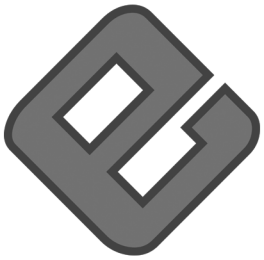
e-Informatica

software engineering journal

2010 | volume 4 | issue 1



e-Informatica

Wrocław University of Technology

Editors

Zbigniew Huzar (*Zbigniew.Huzar@pwr.wroc.pl*)
Lech Madeyski (*Lech.Madeyski@pwr.wroc.pl, http://madeyski.e-informatyka.pl/*)

Wrocław University of Technology
Institute of Applied Informatics
Wrocław University of Technology, 50-370 Wrocław, Poland

e-Informatica Software Engineering Journal
http://www.e-informatyka.pl/wiki/e-Informatica/

Printed in the camera ready form

OFICYNA WYDAWNICZA POLITECHNIKI WROCŁAWSKIEJ
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

ISSN 1897-7979

Drukarnia Oficyny Wydawniczej Politechniki Wrocławskiej. Order No. 418/2010.

Editorial Board

Contents

Editorial

It is a pleasure to present to our readers the fourth issue of the e-Informatica Software Engineering Journal (ISEJ). The mission of the e-Informatica Software Engineering Journal is to be a prime international journal to publish research findings and IT industry experiences related to theory, practice and experimentation in software engineering. The scope of the journal includes methodologies, practices, architectures, technologies and tools used in processes along the software development lifecycle, but particular interest is in empirical evaluation.

The current issue of the journal includes eight papers. The first of the papers by Felkner and Sacha defines formal language that enables handling trust in distributed control systems. The sound and complete deductive system deriving credentials from initial credentials is presented and explained.

The second of the papers by Martenka and Walter is a contribution extending factor-strategy model proposed by Marinescu. It enables more comprehensive and traceable information concerning detected potential anomalies to the designer, resembling the human way of cognition.

The third of the papers by Pahl and Barrett presents a modelling and transformation technique for service-centric distributed systems. Authors capture behavioural aspects and associates quality of architectural structures at different levels of abstraction through patterns. Positive effect of the technique application is illustrated by a case study including design, maintenance and evolution of a system that has been developed by more than 20 people and maintained for more than ten years.

The objective of the fourth paper by Yu is to understand the changing patterns of software complexity. Common coupling is a measure of the system complexity but also it gives insight into software flexibility. How the coupling changes with the evolution of a software system is the subject of study on Apple Darwin, an open-source operating system.

The fifth paper by Hnatkowska and Kasprzyk proposes an approach to business logic implementation that enables easy response to business rules changes. Separation of business logic layer from business rule layer by introducing an integration layer is the core of the idea. The proof-of-concept implementation of the integration layer is presented in the aspect oriented language.

The sixth paper by Rychlý is an interesting application of Milner's π-calculus to describe behaviour of components in service-oriented architecture. A case study of the architecture for functional testing of complex safety-critical systems is presented.

The seventh paper by Staron and Meding presents methods for constructing prediction models of trends in defect inflow in large software projects. Two models are considered. The first one, so called short-term prediction model, is used to predict the number of defects discovered in the code up to three weeks in advance. The second one, long-term prediction model, provides the possibility of predicting the defect inflow for the whole project. The initial evaluation of these methods in a large software project at Ericsson shows that the models are sufficiently accurate and easy to deploy.

In the last paper Alhroob, Dahal and Hossain present a new technique of test cases generation extending the Integrated Classification Tree Methodology. The stress is put on extraction of legitimate test cases by removing the duplicate test cases and those incomputable with the software specifications. Large amounts

of time would have been needed to execute all of the test cases; therefore, a methodology is aimed to select the best testing path which guarantees the highest coverage of system units and avoids using all generated test cases.

We look forward to receiving quality contributions from researchers and practitioners in software engineering for the next issue of the journal.

Editors
Zbigniew Huzar
Lech Madeyski