Remarks on the Efficiency of Information Systems

András Keszthelyi

Óbuda University, Budapest, Hungary
keszthelyi.andras@kgk.uni-obuda.hu

Abstract: In Hungary there exist two big and well-known scholar information systems (SIS). Both of them have become part of everyday life in the administration of higher education. These SISes have made quite a long journey during their evolution. Even so, they have some annoying disadvantages even today; e.g. they serve too few users at a time too slowly in the case of critical activities, such as during registration for courses or for examinations. There are many circumstances which can result in such poor efficiency. In this paper I try to investigate the role of three-level data modelling in an indirect way. I, with a colleague of mine at Budapest Tech, planned and executed a measurement which shows that (much) better performance can be reached based on a 'good' data model, even in a poorer environment.

Keywords: database efficiency; scholar information system

1 Scholar Information System - SIS

The administration of the scholastic records of students has become a great task for today, one which would need great resources if the administration were done in the historical manner, that is, using a paper-based system. This is due to not only the increasing number of students but also the complexity of the credit system as well.

At this point, it is necessary to clearly see the difference between a 'database' and an 'information system'.

A database is a finite amount of data stored in a suitable manner in order to manage the needed administrative tasks in the shortest time possible. There were no theoretical reasons for demanding that a 'database' should be computer-based, but for historical reasons, we do not use this expression for a paper-based filing cabinet. In a computer-based environment, a 'database' is a finite amount of data stored in a proper *structure* according to the data model. These data represent a) entities, b) the attributes of the entities, and c) the connections between the entities.

An 'information system' is more than the database itself. Of course the heart of an information system is a database. We need some or additional *infrastructure* and technical/administrative *people* to manage the database itself and to serve the administrative tasks or jobs and the users. We need some or more special *rules* in order to work in a correct way and to have the least possible number of errors as far as possible. These all can be called an 'information system'.

In the “old” times, the curriculum was well-defined, which was mandatory for all students in only one possible way. Nowadays, the curriculum is well defined, too, but there are a large number of possible ways to fulfill the requirements. Students themselves can decide the manner and timing of their studies. There is only one main rule, which is a logical one: a student is allowed to register for a course if the prerequisites of that course have been fulfilled; for example, one is not allowed to register for Mathematics II when he or she has not yet succeeded in Mathematics I.

According to the above, it is natural that the students select not only the subjects they want to study in a given semester, but they also personally choose one of the courses of that particular subject as well (if the number of the subscribed students to the given course is less than the maximum number allowed). Course selection can be a complex, iterative process until such time as the students are able to compile suitable timetables suitable for them. The registration for examinations has almost the same attributes as the above-mentioned registration for courses.

Storing the students’ results in their different subjects is simpler.

So, managing the administration by hand can scarcely be imagined. Almost everywhere this task is solved by computer-based information systems. Such a system can be called a 'Scholar Information System'.

1.1 Main Questions about a SIS

There are many questions that can be asked in connection with an SIS. These can be grouped in many ways. From the point of view of the end-users, the main questions are the following:

Is the system able to manage a large number of administrative tasks simultaneously? Is it able to serve all or nearly all the students in a given time period who want to or are obliged to do a task in the administrative field. This problem can occur in two typical situations: the first one is the registration for examinations at the end of the semesters; the second one is the registration for courses at the beginning of the semesters.

Is the system realistic? Does it work in accordance with real life? Firstly, does it know and serve the administrative rules of a given institute? Secondly (last but not least), are these rules themselves realistic? Are they in accordance with the rules of logic; are they practical?

Is the system ergonomic? How much time is needed to perform a given task? How many mouse-clicks are needed to perform the most frequent activities?

Can the system handle all the personal and scholastic data of the students in a secure enough way?

In this paper, I examine the first question mentioned above, the load-ability of a database that could be that of a SIS in a typically problematic situation. I planned and executed a quantitative measurement in order to determine how many administrative tasks can be performed almost simultaneously. I have chosen one of the two most critical activities: the registration for examinations. According to our everyday experiences and to a student questionnaire, this task usually forms a bottleneck.

