Realtime-enabled Workflow Management in Service Oriented Infrastructures

Spyridon V. Gogouvitis[1]
sgogouvitis@telecom.ntua.gr

George Kousiouris[1]
gkousiou@telecom.ntua.gr

Kleopatra Konstanteli[1]
kkonst@telecom.ntua.gr

Theodoros Polychniatis[1]
thodpol@telecom.ntua.gr

Andreas Menychtas[1]
a_menychtas@telecom.ntua.gr

Dimosthenis Kyriazis[1]
dkyr@telecom.ntua.gr

Theodora Varvarigou[1]
dora@telecom.ntua.gr

ABSTRACT

Many applications, and especially the ones implementing multi-user collaborative environments, fall within the context of soft real time systems in which only small deviations from timing constraints are allowed. Furthermore, there is a growing trend that such applications are deployed in Service Oriented Infrastructures (SOIs), so as to take advantage of the capabilities that SOIs provide. In such an environment, the aforementioned applications are divided into multiple services within workflows. The dynamic nature of SOIs and their inability to adjust to the ever-changing resource availability and user demand poses an extra burden as workflow systems try to carry out more complex and mission-critical applications. In that frame, we discuss the work that has been carried out in the area of workflow management - developed to serve the needs of applications deployed in distributed environments, along with their limitations and potentials. Based on that, we present the research effort of the IRMOS EU-funded Project which advances the field of research in workflow management in order to support interactive real time applications on SOIs.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures, D.2.2: Design Tools and Techniques, D.2.9 Management.

General Terms

Design, Management

Keywords

Workflow Management, Real Time, QoS, Service Oriented Architecture, Event-driven, Advance Reservation, Multimedia.

AREA'08, October 31, 2008, Vancouver, British Columbia, Canada.

1. INTRODUCTION

Since workflow is a wide concept in technology, the terminology regarding workflows that is used afterwards in this paper is defined. Workflow Management Coalition (WfMC) provides the following general definition [1]: "Workflow is the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules".

Service Oriented Architecture (SOA) [2] is an architectural style that emphasizes implementation of components as modular services that can be discovered and used by clients. Infrastructures based on the SOA paradigm are called Service Oriented Infrastructures (SOIs).

Managing the application workflow operations within a SOI requires the orchestration of the distributed resources [3]. In that frame, workflow is an important factor for application composition promoting inter-organizational collaborations by integrating the teams involved in managing of different parts of a workflow. Besides, literature [4] describes additional advantages of the workflow management such as the utilization of resources to increase throughput or reduce execution costs and the ability to build dynamic applications which orchestrate these resources.

Many applications rich in multimedia content are deployed in such infrastructures benefiting from the aforementioned advantages. These have strict timing requirements and can be classified as real time systems [5]. A real time system is one in which its correctness is defined not only by its final result but also by the time that this is produced [6]. When any deviation from the timing constraints is detrimental for the system then it is considered as a hard real time system. In cases where some deviations from the timing requirements are acceptable, as long as these are few in occurrence and within some predefined boundaries then the system can be considered as soft real time. Traditional workflow management systems are not suitable for soft real time applications, as they do not provide the capabilities needed.

An infrastructure following the SOA principles is the service-oriented Grid. Grid computing [7] is increasingly considered as an infrastructure able to provide distributed and heterogeneous resources in order to deliver computational power to resource demanding applications in a transparent way [8], [9]. Built on

[1]National Technical University of Athens

9, Heroon Polytechniou Str 15773 Athens, Greece

Tel. +302107722546

pervasive internet standards, Grids allow organizations to share computing and information resources across department and organizational boundaries in a secure and highly efficient manner. Grids support the sharing, interconnection and use of diverse resources, integrated in the framework of a dynamic computing system.

In Grid environments, workflow can be defined as the orchestration of a set of activities to accomplish a complicated goal including application processes, business processes, and infrastructure processes [10]. Workflow is an architecturally important factor for dynamic interoperability and adaptation to different business models, which can be addressed as workflow policies, and deployment contexts.

