
Editorial

João Paulo A. Almeida*

Departamento de Informática, CT-VII,
Universidade Federal do Espírito Santo,
Av. Fernando Ferrari, s/n, 29060-970 Vitória,
Espírito Santo, Brazil
E-mail: jpalmeida@ieee.org
*Corresponding author

Luís Ferreira Pires and Marten van Sinderen

Centre for Telematics and Information Technology,
University of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands
E-mail: l.ferreirapires@ewi.utwente.nl E-mail: m.j.vansinderen@ewi.utwente.nl

Maarten W.A. Steen

Novay, P.O. Box 589,
7500 AN Enschede, The Netherlands
E-mail: maarten.steen@novay.nl

Biographical notes: João Paulo A. Almeida obtained his PhD from the University of Twente, The Netherlands, in the area of model-driven design of distributed applications. He is currently an Associate Professor of the Computer Science Department of the Federal University of Espírito Santo, Brazil and member of the Ontology and Conceptual Modelling Research Group (NEMO). His research focuses on the model-driven design of enterprise systems. Key research areas include model-driven design, service-oriented design, enterprise architecture, conceptual modelling, business process modelling and enterprise computing. He currently serves as Program Chair of IEEE EDOC 2009.

Luís Ferreira Pires obtained his PhD from the Faculty of Electrical Engineering of the University of Twente, Enschede, the Netherlands, in September 1994. He is currently an Associate Professor of the Faculty of Electrical Engineering, Mathematics and Computer Science of the University of Twente. He has been working on the application of model-driven architecture and semantic web technologies to the design of context-aware applications and services. He is currently involved in the organisation of the Fifth European Conference on Model-Driven Architecture Foundations and Applications (23–26 June 2009, University of Twente, Enschede), as Workshops and Tutorials Chair.

Marten J. van Sinderen obtained his PhD from the Faculty of Computer Science of the University of Twente, Enschede, the Netherlands, in 1995. He is currently an Associate Professor of the Faculty of Electrical Engineering, Mathematics and Computer Science of the University of Twente. He is also a Coordinator of ASSIST, a research program on service architectures and health applications, on behalf of the CTIT Research Institute of the University of Twente. His research interests include context-aware mobile systems, service-oriented architectures, model-driven design and enterprise interoperability. He currently leads the Dutch GenCom U-Care project (IGC0816) on user-tailorable healthcare services for the home environment.

Maarten W.A. Steen is a Senior Researcher and Consultant at the Novay, an independent institute for ICT-related innovation formerly known as Telematica Instituut. His work focuses on methods, tools and techniques for model-driven architecture and development (MDA/MDD) both in a research setting to advance the state of the art and applied to the domains of enterprise systems development, mobile service development and service-oriented architectures (SOA). He is especially interested in metamodelling, model transformation, multi-viewpoint specification and the combination of ontological reasoning and MDD. He holds a PhD in Computer Science from the University of Kent at Canterbury, UK.

1 Introduction

Model-driven approaches to software development have proliferated in recent years owing to the availability of techniques based on metamodelling and model transformations, such as the meta-object facility (MOF) and the query view transformation (QVT) standards. During the same period, service-oriented architecture (SOA) has become the major trend towards the development of distributed applications. SOA is an architecture style for developing distributed applications in which application parts can only interact by referring to each others' services. SOA has many potential benefits, such as service reuse and the possibility of enacting business processes through orchestration, using executable languages like Business Process Execution Language (WS-BPEL). Model-driven approaches allow the separation of implementation issues from the essential purposes of software systems and in this respect, they match the purpose of identifying services and separating them from their implementations advocated by SOA. Therefore, there are good reasons to apply model-driven approaches to the development of services and SOAs. The combination of these two research areas is the main topic of this special issue, which addresses advances in methods and modelling techniques for services and their compositions (including design concepts, languages, metamodels, profiles and specification frameworks); advances in methodological guidance for the model-driven service development process and advances in model transformations (or mappings) targeted to deploy services and their compositions in services platforms.

2 Historical background

Both model-driven development and SOA are relatively new research areas, but the underlying ideas can be traced back to previous and ongoing developments in software system design and distributed computing.

Model-driven approaches stem from the OMG's model driven architecture (MDA), which is an attempt started around 2001 to provide a vendor-neutral approach to address the challenge of business and technology change (OMG, 2003). MDA is based on the idea that diversity of systems is unavoidable and that instead of trying to force acceptance of a single system solution, it is more realistic to pursue agreement on models of systems and transformations between these models. For this purpose, OMG identified three kinds of (viewpoint) model, namely computation-independent models (CIMs), platform-independent models (PIMs) and platform-specific models (PSMs) and possible transformations between these models. On the one hand, model-driven approaches based on OMG's MDA technologies can be seen as the next step in moving software development towards a higher level of abstraction (Frankel, 2003), thus, achieving better focus on application concerns and avoiding error-prone low-level programming. On the other hand, model-driven approaches apply a separation of concerns and provide support for

reconciling these concerns in the development process. The principle of separation of concerns has been characterised and used already for a long time, in different forms (Parnas, 1972; Vissers and Logrippo, 1985; Silva et al., 1995; Melliar-Smith et al., 1997), to master the complexity of systems, shorten development time and facilitate assessment and maintenance. The main contribution of MDA has been the separation of viewpoints corresponding to business, application and technology concerns, respectively and advances in metamodelling and transformation technologies.

