GENERIC USAGE MONITORING OF PROGRAMMING STUDENTS

Richard Thomas
School of Computer Science & Software Engineering
The University of Western Australia, Australia
richard@csse.uwa.edu.au

Gregor Kennedy
Biomedical Multimedia Unit
University of Melbourne, Australia
gek@unimelb.edu.au

Steve Draper[2], Rebecca Mancy[3], Murray Crease[1], Huw Evans[1] & Phil Gray[1]
Department of: [1] Computing Science [2] Psychology [3] Eduction
University of Glasgow, U.K.
{murray, huw, pdg, rebeccamancy}@dcs.gla.ac.uk, s.draper@psy.gla.ac.uk

Abstract

It is becoming feasible and practical to monitor the generic computer usage of students for extended periods, recording low level actions such as mouse clicks, typing and window changes. This paper presents a case study on the deployment of GRUMPS technology during a period of six weeks when 4.7 million such actions were collected from 141 first year university students learning Ada programming. We suggest our approach can be characterised as REDDI, Rapidly Evolving Digitally-Derived Investigations. Data preparation and cleaning is noted as a bottleneck in generic data collection, but seems less so as techniques are developed and understood.

Keywords

usage monitoring, programming laboratory, data repository, data preparation, first year student, GRUMPS, REDDIs, computer science

Introduction

With the pervasive use of electronic media in teaching and learning in higher education, electronic records of students' learning activities are increasingly being seen as an attractive research and evaluation tool (Evans et al 1999; Judd & Kennedy, 2001; Peled & Rashty, 1999; Phillips et al 2002). In this paper we argue that collecting generic data about students' interactions with software is feasible and discuss the GRUMPS architecture for achieving this. We present a case study from a course in first year programming that demonstrates these generic techniques, and examine emergent issues associated with preparing this type of data for both teachers interested in evaluation, as well as educational researchers.

REDDIs and GRUMPS

The Generic Remote Usage Measurement Production System (GRUMPS, 2001) is being developed at Glasgow University (Evans et al, 2003). The GRUMPS project aims to use current and emerging technical possibilities to develop a qualitatively new level to what used to be the simple and limited business of the collection and use of logfiles e.g. of logins, command calls, web page visits, etc. We call these REDDIs: Rapidly Evolving Digitally-Derived Investigations. They concern explorations where the question evolves as much as the answer. Calling them "investigations" rather than "experiments" underlines that they may or may not have a prior hypothesis, and do not usually control or manipulate the circumstances being studied. These investigations depend upon computer-derived data, and the possibility of collecting it easily: they are as much about what it is possible to learn from these sources as about answering prior questions. To call them "digitally-derived" is to further emphasise their orientation to a particular technical opportunity. On the other hand, and in contrast to their logfile precursors and to data mining, it is now possible to make rapid changes to actual data collected, in response to new interests, guesses, and hypotheses by the investigator. Hence such investigations are potentially "rapidly evolving", and so do not depend on a static dataset collected before analysis began.

The case study we present here relies on data collected from the User Action Recorder (UAR). It runs on Windows workstations and is visible to the user on the task bar. It has the potential to monitor all window activation, mouse and keyboard events, but can be restricted by the user or investigator when required. Restrictions on collection allow the user control over the level of privacy.

The repository is designed for flexibility and initial rapid storage in a relational database system. The database software is very simple, containing two main tables for actions (events) and sessions; see examples in Figure 1. A feature is the use of XML attributes (columns) in order to allow storage of a variety of information according to, for example, the type of action, all within a uniform structure. The repository design has proved to be very robust, and well adapted for rapid collection of large volumes of data (Thomas et al 2001). Thus, collecting large volumes of low level data has become technically feasible. Retrieval is an area of difficulty and a number of researchers have noted that generic data collection often generates unwieldy and unmanageable data from which it is difficult to extract meaningful information (Misanchuk & Schwier, 1992; Reeves & Hedberg, 2003).

