

Grid-enabled Workflows for Industrial Product Design

Moustafa Ghanem, Nabeel Azam
InforSense Ltd,
London, UK.
{mmg, nabeel}@inforsense.com

Mike Boniface, Justin Ferris
IT Innovation Centre,
University of Southampton, UK.
{mjb, jf}@it-innovation.soton.ac.uk

Abstract

This paper presents a generic approach for developing and using Grid-based workflow technology for enabling cross-organizational engineering applications. Using industrial product design examples from the automotive and aerospace industries we highlight the main requirements and challenges addressed by our approach and describe how it can be used for enabling interoperability between heterogeneous workflow engines.

1. Introduction

The development of complex products and services poses many challenges to industry today. Product complexity is increasing whilst greater competition is requiring shorter design cycles and “right first time” design decisions earlier in the design cycle. The design process is a long-term endeavour with many development phases involving people from various disciplines accessing rich and diverse business assets controlled by different organizations.

Industrial companies have existing investments in workflow systems that support their product design processes; however, the technologies do not readily support business partnerships. Increasingly, Grid technology is being used by these companies to enable successful partnerships, helping them to achieve complex design goals. These business-to-business relationships allow organizations to participate in distributed and managed design processes whilst still enforcing commercial level security policies for the protection of intellectual property rights (IPRs) and business objectives.

In this paper, we describe some of the challenges facing the development and use of workflow systems within a Grid-based environment to support industrial product design and present an approach to support cross-organizational design processes. Our approach builds on extending existing workflow systems

InforSense KDE [1,2] and Taverna [3] through their integration with the GRIA [4] middleware to support secure and managed business partnerships. These infrastructure developments are driven by business requirements from key application sectors (aerospace, automotive, pharmaceuticals) within the EU IST SIMDAT project [5].

2. Background

Workflows provide a natural way of describing the execution order of tasks (work units) and dependency of relationships between them. Informally, a workflow is an abstract description of the steps required for executing a particular real-world process, and the flow of information between them. Work passes through the flow from start to finish and activities could be executed by people or system functions (See Figure 1).

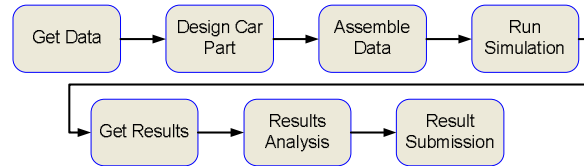


Figure 1: A simplified conceptual workflow for automotive design

Industrial workflow systems typically allow users to describe their workflows using a visual interface. Once designed, a workflow is submitted to workflow execution engine that controls the invocation and data transfer between the different activities.

Since the late 1980s, workflow technology has been used in many application areas, where the primary driver has been to enable automated execution of repetitive tasks, with many applications in office automation and business processes specification and execution. More recently workflow systems have been used for remote application integration where a workflow provides the glue to integrate distributed applications, typically accessed via a Web Service

interface (e.g. [6]). Within a Grid computing setting, workflow systems are typically used as a means for describing the steps required for submitting and executing jobs to high performance computing resources, typically deployed within a single administrative domain, [7, 8]. In this paper we focus on the use of workflow technology for coordinating remote Grid Services across organizational boundaries.

2.1. Industrial application drivers

The SIMDAT project focuses on four key industrial application areas each providing a challenging model problem to drive workflow and Grid infrastructure developments. Such applications provide the use cases and requirements for designing, developing and evaluating Grid technologies within a business context. This paper focuses on two model problems within the aerospace and automotive sectors. The scenarios provided below have been designed to show how Grid technologies can support the aggregation of distributed multi-disciplinary capabilities operating across organizational boundaries.

Collaborative automotive design: The automotive scenario focuses on the validation of car design and crash simulation models that require the partnership between a car manufacturer and a subcontracted design supplier. Workflow technology is used to co-ordinate Grid services for a crash simulation process in a logical order with a simple user interface to guide the user through each individual stage required. The workflow must hide the complexity of the underlying, Grid infrastructure only showing the details of the engineering design process to the engineers. There are a number of concerns regarding the security of data and the protection of IPR between potentially competing suppliers. These include unauthorized access to global or partial models and data both within and outside of the business partnership.

