ISBN 3-540-67978-2

9 783540 679783

ISSN 0302-9743

http://www.springer.de

Database and Expert

2000

Lecture Notes in Computer Science 1873

Mohamed Ibrahim Josef Küng
Norman Revell (Eds.)

Database and Expert Systems Applications

11th International Conference, DEXA 2000
London, UK, September 2000
Proceedings

DEXA 2000

Springer

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Singapore
Tokyo

Mohamed Ibrahim Josef Küng
Norman Revell (Eds.)

Database and Expert Systems Applications

11th International Conference, DEXA 2000
London, UK, September 4-8, 2000
Proceedings

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Mohamed Ibrahim
Maritime Greenwich University Campus
Department of Computing and Mathematical Sciences
30 Park Row, London SE10 9LS, UK
E-mail: M.T.Ibrahim@gre.ac.uk

Josef Küng
University of Linz, FAW
Altenbergerstr. 69, 4040 Linz, Austria
E-mail: jkueng@faw.uni-linz.ac.at

Norman Revell
Middlesex University
Bounds Green, London, UK
E-mail: n.revell@mdx.ac.uk

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Database and expert systems applications : 11th international conference ; proceedings / DEXA 2000, London, UK, September 4 - 8, 2000. Mohamed Ibrahim ... (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 2000
(Lecture notes in computer science ; Vol. 1873)
ISBN 3-540-67978-2

CR Subject Classification (1998): H.2, I.2.1, H.3, H.4, H.5

ISSN 0302-9743
ISBN 3-540-67978-2 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

Printed in Germany

Typesetting: Camera-ready by author
Printed on acid-free paper SPIN 10722345 06/3142 5 4 3 2 1 0

Preface

The Database and Expert Systems Applications (DEXA) conferences have established themselves as a platform for bringing together researchers and practitioners from various backgrounds and all regions of the world to exchange ideas, experiences and opinions in a friendly and stimulating environment. The papers presented at the conference represent recent developments in the field and important steps towards shaping the future of applied computer science and information systems.

DEXA covers a broad field: all aspects of databases, knowledge based systems, knowledge management, web-based systems, information systems, related technologies and their applications. Once again there were a good number of submissions: out of 183 papers that were submitted, the program committee selected 92 to be presented.

In the first year of this new millennium DEXA has come back to the United Kingdom, following events in Vienna, Berlin, Valencia, Prague, Athens, London, Zurich, Toulouse, Vienna and Florence. The past decade has seen several revolutionary developments, one of which was the explosion of Internet-related applications in the areas covered by DEXA, developments in which DEXA has played a role and in which DEXA will continue to play a role in its second decade, starting with this conference.

I would at this point like to express thanks to all the institutions which actively supported this conference and made it possible; namely:

- University of Greenwich
- Institut für Anwendungsorientierte Wissensverarbeitung (FAW), Universität Linz
- Austrian Computer Society
- DEXA Association

Many people contributed numerous hours to the success of this conference. Special thanks go to Maria Schweikert (Technische Universität Wien), Monika Neubauer and Gabriela Wagner (FAW, Universität Linz). We must also thank all members of the program committee, whose careful reviews are important to the quality of the conference, and the local organisers, who have put a lot of time and effort into making DEXA feel at home in Greenwich.

July 2000

Gerald Quirchmayr, Universität Wien, Austria
General Chair

Program Committee

General Chairperson:
G. Quirchmayr, University of Vienna, Austria

Conference Program Chairpersons:
M. Ibrahim, University of Greenwich, UK
J. Küng, FAW, University of Linz, Austria
N. Revell, Middlesex University, UK

Publication Chairperson:
V. Marik, Czech Technical University, Czech Republic

Workshop Chairpersons:
A M. Tjoa, Technical University of Vienna, Austria
R.R. Wagner, FAW, University of Linz, Austria
A. Al-Zobaidie, University of Greenwich, UK

Executive Chairperson
M. Everitt, University of Greenwich, UK

Local Arrangements
R.D. Finney, University of Greenwich, UK

Program Committee Members:
H. Afsarmanesh, University of Amsterdam, The Netherlands
A. Al-Zobaidie, University of Greenwich, United Kingdom
J. Albrecht, University of Erlangen-Nuremberg, Germany
B. Amann, CNAM&INRIA, France
F. Andres, NACSIS, Japan
K. Bauknecht, University of Zurich, Switzerland
T. Bench-Capon, University of Liverpool, UK
B. Bhargava, Purdue University, USA
J. Bing, NRCCL Oslo, Norway
P. Bosc, ENSSAT, France
O. Bukhres, Purdue University, USA
L. Camarinha-Matos, New University of Lisbon, Portugal
A. Cammelli, IDG-CNR, Italy
W.S. Cellary, University of Economics at Poznan, Poland
S. Chakravarthy, University of Florida, USA
S. Christodoulakis, University of Crete, Greece
P. Chrysanthis, University of Pittsburgh, USA
P. Ciaccia, University of Bologna, Italy
C. Collet, LSR-IMAG, France
C. Combi, University of Udine, Italy
D. Cowell, University of Greenwich, United Kingdom
W.B. Croft, University of Massachusetts, USA
J. Debenham, University of Technology, Sydney, Australia
M. Deen, University of Keele, UK
J. Eder, University of Klagenfurt, Austria
T. Eiter, Technical University of Vienna, Austria
G. Engels, University of Paderborn, Germany

