# Details of Grant

| | | | | | |
|---|---|---|---|---|---|
| EPSRC Reference: | **GR/S86211/01** | | | | |
| Grant Title:: | **New-Generation Symbolic Model Checkers for Verifying Asynchronous Systems** | | | | |
| Principal Investigator: | **Dr G Luettgen** | | | | |
| Other Investigators: | | | | | |
| Recognised Researchers: | | | | | |
| Project Partners: | **College of William and Mary** | | | | |
| Department: | **Computer Science** | | | | |
| Organisation: | **University of York** | | | | |
| Scheme: | **Standard Research** | | | | |
| Starts: | **01 September 2004** | Ends: | **31 August 2007** | Value (£): | **184,746** |
| EPSRC Research Topic Classifications: | **Software Engineering** | | | | |
| EPSRC Industrial Sector Classifications: | **Software** | | | | |
| Related Grants: | | | | | |
| Panel History: | | | | | |

## Summary

This proposal is devoted to the development, implementation, and assessment of new-generation symbolic model checkers for verifying event-based asynchronous systems, such as distributed embedded software. It involves a research collaboration of the PI with the US co-author, for whose activities the US National Science Foundation has recently awarded funding. The project's principal aim is to overcome the severe time- and memorydeficiencies of current BDD-based model-checking techniques when verifying asynchronous systems. This will be achieved by employing MDDs and Kronecker-based data structures for storing state spaces and system transitions, respectively, which will permit one to systematically exploit the event locality that is inherent in asynchronous systems in such a way that model checking only requires local manipulations of these data structures. Key project goals include the devising of parallelisations of model-checking algorithms based on these ideas, their formal proof of correctness, and their implementation in the form of C++ packages that will also be made web-accessible for remote execution. The performance of the developed model checkers will be assessed by extensive benchmarking and by comparison to existing state-of-the-art model checkers, and our algorithms' practicality will be tested by applying them to a realistic case study involving the formal analysis of human-computer-interaction aspects in aircraft cockpits.

| Final Report Summary | |
|---|---|
| Model checking is today's key tool for automatically analysing logic properties of the state spaces underlying complex digital systems. This research project has advanced model-checking algorithms that employ decision diagrams (DDs) as data structure for efficiently storing state spaces, using the Saturation algorithm as prime example. Saturation is aimed at the analysis of asynchronous systems, such as distributed software and protocols, and distinguished in the way it searches through state spaces and manipulates DDs.<br>As its first main outcome, this project has advanced Saturation in several ways. Firstly, it has demonstrated via careful evaluation and benchmarking that Saturation is often orders of mangitude more time- and memory-efficient than competing algorithms. Secondly, it has developed a heuristics that exploits state invariants of systems for achieving compact DD encodings. Thirdly, Saturation has been adapted to `bounded' model checking and shown to be competitive to modern SAT-based model checkers. Together, these results have significantly advanced the practical utility of DD-based model-checking techniques.<br>As its second main outcome, the project has devised, implemented and evaluated parallelisations of Saturation. In contrast to related work, the aim has been to achieve speed-ups on multi-core PCs rather than to utilise the large memory available on PC clusters. Parallel Saturation exploits Saturation's unique way of searching through state spaces and has been implemented twice: once using the Pthreads library for parallel programming, and once using the multi-threaded language Cilk. Both implementations show the significant potential of parallelisation, with up to 50% speed-ups on a quad-core PC for practical examples such as NASA's Runway Safety Monitor. The project has also proposed novel approaches to benchmarking and evaluating parallel algorithms, and conducted its own case study of a Unix file system. Together, these results have significantly increased the knowledge in parallel model checking. | |
| Further Information: | |
| Organisation Website: | **http://www.york.ac.uk** |