

GRID SERVICES FOR COMMERCIAL SIMULATION PACKAGES

Navonil Mustafee

Swansea University
School of Business and Economics
Singleton Park
Swansea, SA28PP, Wales, UK

Simon J. E. Taylor

Brunel University
Distributed Systems Research Group, School of
Information Systems, Computing and Mathematics
Uxbridge, UB8 3PH, UK

ABSTRACT

Collaborative research has facilitated the development of distributed systems that provide users non-trivial access to geographically dispersed resources that are administered in multiple computer domains. The term grid computing is popularly used to refer to such distributed systems. Scientific simulations have traditionally been the primary benefactor of grid computing. The application of this technology to simulation in industry has, however, been negligible. This research investigates grid technology in the context of Commercial Simulation Packages (CSPs). Towards this end, the paper identifies (a) six CSP-specific grid services, (b) identifies grid middleware that could be used to provide the CSP-specific grid services, and (c) list CSPs that include vendor-specific solutions for these grid services. The authors hope that this research will lead to an increased awareness of the potential of grid computing among simulation end users and CSP vendors.

1 INTRODUCTION

Grid computing has the potential to provide users on-demand access to large amounts of computing power to speed up applications (Baker, Buyya, and Laforenza 2002). Simulation modeling is an Operational Research (OR) technique that can benefit from this, as computing power can be a bottleneck in simulation projects (Robinson 2005a). *Discrete Event Simulation (DES)* is arguably the most frequently used classical OR technique that is applied across a range of industries such as manufacturing, travel, finance and healthcare (Hollocks 2006). *Monte Carlo Simulation (MCS)* is another OR technique that is extensively used in application areas such as finance and insurance (Herzog and Lord 2002). *Commercial Simulation Packages (CSPs)* are widely used in industry to facilitate DES and MCS (Tewoldeberhan et al. 2002). Taylor and Robinson (2006) identify the use of grid computing as a priority research area in simulation modeling. The focus of this research is to investigate how simulation users in industry using such CSPs can benefit from grid computing.

The rationale of this research is based on the recognition that the development in simulation has been closely allied to the advances in the field of computing (Robinson 2005a). It can therefore be expected that simulation software will continue to rely on the latest advances in computing to support increasingly large and complex simulations (Pidd and Carvalho 2006). Grid computing is arguably one of the more recent advancements in the field of distributed computing, perhaps superseded only by *cloud computing* (Weiss 2007). The rationale of this research is that, as some previous developments in computing have been adopted by simulation users and they have benefited from it, similarly grid computing technologies might provide an opportunity to further the practice of simulation in industry.

The research has the following four objectives, (a) using relevant literature in grid computing, to identify higher-level grid services that could be used for leveraging the practice of simulation in industry; (b)

identify grid computing middleware that could be used to provide the CSP-specific grid services to simulation end users; and (c) identify the DES and MCS CSPs that have vendor-specific support for some of the identified CSP-specific grid services.

The rest of the paper is structured as follows. Section 2 presents a short introduction to CSPs. Section 3 focuses on grid computing. Section 4 discusses six CSP-specific grid services. This is followed by a discussion on the implementation aspects of the CSP-based grid services in Section 5. Section 6 is the concluding section. It summarizes the contribution of this paper, its limitations and future research directions.

2 COMMERCIAL SIMULATION PACKAGES (CSPs)

This paper investigates the application of grid computing to support the practice of CSP-based DES and MCS in industry. In a DES the behavior of a model, and hence the system state, changes at an instant of time (Brooks, Robinson, and Lewis 2001). Two approaches that can be used to control the flow of time in a DES are the Time Slicing approach, where time is moved forward in equal time intervals, and the Next-Event approach, where time is moved at variable time increments from event to event, i.e., from one state change to the next state change (Pidd 2004). Examples of DES packages include Witness[®], Simul8[®], AnyLogic[®] and Arena[®]. MCS, on the other hand, is a simulation procedure that uses a sequence of random numbers according to probabilities assumed to be associated with a source of uncertainty, for example, stock prices, interest rates, exchange rates or commodity prices (Chance 2004). MCS may be modeled in a visual environment using spreadsheet software such as Excel[®], Lotus 1-2-3[®]; spreadsheet add-ins, for example @Risk[®], Crystal Ball[®]; or through MC-specific simulation packages such as Analytica[®] and Analytics[®].

Swain (2005) made a comprehensive survey of commercially available simulation tools based on the information provided by vendors in response to a questionnaire requesting product information. This list consists of a total of 45 CSPs (12 MCS CSPs and 33 DES CSPs) and features the most well known CSP vendors and their products (Swain 2007). Of these, all the CSPs are supported by the Windows platform, 15.56% (approx.) are supported by UNIX and Linux platforms, and only 13.33% (approx.) are supported under the Apple Macintosh Operating System (Mustafee 2007). Platform support for CSPs is important when considering different grid technologies that can be potentially used with existing CSPs. Platform support for CSPs is explored further in section 3.1.

