Enhancing the Performance Predictability of Grid Applications with Patterns and Process Algebras

Anne Benoit, Murray Cole, Stephen Gilmore and Jane Hillston

THE UNIVERSITY OF EDINBURGH

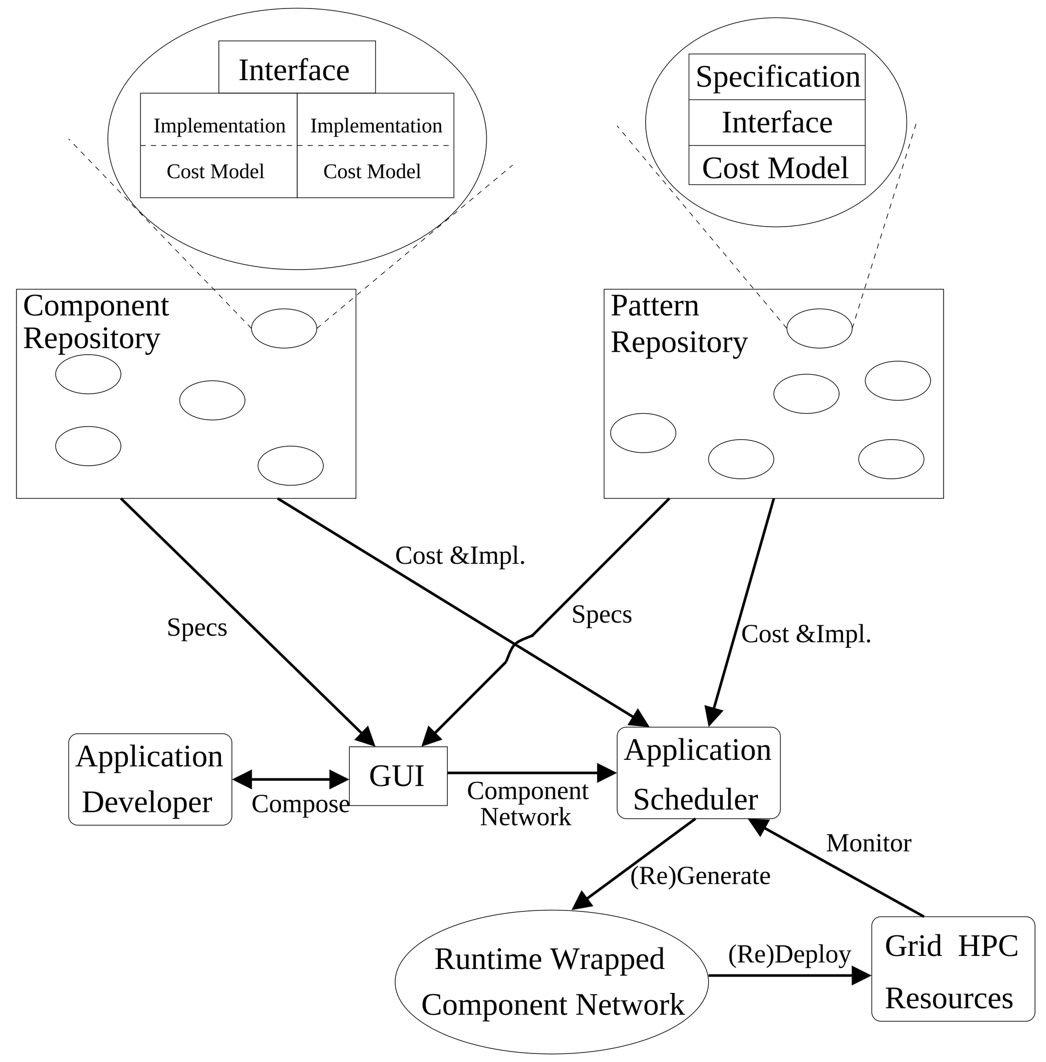
Abstract: The allied problems of programmability, cost predictability and scheduling are exacerbated in a Grid context by the unpredictable and variable availability and performance of resources. The Enhance project investigates the possibility of augmenting a component based Grid High Performance Computing framework with a mechanism for constructing applications as instantiations of pre-defined application patterns. Each pattern encapsulates the logical behaviour of a familiar pattern of multi-component interaction, at a level of abstraction which simplifies the developer's task, without over constraining the selected implementation. The definition and manipulation of semantic and performance behaviour of components and patterns are facilitated by the use of PEPA, a performance oriented process algebra.

