# Undergraduate Database Instruction with MeTube

James Z. Wang   Timothy A. Davis   James M. Westall   Pradip K Srimani
School of Computing
Clemson University
Clemson, SC 29634, USA

## ABSTRACT

This paper describes the principles, implementation, and outcomes of a new undergraduate database course that uses a semester-long project, known as MeTube (a variation of the well-known YouTube system), to motivate and foster interest and creativity in students, while providing adequate complexity to introduce DBMS concepts and techniques. Included in our discussion are experiences in two course offerings, as well as detailed course assessment results.

## CATEGORIES & SUBJECT DESCRIPTORS

H [**Information Systems**]: Models and Principles, Database Management, Information Storage and Retrieval.

## GENERAL TERMS

Documentation, Performance, Design, Experimentation, Security, Human Factors, Standardization.

## KEYWORDS

MeTube, TEXNH, CPATH, DBMS, Problem-based Instruction.

## 1. INTRODUCTION

The traditional computer science educational experience is notoriously defined by the countless hours students spend in and out of class developing software – writing, debugging and testing computer programs. Given the availability of computer-related social networking and entertainment outlets, often accompanied by exciting visual components, the fact that today's student appears to be no longer excited nor motivated to build a toy grade averaging program or an inventory control database, or questions the relevance of implementing an AVL tree to print a collection of integers in sorted order, should not be surprising. This perceived lack of excitement and relevance has doubtlessly contributed to the problems of attracting, motivating, and retaining students. A related but separate problem confronting the curriculum designer in a non-elite university is how to attract and maintain the interest of the best students in a class, while not leaving others behind. Assigning problems that challenge the most skilled can lead to increased withdrawal and failure rates among average students, which is unacceptable in times of declining enrollments. Conversely, problems that target the average students may cause the best students to lose interest in the course.

To address such issues, we have redesigned our freshman and sophomore computer science courses to leverage problem-based instruction in the domain of computer generated visual media through the NSF-supported τέχνη project [4]. We believe that computer graphics is a natural and effective problem domain to exploit for instruction since it is not only compelling and relevant to college-level students, but also inherently possesses a rich set of problem types of varying complexity naturally suited to problem-base learning [3] [5], the solutions of which can be evaluated quickly through visual feedback.

In our undergraduate course revitalization effort, problem-based instruction is used as a vehicle to capitalize on the interest and experience students possess in computer-generated visuals. While utilizing computer graphics to teach introductory computer science courses has been reported previously [2] [6], course design in τέχνη differs from other approaches in the depth, scope, and open-ended nature of the problems employed, especially applied to upper-level courses. Problem domains are non-traditional and selected specifically to attract and motivate students, while supporting the learning goals of the course. Problems are open-ended so that the best students are challenged, and the average students are provided with a pathway to the highest level of success which he or she is capable of achieving. In our quest to complete the curriculum for the junior and the senior year, we are in the process of redesigning our advanced courses, with the support of an NSF CPATH grant.

The purpose of this paper is to describe the redesigned database management systems course in detail – problem domain, motivation, objectives and learning outcomes, course content, and early assessment results. An embedded assessment mechanism has been introduced in this course to measure the degree to which stated learning outcomes are attained. The work of Walker and Fraser [10] and the dynamic assessment of Brown [1], which is based upon Vygotsky's zones of proximal development [9], serve as the foundation on which the assessment mechanisms are built.

## 2. STUDENT LEARNING OUTCOMES

To extend the τέχνη instruction approach [4] to our current senior-level undergraduate database course, *CPSC 462: Database Management Systems*, we designed a semester-long multimedia database project and adopted a problem-based teaching method. Students are required to design and implement a web-based multimedia database system, MeTube, throughout the semester while the instructor teaches concepts, techniques and tools to address the needs of different project phases. This aspect of the course allows students to learn new topics as they naturally arise during the implementation of the project, which strongly supports the concepts of constructivism [8] and intentional learning [7].