1.2 Existing SISes

In Hungary there are two well-known scholar information systems that are used in higher education: the ETR (Egységes Tanulmányi Rendszer – the Uniform Scholar System) and Neptun.

We have been using Neptun for nearly a decade at Budapest Tech.

At the beginning there were serious problems even in the functionality, e.g. there was no possibility to administer if a student had a dispensation. The database was not able to cope with the load caused by registration periods, as it should have been expected. After nearly a decade, we are using the third main version of Neptun. Of course it has developed since then, but it has its own weaknesses in the field of load-ability even today. Our nearly ten thousand students are divided into different sets, and each set of students can start their registration on different days even today in order to lower the load on the database.

According to a student questionnaire created by this author in 2008, the most frequent problems observed by the students were: aborted connections, short timeout and slowness.

ETR has the same problems in efficiency, as can be read about even in the Hungarian-language wikipedia (http://hu.wikipedia.org/wiki/ETR).

These problems are widely known and these are the problems which make most people very angry in most cases. So it is reasonable to investigate the problem of the efficiency of information systems.

2 Efficiency of Databases

What can we call the efficiency of a database? It is the capability to cope with high loads. This capability is determined by several very different factors: the hardware environment, the software environment (the operating system, the relational database management system, the application programming language and tools, the application programs themselves), and the quality of the data model.

Of course, the influence of the hardware environment is very important. This is the first circumstance which comes into one's mind, but it must be declared that to increase the performance of the hardware in order to have a higher software performance is a 'brute force' method: the more money you have, the higher performance you will get.

There are more sophisticated and, of course, cheaper methods which result in higher software performance.

Let us look at the software environment. The operating system, the relational database management system and the application itself are the most important elements in this field. The first two can only be chosen from a given set based on various ratings of their most important technical and co-operational features. How some technical aspects influence the performance I investigated before and presented some years ago. [9]

In the case of the third element, the application, there are more possibilities to influence the performance. After choosing the programming language and tools, there are two main fields which determine the performance of the developed program(s). These are the quality of the applied algorythms and the quality of program coding. In the case of databases, the 'algorythm' has a more special meaning than generally: the quality of the data model is included as the most important; a necessary but not sufficient circumstance.

2.1 The Quality of Data Models

The main steps in developing an information system are: determining what is wanted as precisely as possible; data modelling (i.e. determining the data structure); and determining the functions to operate on the data structure. In the case of data-intensive systems, the data structure is more important and determines the functionality. [1] (p. 541)

So data modelling is the basis, one which is necessary but of course not sufficient for success. The basis is only a possibility on which a good information system can be constructed. In order to succeed, it is necessary to have three-level data modelling and planning, according to Halassy. [2] (pp. 28-33) These levels are the conceptual, the logical and the physical levels. The names of these levels, and even the “three-level” label, have been widely used, but in most cases without the

appropriate meaning. In the early times of databases, Codd wrote that even the SPARC committee of ANSI used these words without defining them precisely. “The definitions of the three levels supplied by the committee in a report were extremely imprecise, and therefore could be interpreted in numerous ways.” [3] (p. 33)

Unfortunately, the field of data modelling is not, and has never been, in focus. In previous times, at the beginning of relational databases, Peter Chen introduced his entity-relationship model [4], which can be considered the basis of data modelling. There are no books even today which discuss data modelling in a scientific way, except books by Dr. Halassy in the Hungarian language, and there are no books which discuss data modelling in its fullness. This is the second reason why I have investigated database efficiency and the role and importance of data modelling in this field.

As Dr. Halassy states, the conceptual level data model is the one in which are described the entities of reality, their properties and relations, or linkages in natural concepts and corresponding to reality. The logical level is the one where the data structure of the database is planned according to the circumstances and constraints of the technical aspects, accessibility and efficiency. Defining the exact type and size of the data elements, the way they are stored in the storage equipments, and the way they are accessed are described in the physical level plan.

There are general prerequisites of the quality of data models. At the conceptual level, a good data model needs to be understandable, unambiguous, realistic, full and *minimal*. [5] (p. 192) Of these properties, minimality is the one which can be precisely examined by mathematical tools.

