

# Use of Large Databases for Group Projects at the Nexus of Teaching and Research

Richard C Thomas  
Computer Science & Software Engineering  
The University of Western Australia  
M002, 35 Stirling Hwy, CRAWLEY 6009, Australia  
+61 8 6488 2733

richard@csse.uwa.edu.au

Rebecca Mancy  
Centre for Science Education, Faculty of Education  
The University of Glasgow  
Glasgow G3 6NH, Scotland  
+44 7967 730987

rebeccamancy@dcs.gla.ac.uk

## ABSTRACT

Final year, group (capstone) projects in computing disciplines are often expected to fill multiple roles: in addition to allowing students to learn important domain-specific knowledge, they should reinforce computing and software engineering concepts and provide for the acquisition of transferable skills. For motivational and pedagogical reasons, it is clearly preferable that such projects respond to real needs, be those in research or industry. We describe two student projects based on a large repository of usage data and integrated into a course in Professional Computing. These projects fulfilled the objectives outlined above and were closely linked to the research of the first author. We suggest that similar projects based on large databases may offer a transferable paradigm for others to follow. Finally, we outline some important elements for a successful group project based on a large database.

## Categories and Subject Descriptors

H3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *Query formulation*. H2.8 [Database Management]: Database Applications – *Data mining, Scientific databases*. K3.2 [Computer and Information Science Education]: *Computer Science Education*

## General Terms

*Measurement, Experimentation, Human Factors.*

## Keywords

*GRUMPS, SQL, stream data, keystroke times, capstone course.*

## 1. INTRODUCTION

Capstone courses are typically compulsory, final year courses that provide a significant integrative, educational experience [1]. They may, for example, take the form of a large group software engineering project. They have long been recognised as

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
*ITiCSE '04*, June 28-30, 2004, Leeds, UK.  
Copyright 2004 ACM 1-58113-000-0/00/0000...\$5.00.

important in the teaching of computer science and indeed many professional accreditation requirements state that such courses and projects must be undertaken. However, finding suitable projects can be difficult, as multiple requirements have to be satisfied. Project work should be a learning experience for students, both in terms of the actual knowledge gained and consolidated, but equally in that it should allow them to acquire generic skills such as teamwork and time management. Furthermore, projects should be seen to be useful, either to research or in a real-world, commercial situation; contrived assignments usually result in a lack of student motivation.

On the technical side, data repositories are increasing in size and also in the range of applications. For example Terabyte databases are being accumulated for the Sloan Digital Sky Survey project in astronomy [4]. In human computer interaction relational databases, rather than flat files, are being used to store log data. We have found that the data thus generated is useful for group projects in a final year Professional Computing course. This paper describes such projects and the nature of the benefits to students, as well as the interplay between research and teaching that has given rise to this work.

## 2. LARGE DATASETS

Recent advances in technology and the falling cost of data storage mean that it is becoming feasible to log and use large quantities (gigabytes) of recorded data. This is apparent in the research domain, for example in the Sloan Digital Sky Survey project in astronomy [4], but is similarly true in the commercial arena, where companies retain such information as full transaction details, with the aim of analysing customer purchase patterns [9].

As large databases become more common, it is important that universities offer students the opportunity to gain the understanding, skills and techniques necessary to work in these domains. More generally, use of large datasets allows students to appreciate the importance of optimisation, as system capacity becomes a limiting factor in execution time. It is therefore desirable that students work directly with datasets of this order of magnitude.

## 3. GRUMPS – A LARGE DATASET

### 3.1 Introduction to GRUMPS

The Generic Remote Usage Measurement Production System (GRUMPS) [13] is being developed at Glasgow University [2]. The goal is to provide general purpose mechanisms for the

capture of user actions for subsequent analysis and mining. An initial application area is expected to be educational research. For the present work a reliable, low complexity version called GRUMPS-Lite was used. It includes a User Action Recorder (UAR) that runs under Windows, collecting low-level actions such as mouse clicks and keystrokes. There is a transport mechanism to harvest collected data and store it in the repository, a SQL Server database. The repository schema is extremely simple as shown in Figure 1(a); it has proved to be very robust for data collection. The XML fields store tagged data appropriate to the circumstances, such as is shown in the Figure 1(b) for a Change Window Focus event.