Collaborative aerospace design: The aerospace scenario simulates the multi-disciplinary collaborative design of a low-noise, high-lift wing configuration. The scenario is typical of sub-system design problems in the context of a future-concept, unmanned cargo vehicles that need to use airfields in noise-sensitive locations. Again, the workflow must hide the complexity of the underlying, Grid infrastructure and must address the security concerns of cross-organizational collaborations.

2.2. Requirements for workflow systems

Grid technologies offer the potential to allow business to share distributed assets within the constraints of organization IPR and operational security policies. Workflow systems accessing resources using a Grid infrastructure must fit within this philosophy. We briefly summarize the main business requirements that need to be supported.

Collaborative cross-organizational workflows: One of the key features of product design workflows within SIMDAT is their need to cross organizational boundaries. Most modern workflow systems provide support for coordinating remote web service invocations. Such remote web services may be used as interface for providing access to remote high performance computing resources. However, such systems provide little or no support for executing secure and managed services across organizational boundaries, and provide no uniform way for managing the remote resources. Within industrial product design applications, security measures provided by the underlying Grid infrastructure must be respected. In addition, a workflow system operating within a Grid computing environment must admit the heterogeneity of such services and the fact that they are not subject to centralized control. Proper management of services and policies including costing and accounting of remote services must also be respected by the underlying workflow engines used. All this must be achieved using uniform and standardised interfaces.

Interoperability between workflow systems: Within business partnerships, it is essential to admit the existence of different workflow management tools used by end users and within various organizations. This is exemplified in SIMDAT through the existence and use of at least four workflow systems (InforSense KDE, Taverna, Optimus and Model Centre). Achieving run-time interoperability between these different workflow systems is essential to ensure design processes can be exploited by business partners. Furthermore, the interaction between workflow systems must be based on using standardized protocols.

Industrial strength operation: Some of the engineering workflows used within the project invoke long-running jobs that require execution times in order of weeks. The underlying workflow execution engines must be robust enough to support such a mode of execution. This must include their ability to support monitoring the execution of these long running tasks, support caching of intermediate results to avoid

lengthy re-executions, and also support interactive control by the end users at various decision points.

3. Grid-enabled workflow systems

Industrial partners in SIMDAT use workflow systems for constructing and automating distributed end user applications through the composition of distributed data and computational services. In this section, we review two of these systems and follow with a technical discussion on how the technology can be adapted to satisfy the requirements for business partnerships in product design.

InforSense KDE: The InforSense KDE system is a commercial workflow authoring and execution system that is based on the research outputs of the Discovery Net e-Science project [2]. Within InforSense KDE, end-user applications are constructed visually as workflows that co-ordinate the flow of data and execution between remote services (data sources and/or analysis tools) with known input and output interfaces. These services can execute either on the user's own machine or via remote services accessed using either web service protocols or native interfaces. The workflows are expressed in DPML, an XML-based workflow language. The distributed execution of the workflow including data flow, control flow and invocation of remote services is handled by the InforSense workflow execution engine.

Taverna: The Taverna workflow authoring and execution environment [3] was initially developed within the myGrid project [9]. It provides a graphical workbench for creating, executing and managing workflows and instances of workflows. Taverna workflows are developed using a graphical interface and represented in an XML-based language XSCUFL. They can orchestrate a combination of local and remote resources that are composed and linked with specific data flow and other constraints to form a reusable workflow process. Taverna uses the Freefluo workflow enactment engine for workflow execution [10].

3.1. Workflow system interoperability

InforSense KDE and Taverna provide similar capabilities at a conceptual level. However, they use different languages to express workflows and their visual interfaces and execution engines are optimized to support different applications. A key aspect in supporting the requirements defined in section 0 is for the different workflow engines to fit within a uniform

model for coordinating the execution of Grid services. The Grid services can be used to wrap both legacy applications but also other workflows, providing the basis for achieving interoperability between the different workflow systems. A workflow written in a specific language and tied to a particular system can therefore be called by another workflow system as a remote Grid service. In addition, the encapsulation of a workflow as a service means that a client tool does not have to understand the service provider's workflow language when executing a remote design process. This allows organizations to develop workflows using the environment of their choice and interoperate with other workflow systems through a loosely coupled fashion.