The origin of SOA is less clear, since not a single organisation can claim the invention of the term and the definition of the underlying concepts (although, OASIS more recently published a standard that unifies the SOA concepts (OASIS, 2006)). SOA is often seen as an evolution of the client-server architecture. The client-server architecture distinguishes fixed client and server roles, which can interact in a point-to-point manner. SOA extends this structure by supporting discoverable services which are offered by service providers and used by service consumers in a way that resembles a commercial marketplace with matching of capabilities and needs (Papazoglou and van den Heuvel, 2007). SOA can also be seen as a next step in the development of software 'black boxes' that shield internal details of application components, providing in this way coarser level functionality through service interfaces. Services, in this sense, are the next evolution step after objects and components (Dodani, 2004), which have focused on reusing external behaviour and hiding implementation details. SOA stresses interoperability, as opposed to the internal operation of individual components (or objects). The uptake of SOA has been boosted by the success of web services, which consist of a collection of technologies (SOAP/XML, WSDL and UDDI) that nicely support the SOA concepts and principles. Although, web services technologies are still evolving they are already very significant for SOA as they demonstrate its technical feasibility, by providing the technology to implement architectures that are to some extent 'service-oriented'.

3 Some challenges

Some important challenges can be observed with respect to the application of model-driven approaches to the development of services and SOAs:

- 1 extending the scope of model-driven approaches towards the business level
- 2 coping with behavioural aspects from different viewpoints (e.g., choreography and orchestration) at different abstraction levels (e.g., business and technical) in modelling and transformations
- 3 coping with non-functional properties in the development process and propagating these properties to the final implementation.

So far, most model-driven approaches have focused on the creation and transformation of models at the application level (i.e., PIMs) or models at the technological level (i.e., PSMs) or on code generation through transformations. There is still a considerable gap between business understanding and technology design, making it difficult to derive proper technology solutions for a given business problem. Even harder is the challenge to keep the business and technology aligned in the face of constant changes, which could be a benefit of using MDA in this area. Although, service-orientation is already an important advance as a structuring approach, serious complexity challenges remain due to various viewpoints of involved stakeholders and the business-technology abstraction gap (Johnson and Brown, 2006; Wegmann et al., 2008). MDA prescribes that business aspects should be captured in CIMs. However, there is still little agreement on which aspects should be represented in CIMs and on the techniques that should be used to represent and transform CIMs into PIMs (Kherraf et al., 2008).

An important aspect of service-oriented applications is the definition of the service behaviour. Particularly, choreography and orchestration are important to guarantee interoperability and correctness in service interactions (Quartel and van Sinderen, 2007). Model-driven approaches to SOA should therefore be able to capture and represent service behaviours from different viewpoints and at different levels of abstraction. However, considering UML, which is the modelling notation of choice of OMG for MDA, we observe that behaviour can be represented in different and possibly in mutually inconsistent ways and that notions of refinement and transformations are still lacking in these techniques. Since other behaviour modelling languages and formalisms could be applied in the scope of MDA as long as they have a MOF-compliant metamodel, we envisage an opportunity to investigate how other modelling techniques than UML could be applied to support these modelling and transformation needs.

Non-functional properties, such as performance, reliability, availability, etc., should be properly considered and modelled in the development process of service-oriented systems, at different abstraction levels. Models can also be used at runtime, especially if the desired system functionality and behaviour cannot be fully foreseen at design time, as is the case with systems that have to adapt to changes in their environment (Parzyjeglja et al., 2008). Models provide then a semantic base at runtime for evaluating monitored system states and for decision making with respect to corrective measures in case an undesirable state is detected. In this way, non-functional properties can be improved. Taking availability as an example, models of alternative system configurations, especially service compositions when considering SOA-based systems, can be used at runtime to reconfigure the system after system failures.

4 Contents of this special issue

This special issue contains three papers that were submitted as responses to our call for papers and two papers that were submitted as regular contributions to *IJBPM*.

The paper by Brahe and Bordbar, entitled ‘A methodology for domain-specific business process modelling and implementation’, tackles especially the challenge of extending the scope of model-driven approaches towards the business process level. This paper proposes domain-specific languages that allow different experts to specify their aspects of interest using domain concepts to which they are familiar. Transformations are defined between these models, in order to enforce their consistency and speed up the development process. The paper illustrates the proposed approach with a use case example and discusses the prototype tools that have been built to demonstrate the approach.

The paper by Quartel et al., entitled ‘Model-driven design, simulation and implementation of service compositions in COSMO’, addresses particularly the challenge of coping with behavioural aspects of service specifications at different abstraction levels. The paper discusses how services and composition of services can be designed, simulated and implemented, using a unifying framework called COSMO. Special attention is given to the support to behaviour refinement, which starts from a high-level definition of goals for the mediation between services and results in an orchestration behaviour that executes the mediation. The proposed approach is illustrated with an example of the semantic web services challenge.