P. Fankhauser, GMD, Germany
E. Fernandez, Florida Atlantic University, USA
S. Field, IBM Zurich, Switzerland
P. Funk, Mälardalen University, Sweden
A.L. Furtado, University of Rio de Janeiro, Brazil
G. Gardarin, INRIA, France
S. Ghandeharizadeh, University of Southern California, USA
P. Godfrey, York University, Canada
G. Gottlob, Vienna University of Technology, Austria
P. Grefen, University of Twente, The Netherlands
U. Hahn, University of Freiburg, Germany
J.-L. Hainault, University of Namur, Belgium
A. Hameurlain, Université Paul Sabatier, France
I. Hawryszkiewycz, University of Technology, Sydney, Australia
Y. Kambayashi, University of Kyoto, Japan
M. Kamel, Naval Postgraduate School, USA
N. Kamel, American University in Cairo, Egypt
G. Kappel, University of Linz, Austria
D. Karagiannis, University of Vienna, Austria
K. Karlapalem, University of Science and Technology, Hong Kong
R. Keller, University of Montreal, Canada
M.H. Kim, KAIST, Korea
B. Knight, University of Greenwich, UK
J. Kouloumdjian, LISI-INSA, France
P. Kroha, Technical University of Chemnitz, Germany
J. Lazansky, Czech Technical University, Czech Republic
M. Leonard, University of Geneve, Switzerland
T.W. Ling, University of Singapore, Singapore
W. Litwin, University Paris 9, France
M. Liu, University of Regina, Canada
F. Lochovsky, University of Science and Technology, Hong Kong
P. Loucopoulos, University of Science and Technology in Manchester, United Kingdom
S.K. Madria, Purdue University, USA
A. Makinouchi, Kyushu University, Japan
H. Mannila, Microsoft Research, USA
V. Marik, Czech Technical University, Czech Republic
S. Marinai, University of Florence, Italy
S. Mazumdar, New Mexico Institute of Mining and Technology, USA
R. Meersman, Free University of Brussels, Belgium
E. Metais, University of Versailles, France
M. Mohania, Western Michigan University, USA
S. Monties, EPFL, Switzerland
T. Morzy, Poznan University of Technology, Poland
G. Müller, University of Freiburg, Germany
W. Naqvi, Raytheon Systems Company, USA
E. Neuhold, GMD-IPSI, Germany
G. Ozsoyoglu, University Case Western Research, USA
G. Pangalos, University of Thessaloniki, Greece
B. Pernici, Politecnico di Milano, Italy
G. Pernul, University of Essen, Germany
G. Quirchmayr, University of Vienna, Austria
F. Rabitti, CNUCE-CNR, Italy

I. Ramos, Technical University of Valencia, Spain
H. Reiterer, University of Konstanz, Germany
N. Revell, Middlesex University, UK
C. Rolland, University Paris I, France
E. Rundensteiner, Worcester Polytechnic Institute, USA
M.E. Rusinkiewicz, MCC, USA
D. Sacca, University of Calabria, Italy
M. Savonnet, University of Bourgogne, France
E. Schweighofer, University of Vienna, Austria
T. Sellis, NTU Athens, Greece
M. Smith, University of California, USA
G. Soda, University of Florence, Italy
H. Sonnberger, EUROSTAT, Luxemburg
U. Srinivasan, CIRO, Australia
O. Stepankova, Czech Technical University, Czech Republic
D. Stott Parker, University of Los Angeles, USA
V.S. Subrahmanian, University of Maryland, USA
J. Suzuki, Keio University, Japan
M. Takizawa, Tokyo Denki University, Japan
K. Tanaka, Kobe University, Japan
Z. Tari, RMIT, Australia
S. Teufel, University of Oldenburg, Germany
J. Teuhola, University of Turku, Finland
B. Thalheim, Technical University of Cottbus, Germany
J.M. Thevenin, University Paul Sabatier, France
C.H. Thoma, IT pro, Switzerland
R. Traunmüller, University of Linz, Austria
P. Triantafillou, Technical University of Crete, Greece
A. Tsalgatidou, University of Athens, University of Jyväskylä, Greece, Finland
S. Urban, Arizona State University, USA
K. Vidyasankar, Memorial University of Newfoundland, Canada
Y. Wand, University of British Columbia, Canada
M. Wing, Middlesex University, UK
W. Winiwarter, University of Vienna, Austria
A. Zaslawsky, Monash University, Australia

External Referees

I. Pramudiono, University of Tokyo, Japan
I. Karali, University of Athens, Greece
A. Pikrakis, University of Athens, Greece
C. Vasilakis, University of Athens, Greece
M. Nikolaidou, University of Athens, Greece
E. Berki, University of Jyvaskyla, Finland
J.H. Canós, Universitat Politecnica de Valencia, Spain
N. Rodriguez Brisaboa, Universitat Politecnica de Valencia, Spain
J.C. Casamayor, Universitat Politecnica de Valencia, Spain
E. Mena, Universitat Politecnica de Valencia, Spain
E. I. Benitez-Guerrero, IMAG-LSR Laboratory, France
G. Vargas-Solar, IMAG-LSR Laboratory, France
A. Lefebvre, France Telecom R&D, France

well as the perceptive factors related to image evaluation. The ranking works the best for a parameter combination which favors the perception factors.

