REP report

M. Danelutto (UNIPI) visiting QUB (P. Kilpatrick) (9/03 to 20/03/2008)

The activities performed in Belfast during the REP can be summarized in the framework of two distinct research tracks, all related to the activities of the Programming model Institute: autonomic management of GCM component (assemblies) and semi formal models for the support of grid application design, development and tuning.

Autonomic management of GCM component assemblies.
Within the Programming model Institute, autonomic management strategies for notable composite GCM components (behavioural skeletons) have already been investigated. In particular we (UNIPI, ISTI/CNR and QUB partners of the Programming model Institute) defined the strategies to be used to take care of performance of behavioural skeletons modelling functional replication parallel patterns in an autonomic way. During the REP, I discussed with P. Kilpatrick the problems related to the hierarchical composition of autonomic managers in the GCM behavioural skeleton context. In particular, we discussed how managers belonging to different layers in the behavioural skeleton tree of an application can interact each other to accomplish the global autonomic strategy of the overall grid application. The problem here is how to manage active and passive autonomic behaviour in the different managers and how to decide when a manager should move from active behaviour to the passive one (and vice versa) depending on the interactions with the parent autonomic managers. Several ideas arose from these discussions. We developed a much better view of the problem and we made some progress in the process of formalizing manager interaction.

Semi formal models for the support of grid application design, development and tuning.
Within this second track of activities we made some progress in two distinct directions:

1. *ORC as a semi formal model to support grid/distributed program development*
We prepared the layout of a journal paper summarizing our experience (UNIPI and QUB) concerning the usage of the ORC model from Misra and Cook as a semi formal model supporting reasoning about grid/distributed programs. This experience has been developed during the REP of P. Kilpatrick in Pisa (06-07) and during the REP of M. Aldinucci in Belfast (Nov.-Dec. 07) as well. The paper layout hosts results that have been already published in Conference papers (Europar'07, CoreGRID Sympoisum'07, Parco'07) as well as new examples and research results. In particular, during the REP, we agreed on a new sample case that will be used as the "skeleton" of the whole paper. It will be introduced at the very beginning of the paper, to exemplify the usage of ORC as a modelling tool, then used to show how ORC can be used to derive (in a semi formal way) valid transformations of grid/distributed applications and eventually to explain the possible usages of ORC in combination with proper metadata associated to the ORC model of the computation at hand. This was a missing item in the paper layout that we managed to prepare during the REP. In addition, we discussed how to refine the existing pieces/results already appeared in the conference papers in such a way we can show within the paper the whole semi formal methodology (as incrementally designed in the last year and half) supporting grid/distributed program design. Last but not least, we investigated which kind of journal can be a suitable target for the paper and we identified a couple of possibilities.

2. *Automatic prototyping of ORC modelled applications*.
   We further investigated the possibilities offered by O2J, the prototype we designed and implemented that allows to "implement" ORC code on a workstation network/grid in Java. The prototype, as developed up to now and as described in the paper accepted to the Heraklion CoreGRID IW'08, provides a runt time support (as a JAVA library) that allows users to run ORC processes and sites in a distributed framework. The original prototype was conceived to utilize ProActive as the grid middleware. During the REP, we completely rewrote the remote node run time support to be able to exploit plain TCP/IP grids (with just a single port opened on the involved firewalls) in such a way the ORC code can be run on a grid of processing elements supporting Java (1.5 or later) in a much more "lightweight" way. With the new version of the prototype we prepared a couple of examples to be shown in Heraklion during the O2J work presentation and we managed to have a simulation of the two examples running on a network emulated on a single machine as well, just to be able to have a demo even in case of network failure/absence.

The activities during my REP in Belfast also included the finalization of a submission to CoreGRID Symposium 2008. With P. Kilpatrick (and with the "remote" cooperation of M. Aldinucci) I prepared the paper "Advances in autonomic components & services". The papers discusses how GCM/behavioural skeleton autonomic managers can be suitably implemented exploiting rule based engines and eventually discuss how an implementation of the GCM task farm behavioural skeleton has been ported to the service framework on top of SCA/Tuscany. The implementation of the GCM task farm behavioural skeleton is *de facto* a plain web service that can be accessed through its WSDL from any service context. The task farm service (WorkPoolService) implements a sort of "batch" task processor (tasks should be all independent) and exploits JBoss rule engine to implement autonomic management strategies. The papers builds on top of the partial results already presented to the Programming model Technical meeting held in Paris, January 2008 and on the results that will appear in the PAPP'08 proceedings (June 08, LNCS).

Pisa,March 28th, 2008 M. Danelutto