**PhD Project Proposal**

School of Electronics, Electrical Engineering and Computer Science

|  |
| --- |
| **Proposed Project Title: Autonomic Management in Parallel/Distributed Systems** |
| **Principal Supervisor: Dr P L Kilpatrick Second Supervisor: Dr D Greer** |
| **Project Description:**  **Introduction**  In modern computing system such as cloud or embedded systems, it is increasingly the case that the application programmer is executing his/her program in an environment to which the programmer has restricted access and consequently limited (if any) control. Thus the programmer is less able to adapt the program to changes in the environment than would have traditionally been the case. Even where access is available, the sheer scale of some HPC environments, with perhaps hundreds of thousands of cores, means that manual intervention in the case of, for example, node/process failure, is difficult or impossible. As a consequence, means must be found by which an application program can manage itself with respect to changes in its environment, or indeed in relation to significant changes to the nature or volume of data being processed. This is particularly the case for long-running programs.  **Problem Area and Approach**  The concept of autonomic management has been introduced, initially by IBM, to address such self-management. A program is equipped with an autonomic manager which monitors and adjusts program behaviour with respect to non-functional concerns such as performance, security, energy usage, etc. Such autonomic management presents a huge challenge, even in the case of sequential programs; for parallel/distributed programs the task is even greater.  In an ideal vision, an application programmer equips his/her program with an autonomic manager and a contract specifying constraints on non-functional concerns of the program; the manager ensures maintenance of the contract throughout the lifetime of the program without human interaction. This is a very demanding target. Work to date at QUB, in collaboration with colleagues in Italy, has provided some early steps toward this vision. The concept of behavioural skeletons has been introduced, that is a parallel pattern/skeleton (such as pipeline, farm, etc.) together with an autonomic manager. The idea is that a programmer constructs a parallel program as a composition of behavioural skeletons. This separates the programming of the core business code from the handling of parallelism (taken care of by the skeleton structure) and the handling of the non-functional concerns (taken care of by the manager).  So far effort has mostly focused on theoretical aspects of the problem, occasionally augmented by ad hoc prototypes. Moreover, little or no thought has been given to software engineering aspects of such an autonomic approach. To monitor and react to changes in non-functional constraints, code instrumentation is required to implant in the application code sensors and actuators which, respectively, capture key metrics and effect changes to parameters and/or structure of the code. Adding this without polluting existing code is a significant research challenge. To address this issue we will draw upon results in Behaviour Driven Design, Coordination Contracts and Aspect Oriented Programming In addition, we may consider if Agent Technology might be pertinent in negotiating adaptions in the face of ‘competing’ concerns (for example, energy v performance).  The work will be carried out in the context of the OpenSource FastFlow [1] framework which provides a stable environment in which to create skeleton-based applications.  [1] http://calvados.di.unipi.it/dokuwiki/doku.php/ffnamespace:about |
| **Contact details**  Supervisor Name: Dr P L Kilpatrick Tel: +44 (0)28 9097 4648  QUB Address: Room 01.011, CSB Email: p.kilpatrick@qub.ac.uk |