Generally, the best parameter combination will depend on the document collection and on the user preference. A user study such as the one carried here only serves to evaluate the global performance of the system, and for future improvement; for efficiency reasons, it cannot be performed very often and for all the collections. An intelligent information retrieval system should be able to adapt to the working environment, i.e. to the document collection and to the user profile. In order to find the best parameter combination for a given user, the system must dynamically update the parameter values, starting from a certain default combination. Relevance feedback is a technique already used in information retrieval, to answer to assessments made by users during interactive sessions [8]. An application of this technique to our case would involve the search in an n-dimensional space, where n is the number of parameters. Given the current parameter value combination and a user feedback, that classifies a certain image before another one, the system must advance in the multidimensional space, towards a convergence point that represents the best combination. The next development system will focus on such aspects. Our final purpose is to build a user-centered image retrieval system.

References

1. J. May and P. Barnard. Modelling multimodal interaction: A theory-based technique for design analysis and support. In *Human-Computer Interaction (INTERACT '97)*, pages 667–668, London. Also in www.shef.ac.uk/ pc1jm/guide.html, 1997. Chapman & Hall.
2. A. Plante, S. Tanaka, and S. Inoue. Evaluating the location of hot spots in interactive scenes using the 3R toolbox. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'98)*, pages 117–123, Los Angeles, USA, 1998. ACM Press.
3. M. Mechkour. EMIR2. An extended model for image representation and retrieval. In *DEXA'95. Database and EXpert system Applications, London.*, pages 395–404, September 1995.
4. I. Ounis and M. Pasca. RELIEF: Combining expressiveness and rapidity into a single system. In *Proccedings of the 21th ACM SIGIR Conference, Melbourne, Australia*, pages 266–274, 1998.
5. I. Ounis and M. Pasca. Modeling, indexing and retrieving images using conceptual graphs. In *Proceedings of the 9th DEXA International Conference on Database and EXpert Systems Applications*, pages 226–239, Vienna, Austria, August 1998.
6. I. Ounis. A flexible weighting scheme for multimedia documents. In *Proceedings of the 10th DEXA International Conference on Database and EXpert Systems Applications*, pages 392–405, Florence, Italy, August 1999.
7. Y. Chiaramella and M. Mechkour. Indexing an image test collection. Technical report, FERMI BRA 8134, April 1997.
8. J.J. Rocchio Jr. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART retrieval System-Experiments in Automatic Document Processing*, chapter 14, pages 313–323. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

Measuring for Database Programs Maintainability

Mario Piattini[1], Antonio Martinez[1,2]
[1]RESEARCH GROUP ALARCOS
DEPARTAMENTO DE INFORMÁTICA
UNIVERSITY OF CASTILLA-LA MANCHA
RONDA DE CALATRAVA, 5
13071, CIUDAD REAL. SPAIN.
{mpiattin, amartinez} @inf-cr.uclm.es
[2]EXCMA. DIPUTACION DE CIUDAD REAL
CALLE TOLEDO, 17
13001, CIUDAD REAL (SPAIN)
amartinez@dipucr.es

Abstract. Nowadays, relational databases are introduced in most of the Information Systems, becoming their essential core. The relational database language, SQL, is being increasingly used for application development. Software engineers have been putting forward huge quantities of measures for software products, processes and resources. Unfortunately, almost all the measures proposed until now, have focused on 3GL program characteristics disregarding databases and their associated languages. Only a few author have proposed some metrics for 4GL effort estimation. In this article we describe three simple measures for assessing SQL code maintainability. Both an experiment with students and a real case in a state owned organization have empirically validated these measures as maintainability indicators.

1 Introduction

Nowadays, relational databases are introduced in most of the Information Systems, becoming their essential core. The relational database language SQL[1], is used in almost all the organizations which have management information systems, either embedded in 3GL (Third Generation Language), mainly COBOL, programs or in more modern productivity-enhancing tools such as 4GL (Fourth Generation Languages).

Information Technology (IT) organizations must improve the quality of software products, especially maintainability, because maintenance cost are the most important

[1] ISO/IEC and ANSI standarizarion committes have recently proposed a new version of the SQL know as SQL:1999 (formerly SQL3) [1]. This paper focus on the previous versions (SQL'89).

problem of software development, ranging between 60 and 90 percent of life-cycle costs [2],[3]. Software measurement is widely recognized as an effective means to understand, monitor, control, predict and improve software development and maintenance projects [4]. Software engineers have been putting forward hundreds quantities of measures for software products, processes and resources [5],[6]. Unfortunately, almost all the measures proposed until now, have focused on 3GL program characteristics disregarding databases and their associated languages [7], or object oriented environment [8]. Only some works attempt to estimate the effort of developing 4GL programs [9],[10],[11] but we have not came accross any efforts to try to assess SQL code maintainability. We adopted the definition of maintainability as the capability of the software product to be modified. Modifications may include corrections, improvements or adaption of the software to changes in environment, and in requirememnts and functional specifications [12]. Maintainability is achieved by three factors: understandability, modifiability and testability [13].
In this paper, three simple measures for assessing SQL program maintainability are proposed. In next section we describe the three measures. In section 3, an empirical validation experiment is presented. In section 4 the results of a real case study are exposed. Finally, in section 5 we summarize the paper and present the conclusions.