A Workflow Model / Specification is used to define a workflow both in task and structure level. There are two types of workflows, namely Abstract and Concrete [11], [12] while concrete workflows are also referred to as executable workflows in some literature [13] . In an abstract model, the tasks are described in an abstract form without referring to specific resources for task execution since it provides the ability to the users to define workflows in a flexible way, isolating execution details. Furthermore, an abstract model provides only service semantic information on how the workflow has been composed and therefore the sharing of workflow descriptions between users is feasible, which is of major importance for the participants of Virtual Organizations (VOs)[8].

In the concrete model, the tasks of the workflow are bound to specific resources and therefore this model provides service semantic and execution information on how the workflow has been composed both for the service instances and for the overall composition (e.g. dataflow bindings, control flow structures). In correspondence with the abstract model and the relationship to the VOs, the tasks included in a concrete model may also refer to data movement requests in order to publish newly derived data into VO [12]. There has to be mentioned that the service instances do not necessarily correspond to resources since within a resource more than one service instances may be available and executable.

A workflow mapping mechanism is an integral part of the QoS provisioning, and especially end-to-end timing requirements, since this is the only way to estimate, calculate and conclude to the mapping of workflows and the selection of the available service types and instances in order to deliver an overall quality of service across a federation of providers. It is also very important that the workflow mechanism is aware of events that are possible to occur and can react in accordance. This can help meet the time requirements of the workflow even in cases where individual tasks fail to meet their deadlines or additional requirements are imposed “on the fly”. The mechanism needs to be able to produce a new concrete workflow taking into consideration new requirements per task and do so in a short time after an event is raised.

In this paper we describe the barriers of workflow management systems in SOIs to address the real-time needs of applications rich in multimedia content; and we present an innovative event-driven approach to overcome the aforementioned issue. This approach is part of the research effort in the framework of the IRMOS EU-funded project in the area of workflow management.

The remainder of the paper is structured as follows. Section 2 presents related work in the field of workflow management in Service Oriented Infrastructures. Section 3 presents the mechanism that is going to be used within IRMOS to tackle the identified problems in various workflow management systems thus far, while Section 4 concludes the paper.

2. WORKFLOW MANAGEMENT IN SOIs

The following sections present the work done in the field of workflow management in Service Oriented Infrastructures. The limitation of the presented approaches will be addressed within the framework of IRMOS project [14].

2.1 Workflow Enactment

A very interesting and well documented survey is found in [15], indicating the characteristics of some of the major players in workflow management systems worldwide. One of the most interesting among them is GrADS [16]. It is based on the Globus Toolkit, one of the most prevalent Grid middleware, and it aims specifically at applications with large computational and communication load. It supports a Directed Acyclic Graph (DAG) approach for workflows which means that dependencies between tasks are analyzed and their parallelization can be performed for enhanced scheduling purposes. It also supports QoS constraints through estimating the application execution time. This estimation is performed through analytical modeling and historical data. Furthermore it is based on a centralized architecture with both global and local decision making policies, a prediction based planning scheme and a task level rescheduling for fault tolerance purposes.

Another worth to mention implementation is that of Askalon project [17], where attention is given to performance oriented applications and is based on the Globus Toolkit too. Performance estimation is also attempted within this project and a positive feature is the decentralized architecture, but with a global decision making mechanism. In ICENI [18], another GT oriented implementation with a computationally intensive scope, the estimation is less efficient than the previous ones but it has a more market driven approach, with a global decision making policy and a centralized architectural scheme. In all of the above cases a very positive feature is that the workflow composition system can be extended by the user.

UNICORE [19] is an approach which is becoming quite popular lately, mainly due to its extremely robust nature. It is mostly based on internal components rather than already circulating solutions for Grid integration and it comes with a number of disadvantages such as concrete workflow models. In a concrete model the user specifies which part of the workflow must be executed in a specific resource. This feature not only limits scheduling performance, as in the case when many users want to access a single resource while others are idle, but also is quite risky in dynamic environments where resources come and go unexpectedly. Nevertheless, one popular characteristic of UNICORE is that many of its mechanisms such as decision making, planning policy and strategy are designed in order to be extensible by the user.