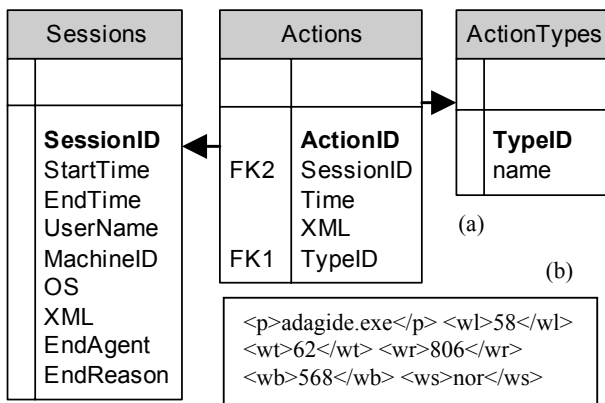


Figure 1(a). The repository schema. (b) XML for a change of window focus event.

### 3.2 The dataset

In January 2003, ethical clearance was granted to collect data from the first year, Level 1 Ada programming laboratories in Computing Science at the University of Glasgow. Participating students signed a consent form, were assured that their anonymity would be upheld and were informed that for ethical and security reasons, full keystroke information would only be recorded while in the AdaGide tool used for Ada programming. Summary statistics for the collection are shown in Table 1.

Table 1. Summary Statistics for the Level 1 Database

Period of collection	6 weeks from February 2003
Number of participants	141
Hours of interaction	1767
Number of sessions	2655
Number of actions	4.7 M
Number of machines	84
Size of database	~2GB

## 4. GRUMPS DATA MANIPULATION

### 4.1 Research questions

The authors' research questions were concerned with aspects of the data such as typing speeds and set-up times [12]. Other participants of the GRUMPS project were interested in aspects such as students' persistence with programming tasks. In order to answer these questions, the data had to be extracted in such a

way as to be immediately meaningful in terms of the research question, as such requiring a great deal of manipulation.

### 4.2 Data manipulation in the database

Research groups such as ours have come to realise that extracting the data in its raw form into flat files is inappropriate for work with large datasets. In addition to the cost of transferring the data from the database to a file, tools such as *grep*, designed for use with flat files, process data sequentially and the time taken grows rapidly with the size of the dataset [3]. Database technologies, developed over four decades, have the capacity to work with data in parallel and are therefore much faster as datasets grow.

### 4.3 Our work with the database

The authors worked initially on generating summaries of the data for analysis by a psychologist [11]. For the reasons outlined above, the decision was therefore taken to process the data within the database as far as this was possible. However, doing so brought new challenges. In order to calculate durations, time differences between actions were required. In sequential files, sequence is recorded implicitly and time differences are thus calculated easily by looping over the file.

In a database, actions are carried out in parallel on sets of (essentially unordered) data. This can be circumvented through the use of cursors allowing line-by-line treatment, but this marks a return to sequential handling and is notoriously slow in databases. Relational techniques were thus developed to optimise the execution time of the data manipulation.

These techniques included calculating durations using self joins, determining the context (usually the application used) for each action and expanding the XML fields used to record various attributes of the actions. The use of indexes and normalisation strategies was tested to compare performance. Some of these techniques required the use of T-SQL, a procedural language extension to SQL [16].

Subsequently it became clear that certain SQL concepts were encountered by others in the processing of stream databases [5]. Following the success of this work we felt we had enough knowledge to be able to supervise student projects using T-SQL.