3 GRID COMPUTING

Computational Grid was defined by Ian Foster and Carl Kesselman in their edited book “The Grid: The Blueprint for a New Computing Infrastructure” as a hardware and software infrastructure that provides access to high-end computational resources (Foster and Kesselman 1998). It was further stated that this access should be dependable, consistent, pervasive and inexpensive. This definition of Grid was further qualified to include resource sharing among various organizations, e.g., the Universities, Government-sponsored research centers and computing centers, that come together to do collaborative work on a problem for a specific time. These organizations are called Virtual Organizations. Thus, Grid computing enabled “coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations” (Foster, Kesselman, and Tuecke 2001). A broader definition of grid computing includes the use of computing resources within an organization for running organization-specific applications. This research is in the context of using grid computing within an enterprise to maximize the use of available hardware and software resources for processing enterprise applications.

Desktop grid computing or desktop grids addresses the potential of harvesting the idle computing resources of Windows-based desktop PCs for processing of parallel, multi-parameter applications which consist of a lot of instances of the same computation with its own input parameters (Choi et al. 2004). In this paper the use of a desktop grid within the enterprise is termed as *Enterprise-wide Desktop Grid Computing (EDGC)*. The discussion in section 2 has shown that all packages are supported on the Windows

platform. This shows the prevalence of Windows-based CSPs in industry. Furthermore, it is a widely accepted observation that employees generally use the Windows-based systems at their workplace. It is therefore arguable that for this research to be widely relevant to the practice of CSP-based simulation in industry, it should, first and foremost, focus on Windows-based grid computing solutions.

3.1 Enterprise-wide Desktop Grid Computing (EDGC)

EDGC refers to a grid infrastructure that is confined to an institutional boundary, where the spare processing capacities of an enterprise's desktop PCs are used to support the execution of the enterprise's applications (Chien et al. 2003). User participation in such a grid is not usually voluntary and is governed by enterprise policy. Applications like Condor (Litzkow, Livny, and Mutka 1988), BOINC (BOINC 2010), WinGrid (Mustafee and Taylor 2009) and WinGrid-WS (Anders 2006; Mustafee et al. 2006) are all examples of EDGC. The reader should note that BOINC is a Public Resource Computing middleware (see section 3.2.2 below). However, when BOINC is used for processing enterprise-applications and is executed within organizational boundaries, it can be considered as an EDGC middleware.

3.2 Middleware for Enterprise-wide Desktop Grid Computing (EDGC)

For the purpose of this paper, the following EDGC middleware will be considered – Condor, BOINC, WinGrid and WinGrid-WS. The reader is referred to Mustafee (2007) and Mustafee and Taylor (2009) for the justification of the criterion used to select the middleware.

(1) Condor: Condor (Condor 2010) is an opportunistic job scheduling system that is designed to maximize the utilization of workstations through identification of idle resources and scheduling background jobs on them (Litzkow, Livny, and Mutka 1988). A collection of such workstations is referred to as a Condor pool. Over the years the functionality provided by Condor has steadily increased. Three key Condor features are Condor Universe, Condor MW and Condor DAGMan.

Condor universe: This is an execution environment for jobs that are submitted by the users. There are several Condor Universes, for example, Java universe (for executing Java programs on Condor) and PVM and Parallel universe (supports the execution of programs written for the PVM and MPI environments).

Condor MW: Condor has a MW (Master Worker) software library that enables users to create master-worker type applications. This C++ library consists of a set of source files that need to be compiled with a user application before the Condor system can be used for the master-worker type computations.

Condor DAGMan: Condor Directed Acyclic Graph Manager (DAGMan) is a workflow management system. It is a meta-scheduler for Condor that operates at a higher-level than the Condor scheduler and manages dependencies between jobs.

(2) BOINC: Public Resource Computing (PRC) refers to the utilization of millions of desktop computers primarily to do scientific research (Anderson 2004). Berkeley Open Infrastructure for Network Computing (BOINC) is the most widely used PRC application that supports scientific projects with diverse objectives such as studying climate change (Stainforth et al. 2002) and improving the design of particle accelerators (LHC@Home 2010). Although BOINC was originally designed to support PRC, the same software can be reconfigured to support desktop grid computing. The widespread availability of desktop PCs in organizations makes the deployment of such an enterprise-wide BOINC infrastructure an even more attractive option. Thus, it may be possible to implement and deploy BOINC-based projects for use exclusively within an enterprise, such that it is geared up to support the execution of the enterprise's applications.

(3) WinGrid: WinGrid is an EDGC middleware that is targeted at the Windows operating system. WinGrid is based on the master-worker distributed computing architecture. WinGrid implements the “push” scheduling approach (master pushes the job to the workers) by starting a server process for each worker. For more information please refer to Mustafee and Taylor (2009).

(4) WinGrid-WS: The architecture of WinGrid-WS (Anders, 2006; Mustafee et al. 2006) extends the original WinGrid architecture through the addition of the WinGrid Shared Repository (WSR). WSR is

server software that needs to be installed on only one desktop grid node. WinGrid-WS implements the “pull” scheduling approach (workers pull jobs from the WSR). For more information please refer to (Anders 2006; Mustafee et al. 2006).