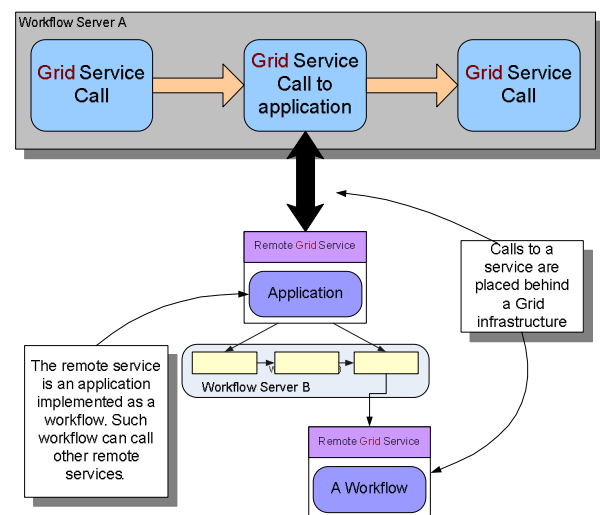


Figure 2: Workflow interoperability model within a Grid environment

Figure 2 shows how workflow execution can be federated across administration domains by simply deploying specific workflows as applications behind a Grid service. The initial workflow engine invokes a service call passing references to data inputs and outputs to application that is implemented using the second workflow engine. The invocation mechanism is similar to existing job submission systems with no understanding of the underlying language or execution environment of the remote server.

Figure 3 illustrates another approach to Grid-enabled workflow interoperability based on passing the description of a workflow from the initial workflow engine to the second. As in the previous case, the submitting client does not need to be able to process and interpret the workflow submitted to the remote engine itself. It simply handles it as an XML file passed as input together with the required input data

and parameters. However, as opposed to the previous case, the second engine must accept submissions of workflow definitions via a Grid service interface.

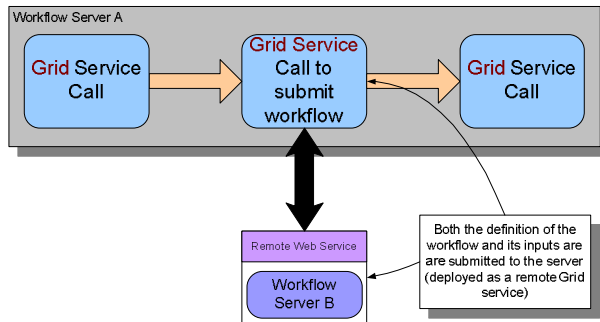


Figure 3: Deploying a workflow server as a service

3.2. Integrating workflow systems and GRIA

Within our work, GRIA 4.1 has been used to provide B2B service provision across organizational boundaries. GRIA uses a fully decentralized management approach, with minimal dependency between sites. Each site offering GRIA services makes its own business decisions about which users to trust and on what terms, and is responsible for enforcing its own access policies and deciding which applications to support. Sites can interact with each other, but this is driven by their common consumers, and those consumers are responsible for managing the resulting dependencies. There are no global agreements to set up, and no virtual organizations need be established, though users can interact according to virtual organization models if they want.

GRIA provides a Java API to build client applications. The API supports a Java binding for service operations, helper classes to encapsulate complex tasks and Swing interfaces used for browsing allocations and monitoring jobs. The process for executing distributed applications installed at a GRIA service provider consists of set of simple tasks for establishing a resource allocation context and executing data transfer and processing activities. Each of these tasks described in more detail below.

Allocate Resources: allows users to negotiate QoS with available service providers. Users request proposals by sending a QoS requirements specification in XML to trusted service providers. An API call is invoked to locate the suppliers that can and are willing to fulfil the request. An allocation is created once a user selects a supplier. The task returns a result set consisting of the repository path, account and a unique resource allocation identifier.

