Communicating Process Architectures 2009
P.H. Welch et al. (Eds.)
IOS Press, 2009

Preface

This thirty-second *Communicating Process Architectures* conference, CPA 2009, takes place as part of *Formal Methods Week*, 1-6 November, 2009. Under the auspices of WoTUG, CPA 2009 has been organised by TASS (formerly Philips TASS) in co-operation with the Technische Universiteit Eindhoven (TU/e). This is for the second time – we had a very successful conference here in 2005 and are very pleased to have been invited back.

We see growing awareness of the ideas characterized by *"Communicating Processes Architecture"* and their growing adoption. The complexity of modern computing systems has become so great that no one person – maybe not even a small team – can understand all aspects and all interactions. The only hope of making such systems work is to ensure that all components are correct by design and that the components can be combined to achieve scalability and predictable function.

A crucial property is that the cost of making a change to a system depends linearly on the size of that change – not on the size of the system being changed. This must be true whether that change is a matter of maintenance (e.g. to take advantage of increasing *multicore* capability) or to add new functionality. One key is that system composition (and disassembly) introduces no surprises. A component must behave consistently, no matter the context in which it is used – which means that component interfaces must be explicit, published and free from hidden side-effect. Our view is that concurrency, underpinned by the formal process algebras of Hoare's *Communicating Sequential Processes* and Milner's *π-Calculus*, provides the strongest basis for the development of technology that can make this happen. Many current systems cannot be maintained if they do not have concurrency working *for* them and not *against* – certainly not if multicores are involved!

We have again received an interesting set of papers covering many different grounds: system design and implementation (for both hardware and software), tools (concurrent programming languages, libraries and run-time kernels), formal methods and applications. They have all been strongly refereed and are of high quality. As these papers are presented in a single stream you won't have to miss out on anything. As always, we will have plenty of space for informal contact and do not forget the evening *Fringe Programme* – where we will not have to worry about the bar closing at half past ten!

We are very pleased this year to have *Professor Michael Goldsmith* of the e-Security Group (WMG Digital Laboratory, University of Warwick) as our keynote speaker. He is one of the creators of the CSP model checking program FDR, which is used by many in industry and academia to check concurrent systems for deadlock, livelock and verify correct patterns of behaviour through refinement checks – all before any code is written.

We thank the authors for their submissions and the Programme Committee for their hard work in reviewing the papers. We also thank Tijn Borghuis and Erik de Vink for inviting CPA 2009 to join FMweek 2009 and for making the arrangements with the TU/e.

Peter Welch *(University of Kent)*, Herman Roebbers *(TASS, Eindhoven)*,
Jan Broenink *(University of Twente)*, Frederick Barnes *(University of Kent)*,
Carl Ritson *(University of Kent)*, Adam Sampson *(University of Kent)*,
Dyke Stiles *(Utah State University)*, Brian Vinter *(University of Copenhagen)*.