

## Preface

We are at the start of a new CPA conference. *Communicating Process Architectures 2005* marks the first time that this conference has been organized by an industrial company (Philips) in co-operation with a university (Technische Universiteit Eindhoven). We see that this also marks the growing awareness of the ideas characterized by ‘Communicating Processes Architecture’ and their growing adoption by industry beyond their traditional base in safety-critical systems and security.

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. 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. Of course, this must be true whether that change is a matter of maintenance (e.g. to take advantage of upcoming multiprocessor hardware) or the addition of 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  $\pi$ -*Calculus*, provides the strongest basis for the development of technology that can make this happen.

Once again we offer strongly refereed high-quality papers covering many differing aspects: system design and implementation (for both hardware and software), tools (concurrent programming languages, libraries and run-time kernels), formal methods and applications. These papers are presented in a single stream so you won’t have to miss out on anything. As always we have plenty of space for informal contact and we don’t have to worry about the bar closing at half ten!

We are pleased to have keynote speakers such as Ad Peeters of *Handshake Solutions* and Guy Broadfoot of *Verum*, proving that you can actually make profitable business using CSP as your guiding principle in the design of concurrent systems, be they hardware or software. The third keynote by *IBM* Chief Architect Peter Hofstee assures us that CSP was also used in the design of the communication system of the recent *Cell* processor, jointly developed by *IBM*, *Sony* and *Toshiba*. The fourth keynote talk is by Paul Stravers of *Philips Semiconductors* on the *Wasabi* multiprocessor architecture.

We anticipate that you will have a very fruitful get-together and hope that it will provide you with as much inspiration and motivation as we have always experienced.

We thank the authors for their submissions, the Programme Committee for their hard work in reviewing the papers and Harold Weffers and Maggy de Wert (of *TUE*) in making the arrangements for this meeting. Finally, we are especially grateful to Fred Barnes (of the University of Kent) for his essential technical expertise and time in the preparation of these proceedings.

Herman Roebbers (*Philips TASS*)

Peter Welch and David Wood (*University of Kent*)

Johan Sunter (*Philips Semiconductors*)

Jan Broenink (*University of Twente*)