Study at Glasgow

To investigate the use and usefulness of the GRUMPS software we applied it in a first year programming course at the University of Glasgow. Students in the compulsory CS1P unit took fortnightly, two hour laboratories designed to practise and develop programming skills. Students used the programming environment *AdaGide* to complete practical programming problems, tasks and exercises. We had a broad idea of the required GRUMPS-derived interaction data and integrated this with other questionnaire data, previous academic results and marks for CS1P.

Having obtained Faculty Ethics Committee approval, student participation was sought in a lecture and subsequent tutorials. Monitoring commenced on February 10, 2003 and lasted for about 6 weeks. Exactly 141 students participated and 2655 UAR-sessions were recorded from 84 machines. There were 4,701,311 actions over 1767 user hours of interaction – about 2 hours per student, approximating the nominal lab time. An abbreviated session record is shown in Figure 1 with a window focus action in that session.

SessionID	StartTime	EndTime	UserID	MachineID	UARExitReason
5253	1045142859063	1045144730173	87858268	bo715-11-02	User Logged Out

ActionID	Session	Time	XML	Type
1251079	5253	1045143002268	adagide.exe <wl>58</wl> <wt>62</wt> <wr>806</wr> <wb>568</wb> <ws>nor</ws>	9

Figure 1: Above, an abbreviated Session record and below an Action record belonging to this session. The action, of type 9, changed the window focus to the adagide.exe *process. The XML also shows the co-ordinates of that window and that its size was normal rather than minimised or maximised. Times are milliseconds from January 1, 1970. Usernames are anonymised.*

Data Cleaning, Preparation and Output

Figure 1 illustrates an example of what can be conveniently collected, while Figures 2 and 3 show a version of what the investigator wanted in order to even begin thinking and deducing about student-oriented issues. There was a major work phase to convert the collected data from one to the other.

The repository took about 2Gbytes of data which was cleaned and prepared for use by researchers. The Action table contained a plethora of detail such as the process e.g. *adagide.exe*, the screen location of every mouse click and the size of the window, a timestamp of each key depression and so on. The task of data preparation was to reduce this to two summaries.

The data preparation phase of the investigation is a large bottleneck, but substantial reduction of this has been achieved. Data preparation took about a full person-month, but this might be reduced given a more experienced T-SQL programmer. A number of techniques were developed: to calculate durations of actions with self-joins; to find the context of an action; to expand and normalise relevant parts of the XML data. A crucial step was to understand when and how to index tables to optimise performance: proper index strategies achieved more than an 80-fold reduction in execution time. Similarly, data preparation would have been facilitated by an underlying structure which better utilised the extraction capabilities of the database, and it is clear that there is an important tradeoff between the optimal model for data collection and for data preparation.

Sess	User	Sess Time	Date	IE Time	Ada Time	IE F	Ada F	Comp	Build	Run	Ada Pause
5253	87858268	1871	Feb13 13:27	271	1276	15	10	3	19	16	676
5612	87868109	10256	Feb19 16:56	3170	3	12	1	0	0	0	NULL

Figure 2: Two lines from the Summary Spreadsheet report on student sessions.

Examples of reports for an investigation are shown in Figures 2 and 3. The formats were based on outputs from previous studies with top-down data collection, which served as approximate requirements specifications for the data preparation phase of this study. Thus they demonstrate that the same sort of information can be generated from both top-down and generic, REDDI approaches.

Action	What	TimeOffset	…		
1250975	UAR.exe	2	…		
1250978	Explorer.EXE	8	1251704	Ada	954
1250983	iexplore.exe	16	1251705	Activity	957
1251031	Explorer.EXE	87	1251709	Build	961
1251032	eudora.exe	138	1251840	leave gexecute	1187
…					

Figure 3: Part of a more detailed report showing on-task and off-task behaviour.

Things we have learned

There are several things we think we have learned. The first is our demonstration of the collection of large volumes of data "bottom-up" i.e. using a method independent of the particular use that will be made of the data. This relied on a data model with: Persistence i.e. no allowance for updating records, just adding new ones; and minimal fixed attributes by using XML fields – the semi-structured data illustrated in Figure 1.