This course has six expected learning outcomes:

1. understanding of conceptual modeling and ability to use Entity-Relationship (ER) Models to design database applications
2. familiarity with relational data models and the ability to design relational database schemas from ER diagrams
3. ability to use an industry standard query language (SQL) to query relational databases
4. understanding of the basic concepts of query optimization and simple query optimization techniques
5. experience in designing and implementing web-based database systems
6. improvement of oral and written communication skills through project presentation

These course outcomes also collectively contribute to the program outcomes of the Bachelor of Science degree in Computer Science at Clemson University.

## 3. MeTube SYSTEM DESCRIPTION

The MeTube system is a modified version of the popular YouTube system (http://www.youtube.com). We believe that a class project based on such an extremely popular, culturally relevant system will motivate students to acquire and appreciate skills in database management systems and web technologies. We also conjecture that by implementing the MeTube system, students will attain the expected learning outcomes more efficiently since they will be provided opportunities for creative and artistic expression – rarely found in a traditional database course. A minimal set of requirements is specified to ensure course goals are met; however, the open-ended aspect of the assignment allows students to identify their own MeTube system requirements by exploring the existing YouTube system. Early results of this work can be found in [11].

Like YouTube, MeTube provides a web-based platform for users to share multimedia files. The requirements of MeTube include:

- **User accounts** A user must register for an account to use all MeTube system functions. Accordingly, students must implement basic account functions, including registration, sign-in, and profile update. Advanced features include contact list management, friend/foe list management, user blocking, etc.
- **Data sharing** A signed-in user should be able to use a web interface to upload multimedia files to MeTube. Any internet user should be able to download and view media files available in MeTube. Besides implementing the basic upload/download functions, students may elect to implement more advanced features. For instance, a signed-in user can set the sharing methods for media files he/she uploaded or block certain users from downloading or viewing media files he/she uploaded.
- **Media organization** All users should be able to browse media files by category. Signed-in users should be able to organize their uploaded media files as well as other media files, in different ways through playlists, favorite lists, etc. Students may also implement advanced features such as displaying most-viewed media files or most-recently uploaded files.
- **User interaction** Signed-in users should be able to interact with each other by exchanging messages and commenting on media files. Students may also implement advanced features such as media rating and group discussion.
- **Search** Users should be able to search for media files based on keywords through a YouTube-like search interface. Advanced features such as word cloud, media recommendation, and feature-based media search, may also be implemented.

Although MeTube is used to educate students in developing database applications with a DBMS system, its web-based nature requires students to learn web technologies and related programming languages. While similar to some past efforts [12] [13] [14], our approach differs in that the web-based aspects are integrated into a semester-long project using current web technology.

Since the main objective of the course is to teach database concepts and techniques, any web programming language used must be easy to learn and require no prior knowledge in web technologies. To this end, we selected PHP, a widely-used general-purpose scripting language especially suited for web development and easily embedded in HTML. Much PHP syntax is borrowed from C, Java and Perl with a set of specific features added. Since most of our students had prior experiences with C or Java programming, they had few problems learning PHP on their own. Further, following our goal of staying within an open-source paradigm, we chose MySQL as the DBMS system for our MeTube project. MySQL and PHP are key components of LAMP (Linux, Apache, MySQL, PHP/Perl/Python), a fast growing open-source enterprise software stack. Students developing the MeTube system with MySQL and PHP will therefore gain experience in open-source application development.

## 4. INSTRUCTION APPROACH

To allow students to begin their projects as early as possible, we divided the semester-long project into six progressive phases. The basic concepts and techniques needed in each phase of the project are introduced in classroom lectures according to students' progress. The project deliverable in each phase allows the instructor to continuously monitor the progress of the project.

### 4.1 Phases of the MeTube Project

The MeTube project consists of the following six phases:

**Phase I:** Identify system requirements by exploring the YouTube system and by studying the project requirement specification provided by the instructor; use the ER model to complete a conceptual design of the MeTube database.

**Phase II:** Convert conceptual designs into relational schemas and design the database tables for MeTube using the MySQL database management system; normalize database tables using normalization rules.

**Phase III:** Design complex SQL queries to insert and retrieve information in a multimedia database; evaluate performance of SQL queries.

**Phase IV:** Design the web interface for users to upload information to the multimedia database; retrieve information from the multimedia database; present data in appropriate formats.