2 Proposed measures for assessing SQL code

SQL language is composed of different kinds of statements: definition statements (e.g. CREATE TABLE), manipulation statements (e.g. INSERT) and control statements (e.g. COMMIT). Four different manipulation statement can be found: SELECT, INSERT, DELETE and UPDATE. The first one is the most used and so we focus our work on it.
We propose the following three measures for characterizing SELECT statement:

NT measure

Number of tables referred in the SELECT statement.

NN measure

Number of nesting, considers the number of "SELECT" in the SELECT statement.

G measure

This measure indicates whether exist (1) or not (0) a GROUP BY clause in the SELECT statement.


```
select f.name_emp, p.number_fic, p.date
       from control_employee p, employee f, h_employee h
       where p.id_emp not in
(select h.id_emp
              from control_employee p, employee f, h_employee h
                where p.number_fic=h.number_fic
                       and f.id_emp=h.id_emp
                       and p.id_emp=f.id_emp
                       and p.date='171298'
                       and p.control='SM'
                       and p.status='A'
                       and f.sex='V'
                       and p.hour in
(select hour
from control_employee p, employee f, h_employee h
                          where p.number_fic=h.number_fic
                                   and f.id_emp=h.id_emp
                                   and p.id_emp=f.id_emp
                                   and p.date='171298'
                                   and p.control='SM'
                                   and p.tipe='AO'
                                   and h.remaindert=0
                          )
               )
                 and p.date='151298'
                 and p.number_fic=h.number_fic
                 and p.id_emp=f.id_emp
                 and f.id_emp=h.id_emp
group by f.name_emp, p.number_fic, p.date
```

Fig. 1. In the example the values are NT=3, NN=3 and G=1.

These measures were proposed based on intuition and experience with SQL programs development and maintenance. The number of tables is likely to influence all the three maintainability factors as SQL statements will be more difficult to understand, to modify and to test if they include more tables.
The number of nesting is likely to also influence the maintainability of the SQL code, as each nesting demands a new level of thinking similar to a new call in 3GL or a level of inheritance in object-oriented programs [14]. The earlier relational optimizers had also been influenced by the SELECT nesting, and vendors recommended not to nest beyond 3 levels for performance reasons.
We believe that grouping rows for calculating values also influences maintainability of SQL programs as it implies an additional operation which must be carried out over a set of rows.

3 Empirical validation of the proposed measures

In this section we summarize an experiment done in order to validate NT, NN and G measures. This empirical validation has been carried out following the experimental method applied to software engineering [9],[15].
Our aim is to demonstrate that the proposed measures can be used for measuring the understandability of the SQL statement which influences its maintainability. Understandability is the the capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use [12].

3.1 Hypotheses

The formal hypotheses are:
Null hypothesis: Different values of the three measures do not affect the understandability of the SELECT statement.
Alternative hypothesis 1: The value of the NT measure affects the understandability of the SELECT statement.
Alternative hypothesis 2: The value of the NN measure affects the understandability of the SELECT statement.
Alternative hypothesis 3: The value of the G measure affects the understandability of the SELECT statement.
Alternative hypothesis 4: The combination of NT and NN measures affects the understandability of the SELECT statement.
Alternative hypothesis 5: The combination of NT and G measures affects the understandability of the SELECT statement.
Alternative hypothesis 6: The combination of NN and G measures affects the understandability of the SELECT statement.
Alternative hypothesis 7: The combination of NT, NN and G measures affects the understandability of the SELECT statement.

3.2 Subjects

The participants of the field test were Computer Science students at the University of Castilla-La Mancha (Spain), who were enrolled in a database course lasting two semesters. Until the day of the experiment, the students did not known that they were to do it. The experiment was developed by 34 students, but only 19 of them were selected, because it answered rightly all the questions.
We have tried to minimize variability among participants by choosing people of the same degree, in particular from the third year.

3.3 Experimental materials

Eight separate SQL statements were required to test the hypotheses. In each one values of the three measures were different. There were two possible values for NT (one or three), two for NN (one or three) and two for G (zero or one). The documentation accompanying each design was approximately twelve pages long including the tables and the queries. In appendix A the tables and the SQL statements are shown.
The same table was used to reduce semantic variability in all the eight cases. In order to avoid learning errors, the eight cases were in different order for each subject, and the subjects were forced to follow the order in which cases appeared in the experimental material.
The subjects were asked to write down the initial and the final time, and the result for each query. We only counted the time employed in giving correct answers.

3.4 Experimental design.

Each level of one factor appears with each level of the other one, so we have selected the crossing design NT x NN x G. See table 1.

		FACTOR NT			
		LOW		HIGH	
		FACTOR NN			
		LOW	HIGH	LOW	HIGH
FACTOR G	LOW	1,1,0	1,3,0	3,1,0	3,3,0
	HIGH	1,1,1	1,3,1	3,1,1	3,3,1

Table 1. Crossed Design for the experiment

To increase the power of the test, α has been set to 0.1 instead of 0.05 level which is more common [16].