3.3 Basic grid service

Users perceive distributed grid resources as one single system that is capable of processing their computation and data intensive jobs. By logging into one computer, users expect to seamlessly access the underlying grid resources like computing clusters, disk arrays, applications, instruments, databases, etc. Baker, Buyya, and Laforenza (2002) identify the following five basic grid services that can be provided by grids.

- (1) Computation Services: These services allow user jobs (these can be considered as executable programs written by the user) to be run on distributed computational resources.
- (2) Data Services: These services provide secured access to datasets. In order to create the illusion of a mass storage, these datasets can be replicated, catalogued or even stored in different locations.
- (3) Application Services: These services provide access to remote software and libraries. They build on computational and data services that are provided by the grid.
- (4) Information Services: These services use the computational, data and application services to present data with meaning (i.e., information). For example, the simulation output can be visualized.
- (5) Knowledge Services: Data grids can be used to mine for knowledge using data that is present in the databases.

3.4 Higher-level grid services

The basic grid services can be used to offer higher-level, grid-supported functionality to the user applications. Some of these higher-level services are described next.

3.4.1 Parallel computation service

Using multiple grid nodes that are installed with Parallel Virtual Machine (PVM) and / or parallel computing libraries based on Message Passing Interface (MPI), the user is generally able to execute parallel applications over the grid. For example, Huang, Song, and Zheng (2006) have implemented a grid-based parallel visualization service to visualize massive datasets of scientific data in parallel. They have used the MPICH-G2 (Karonis, Toonen, and Foster 2003) implementation of MPI over Globus middleware (Globus 2010) for parallel execution of their application. Thus, it can be argued that they utilize three basic grid services, namely, computational service (for parallel processing), data service (to make available scientific datasets) and application service (for accessing MPICH-G2 libraries installed over different grid nodes), to provide a high-level information visualization service that abstracts the underlying basic grid service. Grid computing middleware that provides parallel computation support to user applications include Globus and Condor (PVM and Parallel universe – *section 3.2*).

3.4.2 Task farming service

Grid computing provides access to multiple computing resources and therefore it is generally possible to execute different applications over various grid nodes. This is different from parallel computation service where one application is executed co-operatively by multiple grid resources. The ability to run different applications concurrently over grids facilitates the execution of applications that are based on the master-worker distributed computing architecture. This architecture (also referred to as task farming architecture) consists of one master entity and multiple workers entities, wherein the master entity decomposes the problem into small tasks, distributes these tasks among multiple worker processes and gathers the partial results to produce the final result of the computation; and the worker entities receive messages from the master with the next task (or request next task from the master), process the task and send back the result to the master (Heymann et al. 2000).

3.4.3 Computational steering service

This is yet another high-level service that can be composed of basic grid services. Unlike traditional non-interactive programs that are executed over the grid, computational steering provides a way for the users to interact with grid applications while they are running (Brooke et al. 2003). This allows a user to steer the execution of a remote application based on the intermediate outputs being generated by it.

3.4.4 Grid-facilitated integration service

Grid computing technologies can be used to integrate previously uncoupled resources and applications such as sensor networks, High Performance Computing (HPC) resources, simulation and visualization applications, distributed datasets, command and control systems, etc. This is referred to as grid-facilitated integration service in this thesis. The FireGrid project, for example, utilizes the integration capability of the grid to develop real time fire emergency response systems (Berry et al. 2005).

3.4.5 Grid portal service

A grid portal is a web-based application that is enhanced with the necessary software to enable it to communicate with grid services made available by the grid middleware (Novotny 2002). It provides the users with higher-level abstraction to the underlying grid services. The web browsers provide an easy-to-use, graphical environment through which the users can interact with the grid middleware. An example of grid portals include is the P-GRADE portal (Németh et al. 2004).

3.4.6 Workflow service

The applications that are executed over grid resources can have dependencies among them. For example, the output of one application can be the input to another application (sequential dependency). Such dependencies between applications can be maintained using workflows and workflow management systems. Workflows are concerned with the automation of procedures whereby files and data are passed between applications following a defined set of rules to achieve an overall goal; and workflow management systems are responsible for defining, managing and executing such workflows over computational resources (Yu and Buyya 2006). Examples of workflow management systems include Condor DAGMan (*section 3.2*) and Taverna (Oinn et al. 2004).

3.4.7 Collaboration service

Grid computing facilitates collaboration among virtual organizations. This collaboration can take various forms. At the most basic level it can be collaboration through co-operative use of grid resources. Collaboration in the grid environment can also take the form of users publishing their user-developed web services (think of these as user applications that can be accessed using standard Internet protocols and open standards) for other users to access. Web services are a web-based technology that is increasingly being used to implement Service Oriented Architectures (SOA). OGSA (Open Grid Services Architecture)-complaint grid middleware like GT-4 (Globus 2010.) usually provide containers (hosting environments) to host the user-developed web services, and provide mechanisms for service providers to register their web services through use of service registries (service publication), mechanisms for service consumers to search for services in the registries (service discovery) and mechanisms to invoke the services when a suitable match is found (service invocation).