**Phase V**: Test the MeTube system using various use cases; evaluate system performance and usability; improve the system and optimize database and PHP programs.

**Phase VI**: Prepare a project report to demonstrate the MeTube system; conduct an oral presentation of the project.

These phases cover not only the various aspects of the project, but the course material as well.

## 4.2 Course Content Outline

The MeTube project drives the sequence of course content, which in turn supports the continued development of the semester-long system. While content represents traditional concepts and topics of a database course, they are presented within a context that we believe will make them engaging and relevant to today's students. These concepts and techniques include:

1. *Database design specification (1 hour*): database concepts; database architecture and design considerations.
2. *Conceptual model and ER diagram (2 hours*): the ER model; EER diagrams; ER models used to design database systems.
3. *Relational data model and database schema (2 hours*): relational data model; conversion of ER models to relational database schema.
4. *Database normalization (2 hours*): functional dependency concepts; normal forms of relations; techniques for performing database normalization; associated performance tradeoffs.
5. *SQL language through MySQL (3 hours*): SQL language using the MySQL DBMS system; installing and managing the MySQL database; writing SQL queries for retrieving information from the database.
6. *Index, key, and other constraints (2 hours):* indexing; key constraints; database integrity.
7. *Query optimization (2 hours):* query processing and query optimization techniques; DBMS query processing; options for query optimization; normalization with emphasis on performance tradeoffs.
8. *Web servers, HTTP, HTML, PHP (4 hours):* web servers; HTTP protocols; HTML scripts; PHP programming.
9. *HTTP form, script design and Multimedia presentation (3 hours)*: web form design and JavaScript; multimedia data presentation on the web.
10. *Views, triggers, and stored procedures (3 hours):* views, triggers and stored procedures; transaction processing; techniques for maintaining data consistency.
11. *Performance evaluation and system improvement (3 hours):* test case design; performance evaluation methods; system improvement techniques.

## 4.3 Integration of the MeTube Project and Course Lectures

Table 1 shows the mapping of course content, the associated MeTube project phases, and the expected learning outcomes. This table can be used as a guide for scheduling lectures.

**Table 1: Mapping between project and course content.**

| *Project Phase* | *Course Content* | *Learning Outcomes* |
|---|---|---|
| Phase I | 1, 2 | 1 |
| Phase II | 3, 4 | 1, 2 |
| Phase III | 5, 6, 7 | 3, 4 |
| Phase IV | 8, 9 | 3, 4 |
| Phase V | 10, 11 | 5 |
| Phase VI | 1 ~ 11 | 6 |

One benefit of dividing the MeTube project into six phases is to allow the instructor to evaluate student progress in a timely manner and in turn, provide necessary feedback. Because students understand the concepts and techniques taught in the classroom more effectively when an instructor provides timely feedback on project deliverables at each phase, students are able to improve the design and implementation of their MeTube systems.

## 4.4 Student Projects

At the end of the semester, students had completed working versions of their MeTube systems, available for public access on the web. Figure 1 shows screen shots of two of these systems. Note that all components were designed by the students, who for the most part, desired their systems to mimic YouTube as closely as possible.

# 5. ASSESSMENT RESULTS

Since assessment is often a difficult issue, we have performed a systematic and detailed assessment of our approach using both direct measures (direct examination of student knowledge and skills against measurable learning objectives) and indirect measures (student surveys) that ascertain the opinion or perception of the value of learning experiences.