3.5 Experimental results

Due to the type of experiment used, F statistic was applied to obtain the results.
SPSS v. 7.5 software was used for the calculations. Table 2 shows the results for the F-statistic.

Source of variation	Sum of Squares	DF	Mean Square	F	Sig of F
Source of Variation	1.392,132	3	464,044	382,878	0
NT	796,737	1	796,737	657,380	0
NN	553,289	1	553,289	456,514	0
G	42,105	1	42,105	34,741	0
2 Way Interactions	43,184	3	14,395	11,877	0
NT NN	42,105	1	42,105	34,741	0
NT G	0,658	1	0,658	0,543	0,462
NN G	0,421	1	0,421	0,347	0,557
3 Way Interactions	2,132	1	2,132	1,759	0,187
NT NN G	2,132	1	2,132	1,759	0,187
Explained	1.434,447	7	205,350	169,432	0
Redisual	174,526	144	1,212		
Total	1.611,974	151	10,675		

Table 2. Results of the experiment

Comparing these values with $F_{1,151} = 2.71$, we can ensure that:

Alternative Hypothesis 1: The value of the NT measures affects understandability of SELECT statement. As 657.380>2.71, NT affects results of experiment, so that the alternative hypothesis 1 is valid.
Alternative Hypothesis 2: The value of the NN measures affects understandability of SELECT statement. As 456.514>2.71, NN affects results of experiment, so that the alternative hypothesis 2 is valid.
Alternative Hypothesis 3: The value of the G measures affects understandability of SELECT statement. As 34.741>2.71, G affect to results of experiment, so that the alternative hypothesis 3 is valid.
Alternative Hypothesis 4: Combination of NT and NN measures affects understandability of SELECT statement. As 34.741>2.71, the interaction of NT and NN affects results of experiment, so that, the alternative hypothesis 4 is valid.
Alternative Hypothesis 5: Combination of NT and G affects understandability of SELECT statement. As 0.543<2.71, there is no significant effect of interaction between NT and G
Alternative Hypothesis 6: Combination of NN and G affects understandability of SELECT statement. As 0.421<2.71, there is no significant effect of interaction between NN and G
Alternative Hypothesis 7: Combination of NT, NN and G affects understandability of SELECT statement. As 1.759<2.71, there is no significant effect of interaction between NT, NN and G.
We can conclude that the three kinds of measures proposed have proved to be solid indicators of SQL programs understandability. These three measures are very easy to calculate and could be very useful to predict SQL maintainability.

4 Case study

4.1 General characteristics of the system.

The system is composed of 143 programs developed during a period of one year at the Data Processing Center of the state owned organization (Diputacion of Ciudad Real). The system is a transaction processing system for data maintenance. The programs with embedded SQL are all of small to medium size: each SELECT statement included an average of three tables, two-level nesting and one grouping.
More importantly, the system developed was functionally sound, providing an actual working solution to an actual organizational problem.
One of the most positive aspects of the system is the fact that it was constructed completely by the same team, employing the same methodology and the same developing environment (CA-OpenIngres/4GL).These common factors are advantageous in they can be considered as constants in the analysis, a condition not often encountered in software size research. When they vary, factors such as these can have an obvious impact on system size. Given that these potential contributors may be treated as constant, the degree of confidence adopted in regard to any size relationships supported by the data will consequently be greater [17].

4.2 Data collection

The maintenance process includes adding functionality to the software (adaptive maintenance) and correcting defects discovered in the systems (corrective maintenance).
Our study examined data from a maintenance period beginning with the original installation of the product and ending the product's second release. The required changes varied in magnitude from a simple command line option change to a more complex one. During the maintenance period the programmer recorded the daily effort of product maintenance, descriptions of the faults encountered as a consequence of enhancement activities and the time spent correcting these faults. All these dates are recorded in a specific tool developed in CA-OpenIngres. (see fig. 1 and 2).