The workflow management service within the GRIDCC project [20] is tasked with optimizing the workflows and ensuring that they meet the pre-defined QoS requirements specified upon them. The project aims at utilizing instruments through Grid infrastructures. It also focuses on Web Services and SOA and implements a partner language to use with BPEL4WS. Instead of defining a new language for workflows with QoS requirements, or embedding QoS requirements within a language such as BPEL4WS, GRIDCC uses a standard BPEL4WS document along with a second document which points to elements within the BPEL4WS document and annotates this with QoS requirements. This allows to take advantage of standard BPEL4WS tooling for execution and manipulation as well as to provide QoS requirements. XPath notation to reference elements is used within the BPEL4WS document; therefore, this approach can be easily adapted to other (workflow) languages based on XML. The end-to-end workflow pipeline takes a user's design and implements it within the Grid, through reservation services and performance repository. Workflows are defined through a web based editor which allows the augmentation of QoS requirements by defining the user's expectations for the execution. The WfMS (Workflow Management Service) provides a mechanism for building QoS on top of an existing commodity, based on BPEL4WS engine, thus allowing the provision of a level of QoS through resource selection from a priori information together with the use of advanced reservation.

In [21] a QoS-aware Grid Workflow Language (QoWL), subset of Business Process Execution Language (BPEL) with QoS extensions and QoS-aware Grid Workflow Engine (QWE) is presented. Users may specify different QoS constraints addressing the overall workflow or individual workflow tasks. QWE comprises of a workflow planning component, which performs QoS negotiation and service selection, and a workflow execution component which executes the workflow by invoking the selected services. Based on the specified QoS constraints, the QWE negotiates with multiple candidate Grid services to select appropriate services which satisfy the specified QoS constraints. Performance prediction for QoS-aware VGE services is based on application-specific performance models which depend on metadata about the input data (e.g. matrix size) supplied by the client during QoS negotiation. VGE services rely on a generic QoS module which usually comprises of an application-specific performance model, a pricing model, a compute resource manager that supports advance reservation and a QoS manager. VGE relies on standard Web Services technologies such as WSDL, SOAP, WS-Security, Tomcat and Axis. VGE services enclose native HPC applications, usually parallel MPI codes running on a cluster, and expose their functionality via a set of common operations for job execution, job monitoring, data staging and error recovery. One limiting factor is the dependency upon application nature.

2.2 Scheduling

Relevant work in workflow management systems includes [21]. The authors of the specific paper investigate the effect of resource reservation on external applications as well as on local jobs, and introduce the design of efficient task scheduling algorithms considering the tolerance of local jobs to resource reservation. A new performance metric is proposed, the relative slowdown, to quantify the performance impact of resource reservation. The local job process is modeled with an M/G/1 queuing system and the effect of system parameters on relative slowdown is analyzed. They investigate both first-come-first-serve (FCFS) and round-robin (RR) queuing disciplines. Efficient algorithms are designed and implemented considering local jobs' tolerance to reservation. A user-level soft real time CPU scheduler, DSRT, is updated to enable resource reservation in a general computing platform. They also define a new metric, the relative slowdown of local jobs on a resource for a given reservation which is the ratio of the average waiting time with reservation and the average waiting time without reservation. DSRT is the implementation of the CPU server of QualMan middleware. A major component of DSRT is the resource scheduler, named the dispatcher. A priority scheduling mechanism is applied to differentiate the processing of real-time (RT) processes and time-sharing (TS) processes. The dispatcher runs at the highest fixed-priority.