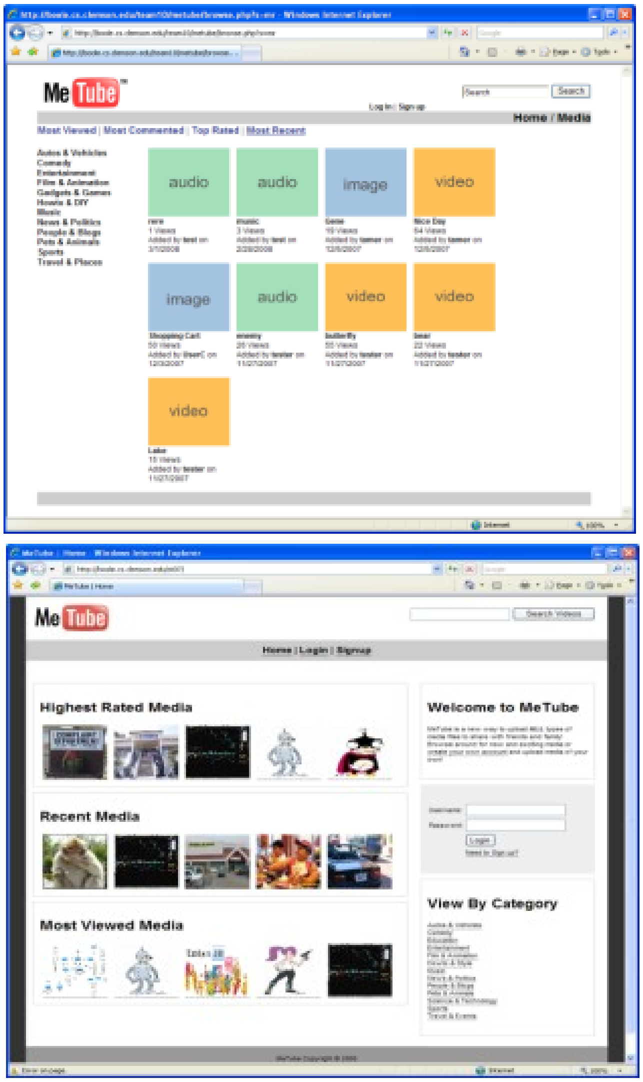

**Figure 1: Screenshots of sample MeTube Systems**

**Course Outcome Assessment**
We used the midterm, final, and semester-long team project to assess the course outcomes in a cumulative manner. The methodology was similar in both offerings of the course; we give the assessment details of the Semester I below and show the results in Figure res 2 and 3.

**Midterm:** The midterm was a comprehensive exam up to the midpoint of the semester with 7 questions – Q2, Q5 and Q7 contributed to assessment of course outcome 1; Q3, Q4, Q5 and Q6 contributed to assessing course outcome 2; and Q5 and Q7 contributed to course outcome 3. For each course outcome, each student's score for a given question was normalized with the maximum score. The normalized scores for all questions used for a given outcome were then added and normalized to compute the composite score for that course outcome.

**Final and the Project:** The final exam had 10 questions – Q2 and Q8 contributed to course outcome 1; Q2 and Q3 contributed to outcome 2; Q2, Q8, Q9 and Q10 contributed to outcome 3; Q1, Q8, Q9 and Q10 contributed to outcome 4; and Q1, Q4, Q5, Q6 and Q7 contributed to outcome 5. The semester-long team project had five interim evaluations (P1, P2, P3, P4, P5), including the final written and oral presentation. P1 contributed to course outcome 1; P2 contributed to outcome 2; and P5 contributed to outcome 5. The composite score for each outcome was computed as previously.

Similarly, the course outcome assessment results for Semester II are presented in Figures 4 and 5.

## 5.1 Student Assessment and Fraser Walker (DELES) Survey

An interesting alternative to the direct assessment is available from the study of educational environments. Walker and Fraser [10] observe that numerous studies report a strong correlation between traditional student outcomes (e.g. grades, test scores) and perceptions of classroom environments. The latter can be measured with unobtrusive and time-saving survey instruments. Walker and Fraser use factor analysis on field tests to develop a survey instrument of 34 ratings on six scales: instructor support; student interaction and collaboration; personal relevance; authentic learning; active learning; and student autonomy. Their instrument was ostensibly targeted at distance education, but if we simply omit the six items in the category of student interaction and collaboration, which is consistent with our cognitive constructivist view, we obtain a factorially valid instrument to measure classroom environments of any type. Thus, we used the resulting 28-item survey. We then used the data provided by these assessment mechanisms to identify characteristics of problem domains that best promote learning. We also used the comments from a Clemson University survey to assess the instructor effectiveness to identify issues to be addressed in future offerings of this course.

