DBMS Evaluation using Quantitative Methods

A. Dasso, A. Funes, M. Peralta, C. Salgado, D. Riesco, G. Montejano, R. Uzal

{arisdas, afunes, mperalta, csalgado, driesco, gmonte, ruzal}@unsl.edu.ar

SEG
Universidad Nacional de San Luis
Ejército de los Andes 950
D5700HHW San Luis
Argentina
http://sel.unsl.edu.ar/
Tel.: +54 (0) 2652 42 4027 ext. 251
Fax: +54 (0) 2652 43 0224

Abstract

Software tools evaluation and comparison is a must for users. Most of the time this is done in an incomplete an informal manner without regards for the economics involved. We present an ongoing project that evaluates different families of software using the Logic Scoring of Preference (LSP) method. This method is very briefly presented and also some of the ongoing evaluation work directed to Data Base Management Systems (DBMS) is explained.

Introduction

Evaluating families of software tools such as Data Base Management Systems (DBMS), programming languages, web browsers, operating systems, etc., is done to choose one particular software among several possibilities or simply to assert one piece of software against others.

Although this activity can have a great economic impact it is not always carry out with the care it should. There are several methods to do this evaluation ranging from the most informal to the more careful and formal, from the simpler form based on the personal opinion of evaluators, to the one that using the opinion of evaluators or users can construct a list of desired characteristics of the software and then analyse them against those characteristics, particularly assigning numerical values for the satisfiability of every desired characteristic for every software being evaluated. The result of this assignment can be a simple addition or more complex and sophisticated methods can be used.

One of them is the Logic Scoring of Preference, which is the method we have adopted, to evaluate different families of software: web browsers, web programming languages and others to come. For more information on the method see [DUJ96], [DuBa97] and [DuEl82].

Past and Current Work

We have already used the method to evaluate web browsers [FDD00] and also web programming languages [DPS03], as well as in the human resources field [DDF03]. We have constructed a list of desired characteristics for both of these evaluations and then used the LSP method to aggregate them and obtain results.

LSP is a method for the realization of complex criterion functions and their application in the evaluation, optimisation, comparison and selection of general complex systems.

As a starting point in the LSP method, it must be clearly determined what are the user requirements, the main attributes of the system and their preference values. These attributes are called performance variables. Each one of these variables is mapped into an elementary preference by defining and applying the corresponding elementary criteria.

In order to develop an exhaustive list of requirements, a hierarchical decomposition process for requirement derivation is applied. At the beginning all major groups of requirements are defined, and then through successive decompositions each group is decomposed into subgroups. By repeating this process the system requirement tree is obtained. The tree leaves correspond to the performance variables.

Elementary criteria are functions that transform real values from a performance variable into a value called elementary preference, which belongs to the [0,1] interval. They represent the degree of fulfilment of the requirements. Therefore, to define the different elementary criteria is necessary to have some previous experience to determine what is the range of acceptable values for each performance variable.

The elementary preferences are used as input for the LSP criterion function. This function yields a single global indicator of the degree of fulfilment of the system requirements. The LSP criterion function is built by aggregating the elementary preferences. To aggregate preferences means to replace a group of preferences (the input preferences) by a single preference (the output preference). It denotes the degree of satisfaction of the evaluator with respect to the group of input preferences. The process starts by aggregating groups of related elementary preferences and generating subsystem preferences. Therefore, the elementary preferences, corresponding to the system requirement tree leaves, are aggregated in new preferences, one by each elementary preference parent. This bottom-up process is repeated with the resulting groups of subsystem preferences until a single global preference can be computed.

If we want to aggregate n elementary preferences $E_1,...,E_n$ in a single preference E, the resulting preference E –interpreted as the degree of satisfaction of the n requirements– must be expressed as a function having the following properties:

1. The relative importance of each elementary preference E_i (i= 1...n) can be expressed by a weight W_i ,
2. $\min(E_1,...,E_n) \leq E \leq \max(E_1,..., E_n)$.

These properties can be achieved using the weighted power means:

$E(r) = (W_1 E_1^r + W_2 E_2^r + ... + W_n E_n^r)^{1/r}$, where

$0 < W_i < 1, \qquad 0 \leq E_i \leq 1, \qquad i = 1, ..., n$

$W_1 + ... + W_n = 1,$

$\infty \leq r \leq +\infty$

The choice of r determinates the location of $E(r)$ between the minimum value $E_{min} = \min(E_1,...,E_n)$ and the maximum value $E_{max} = \max(E_1,...,E_n)$. For $r = -\infty$ the weighted power mean reduces to the pure conjunction (the minimum function) and for $r = +\infty$ to the pure disjunction (the maximum function), giving place to a Continuous Logic Preference (CPL). For a more detailed description of the technique for selection of r see [3], [4].