Another form of grid-facilitated collaboration could be virtual meeting support provided through integration of audio, video and messaging capabilities with grid middleware. An example of this is the Access Grid Collaboration System. Access grid is primarily meant for group-to-group human interaction

through the use of interactive presentation and software environments, remote visualization environments, large-format multimedia displays, among others (Stevens and FLG 2004).

The next section investigates these higher-level grid services in the context of CSP-based simulation. The purpose here is to identify the higher-level grid services which could be potentially used together with CSPs.

4 HIGHER-LEVEL GRID SERVICES FOR CSP-BASED SIMULATION

The following subsections discuss the six higher-level grid services that can potentially be used together with CSPs. The six services are *parallel computing service*, *task farming service*, *workflow service*, *collaboration service*, *web-based simulation service* (this is the grid portal service discussed in the context of the World Wide Web) and *distributed simulation service*. A *Grid-facilitated integration service* is not investigated because CSPs seldom need integration with physical systems, heterogeneous distributed databases, etc. Similarly, a *computational steering service* is not considered appropriate for further investigation because the user will generally need to access the remotely running graphical CSP interface to computationally steer the simulation, and grid middleware do not generally support such remote visualization of user applications that are being executed over various grid nodes. However, groupware such as Microsoft NetMeeting can be used to provide such access (Taylor 2000). A *Distributed simulation service* is the only CSP-specific grid-service which has not been identified as a higher-level grid service in this paper. This service has been included since literature pertaining to distributed simulation identifies several cases in which simulation practitioner that were involved in, for example, creating large and complex models (Mustafee et al. 2009) or modeling supply chains (Gan et al. 2000), benefitted from CSP-based distributed simulation. Each of the higher-level CSP-specific grid services is now discussed.

4.1 CSP-Specific parallel computing service

Parallel computing is the concurrent use of multiple processors to solve a computational problem in the fastest possible time. Parallel computing service in the grid environment has the potential to speed up the execution of a single simulation experiment using multiple processors. The multiple processors taking part in such a computation may include shared-memory and distributed memory multiprocessor computers, network of workstations, etc. The form of grid computing that has been found suitable for grid-enabling CSP-based simulations is EDGC. The computing infrastructures of such grids are generally made up of a network of Windows-based workstations that do not have access to shared memory. However, parallel programs in a distributed memory environment (like desktop grids) can be run using message passing mechanisms like the MPI and PVM. This generally requires that the grid middleware has support for MPI implementation like MPICH2 and/or PVM environment.

4.2 CSP-Specific task farming service

Task farming service for CSPs has the potential to speed up CSP experimentation using multiple distributed processors. In the context of this research, task farming is defined as the execution of multiple individual simulations (scenarios) on PCs that are connected through the network. It is based on the master-worker distributed computing architecture. Unlike parallel computation service, the objective here is not to speed up the execution of one instance of a simulation but to utilize many computers to complete a set of simulation experiments faster. The task farming service for CSPs can potentially support simultaneous execution of multiple sets of simulation experiments, wherein each set consists of one simulation model with associated experiment parameters. The experiment parameters can consist of values for different model-defined variables like processing time for workstations, number of entities in the queue, model warm-up time, the simulation end time, etc.

4.3 CSP-Specific workflow service

Grid-facilitated workflow service has the potential to logically link the execution of different CSPs and software applications that are available on the various grid resources. In the context of CSP-based simulation, workflows can be used, for example, to potentially enable phased execution of different CSP models that represent different parts of the supply chain. For grid computing to support workflow service, it should ideally be possible for the grid middleware to provide mechanisms to execute multiple programs in a phased manner over different grid nodes and to transfer the data generated by the programs amongst the nodes.

4.4 CSP-specific collaboration service

The term “collaboration” can be defined as the cooperation among different individuals to attain common goals. It can therefore be argued that all the higher-level grid services involve some form of collaboration between the modelers since the desktop grid infrastructure being used for the delivery of grid services is composed of the computing resources that are used by the users at their workplace. Thus, by making their resources available over the desktop grid, each user is contributing towards the overall goal of using grid computing technologies to support simulation at their workplace.

However, in this research, the CSP-specific collaboration service is derived from the grid-facilitated higher-level collaboration service. The three potential uses of this service in the context of CSP-based simulation modeling are, (1) collaboration service can facilitate reuse of model components between different users (through search and download of model components), (2) it can facilitate sharing of CSP models (for joint development purposes), and (3) it can facilitate interaction between those involved in simulation studies (through virtual meeting support). These three forms of collaboration service have also been recognized as potential application areas of simulation in a networked environment by Robinson (2005b).