Minimality is a very important property because redundancy is dangerous. If a data structure is redundant, the database built upon it needs (much) more storage. If it needs more storage, it will need more time to be handled. These problems can be solved by 'brute force', by quicker storage equipment and processors. The biggest danger of redundancy is the possibility of data errors: redundancy causes certain undesirable characteristics, the so called insertion, update, and deletion anomalies that can lead to the loss of data integrity. We can suppose that at least some of the experienced problems of the two SISes are rooted in model level errors. I am here focusing on efficiency.

I was considering whether the data model of the SIS used by Budapest Tech (Neptun) meets the above requirements. Of course I was not given the model documentation itself because it is a commercial software, so I had to try another, indirect way. I made a data model for such a scholar system, a data model which is considered to be good enough, at least by this author, to examine that one instead of the original one.

The prerequisite is that if I can reach better or at least not worse results in a poorer environment, I can state that the reason for the difference can be identified in the differences of the data models.

My concept was to identify a function which is critical from the aspects of the response time and of the number of concurrent users to be attended. There are two such functions in a scholar system: registration for examinations and registration for courses at the end and at the beginning of the semesters. In these two cases, nearly ten thousand students would like to use the system, and each one of the students needs to register for an average of about four examinations or about twelve courses.

I chose the first process, the registration for examinations. I made the conceptual data model carefully, as well as the logical and physical level plans based upon it. A colleague of mine implemented the plan and developed the part of the application which is needed to do some efficiency measurements. [7]

3 Questions about the Measurement

3.1 What to Measure?

I chose the registration for exams as a critical field to investigate, as was mentioned above. First it was necessary to decide what I wanted to measure in this field. The exact response time of each registration of each student? The number of retries and/or the response times? The number of successful and unsuccessful tries in a certain time-period? Do I need to make an ABC-assay and to rank the responses into three sets, one of them called 'very good', the other called 'acceptable' and the third one called 'poor'? At what values can I mark the boundaries of these sets?

3.2 Measurement Errors and Mistakes

There are numerous random factors which can and, of course, do influence the measuring. Let us look at at least some of them.

Since the measurement is done in a working computer environment, all the other possible activities of the operating system would be taken into account, e.g. saving data as a response to a given query in a local file needs some (a little but significantly greater than zero) time. This is a random error because the moving of the read-write heads of the hard disks and the puffer usage is unpredictable.

The network traffic which is not part of the measurement activities could be eliminated, closing the subnet for the time of the measurement, but even in this case, there are some factors at the ethernet level which could influence the measurement. This is a quasi-random error because it increases if the network traffic increases.

Last but not least, the measurement also influences itself. To measure some computer activities by computers needs one or more, more or less complex programs to run. These programs also need lesser or more resources, while the total amount of resources is a given constant.

Beyond the above-mentioned errors, there are observational and computational errors as well to cope with.

3.3 The Object of the Measurement

To measure the response times to four significant digits (in seconds) would be an interesting measurement task to plan and execute. In such a case, the correct handling of the above-mentioned measurement errors would not be an easy problem to solve.

Luckily it is not important to know the response times precisely. There are two important questions, which are the following: Could the response times be tolerated by the average student or not, even when a large number of students would like to be served? How many of the registration attempts are fulfilled?

In trying to answer these two questions, the influence of the above mentioned measurement errors are negligible. The borderline between the 'tolerable' and 'intolerable' time requisites cannot be defined precisely in a mathematical manner because it is the subjective opinion of the end-users, in this case the students.

Therefore, I decided to measure the average and the maximum response times and the successfulness of the registration attempts. If the response times and the number of the unsuccessful attempts are significantly lower than in the real system, even in a poorer environment, my above statement could be proven: a better three-level data modelling results in a better database application, in better response times, and in fewer unsuccessful attempts. In general: a correct three-level data modelling results in a growth in efficiency.