## 5. GRUMPS & PROFESSIONAL COMPUTING

### 5.1 Professional Computing

The Professional Computing 307 unit (course) was developed in the then Department of Computer Science at The University of Western Australia in response to the accreditation requirements of the Australian Computer Society. All students who major in the department must complete the unit, taking a quarter of a full semester. The syllabus addresses ethics and other professional topics, but the main component is a capstone group project [1]. Assessment in the unit is via an essay plus the group project [14].

### 5.2 The Group Project

The project is taken by a team of 5 to 7 students, who may or may not be randomly selected. Students within a team are very heterogeneous; they have widely differing abilities and will

have chosen from a variety of optional units as part of their degrees.

There are four Deliverables spaced through the semester: preliminary requirements; project plan, use cases, object and dynamic models, and acceptance tests; the completed system, testing documentation and a time analysis; and final presentation.

Each team is allocated to a Client, usually an academic whose job is to act as a Client in the definition of requirements and acceptance tests. The project is decided by the Client, as in real life. However, unlike most coursework, the project is not precisely defined as this is believed to be useful experience for the students. Usually the Client does not tutor the team, although some technical support may be necessary. For instance, in the current projects, considerable knowledge about the database and appropriate SQL was shared with the students.

The teams each receive three hours of mentoring during the semester. We are fortunate that Motorola, who have a development site adjacent to the campus, volunteered to do this. The feedback from the teams is very positive. Motorola benefits in getting to know all the students completing majors.

The project does, therefore, have some features that mimic the commercial world. One way in which this is not completely possible is in scale, as time is limited by the semester length and other units. On average each student is expected to commit about 70 hours to the project, making around 500 per team. This is by far the largest group project most students ever undertake at university, including engineering majors.

We considered that work on databases such as those generated by the GRUMPS software would be suitable for student projects. Similarly, Szalay [10] has remarked that the SkyServer database could be used to teach students about SQL and real, large databases. We have found that the GRUMPS repository does give students a chance to experience databases on a scale well beyond introductory SQL exercises and coursework. Two teams have now completed such projects, as reported next. The first of these two projects took place before the authors had carried out the work in section 4.3, the second afterwards.

### 5.2.1 Team G

The 6 members of Team G were asked in the second semester 2002 to develop a Java application to analyse keystroke and mouse activity rates. They were given a version of the UAR and collection software, told to collect some data and find out about activity rates, especially typing (tapping) speeds. Considerable effort went into the algorithmic definition of these quantities given the provenance of the data. At this time they were given some advice on SQL but otherwise left pretty much to their own devices. It was always assumed they would export some data for analysis in Java.

The SQL required was limited. Usernames were first obtained from the Session table for display through the Java application window. For subsequently identified usernames, all relevant actions were selected from the Action table. The team found the ordering of this data quite hard.

The task was achieved quite effectively and the team built a neat JDBC interface and automatically invoked Excel to show the results. The project product was not perfect, although it was a good prototype from which the requirements became clearer. Accordingly, given the quality of the students, two of them volunteered to do extra paid development work in the following vacation. The results have been quite useful and the final product, slightly modified by the first author, has been installed in a European research laboratory [6]. This is a particularly strong outcome for a student group project and is a bonus on their CVs. Their professional computing education has been shown to fit the context of the real world, albeit indirectly.

### 5.2.2 Team C

The 7 members of Team C worked on another project a year later in 2003. An important difference was that the authors had spent considerable effort on the Level 1 Data and had successfully used T-SQL for implementation [11]. Thus the level of technical support offered to this team was much greater than for the previous project and the requirements for the data analysis had matured. The team was asked to provide:

- Table of cleaned characters
- Table of digraph timings
- Tables of all words typed and of their occurrences
- Graphs of learning curves of any suitable table or part thereof
- Graph of interkey timing during a typing run
- A regular expression search facility of the cleaned characters

The level of SQL needed here was far greater than before. The XML had to be unpacked and appropriately coded to clean the characters. The digraphs required an understanding of self joins and performance issues required attention.

