Editorial

Specification architecture

K.J. Turner a, G. Scolo b

a University of Stirling, Stirling, Scotland, United Kingdom
b University of Twente, Enschede, The Netherlands

Specification architecture

What is architecture? The word appears frequently in a computing context such as system architecture, communications architecture or processor architecture. The origin of architecture is of course in civil engineering: “a method or style of building” [10]. Etymologically an architect is a master builder. Writings on architecture make it clear that the subject combines both science and art: “architecture is frozen music” [9].

Is such a view of architecture applicable in computing? Some might claim that the term architecture is too grandiose for computing, preferring the term structure. However, structure refers rather clinically to the mere organisation of parts. As in building, there should be more – a method and style, even elegance and artistry. Architectural differences cause us to admire one computer system but to dislike another. Some analogies between architecture in computing and building are discussed in [8, Chapter 13].

Architecture is used in this special issue in the sense of structure plus style. But the focus here is even narrower: the architecture of specifications. A specification necessarily has a structure; even a textual specification presents ideas in a particular order. The organisation of a specification can significantly affect how easy it is to write, to understand, to analyse or to modify. The architecture of a specification concerns the style in which it is structured. Proper attention to architecture can greatly improve a specification. Specification architecture draws on ideas from systems theory, software engineering, formal methods and traditional building architecture.

The parallel notion of design patterns has become popular in object-oriented methods [2]. Interestingly this was inspired by traditional architectural approaches, particularly in town planning [11]. What specification architecture and design patterns share is an interest in how the components of a system are structured and composed. However, specification architecture leans more towards a high-level view whereas design patterns suggest an implementation strategy for reusable software.

In this special issue, specification architecture is considered in a communications context. That is to say, general concepts and questions are investigated as they arise in the high-level specification and design of open distributed systems. The restricted focus does not materially limit the treatment of specification architecture. Indeed it greatly helps to have a definite application of otherwise abstract methodology, resulting in a coherent collection of concrete case studies.

The papers

Specification architecture is important in any application domain. In this special issue the domain is
communications systems. The papers deal with message switching, ODP (Open Distributed Processing [6]), OSI (Open Systems Interconnection [3]), multimedia communication and telephony. Each paper explores the theme of specification architecture from a different perspective. The approaches are diverse, yet they share a common emphasis on architectural matters. Among the formal methods discussed are ESTELLE (Extended Finite State Machine Language [4]), LOTOS (Language Of Temporal Ordering Specification [5]) and SDL (Specification and Description Language [7]).

Specification architecture illustrated in a communications context sets the scene for this special issue. The concept of specification architecture is introduced and its importance is discussed. It is argued that guidelines are needed to achieve a satisfactory architecture. Modularity, generality and simplicity are chosen as examples of architectural principles. Techniques to support these principles are proposed, and also criteria for judging their successful application. A simple message switch is used as a running example of the principles. As a final example, the paper elaborates the high-level design of a high performance, modular network gateway.

On the role of basic design concepts in behaviour structuring investigates the architecture of behaviour descriptions. The aim is to establish the basis of a design culture for open distributed systems. Design concepts underlie a design model that is coupled to a formal model through architectural semantics. Initially the paper discusses the interaction concept and how it is supported by ESTELLE, LOTOS and SDL. The more fundamental concepts of action and causality are then introduced. The paper shows how these concepts can be used to give well-structured descriptions of behaviour. The example of a distributed banking system illustrates the ideas in practice.

Relating architecture and specification examines the nature of architectural semantics. The need to relate architectural concepts and specification concepts is explained. A direct mapping is discussed, relating the architecture domain and specification domain. An indirect mapping via more fundamental concepts is also proposed. A compromise between these approaches is developed, and worked through using examples from Open Systems Interconnection. It is shown how the architectural semantics can be effectively supported using specification templates. To demonstrate the wider applicability of the approach, the use of specification templates in hardware design is briefly explained.

Applying the architectural semantics of ODP to develop a trader specification discusses an architectural semantics in LOTOS for Open Distributed Processing. Selected concepts from Part 2 of ODP (Foundations) are presented along with how they might be formalised in LOTOS. This leads to a discussion of how formalisations of the ODP viewpoints might be related. As a substantial example of the approach, the architecture of an ODP trader specification is elaborated for the enterprise, information and computational viewpoints.

A specification architecture for multimedia systems in Open Distributed Processing deals with distributed multimedia computing. A computational model closely related to ODP is described and formalised. Quality of service plays an important role in this model. A specification architecture is proposed with a clear separation of requirements and behaviour. This supports stepwise development from the specification of requirements and abstract behaviour to an implementation design. The generic approach is instantiated as a method based on LOTOS for behavioural specification and QTL (Quality of Service Temporal Logic) for real-time requirements. Combined verification procedures are presented. The example of video transmission over a network is used to illustrate the approach in general, and with LOTOS and QTL in particular.

Structural models for specifying telephone systems contrasts two different styles for structuring LOTOS specifications, using telephony as the application domain. In the resource-oriented style, processes represent system components. The style can be said to be object-based and can take advantage of component re-use. By way of contrast, the constraint oriented style is used to express logical constraints on the system. The Plain Old Telephone System is specified in each style, with an emphasis on providing a flexible architecture that supports incremental modification. Each specification is extended with user features for call forwarding, originating call screening and three-way calling. The results are analysed, and the strengths and weaknesses of each style are discussed.
Acknowledgements

The guest editors are grateful to the contributors for their enthusiasm in elaborating the topic of this special issue. The editors also extend their thanks to the reviewers for their careful reading of the papers. The advice and involvement of H. Rudin, a senior editor of the journal, was much appreciated.

References

[2] E. Gamma, R. Helm, R. Johnson and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software (Addison-Wesley, Reading, MA, 1995).

Ken Turner graduated in Electrical Engineering from the University of Glasgow in 1970. He was awarded a Ph.D from the University of Edinburgh in 1974 for his research on Pattern Recognition. Until 1986 he was mainly employed by International Computers Ltd. as a data communications consultant. During this period he specialised in systems architecture, data communications and formal methods, leading to his appointment as Professor of Computing Science at the University of Stirling in 1987. His research interests lie in formalising systems architecture, particularly for communications systems, distributed systems and hybrid hardware-software systems.

G. Scollo, Italian, lives in Twente, The Netherlands, and has an extensive background in concurrency, formal methods, process algebras, communication protocols. His latest endeavour has been to establish a private company, based in Twente, that offers consultancy and engineering on mathematical methods in information technology. His main research interest is in algebraic methods in logic and computing -- principal theme of his doctoral dissertation "On the engineering of logics". He was Program Chairman of AMAST'93, the Third International Conference on Algebraic Methodology And Software Technology, and is a member of the AMAST Steering Committee and an Editorial Board member of the AMAST Series in Computing. For seven years until 1994 he was an Associate Professor with the Faculty of Computer Science of the University of Twente, where he introduced new courses on Abstract Data Types and on Protocol Design in the curriculum. Previous experience includes performance measurement and analysis systems for computer networks, protocol design, system design and implementation for laboratory data acquisition, algebraic specification of OSI services and protocols, and design of the formal description technique LOTOS. Dr. Scollo is author or editor of over 50 scientific publications.