Paper [22] presents the concept of Manufacturing Grid (MG) as the application of Grid technology. It follows the Open Grid Service Architecture (OGSA) as the system framework, and Globus Toolkit 3.0 (GT3) as the developing toolkit. A QoS-based Global Process Planning (GPP) and scheduling schema (Manufacturing Grid Resource Scheduling, MGRS) is presented and the corresponding module is implemented with the functions of GPP analyzing, resource discovery, AHP (Analytic Hierarchy Process)-based resource selection and fault-tolerant handling (re-scheduling). The goal of the schema is to be the supplement of the application Grid in manufacturing with better QoS performance, such as higher user satisfaction, product quality and service, as well as the lower failure rate, time-to-market, cost, etc. Its main features are:

- *GPP Analyzing*: Refers to analyzing the submitted task, and decomposing the complicated target task into a few serial or parallel simple, basic manufacturing subtasks, according to its QoS properties
- *Resource Selection*: Once the list of possible resources is known, MGRS will select a resource that is expected to meet the requirements mostly. In order to fulfil the restrictions, it has to gather dynamic information about resource accessibility, machining precision, machining capability, and resource status, etc
- *Fault-tolerant Handling*: This is also called re-scheduling. Manufacturing Grid is inherently a dynamic system where environmental conditions are subjected to unpredictable changes as: resource or network failures, system performance degradation, variations in the cost of resources, etc. Fault-tolerant handling is the efficient way to guarantee that the submitted tasks are completed and that the user's requirements are met.

AHP is particularly useful for evaluating complex multi-attribute alternatives involving subjective criteria. The essential steps in the application of AHP are the following:

1. Decomposing in a hierarchical manner, a general decision problem into sub-problems that can be easily comprehended and evaluated,
2. Determining the priorities of the elements at each level of the decision hierarchy, and
3. Synthesizing the priorities to determine the overall priorities of the decision alternatives

The most limiting feature is the use of UDDI registry for resource discovery. UDDI is generally static and lacks monitoring abilities. Moreover GT3 is considered obsolete, as the current working edition is version 4.

In the Phosphorus project [23], a number of scheduling algorithms are proposed in the context of fairness of resource assignment between users and their tasks. An interesting scheme is the fact that Phosphorus takes into consideration the time for completion and the deadline, in order to meet QoS requirements. Phosphorus also deals task workloads, which are either known or not known a priori, and advance reservation. The MetaScheduling Service (MSS) developed by the specific project can tackle complicated workflows allowing the end-user to execute the individual components of his application using the most appropriate resources available. MSS is able to orchestrate resources of different sites belonging to different administrative domains, while it is also responsible for the negotiation of agreements on resource usage with the individual local resource management systems. The main dependency of this implementation is the use of the UNICORE middleware.

2.3 Fault Detection and Recovery

In a highly dynamic and heterogeneous SOI geographically and organizationally dispersed, heterogeneous resources are incorporated such as computing systems and software, storage systems, instruments, scientific equipment, specialized hardware, communication systems, data sources as well as human collaborators. In such a heterogeneous environment changes are numerous, highly variable and with unpredictable effects. These changes can lead to failure for various reasons: non-availability of required services or software components, overloaded resource conditions, memory shortage, and network fabric failures. For these reasons, workflow management in SOIs, in general, and especially those supporting workflows execution with real-time applications, as in the case of IRMOS platform, should be able to detect and manage failures in order to ensure reliable support of the execution environment.

Workflow failure handling techniques can be divided into two different levels: task-level and workflow-level [25]. Whereas task-level techniques are concerned with masking the effects the failures of the services impose on the entire workflow, workflow-level techniques are focused on the manipulation of the workflow structure in order to deal with erroneous conditions.

The goal of ASKALON [26] is to simplify the development and optimization of Grid workflow applications. The Enactment Service is the central service of the ASKALON middleware responsible for executing a workflow application. The Enactment Service provides fault tolerance for the middleware. It does so in three different levels: Activity level, Control Flow Level and Workflow Level. For the activity level, retry and replication methods are used. For control flow level, light-weight checkpointing and migration are used. With this method, the workflow state and URL references to intermediate data are saved and can be recovered in case of a failure. This is very fast, but poses the disadvantage that the intermediate data are stored on possibly unsecured file systems. For the workflow-level fault tolerance the methods of alternative task, redundancy and checkpointing are used. The last differs from light-weight checkpointing in that it stores backup copies of intermediate data into a database and execution can be restored and resumed at any time from any Grid location. This method posses the disadvantage of increased overhead.

