**CoSMoS: Complex Systems Modelling and Simulation Infrastructure**
Susan Stepney • Peter H Welch
Department of Computer Science, University of York • Computing Laboratory, University of Kent

**project overview**
CoSMoS is a four year project to develop a modelling and simulation process and infrastructure specifically designed to allow complex systems to be explored, analysed, and designed within a uniform framework.

The project involves 2 RAs and 4 RSs, funded by EPSRC, plus one RS funded by Microsoft Research.

**simulation infrastructure**
CoSMoS includes a simulation framework for massively parallel and distributed systems, as code frameworks in occam-\(\pi\), targeting multiple processors. The current implementation infrastructure supports mobile communications, multi-core processors, and distribution over a cluster of Linux nodes.

We are also investigating FPGA architectures that efficiently support complex systems simulations

**CoSMoS process**
We are developing an integrated process, that guides the probing of a complex system in order to build suitable abstract domain models with the help of domain experts, mapping a domain model to the simulation framework, instantiating the framework to produce a simulation, and using the simulation in an analytic and a predictive manner. A key aspect of the process is validation: building a structured argument of the validity of the various process steps.

**case studies:**
These drive the development. They include “textbook” examples of complex systems, such as flocking, brood sorting, and pheromone trails; and also models and simulations of real biological (immunological) processes.

**occam-\(\pi\) technology:**
CoSMoS implementation exploits occam-\(\pi\), a language based on CSP and the \(\pi\)-calculus. We are developing patterns of usage suitable for complex systems simulation. occam-\(\pi\) enables fine grained concurrency (millions of processes), cache sensitive use of multicore and efficient distribution. Below is a snapshot of an evolving process network, showing mobile blood clot processes in mid-collision (about to combine). A 3D version (visualised right) runs to 40 million processes.

**JCSP technology:**
For mainstream language people, JCSP provides a CSP/\(\pi\)-calculus concurrency package for Java. The model is the same as in occam-\(\pi\) and is easy to learn and use. Because of the compositionality of the underlying process algebra, JCSP logic scales well for complex systems. Currently built on standard Java threads, only thousands (rather than millions) of processes are manageable per JVM.

For more information, see http://www.cosmos-research.org/ or contact info@cosmos-research.org