Expert Systems with Applications 40 (2013) 3313–3327

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Neuro-Fuzzy Expert System for evaluating the performance of Distributed Software System Architecture

B.A. Akinnuwesi [a,*], Faith-Michael E. Uzoka [b], Abayomi O. Osamiluyi [c]

[a] Department of Information Technology, Bells University of Technology, Ota, Ogun State, Nigeria
[b] Department of Computer Science & Information Systems, Mount Royal University, Calgary, Canada
[c] Department of Computer Science and Technology, Bells University of Technology, Ota, Ogun State, Nigeria

ARTICLE INFO

Keywords:
Distributed Software System Architecture
Evaluation
Responsiveness
Performance
Expert system
Organizational variable

ABSTRACT

A Neuro-Fuzzy Performance Evaluation Model (NFPEM) proposed in Akinnuwesi, Uzoka, Olabiyisi, and Omidiora (2012) was reviewed in this work with the view of modifying it and thus making it flexible and scalable. The neuro-fuzzy expert system (NFES) reported in this paper is an enhancement to NFPEM with expert system components. NFES can be used to evaluate the performance of Distributed Software System Architecture (DSSA) with user-centric variables as parameters for performance measurement. The algorithm developed for NFES was implemented using Coldfusion programming language and MySQL relational database management system. The prototype of NFES was simulated using some life data and the performance results obtained point to the DSSA responsiveness to the users' requirements that are defined at the requirements definition phase of the software development process. Thus the performance value is a qualitative value representing DSSA (i.e. system) responsiveness.

1. Introduction

A Distributed Software System (DSS) is a complex system used by organizations to deploy services simultaneously to many people online and in real time (Akinnuwesi, 2011). The decisions made at each phase of the DSS development process impact on the quality attributes (e.g. reusability, reliability, modifiability and performance) of software (Lloyd & Connie, 1998). Performance is a pertinent quality attribute of software systems. It is an indicator of the extent to which software system/components meet the requirements of the end users. Performance failure usually results in damaged customer relations, loss of revenue, loss of productivity and cost overruns due to tuning or redesign of system (Lloyd & Connie, 1998). Therefore it becomes imperative to analyze and predict the expected performance of DSS at the architectural design level in order to: avoid the pitfalls of poor quality of software at system implementation level; provide all organizational services and also satisfy the performance expectations of the stakeholders such that all stakeholders get maximum satisfaction from the software system.

A survey of DSS performance evaluation models was carried out in (Olabiyisi, Omidiora, Uzoka, Victor, & Akinnuwesi, 2010), Olabiyisi et al. (2011) and Akinnuwesi et al. (2012) considering performance of the system at both architectural and implementation levels. The authors deduced that none of the existing models considered evaluating DSSA performance using contextual organizational variables and this informed the development of NFPEM that was presented in (Akinnuwesi, 2011 and Akinnuwesi et al., 2012). NFPEM is a neuro-fuzzy algorithm used to measure performance of DSSA based on 31 contextual organizational variables. Though the model did the evaluation as required but in the course of reviewing it, the following were identified as needed to enhance the functionality of the model: (1) Incorporation of components of expert system; (2) Inclusion of values (i.e. input, intermediate results and final output values) reusable components; (3) Making the model dynamic and scalable such that a performance engineer can define the input contextual variables peculiar to each organization as well as the machine function.

Vlahavas, Stamelos, Refanidis, and Tsoukias (1999) used Multiple Criteria Decision Aid (MCDA) framework to build an expert system for software evaluation. The authors evaluated software performance at the implementation level which is not very ideal because evaluation at the architectural level helps to establish the adequacy of the architecture in meeting organizational requirements before developing it to a complete software system (Samuel, 2006; Samuel & Alejandro, 2003; Simonetta, Roberto, & Moreno, 2004). This helps to minimize the risk of tuning and redesigning the system if it has performance failure at the point of implementation by users.

* Corresponding author.
E-mail addresses: boluakinnuwesi@bellsuniversity.edu.ng, moboluwaji@gmail.com (B.A. Akinnuwesi).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.12.039

In this paper, NFES is developed by adopting expert system principle to modify NFPEM algorithm in order to address the aforementioned limitations in both (Akinnuwesi et al., 2012 and Vlahavas et al., 1999). The rest of the paper is organized as follows: Review of related literature is presented in Section 2. The conceptual design of NFES and its algorithm are presented in Section 3. NFES implementation is presented in Section 4. Some conclusions are drawn in Section 5.