In Akogrimo [27] once the execution of a business service has started, it is managed and monitored in order to achieve continuous conformance to the contractual terms of SLAs. In case of execution failure or failure to meet the SLA criteria and depending on the explicit type of failure, EMS is able to reallocate the execution. In order to detect possible failures and maintain the state of the execution between possible reallocations, a fault-detection in conjunction with a recovery scheme is being used. The EMS establishes a WS-Notification mechanism with the business services in order to receive notification messages every time a property of the business resource changes its value. Resource properties provide to the EMS a view on the current state of the resource. Every time the EMS receives a notification message it stores the new value of the resource property. If a failure occurs during the execution of the business service and the creation of new business resource is needed (either on the same or different location), the EMS will set the resource properties to the last known ones before the failure occurs. This approach makes use of the WS-Resource specification in order to maintain the state of the business resources.

3. ADVANCEMENTS IN IRMOS

The IRMOS platform will enable a Service Oriented Infrastructure to be used by applications that need real-time assurances. The platform will be built around the concept of events. Should a new request arrive this will be propagated to a new mapping mechanism in which high level user requirements (e.g. frames per second) will be translated into low level requirements (e.g. CPU cycles). Additionally an abstract user defined workflow will be automatically transformed into a platform specific workflow which takes into consideration the QoS requirements the user has defined. These will be fed into a novel advanced reservation scheme with the ability to provide strong QoS guarantees while at the same time promoting maximum utilization of the resources. The proposed algorithm will estimate the QoS level prior to any additional reservation which will allow for QoS provisioning. The constraints that the user will impose will be related to the workflow as a whole and not per service. The algorithm will be applicable to single-service, sequential and non-sequential workflows. The constraints will also be able to have weights thus giving the user the ability to express their relevant importance. The algorithm will also be flexible in terms of start time. When advance reservation has determined the appropriate resources for the workflow an SLA negotiator will be initiated in order for the creation and signing of SLAs to take place.

Major effort will be put in the Fault Detection and Recovery mechanisms of the platform. This will be **event–oriented**. The user will be able to define events and sets of actions that follow the events defined as well as predefined (or global) events. Should an event, such as unexpected progress, delay, or SLA violation occur, the completion of a workflow according to initial timing requirements may require a rescheduling in response to it. When adaptation is necessary, a set of potential alternatives is generated, with the objective of changing a workflow as it continues to meet initial requirements. For each alternative, prior to actually carrying out the adaptation in a running workflow, it is necessary to estimate its impact on the workflow as a whole. The latter is achieved by not only selecting the optimum service instances within a given workflow but also by characterizing the not-selected instances in order to have a-priori knowledge if they are needed in a rescheduling workflow process. Each time a change is completed in a workflow, the new timing requirements are re-estimated in order to meet the user constraints. As an example, if the event of an SLA violation occurs the platform will be responsible to automatically take steps to mitigate the effects of such a violation for the rest of the workflow. In essence a new SLA will have to be renegotiated taking into consideration the delay that has already occurred and rescheduling the remaining tasks in such a way that the deadline of the entire workflow is met. This will also trigger a process to reschedule the entire workflow.

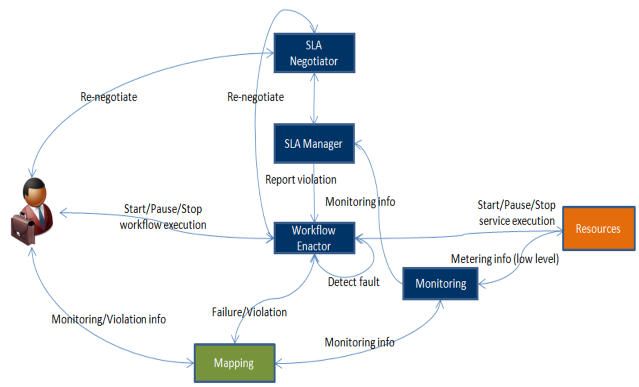

Figure 1. The event-oriented IRMOS approach

In the above figure a simplified version of the IRMOS approach is depicted, containing the most important components. The user can start, pause or stop the execution of a workflow through the workflow enactor, who is responsible of communicating with the resources. A monitoring component constantly monitors the execution of the various tasks and if an event is raised this is reported to the SLA manager and the mapping service. The SLA manager then notifies the SLA negotiator in order for a new SLA to be created. This also leads to the creation of a new concrete workflow through the enactor and the mapping service. The user is also notified. The user is also able to ask for new resources on the fly. This is handled by the SLA negotiator and is propagated through the SLA manager to the Workflow Enactor. It is clear that the IRMOS approach is event oriented.