Again the project products have been satisfactory and indeed are usable as is, for example in Figure 2 the interkey times in milliseconds are displayed for a typing sequence.

The requirements have been refined as a result and vacation work has been offered for polishing the software. The students seem to be pleased with this achievement and its potential.

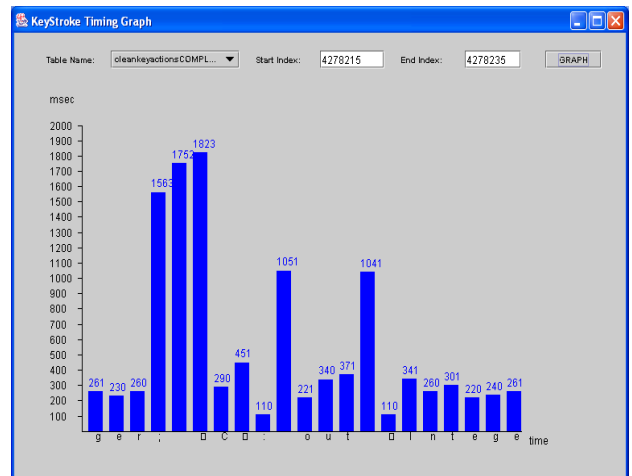


Figure 2. Interkey times typing “ger; C: out Intege”

## 6. DISCUSSION AND CONCLUSION

### 6.1 Learning Outcomes

The learning opportunities were substantial, as evidenced by comments in a special post-course questionnaire. For both teams of students this was their first exposure to anything other than a 'toy' database. Various authors have noted that aspects such as joins and grouping clauses cause problems for students using SQL [7][8]. Indeed, the queries were found to be challenging and the students had not met stored procedures in T-SQL before. The JDBC experience was new to them too, although the principles were not.

Both teams were initially inclined to export as much data as possible, but by the end of the project they had discovered how much could be handled in the database. This helped the students to realise the importance of design; for example, they began to appreciate the possibilities of a frontend/backend architecture and its implications for flexibility. Some also commented on the appropriateness of different paradigms and languages within this framework. Most of the students did not really develop knowledge of indexing and as such were hampered by performance issues. However, this allowed them to understand the importance of optimisation, even if they did not necessarily understand that there are tradeoffs. More generally, some students also report a better awareness of possible technical problems and employment opportunities in commercial database development.

In addition to knowledge specifically oriented towards the technical side of development, students also consolidated more general notions in computer science such as determining a precise project specification from loosely expressed requirements, the value of good documentation and the distinction between verification and validation. Many equally felt they had learnt a great deal about client-developer relations and interactions within the group.

The participants re-used techniques but generally not code, as they felt they "don't learn much in doing so", and would "prefer to write it myself and [...] really UNDERSTAND what the code does". Team G had to make use of the UAR and repository design, and also were influenced by the precursor to GRUMPS that had been used at UWA in 2001. Team C adapted for their own ends some of the JDBC and other Java from Team G. They also drew heavily upon the stored procedures and cleaned tables developed by the authors. The authors themselves had drawn on a few SQL clichés displayed in a prior MSc project in 2002. Finally there has been synergy between Team C, via the author, and a current MSc project at UWA. Therefore a strong interplay has been evident at the nexus between the GRUMPS research and teaching.

### 6.2 Discussion

With ever larger databases emerging as the result of data capture, we feel that it is important to give students the opportunity to work with datasets of this scale. Understanding in these areas is useful both commercially and within research, whilst more generally assisting students in appreciating various aspects of database design and use.

Our experience indicates that projects such as those outlined above have been highly suitable both for meeting these needs, as

well as the broader aims of the Professional Computing course. The skills developed by the project participants have been wide-ranging and have extended to knowledge of information specific to databases, as well as general computing science practice such as re-use and the importance of rigorous testing.