The paper by White et al., entitled ‘Creating self-healing service compositions with feature models and microbooting’, is related to the challenge of considering non-functional properties and propagating them to the implementation level. The paper discusses a mechanism to improve the availability of distributed applications by defining feature models of their service composition and recover applications after failure by using an alternative configuration in which failed service components are replaced. The paper illustrates the approach with Sun’s Java pet store e-commerce case study, often used for the benchmarking of frameworks for implementing distributed applications. The paper also discusses Reflect, which is a middleware layer that implements the proposed approach.

The papers submitted, as regular contributions to this journal, concern the verification of behavioural aspects of service-oriented applications.

The paper by Amme et al., entitled ‘Advanced verification of distributed WS-BPEL business processes incorporating CSSA-based data flow analysis’, presents a technique to verify the data flow of service orchestrations represented using WS-BPEL using models.

The paper by Lohmann et al., entitled ‘Comparing and evaluating Petri net semantics for BPEL’, introduces two Petri net models for WS-BPEL and compares these models, showing that they have been devised by taking different modelling decisions.

The papers of this special issue present some interesting developments related to model-driven SOAs. They also demonstrate that only partial solutions exist for the major challenges that were identified. In order to achieve the benefits of model-driven approaches and SOA, further work is necessary, involving researchers and practitioners of the respective communities. We hope that readers will find the papers of this special issue useful as well as inspiring.

Acknowledgements

João Paulo Andrade Almeida would like to thank FAPES/MCT/CNPq for the support provided in the scope of the INFRA-MODELA project and FACITEC in the scope of the MODELA project. Luís Ferreira Pires, Marten van Sinderen and Maarten Steen acknowledge the support of the Freeband A-MUSE project (<http://amuse.freeband.nl>), which has been sponsored by the Dutch Government under contract BSIK 03025.

References

- Dodani, M.H. (2004) 'From objects to services: a journey in search of component reuse Nirvana', *Journal of Object Technology*, Vol. 3, No. 8, pp.49–54.
- Frankel, D.S. (2003). 'Software industrialization and the new IT', *MDA Journal*, September, pp.2–13.
- Johnson, S.K. and Brown, A.W. (2006) 'A model-driven development approach to creating service-oriented solutions', *Proceedings of the 4th Intl. Conf. on Service-Oriented Computing (ICSOC 2006)*, LNCS, Springer-Verlag, Vol. 4294, pp.624–636.
- Kherraf, S., Lefebvre, E. and Suryan, W. (2008) 'Transformation from CIM to PIM using patterns and archetypes', *Proceedings of the 19th Australian Conference on Software Engineering (ASWEC '08)*, IEEE Computer Society, 2008, pp.338–346.
- Melliari-Smith, P.M., Moser, L.E. and Narasimhan, P. (1997) 'Separation of concerns: functionality vs. quality of service', *Proceedings of the 3rd Intl. Workshop on Object-Oriented Real-Time Dependable Systems*, IEEE, pp.272–274.
- OASIS (2006) *Reference Model for Service Oriented Architecture 1.0.*, October.
- OMG (2003) *MDA Guide Version 1.0.1*, June.
- Papazoglou, M.P. and van den Heuvel, W-J. (2007) 'Service oriented architectures: approaches, technologies and research issues', *VLDB Journal*, Vol. 16, pp.389–415.
- Parnas, D.L. (1972) 'On the criteria to be used in decomposing systems into modules', *Communications of the ACM*, Vol. 15, No. 12, pp.1053–1058.
- Parzyjeglą, H., Jaeger, M.A., Mühl, G. and Weis, T. (2008) 'Model-driven development and adaptation of autonomous control applications', *IEEE Distributed Systems Online*, Vol. 9, No. 11, pp.1–9.
- Quartel, D.A.C. and van Sinderen, M.J. (2007) 'On the interoperability and conformance assessment in service composition', *Proceedings of the 11th IEEE Intl. EDOC Enterprise Computing Conference (EDOC 2007)*, IEEE Computer Society, pp.229–240.
- Silva, A.R., Sousa, P. and Marques, J.A. (1995) 'Development of distributed applications with separation of concerns'. *Proceedings of the IEEE 1995 Asia-Pacific Software Engineering Conference (APSEC'95)*, IEEE, pp.168–177.
- Vissers, C.A. and Logrippo, L. (1985) 'The importance of the service concept in the design of data communications protocols', *Proceedings of the IFIP WG6.1 5th Intl. Conf. on Protocol Specification, Testing and Verification (PSTV'85)*, North-Holland, pp.3–17.
- Wegmann, A., Kotsalainen, A., Matthey, L., Regev, G. and Giannattasio, A. (2008) 'Augmenting the Zachman enterprise architecture framework with a systemic conceptualization', *Proceedings of the 12th IEEE Intl. EDOC Enterprise Computing Conference (EDOC 2008)*, IEEE Computer Society, pp.3–13.