4. CONCLUSIONS

In this paper we have shown that the algorithm that have been proposed thus far concerning workflow management do not provide real-time assurances or end-to-end QoS guarantees. We believe that within the IRMOS project a novel workflow management system will be built that will provide solutions to these problems, thus making it feasible for SOI to be adopted in the industry. This will be accomplished by novel event driven mapping mechanisms, advance reservation and fault detection and recovery. The workflow management system will be tested on three different reference applications, namely Film Post-Production, e-Learning and Virtual and Augmented Reality. All three applications are rich in multimedia content and have high real-time needs and are therefore considered as ideal as proof of concept of our novel event driven design.

5. ACKNOWLEDGMENTS

This work has been supported by the IRMOS project and has been partly funded by the European Commission's IST activity of the 7th Framework Programme under contract number 214777.

6. REFERENCES

[1] Workflow Management Coalition, Terminology & Glossary, Document Number WFMC-TC-1011, Issues 3.0, February 1999.

[2] T. Erl, “Service-oriented Architecture: Concepts, Technology, and Design”, Upper Saddle River: Prentice Hall PTR. ISBN 0-13-185858-0, 2005.

[3] A. Mayer, S. McGough, N. Furmento, W. Lee, M. Gulamali, S. Newhouse, and J. Darlington, “Workflow Expression: Comparison of Spatial and Temporal Approaches”, Workflow in Grid Systems Workshop, GGF-10, Berlin, March 9, 2004.

[4] D.P. Spooner, J. Cao, S. A. Jarvis, L. He, and G. R. Nudd, “Performance-aware Workflow Management for Grid Computing”, The Computer Journal, Oxford University Press, London, UK, 2004.

[5] L. Abeni, G. Buttazzo, “Integrating multimedia applications in hard real-time systems”, Proceedings of the 19th IEEE Real-Time Systems Symposium, Madrid, Spain, 1998.

[6] L.Sha, T. F. Abdelzaher, K. Årzén, A. Cervin, T. P. Baker, A. Burns, G. C. Buttazzo, M. Caccamo, J.P. Lehoczky, A.K. Mok, “Real Time Scheduling Theory: A Historical Perspective”, Real-Time Systems Vol. 28, No. 2-3, pp. 101-155, November 2004.

[7] I. Foster and C. Kesselman, “The Grid: Blueprint for a Future Computing Infrastructure”, Morgan Kaufmann Publishers, USA, 1999.

[8] I. Foster, C. Kesselman, S. Tuecke, “The Anatomy of the Grid: Enabling Scalable Virtual Organizations” International Journal Supercomputer Applications, Vol. 15, No. 3, 2001.

[9] W. Leinberger, V. Kumar, “Information Power Grid: The new frontier in parallel computing?” IEEE Concur., Vol. 7, No. 4, pp. 75-84, October-December 1999.

[10] I. Altintas, A. Birnbaum, K. Baldridge, W. Sudholt, M. Miller, C. Amoreira, Y. Potier, and B. Ludaescher, “A Framework for the Design and Reuse of Grid Workflows”, International Workshop on Scientific Applications on Grid Computing (SAG'04), LNCS 3458, Springer, 2005.

[11] E. Deelman, J. Blythe, Y. Gil, and C. Kesselman, “Workflow Management in GriPhyN”, The Grid Resource Management, Kluwer, Netherlands, 2003.

[12] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M. H. Su, K. Vahi, M. Livny, “Pegasus: Mapping Scientific Workflow onto the Grid”, Across Grids Conference 2004, Nicosia, Cyprus, 2004.