## 5.2 Student Assessment and Fraser Walker (DELES) Survey

An interesting alternative to the direct assessment is available from the study of educational environments. Walker and Fraser [10] observe that numerous studies report a strong correlation between traditional student outcomes (e.g. grades, test scores) and perceptions of classroom environments. The latter can be measured with unobtrusive and time-saving survey instruments. Walker and Fraser use factor analysis on field tests to develop a survey instrument of 34 ratings on six scales: instructor support; student interaction and collaboration; personal relevance; authentic learning; active learning; and student autonomy. Their instrument was ostensibly targeted at distance education, but if we simply omit the six items in the category of student interaction and collaboration, which is consistent with our cognitive constructivist view, we obtain a factorially valid instrument to measure classroom environments of any type. Thus, we used the resulting 28-item survey. We then used the data provided by these assessment mechanisms to identify characteristics of problem domains that best promote learning. We also used the comments from a Clemson University survey to assess the instructor effectiveness to identify issues to be addressed in future offerings of this course.

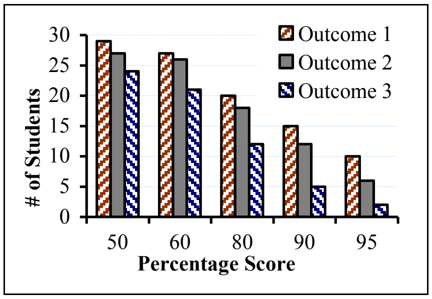

**Figure 2: Course Outcomes (Midterm, Semester I)**

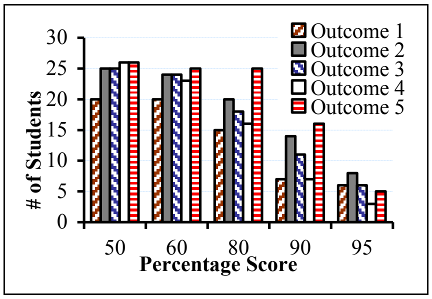

**Figure 3: Course Outcomes (Final/Project, Semester I)**

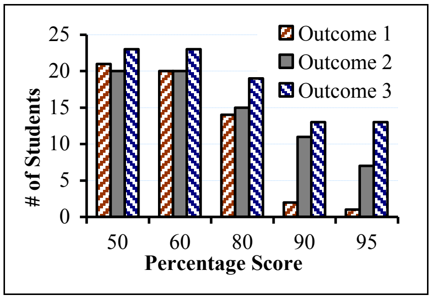

**Figure 4: Course Outcomes (Midterm, Semester II)**

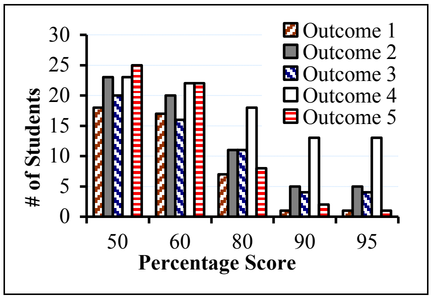

**Figure 5: Course Outcomes (Final/Project, Semester II)**

**Sample student comments from student assessment of the instructor:**

*The project (MeTube) was fun, and I learned a lot.*

*I really like the semester long project approach to teaching. This method provides hands-on experience in using the material that is being taught.*

*The way he teaches and the project he assigned helped a lot in learning web and database programming skill. It is a good course from an industrial point of view.*

*Having a course on database and some technologies that are useful in the real world is nice. There should be more classes dealing with this type of stuff and less classes focused on outdated, old methods and technologies.*

*The main project for this course was very interesting and engaging. It really gave us a glimpse on how database management worked and gave us some experience in working with a real, practical project.*

*The semester-long "metube" project is an effective exercise in putting the course's concepts into practice.*

*I believe that the project (MeTube) was too ambitious. It's just a large undertaking that requires a lot of time.*

*The main purpose of this course is to develop a MeTube system within the 16 weeks. The problem is that much of the material necessary to know is not taught until the very end.*

**DELES Survey Summary (Qualitative):**

1. *Student Interaction and Collaboration*: A sizable majority of students reported that they interact and collaborate with others and the interaction helped their understanding of the course material and the project design. They required some facilities, such as a discussion board, to share information with others to improve interaction.
2. *Active Learning***:** Most students learned new material on their own to complete the project phases in time; in many cases, students *wanted* to learn this new material to advance their projects.
3. *Personal Relevance:* Some students connected class material with their personal experiences (possibly due to prior familiarity with YouTube), while some reported not finding the connection.
4. *Authentic Learning***:** The semester-long project provided students an opportunity to deal with real-world problems via class activities. They were able to participate in teamwork and improve their skills in written and oral communication via writing project reports and presenting their projects orally to the entire class.

