Communicating Complex Systems

Technical paper for ICECCS 2006

Peter H. Welch, Frederick R.M. Barnes and Fiona A. Polack

P.H. Welch and F.R.M. Barnes are members of the Computing Laboratory, University of Kent, Canterbury, Kent, CT2 7NF, England. email: {phw,frmb}@kent.ac.uk.

F.A. Polack is a member of the Department of Computer Science, University of York, Heslington, York, YO10 5DD, England. email: fiona@cs.york.ac.uk.

Abstract: **This paper outlines a new language and run-time technology offering formalised design and effiecient implementation for highly concuurent dynamic systems. It is based on a careful combination of ideas from Hoare's CSP (giving compositional semantics, refinement and safety/liveness analysis) and Milners's pi-calculus (giving dynamic network construction and mobility). We have been experimenting with systems developing as layered networks of self-organising neighbourhood-aware communicating processes, with no need for advanced planning or centralised control. The work reported is part of our TUNA (*'Theories Undepinning Nanite Assemblies'*) project, in parnership with colleagues from the Universities of York and Surry, which is investigating formal approaches to the capture of *safe emergent behaviour* in highly complex systems. A particular study modelling artificial blood platelets is described. The software technology scales to millions of processes per processor and distributes over common multiprocssor clusters.**

Keywords: **concurrency, modelling, emergent behaviour, safety, occam-pi, CSP, pi-caculus**

Communicating Complex Systems

Peter H. Welch, Frederick R.M. Barnes and Fiona A. Polack

***Abstract*—This paper outlines a new language and runtime technology offering formalised design and effiecient implementation for highly concuurent dynamic systems. It is based on a careful combination of ideas from Hoare's CSP (giving compositional semantics, refinement and safety/liveness analysis) and Milners's pi-calculus (giving dynamic network construction and mobility). We have been experimenting with systems developing as layered networks of self-organising neighbourhood-aware communicating processes, with no need for advanced planning or centralised control. The work reported is part of our TUNA (*'Theories Undepinning Nanite Assemblies'*) project, in parnership with colleagues from the Universities of York and Surry, which is investigating formal approaches to the capture of *safe emergent behaviour* in highly complex systems. A particular study modelling artificial blood platelets is described. The software technology scales to millions of processes per processor and distributes over common multiprocssor clusters.**

***Index Terms*—concurrency, modelling, emergent behaviour, safety, occam-pi, CSP, pi-caculus**

I. Introduction and Motivation

... to be completed ...

Introduction to the TUNA project and the complex systems we are simulating. Also why we're using occam-π [1] to do it — i.e. to support self-organising dynamic systems that we can formally reason about (safety element); and why we would want to reason about them in the first place. Performance details (memory and run-time overheads) and why this enables masssive systems to be realised that give a chance for interesting, complex and unplanned behaviour to emerge.

II. Overview of occam-pi

Software systems programmed in occam-π are built as layered networks of communicating processes. Processes in occam-π are self-contained, interacting with their external environment through communication channels and a few other synchronisation mechanisms (all built from CSP *events*). Combined with strict parallel-usage and alias checks performed by the compiler, this gives occam-π programs strong safety guarantees — that can be established formally using Hoare's CSP [2].

The following subsections provide an overview of the different occam-π mechanisms relevant to this work. The language encompasses much more, however, and we give some hints ...

A. Processes, channels and networks

... to be completed ...

P.H. Welch and F.R.M. Barnes are members of the Computing Laboratory, University of Kent, Canterbury, Kent, CT2 7NF, England. email: {phw,frmb}@kent.ac.uk

F.A. Polack is a member of the Department of Computer Science, University of York, Yorkshire, YO10 5DD, England. email: fiona@cs.york.ac.uk

Something about basic processes, channels and networks. Some integrators in figure 1. Maybe this can be assumed?

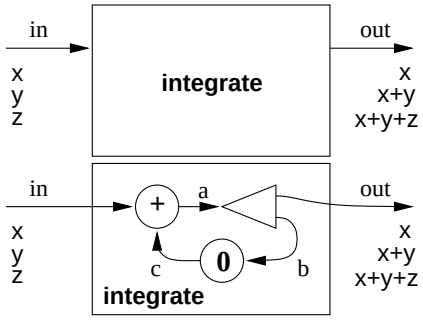

Fig. 1. Serial and parallel integrators

B. Mobile channels, processes and barriers

... to be completed ...

Something about mobile channels (for dynamic construction of process networks), mobile processes (for distributing computation) and mobile barriers (for N-way synchronisation).

C. Dynamic process creation

... to be completed ...

Something about FORKING and FORK (for dynamic growing and shrinking of systems).

III. Simulating Complex Systems

... to be completed ...

How we use the occam-π mechanisms to build simulations – the model of self-contained neighbourhood-aware autonomous agents that roam around a virtual world we create for them. The facilities in occam-π allow us to do interesting things, like changing the 'world' dynamically, including lazy simulation [3]. How we use mobile barriers to time-step the simulation, as well as providing fairly cheap support for visualisation (phased barriers) [4]. Elaborate on how directly such design mirrors the low-level layers of processes (e.g. platelets, chemicals, physical space) in nature and their low-level interactions ... with benefits on the reality of the modelling and its efficiency (processor activity only to support active processes and their immediate environment).

Then talk about the blood-clot simulation and/or implementation of Steve's CSP models.

IV. Formal Design and Analysis

... to be completed ...

Something on the initial design of CSP/B models and their safety/liveness analysis using model checckers. How

these are turned into executable occam-pi (or JCSP) systems.

Something on the CSP models we have for mobile barriers and channels?

V. Conclusions and Future Work

... to be completed ...

What we've done and where we're going.

References

[1] P. Welch and F. Barnes, "Communicating mobile processes: introducing occam-pi," in *25 Years of CSP*, ser. Lecture Notes in Computer Science, A. Abdallah, C. Jones, and J. Sanders, Eds., vol. 3525. Springer Verlag, Apr. 2005, pp. 175–210.

[2] C. Hoare, *Communicating Sequential Processes*. London: Prentice-Hall, 1985, ISBN: 0-13-153271-5.

[3] A. Sampson, P. Welch, and F. Barnes, "Lazy Simulation of Cellular Automata with Communicating Processes," in *Communicating Process Architectures 2005*, ser. Concurrent Systems Engineering Series, J. Broenink, H. Roebbers, J. Sunter, P. Welch, and D. Wood, Eds., vol. 63. Amsterdam, The Netherlands: IOS Press, Sept. 2005, pp. 165–175, iSBN: 1-58603-561-4.

[4] F. Barnes, P. Welch, and A. Sampson, "Barrier synchronisations for occam-pi," in *Proceedings of the 2005 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'2005)*, H. R. Arabnia, Ed. Las Vegas, Nevada, USA: CSREA press, June 2005, pp. 173–179, ISBN: 1-932415-58-0.