[13] B. Ludäscher, I. Altintas, and A. Gupta, “Compiling Abstract Scientific Workflows into Web Service Workflows”, 15th International Conference on Scientific and Statistical Database Management, Cambridge, Massachusetts, USA., IEEE CS Press, pp. 241-244, Los Alamitos, CA, USA., July 09-11, 2003

[14] http://www.irmosproject.eu/

[15] Jia Yu, Rajkumar Buyya, “A Taxonomy of Workflow Management Systems for Grid Computing”, Journal of Grid

Computing, Volume 3, Issue 3 - 4, pp. 171-200, Springer, 2005.

[16] F. Berman, A. Chien, K. Cooper, J. Dongarra, I. Foster, D. Gannon, L. Johnsson, K. Kennedy, C. Kesselman, J. Mellor-Crummey, D. Reed, L. Torczon, and R. Wolski. The GrADS Project: Software Support for High-Level Grid Application Development. International Journal of High Performance Computing Applications(JHPCA), 15(4):327-344, SAGE Publications Inc., London, UK, Winter 2001.

[17] T. Fahringer, A. Jugravu, S. Pllana, R. Prodan, C. S. Jr, and H. L. Truong. ASKALON: a tool set for cluster and Grid computing. Concurrency and Computation: Practice and Experience, 17:143-169, Wiley InterScience, 2005.

[18] S. McGough, L. Young, A. Afzal, S. Newhouse, and J. Darlington. Workflow Enactment in ICENI. In UK e-Science All Hands Meeting, Nottingham, UK, IOP Publishing Ltd, Bristol, UK, Sep. 2004; 894-900.

[19] Unicore Forum. Unicore Plus Final Report: Uniform Interface to Computing Resource. 2003, http://www.unicore.org/documents/UNICOREPlus-Final-Report.pdf [December 2004].

[20] McGough, A. S., Akram, A., Akram, A., Guo, L., Guo, L., Krznaric, M., Dickens, L., Colling, D., Martyniak, J., Powell, R., Kyberd, P., and Kotsokalis, C. 2007. GRIDCC: real-time workflow system. In Proceedings of the 2nd Workshop on Workflows in Support of Large-Scale Science (Monterey, California, USA, June 25 - 25, 2007). WORKS '07. ACM, New York, NY, pp.3-12

[21] Ivona Brandic, Siegfried Benkner, Gerhard Engelbrecht, Rainer Schmidt, "QoS Support for Time-Critical Grid Workflow Applications," e-science, pp. 108-115, First International Conference on e-Science and Grid Computing (e-Science'05), 2005.

[22] Ming Wu, Xian-He Sun, Yong Chen, "QoS Oriented Resource Reservation in Shared Environments," ccgrid, pp. 601-608, Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06), 2006.

[23] Liu Lilan, Yu Tao, Shi Zhanbei, Fang Minglun "A QoS-based Global Process Planning and Scheduling Approach for Manufacturing Grid" in Flexible Automation and Intelligent Manufacturing (FAIM 2004), Toronto, Canada, 2004, 7: 1194-1201.

[24] Deliverable D.5.2., "QoS-aware Resource Scheduling", Project Phosphorus, http://www.ist-phosphorus.eu/files/deliverables/Phosphorus-deliverable-D5.2.pdf

[25] S. Hwang and C. Kesselman. Grid Workflow: A Flexible Failure Handling Framework for the Grid. In 12th IEEE International Symposium on High Performance Distributed Computing (HPDC'03), Seattle, Washington, USA, IEEE CS Press, Los Alamitos, CA, USA, June 22 - 24, 2003.

[26] Thomas Fahringer, Radu Prodan, Rubing Duan, Francesco Nerieri, Stefan Podlipnig, Jun Qin, Mumtaz Siddiqui, Hong-Linh Truong, Alex Villaz´on, Marek Wieczorek. "ASKALON: A Grid Application Development and Computing Environment", The 6th IEEE/ACM International Workshop on Grid Computing.

[27] Deliverable D4.3.3, "Report on the Implementation of the Infrastructure Services Layer", Akogrimo Project, http://www.akogrimo.org/modules.php?name=UpDownload&req=viewdownloaddetails&lid=116