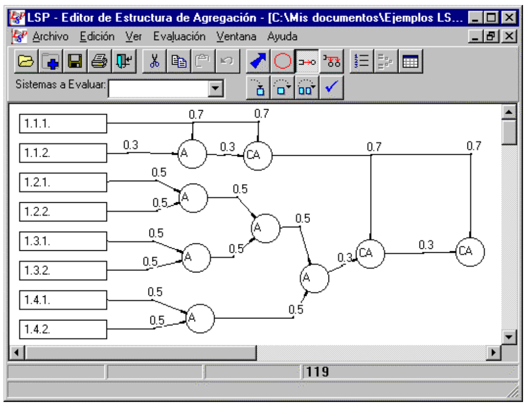
Normally the range between pure conjunction and pure disjunction is covered by a sequence of equidistantly located CPL operators: C, C++, C+, C+–, CA, C–+, C–, C– –, A, D– –, D–, D–+, DA, D+–, D+, D++, D.

In order to perform the evaluation more automatically two of the authors have constructed a tool that implements the LSP method [DFPS01]. An example of a screen of the tool is shown in Figure 1.

Currently we are working in updating the evaluations already done –improving their Preference Tree amongst other things– as well as incorporating other systems from the same family of software to the evaluations. We are currently developing the Preference Tree and working on the desired characteristics that will help us to evaluate different DBMS.

DBMS Characteristics for Evaluation

Some of the characteristics we are using at the moment are Security, with particular emphasis in User Identification and Accesses Permissions. We are analysing the different Authorization Mechanisms implemented for those checks. We are also considering Views as a Protection Mechanism.



One traditional way of evaluating a software system –and in this particular case a DBMS– is through a benchmark. The Transaction Processing Performance Council (TPC) has created a number of benchmarks to evaluate DBMSs and publishes periodically results obtained from these benchmarks. These can be considered as de facto benchmark standards. Using the LSP method we are able to build a model that aggregates the metric results produced by the TPC benchmarks in one number. We can also modify that model to adapt it to different DBMSs user's requirements.

We are also considering the platform used by the different DBMS and obviously their availability and costs. Cost Analysing is an important side of every evaluation and sometimes is done separately due to the complexity of the subject.

Conclusions and Future Work

Up to now we have been gaining experience in the evaluation of different software tools using LSP. We have presented here an outline of our current work in constructing the Preference Tree for DBMS since that is the first step in the evaluation process. Since using the LSP method implies a permanent review process of every step –that can be similar to a spiral model– we expect to continue improving our current model by commencing to define Elementary Criteria and assigning values to the different Performance Variables.

We also plan to continue expanding in the future the use of the LSP method to the evaluation of other software tools.

References

[DDF03] N. Debnath, A. Dasso, A. Funes, G. Montejano, D. Riesco, and R. Uzal, "The LSP Method Applied to Human Resources Evaluation and Selection", Journal of Computer Science and Information Management, Publication of the Association of management/International Association of Management, Volume 3, Number 2, 2003, ISBN 1525-4372, pp.1-12.

[DFPS01] A. Dasso, A. Funes, M. Peralta, C. Salgado, “Una Herramienta para la Evaluación de Sistemas”, Workshop de Investigadores en Ciencias de la Computación, WICC 2001, Universidad Nacional de San Luis, San Luis, Argentina, May 2001.

[DPS03] N. Debnath, M. Peralta, C. Salgado, A. Funes, A. Dasso, D. Riesco, G. Montejano, R. Uzal, "Web Programming Language Evaluation using LSP", Proceedings de CAINE03, Las Vegas, USA, 11-13 de Noviembre, 2003.

[DuBa97] J. J. Dujmovic and A. Bayucan, “Evaluation and Comparison of Windowed environments”, Proceedings of the IASTED Interna Conference Software Engineering (SE' 97), pp 102-105, 1997.

[DuEl82] J. J. Dujmovic and R. Elnicki, “A DMS Cost/Benefit Decision Model: Mathematical Models for Data management System Evaluation, Comparison, and Selection”, National Bureau of Standards, Washington, D.C., No. NBS-GCR-82-374, NTIS No. PB82-170150 (155 pages), 1982.

[DUJ96] J. J. Dujmovic, “A Method for Evaluation and Selection of Complex Hardware and Software Systems”, The 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise Computing Systems. CMG96 Proceedings, vol. 1, pp.368-378, 1996.

[FDD00] A. Funes, A. Dasso, J. Dujmovic, G. Montejano, D. Riesco, R. Uzal, “Web Browsers Performance Analysis using LSP Method”. Proceedings of the International Conference on Software Engineering Applied to Networking and Parallel/ Distributed Computing, SNPD ' 00. Reims, France, May 18-21, 2000.