4.5 CSP-Specific distributed simulation service

A desktop grid middleware that provides distributed simulation support to DES CSPs should generally include mechanisms to enable synchronization of simulation time among different simulation models and to transfer messages between them. The message exchange by models running on multiple desktop grid hosts can be implemented in a *centralized* or a *decentralized* manner. In *centralized* message passing, one central component is responsible for receiving and sending messages from and to different hosts. When each host is responsible for communication with other hosts it is referred to as *decentralized* message passing. Grid computing middleware, such as BOINC and Condor, are not considered appropriate for enabling distributed simulation over a desktop grid because such solutions do not incorporate mechanisms for time synchronization and communication between individually running models (Lüthi and Großmann 2001). The reasons for this are discussed below.

Time synchronization is outside the purview of grid middleware because these are general purpose programs that are designed to support a wide range of user applications, and the vast majority of applications do not require time synchronization mechanisms. Both *centralized* and *decentralized* message passing is also outside the scope of most grid middleware because the focus is usually on executing serial applications over multiple computers. An exception to this is *Condor PVM universe* and *Condor parallel universe*, which support parallel execution through message-passing mechanisms. However, none of these universes have inbuilt time synchronization mechanisms. A distributed simulation middleware may therefore have to be used along with a grid middleware to potentially enable distributed simulation of CSPs over the grid. IEEE 1516 HLA standard (IEEE 2000) is increasingly being used for distributed simulation in industry. As such, this research has considered the grid-facilitated distributed simulation service with reference to HLA-RTI middleware for distributed simulation.

4.6 CSP-Specific web-based simulation service

For the purpose of this research, web-based simulation is defined as simulation in a client-server environment that uses web-based technologies like web browsers, web servers, web services and Java applets, among others. In the context of CSPs, this suggests that the simulation packages are accessible through web pages or through “callable” web services. It is usually possible to create a web-based front-end to a CSP application that exposes package functionality. A simulation user who is able to access a package through a web browser will arguably not have a requirement for a grid-facilitated web-based simulation service. However, this service is only one among six potential CSP-specific grid services. If grid technology is adopted to support the other five services, then it is likely that web-based simulation service will also be used because it standardizes the access to CSPs in a distributed environment.

The two possible ways through which web-based simulation service could potentially support the CSPs are, (1) through use of grid portals and (2) through use of “callable” user-developed web services that expose CSP interfaces and which are hosted in web services containers provided in grid middleware.

5 IMPLEMENTATION OF A CSP-SPECIFIC GRID SERVICE

This section realizes objectives (b) and (c), as mentioned in the abstract and the introduction of this paper. Thus, it provides information on the grid middleware that can be used to provide CSP-specific services to unmodified CSPs (objective b – see *column 5* in *Table 1* below) and it also lists the existing CSPs that have vendor-supported solutions to some of the identified CSP-specific grid services (objective c – see *columns 3 and 4* in *Table 1* below). The data for the latter has been collected from the product information published by the vendors of the CSPs on their websites (Mustafee 2007). With regard to objective (b), *Column 6* in *Table 1* also refers the readers to the published papers that contain more information on the grid middleware-CSP integration approaches that were used for realizing some of the CSP-specific services. Also, middleware/middleware components that could not be experimentally evaluated due to unsupported CSP implementations (like Condor parallel universe execution environment), etc. have been marked for future investigation.

This paper has identified six higher-level grid services that can potentially be used together with the CSPs. As can be seen from columns 3 and 4 of *Table 1* below, only a few CSPs support the identified grid services (for example, *parallel computation service* is supported by only two MCS packages, namely, *@Risk Industrial* and *TreeAge*; no DES CSP supports this). However, some of the CSP-specific grid services can be provided through the integration of grid technology with existing CSPs, for example, *task farming service* is possible through the integration of BOINC, Condor, WinGrid and WinGrid-WS with MCS and DES CSPs (column six refers to particular studies that were conducted in this context).

The columns pertaining to *Table 1* are explained below:

- **Col 1 [CSP-specific services]:** The CSP-specific services.
- **Col 2 [MCS/DES CSP]:** MCS CSPs, DES CSPs or both.
- **Col 3 [CSP support on multiple processor machines]:** CSPs that support CSP-specific services over multi-processor machines using custom solutions. Names of the CSPs are also mentioned.
- **Col 4 [CSP support over distributed processors]:** CSPs that support CSP-specific services over distributed processors using custom solutions. Names of the CSPs are also mentioned.
- **Col 5 [Grid middleware]:** Grid middleware (including specific middleware components like Condor DAGMan, Condor MW, etc.) that have been identified as potential solutions for grid-enabling CSPs with respect to specific services.
- **Col 6 [Comments/References]:** For general comments and for references to published papers that contain the grid-CSP integration solution in question.