From a motivational point of view, the outcome of the projects was strongly linked to the first author's research agenda and we believe that our own enthusiasm must in turn improve the student experience. Furthermore, this aspect of genuine discovery was a motivating factor since the students were interested to see the results that would be generated by their tools. As a testimony of the students' own enthusiasm towards the projects, it is worth noting that several of the students from both groups have been keen to develop the project further and are about to commence a summer placement.

It is similarly encouraging to the students that their product has met with appeal beyond the author's own interests and has been installed in a European research laboratory. Because the software products may be useful, care has been taken to deal with the Intellectual Property Rights properly. This example serves as an illustration of the use of the Professional Computing syllabus material in a real-world context.

Several factors made it reasonably easy to set requirements that were realistic and attainable for a group with a very mixed set of skills, abilities and motivations. Those students who had already studied a course on database technologies worked directly with the database. Extra learning was managed to ensure that it remained tractable, whilst our own recent experience with the database and T-SQL meant that we were aware of many of the pitfalls so could carefully advise the students of these. For other members of the group, a series of integrated requirements was devised that capitalised on other skills, in our case mostly concerned with the user interface; for example, graphing of keystroke times. This is comparable to real-world situations where database specialists develop in SQL and related technologies (back-end) whilst others build other aspects of the application such as the interface (front-end).

In order to create similar projects in other universities, access to a large database must be obtained. The projects described here were carried out using databases generated by GRUMPS-Lite and we would be pleased to supply this tool to academics in other universities so that they can collect their own usage data. Also, other large databases are available on line (for example the SkyServer database [15]) and many more are being developed in various areas of research and commerce. As such, projects using large databases lend themselves to collaboration with other research departments and industry.

For planning, and during the project itself, it is necessary to appoint a supervisor who has a working knowledge of SQL and who, ideally, is familiar with the database. A research question, which can be answered using the information in the database, is posed by a 'client' (who may or may not be the supervisor). Negotiations between the supervisor and the client lead to the elaboration of a capstone project which is both realistic and which addresses the client's question. The supervisor recommends how to split the project into semi-independent parts, according to the students and the course in question. This seems essential as the potential pitfalls in the database side may not be solved by brute force, so the right person and approach is

required. It is important to ensure that the problem itself is easily understood and that the level of work required does not impose an initial learning period that is too long. Also, the required outcome from the early stages should be clear products or tests, although later sections of the project can be left more open.

Students have generally commented very favourably on the value of mentoring reporting that discussions concerning project processes, including milestones and goals, were particularly beneficial. Mentoring also covered issues such as group dynamics and project management, as well as giving insights from a business perspective, all seen to be valuable. However it may be that some students would have gained more from mentoring if the unit co-ordinator had discussed it more beforehand as it is so different from the teaching and learning experiences to which they are accustomed.

### 6.3 Conclusion

We have explained the growing importance of large databases both in research and industry and described the GRUMPS system as an example of such a dataset. This database of student actions has been used as a basis for projects for the Professional Computing 307 course.

Two of these projects are described and are shown to be particularly well suited to meeting the needs of the Professional Computing course, both in terms of gaining specific knowledge and transferable skills, but also for student motivation and real-life applicability. In our case, the projects have played a key role in the effective interplay between research and teaching, contributing to advance the research project from which they originated. The students have found the projects challenging yet rewarding and they have kindled their interest in database technologies.

We believe that projects based on large databases have the potential to be valuable for all participants, and represent a transferable paradigm for others, either using data gathered with GRUMPS-Lite, the online SkySurvey database [15], or alternatively in collaboration with researchers or companies who have generated similar databases.