An important insight was that there are two stages in data preparation (for a REDDI in GRUMPS), not counting the tedious but relatively trivial issue of data cleaning. Step 1 is to find a representation in which data is immediately meaningful to the investigator with respect to the hypothesis (however weakly formulated). Figure 2 gives an example. Step 2 is the process of actually preparing the data to fit the above representation. In practice even Step 1 is a substantial cost in both computation time and programmer time.

Unsurprisingly, with hindsight, we also discovered the need to record event sequence explicitly. This is implicitly given in a sequential file model, but not here. The transport mechanism for data from collection to repository should also preserve sequence. Furthermore, our systems are designed to deal with large volumes of data, perhaps 20 or 50 million actions in say 20 Gbyte databases. At these volumes it does appear appropriate to use a database system as the repository and export subsets as required. Gray (2003) has pointed out the rising time for *ftp* and *grep* (common utilities) as volumes grow.

Finally it is clear that new skills are required to use the generic approach effectively. In particular database design and programming are needed beyond those adequate to issuing traditional queries. The handling of stream data in the relational model is of current interest in other domains such as stock tickers and web logs (Golab & Ozsu, 2003).

Conclusion

Supporting REDDIs by database technology and bottom-up, generic methodologies has introduced new difficulties in comparison with our earlier approaches where data collection is carried out by hand-designing additions to the source code of the program to be monitored and data is stored in sequential files. Several of these difficulties have been outlined in this paper and techniques have been developed to overcome certain issues. Whilst further work is required to identify and resolve further difficulties, doing so will hopefully allow support for rapid changes in analysis of collected data. It will then be possible to apply usage data much more widely, and dependence on access to the source code and modification of the software being monitored will be reduced. Thus, analyses will no longer be limited to testing only hypotheses that were recognised before any data was collected and we are confident that usage data will become more generally attractive as an educational evaluation tool.

References

Evans, G., Dodds, A., Weaver, D. & Kemm, R. (1999). Individual differences in strategies, activities and outcomes in computer-aided learning. In R. Jeffery & S. Rich (Eds), *Proceedings joint conference of AARE and NZARE,* Melbourne, Australia.

Evans, H., Atkinson, M., Brown, M., Cargill, J., Crease, M., Draper, S., Gray, P. & Thomas, R. (2003). Pervasiveness of Evolution in GRUMPS Software. *Software: Practice & Experience*, 33 (2), 99-120.

Golab, L. & Ozsu, M.T. (2003) Issues in data stream management. *SIGMOD Record*, 32 (22), 5-14, June.

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 [25 Sept. 2003].

GRUMPS. (2001). *The GRUMPS Research Project*. [Online]. Available: http://grumps.dcs.gla.ac.uk [16th June 2003].

Judd, T. & Kennedy, G. (2001) Extending the role of audit trails: A modular approach. *Journal of Educational Multimedia and Hypermedia*, 10 (4), 377-395.

Misanchuk, E.R. & Schwier, R. (1992). Representing interactive multimedia and hypermedia audit trails. *Journal of Educational Multimedia and Hypermedia*, 1(3), 355-372.

Peled, A. & Rashty, D. (1999). Logging for success: Advancing the use of WWW logs to improve computer mediated distance learning. *Journal of Educational Computing Research, 21*(4), 413-431.

Phillips, R., Baudains, C. & van Keulen, M. (2002). *An evaluation of students' learning in a web-supported unit of plant diversity*. Proc 19th ASCILITE Conference, Auckland, New Zealand.

Reeves, T.C. & Hedberg, J.G. (2003). *Interactive Learning Systems Evaluation*. Englewood Cliffs, NJ: Educational Technology Publications.

Thomas R.C., Cashman, P., Edwin, D. & Ritchie, M. (2001) Investigations into Privacy and Other Aspects of a Real-Time Distributed Marks Collection System. OzCHI 2001 Conference, Fremantle.

Acknowledgements

We are particularly grateful to the CS1P student participants; also Quintin Cutts and Rob Irving. Richard Thomas thanks the National eScience Centre, Edinburgh. The GRUMPS team gratefully acknowledges the funding provided by the UK's EPSRC (GR/N381141).