Table 1: Custom CSP support and grid middleware support for CSP-specific services (Mustafee 2007)

Col 1	Col2	Col 3	Col 4	Col 5	Col 6
Parallel computation service	MCS	<i>@Risk Industrial, TreeAge</i>	None	(1) Condor parallel universe	<ul style="list-style-type: none"> ▪ (1) is for future investigation -MCS and DES CSPs may need to have MPI/PVM-based implementation
	DES	None	None	(1) Condor parallel universe	
Task farming service	MCS	None	<i>Vanguard Studio, GoldSim</i>	(1) BOINC (2) Condor Java universe (3) Condor MW (4) WinGrid (5) WinGrid-WS	<ul style="list-style-type: none"> ▪ (1) See Zhang et al. (2007) ▪ (2) See Mustafee (2007) ▪ (3) is for future investigation ▪ (4) See Mustafee et al. (2006) ▪ (5) See Alstad (2006)
	DES	<i>Simul8</i>	<i>Simprocess, Simul8</i>	(1) BOINC (2) Condor Java universe (3) Condor MW (4) WinGrid (5) WinGrid-WS	
Workflow service	MCS, DES	None	None	(1) Condor DAGMan (2) WinGrid	<ul style="list-style-type: none"> ▪ (1) is for future investigation ▪ (2) See Mustafee (2007)
Collaboration service	MCS, DES	N/A	None	(1) Access Grid (Stevens and FLG 2004)	<ul style="list-style-type: none"> ▪ (1) is for future investigation
Distributed simulation service	MCS	N/A	N/A	N/A	<ul style="list-style-type: none"> ▪ Distributed simulation is not applicable to MCS CSPs
	DES	None	<i>AutoMod</i>	(1) BOINC with HLA-RTI (2) Condor Java universe with HLA-RTI (3) WinGrid with HLA-RTI	<ul style="list-style-type: none"> ▪ (1) is for future investigation ▪ (2) is for further investigation ▪ (3) See Mustafee et al. (2009)
Web-based simulation service	MCS, DES	N/A	<i>QMS, MineSim, Vanguard Studio, AnyLogic, Agena-Risk, Witness, Analytica, Simprocess</i>	(1) WinGrid-WS (grid portal)	<ul style="list-style-type: none"> ▪ (1) See Alders (2006)

Irrespective of whether or not a bespoke CSP-grid solution can be implemented, from *Table 1* it is aptly clear that there is very little vendor effort towards grid-enabling existing CSPs. Thus, it is worth considering if there is an end-user demand for grid technology for CSP-based simulation or, as has been pointed out earlier - “grid is a solution in search for a problem” (Schopf and Nitzberg 2002), it is being investigated in this research to explore technology-driven possibilities. It is arguable that the suggestion of using multiple networked computers to execute simulations faster is appealing to practitioners, although they may not be aware of the term “grid computing”. This argument is further strengthened by the observations made by the authors during their interactions with simulation end-users. Thus, in the case of distributed experimentation at least, there appears to be some user demand for distributed systems that can support execution of CSP-based simulations on multiple computers. Owing to this demand, simulation package vendor “Simul8 Corporation” has recently added functionality that would allow parallel processing across networks in order to get faster results in CSP SIMUL8[®]. However, the potential of executing experiments in parallel over a network of computers (task farming service) is but one of multiple higher-level services that can be provided through use of grids. The majority of simulation users may be

unaware of these grid-facilitate services, and from this perspective grid computing can be seen as providing a technology-driven impetus to facilitate its possible adoption for CSP-based simulation in industry.

6 DISCUSSION AND CONCLUSION

The contribution of this research is that it identifies six grid-facilitated CSP-specific services that can be potentially provided through the use of specific grid computing technologies. Furthermore, it identifies existing CSPs that provide some of the CSP-specific grid services through custom solutions. A further contribution is the recognition of the form of grid computing, namely Enterprise Desktop Grid Computing (EDGC), which can be used to grid-enable existing CSPs.

One of the limitations of this research is that, although the focus of this research was on end-users who were considered experts in modeling and simulation but were not expected to be IT specialists, the CSP-grid integration technology that has been used in this work requires some knowledge of Java and Visual Basic programming (refer to the references in *Column 6 - Table 1*). Furthermore, the end-users will also need to know the EDGC middleware-specific mechanisms to create jobs, submit jobs, retrieve results, etc. For the wider adoption of grid technology for CSP-based simulation, it may be necessary to develop higher-level tools that would hide the complexity of the CSP-grid integration technology and middleware specific mechanisms, and provide end-users with easy to use graphical interfaces through which they could possibly integrate CSPs with grid middleware. In spite of these limitations, the authors hope that this research will encourage wider understanding of the potential of grid computing among simulation practitioners and the CSP vendors and will motivate further research and commercialization of this technology.