### 7. ACKNOWLEDGMENTS

We are particularly grateful to the CS1P course student participants from Computing Science, Glasgow University 2002-3; Quintin Cutts and Rob Irving; Murray Crease, Huw Evans, Phil Gray, Steve Draper, Gregor Kennedy. The GRUMPS team gratefully acknowledges the funding provided by the UK's EPSRC (GR/N381141). UWA PC307 teams G and C students: Peter Bagas, Francis Barber, Ben Brearley, Carl Butcher, Leigh Carey; Chris Harris, Raymond Hon-man Yuen, Hayden Albrey, Christopher Tzehong Jee, Hernani Binti Nahrawi, Mohammad Yaqoob Siddiqui, Shen Zhang.

### 8. REFERENCES

[1] Clear, T., et al., eds. ITiCSE 2001 Working Group Reports - Resources for Instructors of Capstone Courses in

- Computing. SIGCSE Bulletin, ed. J. Impagliazzo. Vol. 33. 2001, ACM: New York. 93-113
- [2] Evans, H., Atkinson, M., Brown, M., Cargill, J., Crease, M., Draper, S., Gray, P. & Thomas, R. (2003). The Pervasiveness of Evolution in GRUMPS Software. *Software: Practice & Experience*, 33 (2), 99-120.
- [3] Gray, J. (2003) Online Science: The World Wide Telescope as a Prototype for the New Computational Science, talk at NeSC, Edinburgh, July 4, [Online] Available: [http://www.research.microsoft.com/~Gray/talks/WWT\\_ISC\\_2003.pdf](http://www.research.microsoft.com/~Gray/talks/WWT_ISC_2003.pdf) [25 Sept. 2003].
- [4] J. Gray, A.S. Szalay, A. Thakar, P. Kunszt, C. Stoughton, D. Slutz, J. vandenBerg (2002) Data Mining the SDSS SkyServer Database. *Distributed Data & Structures 4: Records of the 4th International Meeting*, pp 189-210.
- [5] Golab, L. & Ozsu, M.T. (2003) Issues in data stream management. *SIGMOD Record*, 32 (22), 5-14, June.
- [6] Karahasanovic, A., Fjuk, A., Sjöberg, D., Thomas, R. (2004) A Controlled Experiment to Evaluate the Reactivity and Usefulness of Data Collection with the Think-Aloud Tool versus Classical Think-Aloud Method. *IRMA 2004 Conference*, New Orleans, USA, May 23-6
- [7] Kearns, R., Shead, S., Fekete, A. (1997) A Teaching System for SQL. *Australasian Computer Science Education ACSE'97*, ACM Press (1997) 224-231
- [8] Mitrovic, A. (1998) Learning SQL with a Computerized Tutor. *SIGCSE Bulletin* 30 (1), 307-311
- [9] Dan Orzech (2003), Rapidly Falling Storage Costs Mean Bigger Databases, New Applications available online at <http://www.cioupdate.com/trends/article.php/2217351>
- [10] Szalay, A. S., Gray, J., Thakar, A. R., Kunszt, P. Z., Malik, T., Raddick, J., Stoughton, C., VanderBerg, J. (2001) The SDSS SkyServer – Public Access to the Sloan Digital Sky Server Data. *ACM SIGMOD 2002*, MSR-TR-2001-104
- [11] Thomas, R.C., Kennedy, G.E., Draper, S.D., Mancy, R., Crease, M., Evans, H. & Gray, P.D. (2003) Generic usage monitoring of programming students. *ASCILITE 2003 Conference*, Adelaide, 7-10 December, 715-9
- [12] Thomas, Richard C. (1998) Long term human-computer interaction. Springer Verlag.
- [13] The GRUMPS Research Project. [Online]. Available: <http://grumps.dcs.gla.ac.uk> [16th June 2003].
- [14] PC307 Professional Computing <http://undergraduate.csse.uwa.edu.au/units/230.307/05Nov03>.
- [15] Skyserver online at <http://skyserver.sdss.org>
- [16] Transact-SQL Overview [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tsqlref/ts\\_sqlcon\\_6lyk.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tsqlref/ts_sqlcon_6lyk.asp) Accessed 17Aug03