The vision of computational and data "Grids" as both tools and enabling fabrics for distributed collaborative computation continues to excite applications designers and challenge the Computer Science research community. While some of the issues raised are familiar from previous environments in distributed and parallel computing, many are new. For example, Grids can be expected to exhibit considerable heterogeneity amongst computational resources and the likelihood of rapidly changing availability and performance characteristics of both computational and communications resources, resulting in a need for rational policies for scheduling and charging for access.

Globus and associated developments such as the Open Grid Services Architecture (OGSA) have already emerged as strong candidates for managing the nuts and bolts of heterogenous resource management, security, information services, and data management. Sitting above this, a number of projects have proposed a component oriented approach to the construction of distributed high performance applications in which existing (or newly developed) HPC applications are packaged as self-contained but composable objects, to be used and re-used in diverse configurations, with the aid of a framework which provides a repository of components and support for mechanisms to combine and distribute them across a Grid.

Two key aspects of any such system are the mechanism through which components are composed and the complementary mechanism through which such compositions are costed, since the former determines the facility with which an application programmer can define new applications and the latter underpins the validity of the scheduling (and run-time rescheduling) decisions made on behalf of the application.

Existing proposals in these areas share the characteristic of allowing arbitrary composition of components (subject to proper typing at interfaces), typically front-ended by a GUI hiding some underlying schematic language. This universality is double-edged. On the one hand it offers maximum freedom to the application developer. Equally, it offers no conceptual support in structuring applications and similarly challenges the cost composition mechanism with the prospect of disentangling arbitrarily complex structures.



The **Enhance** project seeks to address the allied problems of programmability, cost predictability and scheduling by augmenting a component based Grid HPC framework with *a mechanism for constructing applications as instantiations of pre-defined application patterns*. Each pattern encapsulates the logical behaviour of a familiar pattern of multi-component interaction, at a level of abstraction which simplifies the developer's task, without over constraining the selected implementation. The definition and manipulation of semantic and performance behaviour of components and patterns are facilitated by the *use of a performance oriented process algebra* as a modelling tool. In OGSA terms, our project addresses the area of *Workflow services* identified in as "supporting the coordinated execution of multiple application tasks on multiple distributed Grid resources".

Our work underpins the application development process as illustrated. The developer uses selections from a pre-defined set of application patterns to compose selections from a corresponding repository. Selected components and patterns carry stochastic process algebra cost models into an applications scheduler and monitor. This decides, based on the cost models and dynamically acquired performance statistics from the available Grid resources, how best to schedule and re-schedule the application. Our key insights in support of this process are:

- that the use of patterns helpfully constrains the implementation challenge, by providing detailed knowledge of the overall evolution of an application's interactions;
- correspondingly, that patterns allow both more accurate cost prediction (which in turn informs good scheduling decisions) and lighter weight monitoring (because the pattern based model tells us what the crucial aspects are for each instantiation);
- that cost modelling with a formalised, stochastic performance process algebra and associated tools facilitates capture and manipulation of the remaining uncertain aspects of application behaviour.

Working with our colleagues on the ICENI team at Imperial College and within NeSC we aim to demonstrate, by construction of a prototype and experimentation with a number of case studies, that this approach allows simple, intuitive construction of applications, and facilitates runtime scheduling and rescheduling with an effectiveness which surpasses that of less constrained methodologies.

School of informatics