Learning About Data Integration Challenges from Day One

Alon Y. Halevy
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195
alon@cs.washington.edu

ABSTRACT

I describe the format of the new version of an introductory database course that I taught at the University of Washington in Winter, 2003. The key idea underlying the course is to expose the students to some of the challenges that arise when working with and integrating data from *multiple* database systems and applications.

1. BACKGROUND AND MOTIVATION

I have been teaching the introductory database course at the University of Washington for five years, and although it has been moderately successful, I still remained unsatisfied. The quarter-long (10 week) course covers the typical set of topics: I begin with conceptual design and a taste of normal forms, and then teach SQL, including views, triggers and transactions. In the second part of the course I discuss XML, including its data model, basics of XPath and XQuery, and mapping XML to relational stores. Finally, I turn to some database internals, covering indexing techniques, query execution and elementary aspects of query optimization. During the course the students are assigned 4-5 homeworks, some of which requires working with a DBMS, and in the project component of the course, the students are required to choose a database application and build it using a DBMS, usually with a web-based user interface.

My main reason for dissatisfaction was that I felt the course was relatively light intellectually, and did not expose the students to the really hard problems facing data management today. As a result, it was hard to get the strong students in the class excited about the subject matter. The course also seemed to be of a relatively easy load. I was always envious of the Graphics faculty who work their students to tears, but the students rave about the course.[1] Overall, it was an ok, but ho-hum course, leaving a lot to be desired.

My thinking about changing the course was based on the following principles. First, the main goal of the course is to educate the students about the *use* of database systems, either in themselves or as part of more complex systems. I give the students a glimpse of the insides of the system, but the course is not about the internals. Second, to further that goal, I thought one of the main challenges the students are likely to face in the workforce involves multiple database applications: integration of data, data exchange, and sharing data with web services. Third, my past experience in teaching the course revealed that undergraduate students do get excited when they see that the material they are learning is close to the frontiers of research, and even more so when it is the research in my group.

Hence, I decided to introduce into the course some of the real-world challenges that people face while integrating multiple data management systems, and rearrange the course project along those lines. The complete details about the course can be found on its web site at www.cs.washington.edu/education/courses/cse444/03wi/.

2. THE REVISED COURSE

The major change to the course was in the project component. The project was divided into three phases, each taking roughly three weeks. The students were only told about the on-going phase, and not about the subsequent ones.

In the first phase, each student was asked to build a database application in one of three domains (essentially, the same thing they did in ten weeks in groups of three in previous offerings of the course, except that here the domains were chosen for them). The domains were inventory, billing and shipping. The specification described in text what their schemas should cover (and required them to model additional aspects not in the specification). They had to design and implement a database schema, populate the database with a few tens of tuples, and build a simple web-based user interface for a set of specified query types.

The second phase was the brutal one. I arranged the students in groups of three, one from each of the aforementioned domains, forming *companies*. In the next three weeks, the students were asked to do the following. First, they had to create a web site for shopping, where customers can choose from their products, select a shipping method, and arrange their billing. They had to build the web site using the databases from Phase 1 as stand-alone applications. Second, they had to create a *CEO workbench*, in which a manager can pose certain decision-support style queries that spanned the three databases (the queries were given in advance). Finally, the groups had to make available several web services

[1] Granted, it's hard to compete with people whose focus in life is producing pretty pictures, but I wanted some of the same experience in my course.

that accessed their databases.

During Phase 2, in order to simulate a true integration effort, the students were highly discouraged from making changes to their original schemas. If they still wanted to, they had to write a petition and justify that the cost of making the modification to the original database would outweigh the future benefits. I only received a handful of such petitions.

In the third phase, the 16 companies in the course communicated in a peer-to-peer fashion using web services. Specifically, each company made available a book-ordering web service. When a request came in for a book that was not in the company's inventory, the company would contact the peer companies to obtain availability and price quotes for the book. The company could then decide where to get the book from, and whether to add a middle-man's commission. To make sure there was reasonable exchange of books between companies, the companies' inventories were filled by the course staff (also using a web service). To simplify matters, this phase ignored the billing and shipping aspects of shopping.

The material presented in the lectures was changed only very slightly in order to best support the project. Most notably, I starting teaching SQL in the first week, because it was important for phase 1. I taught schema design only after the basics of SQL (and by that time, since the students had already designed one schema, they appreciated the material even more). In general, it was still a bit tricky to ensure that the students know enough in order to perform Phase 1 in time, while keeping with the flow of the course. Slightly later in the course I gave a lecture on web services in time for them to be used for phase 2. In general, the course web site provided as much support and examples as possible so the students could focus on the relevant parts of the project.

3. LESSONS LEARNED

Overall, judging from the students' reactions and course evaluations, the course was a success. The students knew they were in for a 'special treat' from the outset, and were therefore in the right spirit. Unlike previous offerings of the course, some significant class time was spent discussing the project and the challenges facing the students.

Phase 1 of the project went off very smoothly, and with no complaints. The only concern there was that ten weeks of work in previous courses was compressed into three, but that turned out not to be an issue.[2]

Phase 2 was the most challenging and work intensive. As it turned out, having the heaviest phase in the middle of the quarter rather than towards the end was a feature. By far, the challenges the students struggled most with were on the schema interoperation level, since the individual schemas were designed before the students knew they would have to integrate them.[3] In particular, there were mismatches between what keys represented in the different databases; in some cases, data that was represented as multiple columns in one database was represented as a single column in another, and in some cases, some information needed for integration was missing from one of the databases. In contrast, somewhat to my surprise, query processing across multiple databases was not a great concern to the students. For the mostpart, the groups implemented a simple version of dependent joins, where they got a result set from one database, and fed each of the results in the appropriate query to the other database. Given that the course lectures had not discussed query processing strategies at that point, this was a fortunate outcome.

Phase 3 was a bit scaled back after I saw the effort the students put in for the previous phase. Working with web services was a good experience for the students (we were using .NET out of convenience). The main problem in Phase 3 was that as the students were developing their services, they were not able to test it against other groups who were also under development. Clearly, I should have built an example web-service with which the students can test during development. Like Phase 2, there were additional schema incompatibilities that came up as the groups had to conform to a set of web services.

All in all, the project generated very good discussions in and out of class. The students felt that they were dealing with real-world issues, and found it easier to get motivated for the project. One of the topics that generated a lot of discussion was how to design schemas *for* possible future integration. This, of course, is still an open research issue.

4. CONCLUDING REMARKS

This was the first offering of this course (the next will be in January, 2004). Clearly, there is a need for a lot of fine-tuning of the details, but I felt good about the course and was able to impart my enthusiasm to the students. Perhaps, most importantly, I can now add to my collection of quotes from student evaluations (that already includes comments such as "smells good" and "cool shades") the quote: "Halevy rocks!".

5. ACKNOWLEDGEMENTS

This course would not have been possible without two incredibly dedicated teaching assistants: Xin (Luna) Dong (a Ph.D student), and Eric Chu (currently a graduate student at the University of Wisconsin). My more senior students, Jayant Madhavan and Rachel Pottinger were generous in offering some advice throughout the course, as well as helping with designing the project.

[2] It helped that the infrastructure for such projects is already well established.

[3] In future offerings of this course, the students will probably have heard about my scheme. In order to ensure that the schemas are developed independently, I will have the students design them as an in-class quiz.