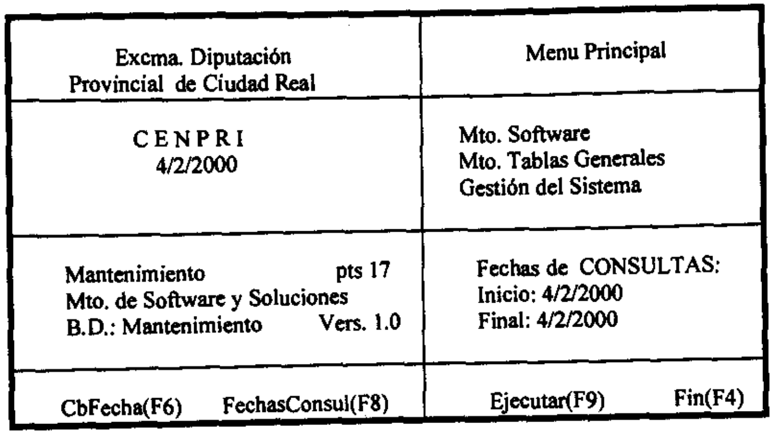

Fig. 2. Screen main of the application

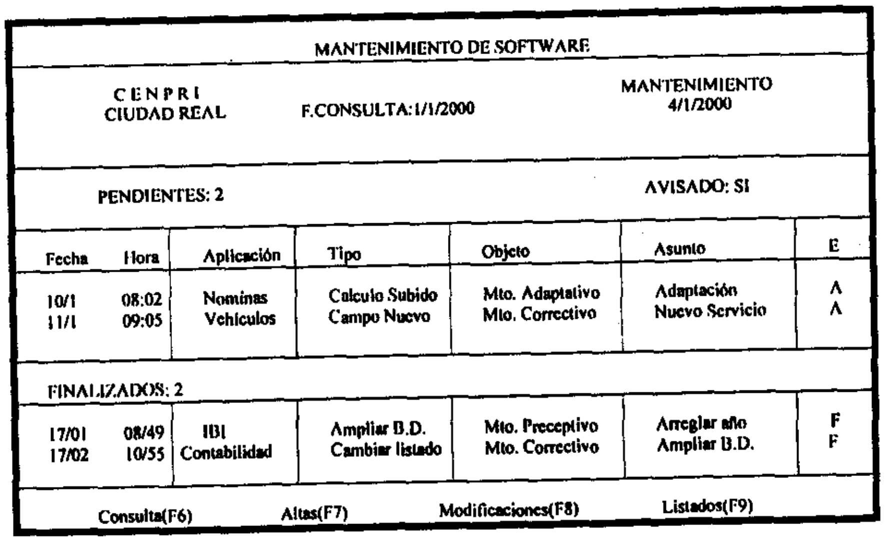

Fig. 3. Tool window showing an initial task

4.3 Data analysis and results

4.3.1 Descriptive statistics

The general descriptive statistics for each one of the variables are shown in table 3.

Variable	Mean	Variance	S.E. Skew	Min	Max	Std Dev	Skewness
NT	6,042	30,364	0,203	0	22	5,510	1,361
NN	1,923	3,438	0,203	0	6	1,854	0,712
G	0,399	0,241	0,203	0	1	0,492	0,419
TIME	74,364	5.838,865	0,203	1	290	76,413	1,205

Valid observations - 143 Missing observations – 0

Table 3: Descriptive statistics for each measure

4.3.2 Correlation analysis

For the test of correlation we use Pearson´s coefficient statistics and Spearman´s non-parametric correlation to identify potentially relationships between the variable time of maintenance (expressed in minutes) and the measures defined, as well as the relationships that could exist between the same variables. The results are shown in tables 4 and 5.

The statistics of both correlation sets evidence strong significant relationships between the variable time of maintenance and the measure defined, except for measure NG. We can observe that the relationships between the specification of the measured NT, NN and the variable TIME of maintenance are significant.

		NT	NN	G	TIME
Pearson	NT	1,000	0,796	0,280	0,986
	NN	0,796	1,000	0,258	0,881
	G	0,280	0,258	1,000	0,284
	IME	0,986	0,881	0,284	1,000
Sig	NT		0,000	0,001	0,000
	NN	0		0,002	0,000
	G	0,001	0,002		0,001
	TIME	0	0,000	0.001	
N	NT	143	143	143	143
	NN	143	143	143	143
	G	143	143	143	143
	TIME	143	143	143	143

Table 4. Pearson's Correlation Coefficients

		NT	NN	G	TIME
Spearman	NT	1,000	0,799	0,286	0,964
	NN	0,799	1,000	0,243	0,908
	G	0,286	0,243	1,000	0,283
	TIME	0,964	0,908	0,283	1,000
Sig	NT		0,000	0,001	0,000
	NN	0,000		0,003	0,000
	G	0,001	0,003		0,001
	TIME	0,000	0,000	0.001	
N	NT	143	143	143	143
	NN	143	143	143	143
	G	143	143	143	143
	TIME	143	143	143	143

Table 5. Spearman's correlation coefficients.

NT and G are also highly correlated, which is logical because each nesting introduces a table, which usually is different from the table of the previous nesting level. We are conscious that maintenance time can depend on several other different factors than the SELECT characteristics. Some programs have, besides SELECT, other statements like INSERT, DELETE or UPDATE; and also different procedural and visual statements. However, due to the type of the programs involved, we think that these results can be a valid first attempt to characterize SQL programs.
The level of grouping (G) is not correlated with the time of maintenance. We cannot find an answer for this now. More case studies and experiments must be considered in order to explain the influence of grouping in SQL understandability, modifiability and testability.

5 Conclusions and future works

More research is needed into the aspects of software measurement [18], both from theoretical and from practical points of view [19]. We think it is very interesting to dispose of measures for relational databases. We have proposed and validated three types of measures for assessing SQL program maintainability: NT,NN and G. We are also developing some similar measures for other clauses of the SELECT statement such as "HAVING", "WHERE" or "FROM".
More experiments and case studies are needed to confirm these measures as valid indicators for SQL program maintainability. Verification of these metrics in some formal frameworks as [16] or [20] is being carried out.
These measures are not enough to evaluate the maintainability of programs developed with 4GL, so different measures must be proposed for other different "sublanguages" besides data manipulation one. Traditional metrics must be adapted or new metrics

must be defined to assess procedural, visual, control, definition and transaction statements.

Acknowledgements

This research has been supported by the MANTICA project CICYT and the European union under grant 1FD97-0168 . We thank paper to Fernando Brito e Abreu and Miguel Goulão for their thoughtful comment which contributed to improve considerably this paper.