Upload Inputs: allows users to upload multiple data inputs to a service provider. An additional parameter is used to allow the user to enter data stager names to identify each input. A comma delimited list of names can be entered and this is checked against the number of input files selected to ensure they match. Upon execution the files are written to the service provider and information from each input data stager is written to a new result set.

Execute Job: allows users to execute a computational job. The user specifies their job requirements within an XML file. A comma delimited list of data stager names also has to be specified. The user must know how many outputs are returned from the service beforehand. While the job is running, a status log is returned that can be viewed by the user. Once the job has finished executing the job information and output data information is appended to a new result set.

Download Output: allows a user to download data from a single GRIA data stager. Users provide a stager ID and a local location for storing the file.

Finish Conversations: allows users to close any open resource allocations to be reimbursed for resources that have not been used. The task loops through an ID list closing each allocation and returns a string informing the user that all conversations were closed successfully.

The above 5 tasks can be easily integrated within any workflow system allowing the workflow user to chain them in various configurations within larger workflows. Figure 4 provides an overview of the general integration method using InforSense KDE as an example.

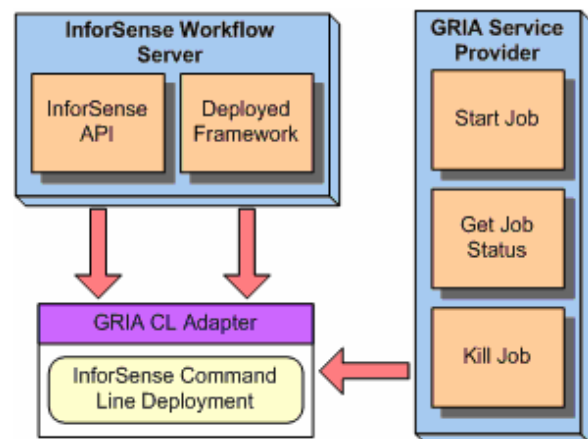


Figure 4: Integration and interfacing to GRIA from the InforSense KDE workflow engine

3.3. Wrapping workflows as GRIA services

Within our framework, workflow processes can also be published as a Grid Service by a service provider allowing service consumers to discover and execute workflows in a similar manner to other Grid jobs. A workflow service is easily developed allowing workflow clients to execute workflows published by service providers and allowing complex inter-domain hierarchical workflows to be orchestrated.

The workflow service maintains accounts with other trusted service providers and negotiates QoS using the GRIA resource allocation service to fulfil the needs of workflows they are enacting on behalf of clients. When the workflow is executed, service providers monitor usage for the entire workflow, which may aggregate usage across many different suppliers. Service providers can then update the client's usage and bill depending upon the business value of the overall scientific process. Deploying GRIA services requires a Grid service wrapper to be developed based on GRIA's security and management infrastructure. These steps, which can be included easily within most workflow systems, are described in more detail below.

Workflow Publication: The workflow service must maintain a workflow repository that stores approved workflows that can be discovered and enacted by clients. GRIA places no requirements on the repository technology or metadata schema used for discovery but does require the repository to assign workflow identifiers that can be used by clients when submitting an enactment request. The naming scheme for workflow identifiers can be defined by service developer but should be unique within the service provider context. The current implementation assigns a URI based on the service provider's domain.

Security Integration: The workflow service must be compliant with WS-I Basic Profile and the WS-I Basic Security Profile. GRIA provides AXIS web service handlers that implement the WS-Security specification providing message-level integrity, confidentiality and authentication. These handlers should be integrated with the workflow service at service deployment time.

Authorization Integration: When accessing GRIA services, clients must provide a context identifier that is grounded in an overall business process for accessing a service. The context identifier, which could be an account, resource allocation, job, or workflow, is used as the policy target for authorization decisions. The workflow service must check that the client is authorized to access the context, which is achieved

using GRIA's Dynamic Authorization Framework (DAF). DAF essentially provides two components a Policy Enforcement Point (PEP) and Policy Decision Point (PDP). The PEP is an AXIS handler that wraps the service and prevents unauthorized users from invoking service operations on resources to which they have not been granted access.

