teacher should not ignore students when note that they are not moving in the understanding of the topic, and it could be acceptable making questions to help students in reading comprehension and information exploration. At this point it is critical not exceed the actual capacity of the group, so that cases should be on issues close to their interests at the beginning, and gradually expose them to greater need for exploration in new issues.

There are several alternatives to assess the performance of this work: socialization of the subject, comparison and discussion of design proposals, elaboration of comprehensive documents, among others. Teacher must influence discreet so that students are not rushing to make design before we get a broad enough understanding of the topic. Studies

suggest the desirability of natural language for the first representation of the information, after giving way to models in more restricted languages.

It is therefore suggested that the software is not used as a tool for creating design, but as a tool for implementing them, after they have been assessed, this in order to foster in students the ability to direct interpretation and evaluation of design.

The second session, referred to the conclusion of the case, and necessarily project-oriented should be included as requirements, aims to generate feedback about the proposed design.

The first feedback should be about the process of inquiry, in which students must give evidence if the search was comprehensive enough. It is important that students arrive at the second session with sufficient background to support a discussion on the subject, and the formal products such as relational design and data dictionary.

In cases that were proposed as problems, the teacher should be open minded to discuss about some proposed design different from those he had foresee, but still are good proposals.

This is expected, since in targeting problems may be divergence of proposals in aspects not provided by the teacher, and if this is the case, the feedback must be very careful to avoid creating frustration in students, and instead of that, generate a discussion about the importance of a rigorous exploration of the topic.

One significant contribution of the teacher is to provide real data for testing all proposals. If there has been a good learning process monitoring, the teacher must strive to identify specific data for cases that breaks the design proposals of students, for reasons such as the determination of inappropriate field sizes, or the confusion between entities and attributes, or other reasons related with comprehension by each group.

For students, the ongoing process to develop projects and problems, has the advantage of forming a collection of cases generated by themselves, suitable of being studied and discussed, and from which

they can identify recurring regularities and structures, which lead to infer from practice some technical concepts that in regular curriculum they are supposed to learn by theory. Then, each student should be encouraged to produce designs, and also to analyze his/her mates designs, to take advantage of CBR approach.

Changing the current curriculum

Some of the indications of incorporation of software in sessions, described in methodology, make it necessary breaking the characterization of the course program databases mentioned earlier, in which SQL was a final theme.

While in principle the data manipulation language and the implementation of the consultation will require the active participation of teachers, they should encourage students to learn their use early in the course.

The resultant course is more holistic than the traditional one. With each project, problem or case, the student goes through a full cycle of design and use of the database, so that the difference between the beginning and the end of the course does not lie in the subjects studied, but in the sophistication achieved in the practical application of concepts.

4. INVESTIGATION RESULTS

As mentioned, it was developed a pilot test of the methodology and software tool in the first academic period of 2009 with 20 students of third semester of a computer science undergraduate career, which participated in seven weekly sessions of working with the assistance of a teacher researcher.

For the second academic period of 2009, a whole course was included in the project for a field test. Preliminary results have been satisfactory in several respects:

- Students use internet as a source of information, to explore great amount of information to introduce in databases.
- Some of the students work in internet is to determine technical issues like data types and length of fields.
- With real data, students have expressed high expectations regarding the efficiency of their design, not only for data storing but for data consulting.

- Because of that expectations, students have reached in 5 weeks, skills for query construction they reached in 12 weeks with traditional course.

- Some students, for whom the design was difficult to understand from the perspective of data storage, it was easier to understand after seeing the results of the queries, a process inverse to the traditional.

- Students have become more identified with their career.

It is expected that final results let the project team formulate more definitive and favorable conclusions, from the systematic comparison of data taken before and after application of the methodology for a significant time.

References

Abdullat, A. A. (2001). "Teaching A Database Systems Design Course: Is It Theory Or Practice?" Information Systems Educator Conference Proceedings. Cincinnati.

ACM Association for Computing Machinery. (2008). Current curricula - Association for Computing Machinery. Retrieved 07 01, 2009, from <http://www.acm.org/education/curricula>

Bogdanovic, M., Stanimirovic, A., Davidovic, N., & Stoimenov, L. (2008). "The Development and Usage of a Relational Database Design Tool for Educational Purposes". Proceedings of the Informing Science & IT Education Conference .

Chen, C. (2001). "An Investigation of Students' Thought Processes in Solving Business Problems". Information Technology, Learning, and Performance Journal , 19 (1), 5-19.

Chen, C. (2000). "Using Realistic Business Data in Teaching Business Problem Solving". Information Technology, Learning and Performance Journal , 18 (2), 41-50.

- Chen, C., & Ray, C. (2004). "The Systematic Approach in Teaching Database" *Information Technology, Learning, and Performance Journal* , 22 (1), 9-21.
- Cheong, F. (2008). "Teaching Database Modeling and Design: Teach an Intelligent Systems Course". *Journal of Information Technology Education* , 7, 47-60.
- Codd, E. (1970). "A Relational model of Data for Large Shared Data Banks." *Communications of the ACM* , 13 (6), 377-387.
- Connolly, T. M. (2005). "Database Technology Professional European Survey Analysis". *International Conference on Interactive Technology in Education*, (pp. 20-22). Hämeenlinna.
- Connolly, T. M., & Begg, C. E. (2005). "A Constructivist-Based Approach to Teaching Database Analysis and Design." *Journal of Information Systems Education* , 17 (1), 43-53.
- Fessakis, G., Dimitracopoulou, A., & Komis, V. (2005). "Improving Database Design Teaching in Secondary Education : Action Research Implementation for Documentation of Didactic Requirements and Strategies". *Computers in Human Behavior* , 21 (2), 159-194.
- Mishra, P., Zhao, Y., & Tan, S. (2000). "From Concept to Software : Developing a Framework to Understanding the Process of Software Design". *Journal of Research on Computing Education* , 32 (2), 220-238.
- Philip, G. C. (2007). "Teaching Database Modeling and Design: Areas of Confusion and Helpful Hints". *Journal of Information Technology Education* , 6, 481-497.
- Steinberg, G. (2007). "Teaching Relational Database Concepts to Computer Literacy Students: The Spreadsheet Metaphor". *The Proceedings of the Information Systems Educator's Conference*. Pittsburgh.
- Van Gorp, M. J., & Grissom, S. (2001). "An Empirical Evaluation of Using Constructive Classroom Activities to Teach Introductory Programming". *Computer Science Education* , 11 (3), 247-260.
- Wals Zurita, I. (2005). *Herramienta Web para la Enseñanza de Bases de Datos*. Escuela Técnica Superior de Ingeniería Informática, Universidad de Sevilla, Ciencias de la Computación e Inteligencia Artificial, Sevilla, España.
- Wright, P. H. (2004). *Introducción a la Ingeniería* (3 ed.). (R. Arrijo Juárez, Trad.) México DF, México: Limusa.