References

1. Eisenberg, A. and Melton, J.: SQL: 1999, formerly know as SQL3. SIGMOD Record, 28 ,(1) (1999) 131-138.
2. Card, D.N. and Glass, R.L. : Measuring Software Design Quality. Englewood Cliffs. USA (1990).
3. Pigoski, T.M.: Practical Software Maintenance. Wiley Computer Publishing. New York, USA (1997).
4. Briand, L.C., Morasca, S. and Basili, V.: Property-based software engineering measurement. IEEE Transactions on Software Engineering, 22(1) (1996) 68-85.
5. Fenton, N. and Pfleeger, S. L.: Software Measures: A Rigorous Approach 2nd edition.. Chapman & Hall.London (1997).
6. Melton, A. (ed.) .: Software Measurement. International Thomson Computer Press. London (1996).
7. Sneed, H.M. and Foshag, O. : Measuring Legacy Database Structures. Proc of The European Software Measurement Conference FESMA 98, Antwerp, May 6-8, Coombes, Van Huysduynen and Peeters (eds.), (1998) 199-211.
8. Henderson-Sellers, B. : Object-Oriented Measures - Measures of complexity. Prentice-Hall, Upper Saddle River, New Jersey (1996).
9. Bourque, P. and Côté, V. : An experiment in software sizing with structured analysis measures. Journal of Systems and Software 15 (1991) 159-172.
10. Dolado, J.J.: A Study of the Relationships among Albrecht and Mark II Function Points, Lines of Code 4GL and Effort. J. Systems Software, 37 (1990) 161-173.
11. Verner J. and Tate G.: Estimating Size and Effort in Fourth-Generation Development. IEEE Trans. on Software Engineering (1988) 15-22.
12. ISO/IEC 9126: Information Technology-Software Product Evaluation-Quality characteristics and guidelines for their use, International Organization for standardisation (1994).
13. Li, H.F. and Cheng, W.K. An empirical study of software measures. IEEE Trans. on Software Engineering, 13 (6) (1987) 679-708.
14. Cant, S. N., Jeffery, D.R., and Henderson-Sellers, B.: A concept model of cognitive complexity of element of the programming process. Information and Software Technology. 37(7) (1995) 351-362.
15. Pfleeger, S. L.: Experimental Design and Analysis in Software Engineering. Annals of Software Engineering .JC. Baltzer AG, Science Publishers 1 (1995) 219-253.

16 Briand, L., Bunse, D., Daly, J. and Differding, C.: An experimental comparison of the Maintainability of Object-Oriented and Structured Design Documents, Proc. Int. Conference on Software Maintenance, Harold, M.J. Visaggio, G. (eds), Bari, 1-3 (1997) 130-138.
17. MacDonell, G., Shepperd, J. and Sallis, J.: Measures for Database Systems: An Empirical Study. IEEE Software (1997) 99-107
18. Neil, M.: Measurement as an Alternative to Bureaucracy for the Achievement of Software Quality. Software Quality Journal 3 (2), (1994) 65-78.
19. Glass, R.: The Relationship Between Theory and Practice in Sotfware Engineering. IEEE Software, November, 39(11) (1996) 11-13.
20. Zuse, H.: A framework of software measurement. Walter De Gruyter (1998).

APPENDIX A

TABLE FIELD DESCRIPTION

TABLE DESCRIPTION: employee

```
Name:        employee
Owner:       ingres
Created:     24/03/1999 14:08:38
Type:        user table
Version:     OPING1.2
```

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
id_emp	varchar	9	no	no	1
namef_emp	varchar	40	no	no	
name_emp	varchar	30	no	no	
direction	varchar	40	no	no	
post_office	varchar	5	no	no	
town	varchar	2	no	no	
village	varchar	3	no	no	
phone	varchar	9	yes	null	
sex	varchar	1	no	no	
status	varchar	1	no	no	
children	varchar	2	no	no	
birth	date		no	no	
birth_town	varchar	2	no	no	
birth_village	varchar	3	no	no	
birth_country	varchar	3	no	no	
number_social_sec	varchar	14	no	no	
number_social_sec1	varchar	6	yes	null	

The table "employee" has 4 rows.

TABLE DESCRIPTION: h employee

```
Name:        h_employee
Owner:       ingres
Created:     20/11/1998 18:05:17
Type:        user table
Version:     OPING1.2
```

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
code_service	varchar	3	no	no	
code_subs_service	varchar	2	no	no	
id_emp	varchar	9	no	no	
number_fic	varchar	4	no	no	
remainder1	float	8	no	no	
remainder2	float	8	no	no	
remainder3	float	8	no	no	
remainder4	float	8	no	no	
remaindert	float	8	no	no	
remainderf	float	8	no	no	
period	integer	1	no	no	
year	integer	1	no	no	
tipe_hour	integer	2	yes	null	
subtipe	integer	2	yes	null	
key_emp	varchar	15	yes	null	
date_certificate	varchar	6	yes	null	
situation	varchar	1	yes	null	
remainderf	date		yes	null	

The table "h_employee" has 3 rows.

TABLE DESCRIPTION: control employee

```
Name:        control_employee
Owner:       ingres
Created:     04/03/1999 13:46:31
Type:        user table
Version:     OPING1.2
```

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
id_emp	varchar	9	no	no	
number_fic	varchar	4	no	no	
date	varchar	6	no	no	
hour	varchar	4	no	no	
code_incidence	varchar	2	yes	null	
control	varchar	2	yes	null	
status	varchar	1	yes	null	
code_center	varchar	2	yes	null	
tipe	varchar	2	no	no	

The table "control_employee" has 72 rows.