Detailed quantitative results of the DELES survey can be found at http://mmlab.cs.clemson.edu/metube/deles.pdf.

# 6. CONCLUSION AND FUTURE WORK

The students' experiences with the τέχνη approach to the course have been generally positive and effective, as noted in the student comments and assessment graphs. Allowing students to study and implement real-world problems is engaging and motivating, as evidenced through student pride in their final projects. While we plan to continue offering the course using the MeTube project, we recognize room for improvement.

The course, in its current scope, is somewhat difficult for undergraduate students, as they must learn a broad set of techniques to implement the MeTube system, not all of which were taught in class. As a result, our plan for future offerings will include the following:

- reducing the features required for the MeTube project
- providing students with sample PHP and MySQL code relevant to the project
- providing partial implementation of some components of the MeTube system
- introducing HTML and PHP to students before advanced database issues are taught so that students can start implementing MeTube features earlier.

We are currently conducting a more thorough evaluation assessment of all τέχνη courses, with results comparing this new approach to the traditional approach forthcoming in a future paper.

# 7. ACKNOWLEDGMENTS

This work was supported in part by the CISE Directorate of the U.S. National Science Foundation under award EIA-0305318 and EAE-0722313.

# 8. REFERENCES

[1] Brown, A., Ash, D., Rutherford, M., Nakagawa, K., Gordon, A, and Campione, J. Distributed expertise in the classroom. In Salomon, G. (ed.). *Distributed Cognitions*, Vol. 7: 188-228. Cambridge Univ. Press, Cambridge, England, 1993.

[2] Burger, K. R. Teaching two-dimensional array concepts in Java with image processing examples. *SIGCSE Bulletin*, 35(1), 2003, 205-209.

[3] Cunningham, S. Graphical problem solving and visual communication in the beginning computer graphics course. *SIGCSE Bulletin*, 34(1), 2002, 181-185.

[4] Davis, T. A., Geist, R. M., Matzko, S., and Westall, J. M. TEXNH: Trial phase for the new curriculum, *SIGCSE Bulletin (Proceedings of SIGCSE 2007)*, March 2007.

[5] Duch, B., Gron, S., and Allen, D. The power of problem-based learning. Stylus Publishing, LLC, Sterling, VA, 2001.

[6] Fell, H. J., and Proulx, V. K. Exploring Martian planetary images: C++ exercises for CS1. *SIGCSE Bulletin*, 29(1), 1997, 30-34.

[7] Martinez, M. Designing intentional learning environments. *Proceedings of the 15th Annual International Conference on Computer Documentation*, ACM Press, 1997, pp. 177-178.

[8] Mordecai, B-A. Constructivism in computer science education. *SIGCSE Bulletin*, 30(1), 1998, pp. 257-261.

[9] Vygotsky, L. S. Mind in society: the development of higher psychological processes. Harvard Univ. Press, Cambridge, MA, 1978.

[10] Walker, S., and Fraser, B. Development and validation of an instrument for assessing distance education learning environments in higher education. *Learning Environments Research*, Vol. 8:289-308, 2005.

[11] Wang, J., Davis, T. A., Westall, J. M., and Srimani, P. Work in progress – MeTube: A novel way to teach database to undergraduates. *IEEE Frontiers in Education*, 2009.

[12] Springsteel, F., Robbert, M. A., and Ricardo, C.M. The next decade of database course: Three decades speak to the next. *SIGCSE Bulletin*, 32(1), 2000, 41-45.

[13] Robbert, M. A., and Ricardo, C. M. Trends in the evolution of the database curriculum. *SIGCSE Bulletin*, 35(3), 2003, 139-143.

[14] Dietrich, S. W., Urban, S. D., and Haag, S. Developing advanced courses for undergraduates: A case study in databases. *IEEE Transactions on Education*, 51(1), 2008, 138-144.