REFERENCES

- Alstad, A. 2006. *Grid system for performance transparency with COTS simulation packages*. Masters thesis. School of Information Systems, Computing and Mathematics Brunel University, UK.
- Anderson, D. P. 2004. BOINC: a system for public-resource computing and storage. In *Proceedings of the 5th International Workshop on Grid Computing*, 4-10. IEEE Computer Society, Washington, DC, USA.
- Baker, M., R. Buyya, and D. Laforenza. 2002. Grids and grid technologies for wide-area distributed computing. *Software - Practice and Experience*, 32(15): 1437-1466.
- Berry, D., A. Usmani, J. Torero, A. Tate, S. McLaughlin, S. Potter, A. Trew, R. Baxter, M. Bull, and M. Atkinson. 2005. FireGrid: integrated emergency response and fire safety engineering for the future built environment. In *Proceedings of the 2005 UK e-Science All Hands Meeting*, 1034-1041.
- BOINC 2010. Berkeley Open Infrastructure for Network Computing project homepage. Available via <http://boinc.berkeley.edu/> [accessed April 28, 2010].
- Brooke, J. M., P. V. Coveney, J. Harting, S. Jha, S. M. Pickles, R. L. Pinning, and A. R. Porter. 2003. Computational steering in RealityGrid. In *Proceedings of the 2003 UK e-Science All Hands Meeting*, 885-888.
- Brooks, R., S. Robinson, and C. Lewis. 2001. Simulation and inventory control. *Operational Research Series*. Hampshire, UK: Palgrave.
- Chance, D. M. 2004. Monte carlo simulation, teaching note 96-03. Available online <http://www.bus.lsu.edu/academics/finance/faculty/dchance/Instructional/TN96-03.pdf>. [accessed April 28, 2010].
- Chien, A., B. Calder, S. Elbert, and K. Bhatia. 2003. Entropia: architecture and performance of an enterprise desktop grid system. *Journal of Parallel and Distributed Computing*, 63(5): 597-610.
- Choi, S., M. Baik, C. Hwang, J. Gil, and H. Yu. 2004. Volunteer availability based fault tolerant scheduling mechanism in desktop grid computing environment. In *Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications*, 366-371. IEEE Computer Society, Washington, DC, USA.

- Condor 2010. Condor project homepage. Available via <http://www.cs.wisc.edu/condor/> [accessed April 28, 2010].
- Foster, I., and C. Kesselman. 1998. *The grid: blueprint for a new computing infrastructure*. San Francisco, CA: Morgan Kaufmann.
- Foster, I., C. Kesselman, and S. Tuecke. 2001. The anatomy of the grid: enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*, 15(3): 200-222.
- Gan, B. P., P. Lendermann, M. Y. H. Low, S. J. Turner, X. Wang, and S. J. E. Taylor. 2005. Interoperating Autosched AP using the high level architecture. In *Proceedings of the 37th Winter Simulation Conference*, ed. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 394-401. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Globus 2010. Globus toolkit homepage. Available via <http://www.globus.org/toolkit/> [accessed April 28, 2010].
- Herzog, T. N., and G. Lord. 2003. Applications of simulation models in finance and insurance. In *Proceedings of the 35th Winter Simulation Conference*, ed. S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 249-257. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Heymann, E., M. A. Senar., E. Luque, and M. Livny. (2000). Adaptive scheduling for master-worker applications on the computational grid. In *Proceedings of the 1st International Workshop on Grid Computing*, 214-227. In R. Buyya, and M. Baker (eds.), *Lecture Notes in Computer Science*, volume 1971, Springer-Verlag, UK.
- Hollocks, B. W. 2006. Forty years of discrete-event simulation - a personal reflection. *Journal of the Operational Research Society*, 57(12): 1383-1399.
- Huang, Z., G. Song, and Y. Zheng. 2006. Proxy-based parallel visualization in a grid environment with PC clusters. In *Proceedings of the 1st International Multi-Symposiums on Computer and Computational Sciences (IMSCC'06)*, 683 - 687. IEEE Computer Society Press, Los Alamitos, CA, USA.
- IEEE. 2000. *IEEE standard for modeling and simulation (M&S) high level architecture (HLA)*. New York, NY: Institute of Electrical and Electronics Engineers.
- Karonis, N. T., B. Toonen, and I. Foster. 2003. MPICH-G2: A grid-enabled implementation of the Message Passing Interface. *Journal of Parallel and Distributed Computing*, 63(5): 551-563.
- LHC@Home 2010. LHC@Home project homepage. Available via <http://lhathome.cern.ch/> [accessed April 28, 2010].
- Litzkow, M., M. Livny, and M. Mutka. 1988. Condor - a hunter of idle workstations. In *Proceedings of the 8th International Conference of Distributed Computing Systems*, 104-111. IEEE Computer Society, Washington, DC, USA.
- Lüthi, J. and S. Großmann. 2001. The resource sharing system: dynamic federate mapping for HLA-based distributed simulation. In *Proceedings of the 15th Workshop on Parallel and Distributed Simulation*, 91-98. IEEE Computer Society, Washington, DC, USA.
- Mustafee, N. 2007. *A Grid Computing Framework for Commercial Simulation Packages*; PhD Thesis. School of Information Systems, Computing and Mathematics, Brunel University, UK. Available online <http://bura.brunel.ac.uk/handle/2438/4009>. [accessed April 28, 2010].
- Mustafee, N., A. Alstad, B. Larsen, S. J. E. Taylor, and J. Ladbrook. 2006. Grid-enabling FIRST: speeding up simulation applications using WinGrid. In *Proceedings of the 10th International Symposium on Distributed Simulation and Real-Time Applications (DSRT 2006)*, ed. E. Alba, S. J. Turner, D. Roberts, and S. J. E. Taylor, 157-164. IEEE Computer Society, Washington, DC, USA.
- Mustafee, N., and S. J. E. Taylor. 2009. Speeding Up Simulation Applications Using WinGrid. *Concurrency and Computation: Practice and Experience*. 21(11): 1504-1523.
- Mustafee, N., S. J. E. Taylor, K. Katsaliaki, and S. Brailsford. 2009. Facilitating the Analysis of a UK National Blood Service Supply Chain Using Distributed Simulation. *Simulation: Transactions of the Society of Modelling and Simulation International*. 85(2): 113-128.
- Németh, C., G. Dózsa, R. Lovas, and P. Kacsuk. 2004. The P-GRADE Grid Portal. In *Proceedings of the International Conference on Computational Science and its Applications (ICCSA 2004)*, 10-19. In A.

