Math. Struct. in Comp. Science (2015), *vol.* 25, *pp.* 480–481. © Cambridge University Press 2014
doi:10.1017/S0960129512000898 First published online 10 November 2014

Preface: Special issue on objects and services

IVAN LANESE[†‡] and DAVIDE SANGIORGI[†‡]

[†] *Focus Research Team, University of Bologna, Italy*
[‡] *INRIA, Sophia Antipolis, France*
Email: ivan.lanese@gmail.com, davide.sangiorgi@gmail.com

Received 30 May 2012

Objects and services are pervasive concepts in modern distributed systems. Objects and services, sometimes generically referred to as components, represent the unit of interaction. They offer mechanisms for abstraction and encapsulation, through a well-defined interface that specifies the way in which a given component can be used from the outside, thus hiding the details of the internal implementation. These features are important for building flexible systems, in which components can be used just by inspecting their interface.

There is not a universal definition of objects and services, and of their difference. Intuitively, when speaking of a service, there is a strong emphasis on network and on network interactions, with the consequence that typical issues of distributed systems become more important, notably heterogeneity, platform-independence, service-discovery, coordination and transactions.

Recent years have seen a lot of research on the foundations of objects and services, or more broadly, of component-based distributed systems. In the light of this, we issued a call for a special issue in Mathematical Structures in Computer Science on the topic. The call has been distributed on various mailing lists and has resulted in eight accepted papers, all published in this issue of the journal.

Classifying the presented papers on the problems tackled, we see that two papers deal with questions related to protection and security; three papers describe structured forms of communications, that is, mechanisms for specifying protocols of interaction among distributed components, including transactional mechanisms; another paper is about adaptability and code re-use. Below we briefly sketch the content of each paper.

The paper ‘Abstract Delta Modelling’, by Clarke, Helvensteijn and Schaefer, presents delta modelling: an approach to facilitate automated product derivation for software product lines. A software product line is a set of software systems with clearly-exposed commonalities and variabilities. The goal of software product line engineering is to exploit these features so to easily develop concrete products, by well managing code reuse. The paper explores delta modelling from an abstract, algebraic perspective. Further, a mechanism is proposed for resolving conflicts and ambiguity during code reuse.

The paper ‘Security Monitor Inlining and Certification for Multithreaded Java’, by Dam, Piessens, Jacobs and Lundblad, analyses the possibility of enforcing security policies by inlining them in multithreaded Java. Differently from the sequential case, only a class of race-free policies can be inlined. This class is characterized, and inliners for it are defined.

Moving to the papers in the service-oriented area, three of them concentrate on calculi for modelling service systems, and two have a strong emphasis on error handling and long running transactions: activities that either succeed, or they are compensated.

The paper 'cJoin: Join with communicating transactions', by Bruni, Melgratti and Montanari, presents cJoin. A main feature of cJoin is that when two transactions interact they are merged, thus they either fail or succeed together.

The paper 'Attribute-Based Transactions in Service Oriented Computing', by Bocchi and Tuosto, considers a similar problem: how services enter a transactional scope upon invocation. The approach is inspired by Java Transaction API, but it is investigated at the level of process calculi. This permit the development of sound typing and observational theories.

The paper 'CaSPiS: A Calculus of Sessions, Pipelines and Services', by Boreale, Bruni, De Nicola and Loreti, presents CaSPiS, a calculus for modelling services based on the concept of binary session. Pipelines are used for orchestrating different sessions. The problem of smoothly closing sessions in case of errors is also considered.

The paper 'Model checking Usage Policies', by Bartoletti, Degano, Ferrari and Zunino, is in the area of security. Usage automata are proposed as a model for specifying policies on the usage of resources. The main contribution of the paper is a model checking technique for usage automata. Interestingly, even if the model is infinite state, a polynomial-time technique is found and presented.

All papers have been reviewed according to the normal standards of Mathematical Structures in Computer Science. We would like to express our deep gratitude to both the authors and the referees for their effort.

https://doi.org/10.1017/S0960129512000898 Published online by Cambridge University Press