4 The Measurement

The test environment consisted of PC computers and free software. The database server had an Intel processor of four cores and 8 GB of RAM. It runs Linux as the operating system, httpd server Apache with PHP as an application interface between the users and the database, and MySQL as a relational database management system. Instead of a large number of workstations one simple but strong PC was used, one which had enough RAM to run the needed amount of offline browser wget. This circumstance has no effect on the measurement: the number of registrations are the same and each of them goes through the network.

The test database contained 8192 students, about as many as BMF's active students, four examinations for each of them. The number of places was one and a half times bigger than the number of the students' examinations. Registrations themselves were made by a PHP script index.html randomly for the test user currently called it via the offline browser of the test client workstation. Each test user registered a date for all examinations.

Normally the selection is done by the students themselves, sitting and thinking in front of their computer. From the point of view of the measurement, there was no difference between a date selection by a human student and a random date selection by a program. The circumstance that this date selection is done at server side increases the load on the server a bit, so the measured results are a bit worse than they would be in reality.

We logged the client system time at the beginning of the connection to the database server and when the response was saved to a local file by the offline browser. The server load was also logged. Test registrations were started almost simultaneously with a two second pause after every one hundred starts. The settings of the offline browser wget were: max 4 retries, 30 seconds timeout, 10 to 30 seconds between two retries.

The test environment and application is described more precisely in [7], [8/a].

4.1 The Measured Results

The results were better than I had expected.

All the registrations of all the students were successful.

The total time needed for the 32.768 registrations of the 8.192 students was 3 minutes and 7 seconds, so the average time needed for a test student was 0.0228 second, with a maximum value of 1 (one) second because the offline browser wget logs its activities in hh:mm:ss format, so fractions of seconds cannot be taken into account.

The maximum value for the server load (1 min load) was 2.85, with an interesting, staircase-of-staircases like diagram as described in [7].

These values are quite good compared to real life experience. Even if the measured values were bigger by a whole order of magnitude (i.e. the 32k registration of the 8k students needed about half an hour) they would be good enough to prove my statement. Therefore, I can state that we have the possibility to develop much more efficient scholar information systems.

Conclusions and Two Open Questions

Summarizing the above, I can state that (much) better results can be achieved based on a 'good' three-level data modelling even in poorer hardware and/or

software circumstances. The quality of the data model influences the quality and the efficiency of the database to such an extent that precise three-level data modelling, according to Dr. Halassy [6] (p. 32), ought to be more important than it is generally considered at present. The quality of the data model is an important variable, if not the most important one. There is no other way to produce good information systems. System development methods and standards alone are not enough for that.

We have been using the two big SISes in higher education for about a decade. So I have two open questions at the end.

The first: Are the faculties of computer sciences in the country of John von Neumann able to, want to and dare to develop a better system? The second: Why has the first question never been asked?

References

[1] Raffai Mária dr.: Információrendszerek fejlesztése és menedzselése. Novadat Bt., 2003

[2] Halassy Béla dr.: Adatmodellezés. Nemzeti Tankönyvkiadó Rt., 2002

[3] Codd Edgar Frank: The Relational Model for Database Management - version 2. Addison-Wesley Publishing Company, 1990

[4] Chen P.: The Entity-Relationship Model -- Toward a Unified View of Data. In: ACM Transactions on Database Systems (TODS), 1976. március, I. évf. 1. szám

[5] Halassy Béla dr.: Az adatbázistervezés alapjai és titkai. IDG Magyarországi Lapkiadó Kft., 1995

[6] Halassy Béla dr.: Ember - információ - rendszer. IDG Magyarországi Lapkiadó Vállalat, 1996

[7] Szikora Péter: Measured Performance of an Information System. 7th International Conference on Management, Enterprise and Benchmarking, Budapest, 2009

[8a] Szikora Péter: The Role of the Tools and Methods of Implementation in Information System Efficiency. 2nd International Conference for Theory and Practice in Education, Budapest, 2009

[8b] Keszthelyi András: The Role of Data Modeling in Information System Efficiency. 2nd International Conference for Theory and Practice in Education, Budapest, 2009

[9] Keszthelyi András: Information management in the higher education -- the role and importance of the different technologies. 3rd International Conference on Management, Enterprise and Benchmarking, Budapest, 2005