- Laganà, M. L. Gavrilova, V. Kumar, Y. Mun, C. J. K. Tan, and O. Gervasi (eds.), Lecture notes in Computer Science, volume 3044, Springer-Verlag, Germany.
- Novotny, J. 2002. The grid portal development kit. *Concurrency and Computation: Practice and Experience*, 14(13-15): 1129-1144.
- Oinn, T., M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li. 2004. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17): 3045-3054.
- Pidd, M. 2004. *Computer simulation in management science (5th edition)*. Chichester, UK: John Wiley & Sons.
- Pidd, M. and M. A. Carvalho. 2006. Simulation software: not the same yesterday, today or forever. *Journal of Simulation*, 1(1): 7-20.
- Robinson, S. 2005a. Discrete-event simulation: from the pioneers to the present, what next? *Journal of the Operational Research Society*, 56 (6): 619-629.
- Robinson, S. 2005b. Distributed simulation and simulation practice. *Simulation*, 81(5): 5-13.
- Stainforth, D., J. Kettleborough, M. Allen, M. Collins, A. Heaps, and J. Murphy. 2002. Distributed computing for public interest climate modelling research. *Computing in Science and Engineering*, 4(3): 82-89.
- Schopf, J. M., and B. Nitzberg. 2002. Grids: the top ten questions. *Scientific Programming*, 10(2): 103-111.
- Stevens, R. and Futures Lab Group. 2004. Group-oriented collaboration: the access grid collaboration system. In *Foster, I. and Kesselman, C. (eds.), The Grid: Blueprint for a New Computing Infrastructure (2nd Edition)*, chapter 15. San Francisco, CA: Morgan Kaufmann.
- Swain J. J. 2005. Gaming reality: biennial survey of discrete-event simulation software tools. *OR/MS Today (December 2005)*. Institute for Operations Research and the Management Sciences (INFORMS), USA. Available online <<http://www.lionhrtpub.com/orms/orms-12-05/frsurvey.html>>. [accessed April 4, 2010].
- Swain J. J. 2007. INFORMS simulation software survey. *OR/MS Today*. Institute for Operations Research and the Management Sciences (INFORMS), USA. Available online <<http://www.lionhrtpub.com/orms/surveys/Simulation/Simulation.html>>. [accessed April 4, 2010].
- Taylor, S.J.E. and S. Robinson. (2006) So where to next? A Survey of the Future for Discrete-Event Simulation. *Journal of Simulation*. 1,1, pp. 1-6.
- Taylor, S. J. E. 2000. Groupware and the simulation consultant. In *Proceedings of the 32nd Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 83-89. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Teweldeberhan, T. W., A. Verbraeck, E. C. Valentin, and G. Bardonnnet. 2002. An evaluation and selection methodology for discrete-event simulation software. In *Proceedings of the 34th Winter Simulation Conference*, ed. E. Yücesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes, 67-75. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Weiss, A. 2007. Computing in the Clouds, *NetWorker*, 11(4): 16-25.
- Yu, J. and R. Buyya. 2006. A taxonomy of workflow management systems for grid computing. *Journal of Grid Computing*, 3(3-4): 171-200.
- Zhang, J., N. Mustafee, J. Saville, and S. J. E. Taylor. 2007. Integrating BOINC with Microsoft Excel: a case study. In *Proceedings of the 29th Information Technology Interfaces Conference, Dubrovnik, Croatia*. June 25-28, 2007. 733 – 738.

AUTHOR BIOGRAPHIES

NAVONIL MUSTAFEE is a lecturer in Information Systems and Operations Management at the School of Business and Economics, Swansea University (UK). His research interests are in grid computing, pa-

rallel and distributed simulation, and healthcare simulation. His e-mail address is n.mustafee@swansea.ac.uk.

SIMON J. E. TAYLOR is the Founder and Chair of the CSPI-PDG under SISO. He is the co-founding Editor-in-Chief of the UK Operational Research Society's (ORS) Journal of Simulation and the Simulation Workshop series. He was Chair of ACM's SIGSIM (2005-2008). He is a Reader in the School of Information Systems, Computing and Mathematics at Brunel and leads the Distributed Systems Research Group. He has published over 100 articles in modeling and simulation. His recent work has focused on the development of standards for distributed simulation in industry. His email address is simon.taylor@brunel.ac.uk.