
Preface

G. Zayaraz

Department of Computer Science and Engineering,
 Pondicherry Engineering College,
 Puducherry 605014, India
 E-mail: paperssaetep@gmail.com
 E-mail: gzayaraz@pec.edu

Biographical notes: G. Zayaraz is currently working as Professor in Computer Science and Engineering at Pondicherry Engineering College, Puducherry, India. He received his Bachelor's, Master's and Doctorate degree in Computer Science and Engineering from Pondicherry University. He has published more than 45 research papers in reputable international journals and conferences. His areas of specialisation include software engineering, software architecture, architecture evaluation, mathematical/statistical analysis of algorithms and Information security. He is a reviewer/editorial member for several reputed international journals and conferences and Life Member of CSE, and ISTE.

We received an overwhelming response to the Call for Papers for this Special Issue on Software Architecture, Evaluation and Testing for Emerging Paradigms. This special issue aims at presenting the current trends in software engineering, ranging from software architecture evolution to testing methodologies for emerging paradigms. The issue will provide an opportunity for academic and industry professionals to discuss recent progress in software engineering processes, techniques, methodologies employed, tools developed, overall assessment of benefits, risks, mitigation and testing processes.

Forty papers from different countries were received, including Egypt, Italy, China, Greece, etc. Ten eminent reviewers, most of them doctorates specialised in software engineering, helped us to review the papers. A three-stage process of review was done for each paper received. The papers received were pre-screened for suitability of content before a plagiarism check. Then, the papers were checked for plagiarism using the Plagiarism Detector V.10 tool. The papers that had more than 95% original content were sent for double-blind review. The reviewers have selected 10 excellent papers from the 40 papers received. We would like to thank all the reviewers at this juncture for their valuable suggestions and support. The selected papers cover major critical areas of software engineering such as impact analysis, testing and architecture design

A new approach to estimating impact analysis and reducing the number of test cases selected for regression by allocating tokens to the changes in the system is presented. The proposed generic and resource-aware test platform

- looks for a minimal set of tests to re-execute written in the TTCN-3 standard notation

- assigns the involved test components in execution nodes while respecting resources and connectivity constraints
- finally performs runtime tests with reduced interference risks by supplying different test isolation strategies depending on testability degrees of each component under test.

To ensure the correctness and trustworthiness after dynamic evolutions in distributed component-based systems, runtime testing is used as an online validation technique. A generic and resource-aware test platform is proposed to improve the test effectiveness and efficiency. First, it computes the affected components by these changes and looks for a minimal set of tests to re-execute written in the TTCN-3 standard notation. Secondly, it assigns the involved test components responsible for executing the selected test cases in execution nodes while respecting resources and connectivity constraints. Finally, it performs runtime tests with reduced interference risks by supplying different test isolation strategies depending on testability degrees of each component under test.

A software architecture that uses aspect-orientation methodologies to integrate applications with appropriate services available on several clouds is proposed. It allows storage-related code within applications to be redefined and provide proper interaction with services available on several clouds. The pointcuts for intercepting calls are implemented for I/O operations. The application-related concerns remain well-separated from cloud-based constructs, hence obtaining modularity. Once weaving has been performed, the execution seamlessly connects local application-related logic and server-side cloud-based code.

An exhaustive survey on load and stress testing is presented. The related solutions to both load and stress testing issues in different fields and emerging paradigms are enumerated. The advantages and shortcomings of each approach are analysed.

Unanticipated changes are changes that appear once the software is delivered. The ontogenetic approach deals with the two types of change in the same way at the running level. The extraction of anticipated changes by considering the goal of extending the RUP through a discipline and deriving associated artefacts to support the ontogenetic approach is proposed. The article shows how three approaches to requirement engineering (features wheel method, goal-oriented requirement engineering and use cases) were combined to support a new discipline in RUP and produce change cases, which are essential artefacts for the modelling of anticipated changes and their handling in subsequent phases.

High-level Petri nets are used to define a web service modelling technique using the G-net formalism. An expressive algebra that permits the combination of the modelled services is proposed. To take benefit from the Maude formal verification tools, a method that permits the translation of a modelled service from a G-net model to an equivalent Maude specification is suggested.

A hybrid automata (HA) model is combined with the amend Muller automata (AMA) and push down automata (PDA) to verify the composed web services for both deterministic and non-deterministic systems. Departed transition and deadlock evasion algorithm has been proposed. The proposed algorithm is implemented and the composed services are validated using real time examples.

A work inspired by bat algorithm is presented. It is ideal for optimising mathematical problems of NP hard complexity. Given a particular software state chart diagram (UML), the 'bats' perform random 'flights' (random walk) and through probabilistic and heuristic-based optimal solutions, and arrive at the global optima. After each random walk, a fitness function decides how 'fit' a particular state is and that is compared with other local optima. This solution works for all possible state diagrams, including those with multiple initial and final states.

Web services allow different applications to talk to each other and share data and services among themselves. With the proliferation of web services, it is necessary to examine the functional and the non-functional aspects (QoS) of a service for efficient service selection. A match-making algorithm for functional aspects and an enhanced Fuzzy Topsis method for web service selection are proposed. The QoS parameters considered for selection are availability, response time, reliability and accessibility. Furthermore, a dynamic replication algorithm has been used to monitor the availability of a service and replicates based on the service level agreement. A diabetes monitoring system has been considered as a case study to investigate the service selection process.

The last two papers contribute towards issues in regression test selection and impact analysis. A novel approach to reduce the number of test cases considering the number of times a module associated with a requirement is changed and the number of times a module has resulted in faults is presented in the first paper. The approach reveals a large percentage of faults in the early stage of regression testing itself. Efforts have been made to employ computed requirement priorities such that regression testing is performed with minimum efforts.