

Letter From the Guest Editors

Adaptive Software Systems

Dilma M. da Silva

IBM T. J. Watson Research Center
dilmasilva@us.ibm.com

Fabio Kon

Department of Computer Science
University of São Paulo
www.ime.usp.br/~kon

The highly dynamic nature of contemporary computing environments characterized by networked heterogeneous hardware and software makes conventional software architectures based on static policies and mechanisms inappropriate. The increasing levels of mobility and ubiquity in software systems together with rapid development cycles require new software architectures to provide support for dynamic reconfiguration and adaptation to both changes in user requirements and variations in the execution environment.

An adaptive software system is built in such a way that allows the dynamic reconfiguration of its internal parameters, components, and software architecture. By monitoring its execution environment and by applying optimizing heuristics, the system is then capable of detecting when it should reconfigure itself, what reconfiguration should take place, and how it should be carried out.

An adaptive video streaming application, for example, could monitor the available network bandwidth in both its client and server sides, the network latency between a specific client and its server, the availability of memory and CPU in the server and the number of concurrent clients accessing the same server. By processing this information, an “intelligent” video server could be able to customize its output streams by selecting appropriate parameters for codifying the video and streaming its data. It could, for example, dynamically load a new video encoder/decoder to translate the video from one format to another, transform the video size and color depth, change the packetizing algorithm used to break video frames into network packets, add a delay to synchronize streams sent to different clients, migrate or replicate the video server to locate it closer to its users, adapt the use of resources (e.g., disk, network, memory) to deal with failures and variations on quality of service requirements, etc.

Adaptive software systems require three major concepts to be implemented: monitoring (to collect information about current load and requirements), reconfiguration (to

enable the changes needed to improve the system), and decision making (to use the monitored information to decide which reconfiguration must be made and when).

The wealth of the area of adaptive software systems resides in its links to multiple fields of Computer Science. Various different classes of applications can benefit from adaptation: multimedia, operating systems, middleware systems, databases, Web-based applications, etc. Knowledge from various Computer Science fields can be leveraged to build effective adaptive software systems; this includes fields such as Artificial Intelligence, Computer Theory, Object-Oriented, Knowledge Management, Software Architecture, etc.

Much of the novel developments in this area have been presented in venues such as the International Conference on Configurable Distributed Systems (1995 to 1998), ACM Reflection (1996 to 2001), and more recently in ACM SOSP, ACM OOPSLA, IEEE ICDCS, and ACM/IFIP/USENIX Middleware and its Workshop on Adaptive and Reflective Middleware.

For this special issue, we received 12 submissions that were carefully reviewed by a group of referees that were selected for their current work on adaptive, configurable, and reflective software systems or more specific topics related to certain submissions. This competitive reviewing process led to the acceptance of four high-quality articles that were further enhanced by their authors based on the feedback provided by the referees.

Loques, Sztajnberg, Cerqueira, and Ansaloni describe their work on separating non-functional requirements through high-level contracts associated with architectural descriptions. The contracts are used to guide configuration adaptations during runtime. The approach has been evaluated through three use-cases.

Ballesteros, Guardiola, Leal, Soriano, de las Heras Quirós, Castro, Leonardo, and Arévalo present Plan B, a new operating system that replaces the traditional file abstraction based on read and write operations with a new

notion of typed data containers associated with constraints. The constraints determine which elements in the system can be used together, and they are used to adapt resource allocation and location to match current requirements.

Gill, Mgeta, Zhang, Torri, Krishnamurthy, Pyrali, and Schmidt describe the design and implementation of a dynamic scheduling framework based on the OMG Real-Time CORBA 1.2 specification. This framework is based on a distributable thread abstraction that propagates quality of service (QoS) parameters across nodes in a distributed system. It enforces the scheduling of multiple distributable threads dynamically using standard CORBA middleware. The article then presents experimental results that show how adaptive dynamic scheduling and management of distributable threads can be enforced efficiently. Finally, it presents results from case studies of multiple adaptive middleware QoS management technologies to monitor and control the quality, timeliness, and criticality of key operations in an avionics system.

Costa and Santos advance the area of adaptive and reflective middleware by reporting their ongoing efforts on

dealing with the problem of consistency of the middleware platform as dynamic adaptations are carried out. Their approach is based on using metainformation techniques to provide a uniform way to specify and manipulate software configurations. Platform configuration and runtime adaptation are specified using a small number of building blocks defined in their meta-model. The authors report their experiments with these ideas both in a Python and a Java prototype and analyze the performance of their reflective middleware with regard to the requirements of distributed multimedia applications.

We gratefully acknowledge the great work performed by the following reviewers that helped us to put up this special issue: Alexandre Sztajnberg, Ana Cristina de Melo, Arnaldo Moura, Carlos Maziero, Carlos Morimoto, Cecilia Mary Fischer Rubira, Edson Moreira, Fábio Costa, Francisco José Silva e Silva, Francisco Reverbel, Hernan Astudillo, Leila Silva, Leliane Nunes de Barros, Manuel Román, Marcelo Finger, Marco Gubitoso, Markus Endler, Orlando Loques, Paulo Masiero, Renato Cerqueira, Robert W. Wisniewski, Roberto Bigonha, Rômulo Silva de Oliveira, Rosângela Penteado, and Thais Vasconcelos Batista.