2. Literature review

In this research we carried out a detail review of Vlahavas et al. (1999), Behrouz, Vani, and Abdel-Halim (2009) and Akinnuwesi et al. (2012) in order to establish the system components needed to enhance NFPEM functionalities.

2.1. Vlahavas et al. (1999)

The authors presented an expert system for performance evaluation based on the Multiple Criteria Decision Aid (MCDA) framework with features of flexibility in problem modeling and built-in knowledge about software problem and software attribute assessment. This means that different kinds of problems can be modeled using the system, thus, the type of problem and the attributes to be considered for performance measurement are selected or determined using the expert assistant component. Fig. 1 presents the architecture of Vlahavas et al. (1999).

Performance measurement attributes were broken down into basic and compound attributes. The compound attributes were broken down into sub-attributes while the basic ones cannot be further analyzed. Focus was on quality and cost attributes. The quality attribute being a compound one was further broken down into functionality, reliability, usability, efficiency, maintainability, and portability attributes; each of which is further analyzed into sub-attributes until basic ones are reached and cannot be further analyzed. Scale of measurement is associated with each basic attribute and at the point of evaluation, a value is provided for each attribute of each software product to be analyzed as specified by the software performance engineer. Results of performance evaluation were stored for future reference by the evaluator and determination of attributes for evaluation of other products at a later time. Profiles of its users were maintained. All these were stored in the knowledge base.

The system becomes difficult or complex to use when attributes become much (i.e. >150). It requires the user of the system to have at least software engineering or performance evaluation skills. All sub-attributes that make up a compound attribute may not be determined and redundancy of basic attributes may even occur. Also, it is not subjective enough because attribute values are determined by the evaluator rather than by a collection of opinions from the different users of the product. The primary aim of software performance evaluation is to determine if requirements specification defined by the client organization (end users) is met. The approach defined in Vlahavas et al. (1999) investigates performance at end users implementation level of the software product and not at the architectural level where performance evaluation is considered very necessary.

2.2. Behrouz et al. (2009)

The authors proposed a system similar to that of Vlahavas et al. (1999) but it is based on the Multidimensional Weighted Attribute Framework (MWAF). They assigned weights and values to attributes and applied the principles of Tukey's pairwise comparison and analysis of variance (ANOVA) tests to them. The systems presented in Behrouz et al. (2009) and Vlahavas et al. (1999) both have the same limitations. Behrouz et al's architecture is presented in Fig. 2.

2.3. Akinnuwesi et al. (2012)

The need for evaluating performance of DSSA using user-centric variables was established in Olabiyisi et al. (2010) and Olabiyisi et al. (2011) and thus NFPEM was proposed in Akinnuwesi et al.

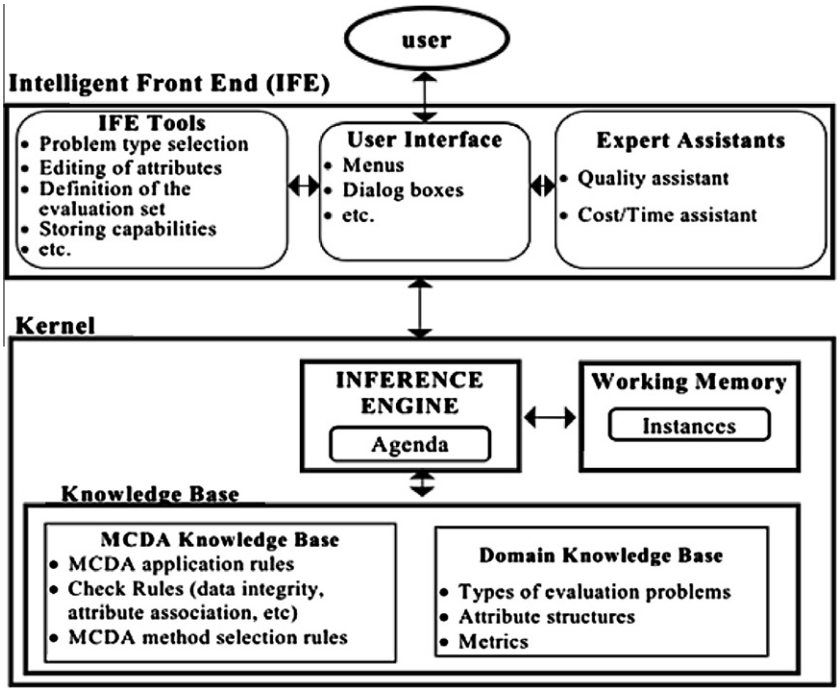

Fig. 1. The structure of ESSE (Vlahavas et al., 1999).