Graphical Composition of Grid Services

Kenneth J. Turner and Koon Leai Larry Tan

Computing Science and Mathematics, University of Stirling, Scotland FK9 4LA
kjt@cs.stir.ac.uk, klt@cs.stir.ac.uk

Abstract. Grid services and web services have similarities but also significant differences. Although conceived for web services, it is seen how BPEL (Business Process Execution Logic) can be used to orchestrate a collection of grid services. It is explained how CRESS (Chisel Representation Employing Systematic Specification) has been extended to describe grid service composition. The CRESS descriptions are automatically converted into BPEL/WSDL code for practical realisation of the composed services. This achieves orchestration of grid services deployed using the widely used Globus Toolkit and ActiveBPEL interpreter. The same CRESS descriptions are automatically translated into LOTOS, allowing systematic checks for interoperability and logical errors prior to implementation.

1 Introduction

1.1 Motivation

This paper presents a unique blend of ideas from different technical areas: distributed computing, software engineering, service-oriented architecture, and formal methods. Grid computing has emerged as a leading form of distributed computing. However, grid computing has largely focused on the development of isolated applications. Service-oriented architecture provides a framework for combining grid services into new ones.

The emphasis of this paper is on integrating software engineering techniques (visual programming, formal methods) into an evolving application area of considerable importance (grid computing). The aim has been to achieve immediate and practical benefits from advanced software techniques. Grid computing is a comparatively new field that has so far focused mainly on pragmatic, programmatic aspects. The work presented here offers a number of advantages:

- As with component-based approaches, grid services are combined into new composite services using BPEL as an emerging standard for web services.
- Grid service composition is described graphically, making it comprehensible to less technical users. Compared to the automatically generated code, the approach is compact and much more attractive than writing the raw XML that underlies it.
- A sound technique has been defined, benefiting from formal methods behind the scenes yet supporting automated implementation.

The approach is therefore application-driven (orchestrating grid services), novel (combining practice and theory), practical (automated implementation and validation), and integrated (complementing existing grid practice).

N. Guelfi and D. Buchs (Eds.): RISE 2006, LNCS 4401, pp. 1–17, 2007.

1.2 Background to Grid Computing

Grid computing is named by analogy with the electrical power grid. Just as power stations are linked into a universal electrical supply, so computational resources can be linked into a computing grid. Distributed computing is hardly a new area. But the architecture and software technologies behind the grid have captured the attention of those who perform large-scale computing, e.g. in the sciences. Grid computing offers a number of distinctive advantages that include:

- support for virtual organisations that transcend conventional boundaries, and may come together only for a particular task
- portals that provide ready access to grid-enabled resources
- single sign-on, whereby an authenticated user can make use of distributed resources such as data repositories or computational servers
- security, including flexible mechanisms for delegating credentials to third parties to act on behalf of the user
- distributed and parallel computing.

Grid computing is governed by OGSA (Open Grid Services Architecture [8]). Open standards for the grid are being created by the GGF (Global Grid Forum). Grid applications often make themselves available via services that are comparable to web services – another area of vigorous development. For a time, grid services and web services did not share compatible standards. The major issue was the need for stateful services that have persistent state. A grid-specific solution to this was developed. However, this was clearly something that web services could also benefit from.

A harmonised solution was defined in the form of WSRF (Web Services Resource Framework [10]). This is a collection of interrelated standards such as WS-Resource and WS-ResourceProperties. WSRF is implemented by various toolsets for grid computing such as GT4 (Globus Toolkit version 4, *www.globus.org*).

1.3 Background to Service Orchestration

This paper emphasises the *composition* of grid services, not the description of *isolated* grid services. Composing services has attracted considerable industrial interest. This is achieved by defining a *business process* that captures the logic of how the individual services are combined. The term *orchestration* is also used for this. A nice feature of the approach is that a composed service acts as a service in its own right.

Competing solutions were originally developed for orchestrating web services. A major advance was the multi-company specification for BPEL4WS (Business Process Execution Language for Web Services [1]), which is being standardised as WS-BPEL (Web Services Business Process Execution Language [2]). BPEL is now relatively well established as the way of composing web services. However, its use for composing grid services has received only limited attention. The work reported in this paper has used ActiveBPEL (an open-source BPEL interpreter, *www.activebpel.org*).

1.4 Background to CRESS

CRESS (Communication Representation Employing Structured Specification) was developed as a general-purpose graphical notation for services. Essentially, CRESS