
Editorial

Saqib Saeed*

Department of Computer Science,
Bahria University,
Islamabad, Pakistan
Email: saqib.saeed@gmail.com
*Corresponding author

C.R. Rene Robin

Department of Computer Science and Engineering,
Jerusalem College of Engineering,
Chennai 100, Tamil Nadu, India
Email: crrenerobin@gmail.com

Biographical notes: Saqib Saeed is an Assistant Professor at the Computer Science department of Bahria University, Islamabad, Pakistan. He has a PhD in Information Systems from University of Siegen, Germany, and a Masters degree in Software Technology from Stuttgart University of Applied Sciences, Germany. He is also a certified software quality engineer from the American Society of Quality. His research interests lie in the areas of human-centred computing, computer-supported cooperative work, empirical software engineering, and ICT4D.

C.R. Rene Robin is currently working as Professor and Head of the Department of Computer Science and Engineering, Jerusalem College of Engineering, Chennai, India. His research interests include artificial intelligence, knowledge management, ontological engineering, and e-learning. He is a recognised supervisor of Anna University, India, to guide PhD and MS (by research) scholars. He is a doctoral committee member of Sathyabama University and VIT Chennai. He is an active Professional Member of ACM. He received the Mentor Award from IBM for TGMC 2011. He got the Young IT Professional Award 2012 (Special Mention) from the Computer Society of India. He received the Award for Teaching Excellence at Indo Global Education Summit 2013.

Software development is a complex activity and most programming languages support only one or a few programming paradigms such as concurrent, object-oriented or functional ontologies. As a result, software developers may have to mix and match between different programming paradigms in different programming languages to accomplish their task, which can be tedious. The addition of ontology paradigms into mainstream programming languages has improved the efficiency of implementing business domains. This special issue explores the development of meta-domain-specific language in current mainstream programming languages. The papers in this special issue have been selected based on peer review.

In the first paper, Rao et al. discuss a system that facilitates knowledge sharing among the information systems community and further scholarly activities that require a deeper understanding of the structure of information in the information system domain.

In the next paper, the authors present a novel approach regarding recording of applied ontology changes in a form of change log. A Layered Change Log Model (LCLM) is presented to deal with customisation and abstraction of ontology-based model evolution.

Third paper, titled 'Designing a novel cross ontology measure to extract resources from knowledge base for an adaptive e-learning system', deals with an adaptive e-learning system, which generates a user specific e-learning content based on his knowledge level and by comparing the similarity of concepts between ontologies.

In the next contribution Alias et al. focus on integrating ontology paradigm in a programming language called Clojure. The ontology paradigms are added as a library in Clojure for reusability and code sharing. The basic approach used for integrating the systems is metaprogramming. The developed system will support ontology, functional, object oriented and concurrent paradigms. The entire business process can be represented using the developed library.

The authors of the next paper present an efficient Domain Specific Language (DSL) architecture that allows structured reuse within the automation domain. The DSL architecture and its benefits and drawbacks are also presented.

The paper titled 'Automating ontologies for e-learning' deals with generating domain ontologies for a given portable document format file that caters to the need of e-learning users with an automatic pre-requisite and follow-up

identification facility. The system extracts the concepts using a term frequency inverse document frequency weighting scheme and then determines the associations among concepts using an appropriate algorithm.

In the next paper, Saraswathi and Venkataramanasam focus on the construction of an automatic ontology tree that is capable of dynamically updating information for efficient management of up-to-date information in the domain of mobile phones. The constructed ontology tree can be queried with domain-related queries, and the ontology tree provides the most relevant result for the query. The proposed system is more suited for domains that make frequent updates because it takes special care on dynamically monitoring changes on the web and performing updates on a par with it.

In the next contribution, Zuñiga et al. present a loosely coupled architecture for develop web services composition applications, in which the architecture's modules can be

integrated in different ways, from inter-organisational sources, supporting different levels of automation and different composition methods, for diverse application domains and kinds of users.

The next contribution, by Kachroudi et al., deals with a new cross-lingual alignment method, designed to take into account the goal of dealing with ontologies expressed in different natural languages. The proposed method implements a strategy of a direct alignment based on the use of external resources.

In the last paper, Kreiner and Krallinger present a framework enabling the Tool Smith to provide reusable constraints for a Domain Expert to create a custom constraint set. The work is an extensible constraint framework, supporting the model-based development process.

We are very thankful to Dr. Miguel-Angel Sicilia, members of reviewing panel and Inderscience staff to help in realising this special issue.