SQL STATEMENTS

1.- **select** hour **from** control_employee **where** number_fic='0959' **and** date='181298'

2.- **select** number_fic, date, count(hour) **as** w_number_fic **from** control_employee **where** number_fic='0800' **group by** number_fic, date

3.- **select** number_fic, date **from** control_employee **where** number_fic **not in** (**select** number_fic **from** control_employee **where** date='171298' **and** control='SM' **and** status='A' **and** hour **in** (**select** hour **from** control_employee **where** date='171298' **and** control='SM' **and** tipe='A0')) **and** date>'131298'

4.- **select** number_fic **from** control_employee **where** number_fic **not in** (**select** Number_fic **from** control_employee **where** date='171298' **and** control='SM' **and** status='A' **and** hour **in** (**select** hour **from** control_employee **where** date='171298' **and** control='SM' **and** tipe='A0')) **and** date>'131298' **group by** number_fic

5.- **select** f.name_emp, h.hour, p.key_emp **from** employee f, control_employee h, h_employee p where f.id_emp=h.id_emp **and** h.number_fic=p.number_fic **and** h.date='171298'

6.- **select** f.name_emp, p.key_emp **from** employee f, control_employee h, h_employee p **where** f.id_emp=h.id_emp **and** h.number_fic=p.number_fic **and** h.date='171298' **group by** name_emp, key_emp

7.- **select** f.name_emp, p.number_fic, p.date, h.date_certificate **from** control_employee, employee f, h_employee h **where** p.id_emp **not in** (**select** h.id_emp from control_employee p, employee f, h_employee h **where** p.number_fic=h.number_fic **and** f.id_emp=h.id_emp **and** p.id_emp=f.id_emp **and** p.date='171298' **and** p.control='SM' **and** p.status='A' **and** f.sex='V' **and** p.hour **in** (**select** hour **from** control_employee p,employee f, h_employee h **where** p.number_fic=h.number_fic **and** f.id_emp=h.id_emp **and** p.id_emp=f.id_emp **and** p.date='171298' **and** p.control='SM' **and** p.tipe='A0' **and** h.remaindert=0)) and p.date='151298' and p.number_fic=h.number_fic **and** p.id_emp=f.id_emp **and** f.id_emp=h.id_emp

8.- **select** f.name_emp, p.number_fic, p.date **from** control_employee p, employee f, h_employee h **where** p.id_emp **not in** (**select** h.id_emp **from** control_employee p, employee f, h_employee h **where** p.number_fic=h.number_fic **and** f.id_emp=h.id_emp **and** p.id_emp=f.id_emp **and** p.date='171298' **and** p.control='SM' **and** p.status='A' **and** f.sex='V' **and** p.hour **in** (**select** hour **from** control_employee p, employee f, h_employee h **where** p.number_fic=h.number_fic **and** f.id_emp=h.id_emp **and** p.id_emp=f.id_emp **and** p.date='171298' **and** p.control='SM' **and** p.tipe='A0' **and** h.remaindert=0))**and** p.date='151298' and p.number_fic=h.number_fic **and** p.id_emp=f.id_emp **and** f.id_emp=h.id_emp **group by** f.name_emp, p.number_fic, p.date

Database Migration in WAN Environments: How Can It Earn Good Performance? *

Takahiro Hara Masahiko Tsukamoto Shojiro Nishio

Dept. of Information Systems Eng., Graduate School of Engineering, Osaka University
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan
Email: {hara,tuka,nishio}@ise.eng.osaka-u.ac.jp

Abstract. In broadband networks, appropriate use of database migration can drastically shorten the average transaction processing time of a distributed database system. In previous work, we have proposed a transaction processing method based on database migration, in which we assumed that the network bandwidth is very broad and almost uniform over the whole network. However, WANs consist of various types of networks, including private LANs, narrowband public networks and broadband backbones; thus, that method cannot be applied directly in WAN environments. In this paper, we propose a database migration scheduling method which gives the shortest communication time for database operations, and is suitable for use in WANs. We also prove that the proposed method can find the database migration schedule which gives the shortest communication time for a given access sequence.

1 Introduction

The recent evolution of broadband networks has had a significant impact on the design of database management systems[2, 3, 9, 10]. For distributed database systems in narrowband networks, the minimization of the transmitted data volume is considered as the primary factor in performance improvement. However, in the case of systems in broadband networks, the effective use of the network is a much more significant factor.

Here, the problem is how broadband networks can be used effectively. A possible answer is the migration of databases from site to site through networks. We call such migration *DB-migration*[6]. DB-migration can be performed in a short time period in broadband networks. Therefore, dynamic database relocation using DB-migration can be used for several purposes, including *transaction processing*[6, 7].

In a conventional distributed database system, each database is fixed at a particular site, and a typical database operation is performed through several *operation request messages*. After the message exchange, the operation is validated

* This research was supported in part by Research for the Future Program of Japan Society for the Promotion of Science under the Project "Advanced Multimedia Content Processing (JSPS-RFTF97P00501)".