Subcontract Relationship: When a published workflow requires access to other service providers the workflow service acts as a GRIA client. Service providers set up trusted business relationships out of band prior to workflow publication using the standard GRIA user interface for managing accounts. These relationships are stored within GRIA's client state repository that is accessed by the workflow service during enactment. The QoS requirements for each GRIA task within the workflow are provided by the workflow author and are stored as additional metadata within the workflow description. The metadata is used by the workflow task at enactment time to decide how the resource allocation context should be created. Various resource allocation strategies are supported including a fixed prior resource allocation or new allocation based on defined QoS terms such as work, memory, storage and transfer.

Billing: Depending upon the business model for workflow enactment, the service developer can provide capabilities to charge for workflow enactment. Possible chargeable items include resources used locally by the service provider, resources used through subcontract relationships and the value of the overall workflow. QoS and billing is provided by GRIA's management capabilities for resource allocation and accounting. The workflow service should check available QoS and records usage using GRIA's resource allocation service. Similarly, the workflow service should check available credit and charge using GRIA's account service

4. Application examples

4.1. Automotive design application

The automotive scenario used in this paper provides an example of cross-organizational collaboration for product design. It focuses on the validation of car design and crash simulation processes which require the collaboration between three organizations; a car manufacturer, a parts design subcontractor/supplier and a computational service provider that provides access to high performance computing facilities.

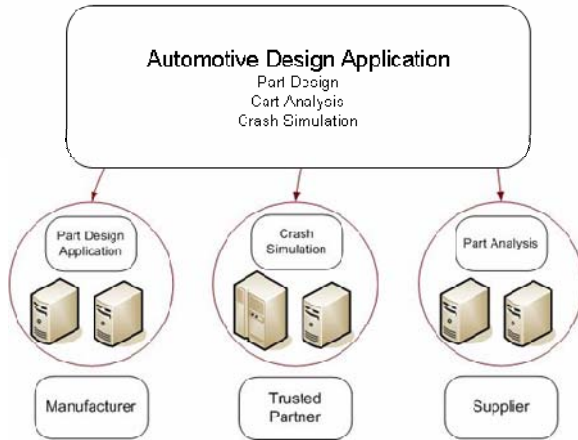


Figure 5: Overview of automotive scenario

Within this scenario (See Figure 5), a car manufacturer initiates the process by generating 3 models. The first model is a global finite element (FE) model of the car that excludes the part that is to be designed by the supplier. The other two models are an FE and a CAD model, which are a subset of the global environment and specifically include the parts of the car that come into contact or are connected to the part that is to be designed by the subcontractor.

The three models are uploaded to the data repository hosted by a computational service provider (trusted partner) who offers access to high performance computing facilities. Access is only granted to the models concerned with the environment around the part to be designed. Furthermore, the models are partially encrypted preventing access to the supplier's confidential information.

At the trusted partner's site, the model for the newly designed part is assembled with the global environment FE model to be used with the crash simulation application. Logs can be analyzed by the supplier to check the stability of the process and allows them to terminate the job prematurely if necessary. Once the simulation is complete, the supplier/subcontractor downloads a subset of the global results with respect to the designed part and analyses the results. Unsuitable results require further iterations of the design and simulation process. Once a set of results is deemed satisfactory, the supplier grants the manufacturer access to the CAD and FE models of the part designed along with the simulation results.

Implementation: The automotive scenario has been implemented using two InforSense KDE installations, one at the manufacturer's site and the other at the supplier's site.

The workflow at the manufacturer site is mainly concerned with data transfer services, which occur

twice in a single prototype, Initiate Project and Get Project Results services. The first instance, which initiates the process, involves using a GRIA service to upload the three models to the trusted partner. The latter service downloads the designed part data and simulation results once permission from the supplier is granted.

The supplier side workflow is shown in Figure 6 and requires three GRIA services which handle data transfer and invoke the simulation process. The data to be transferred between the services as well as resource identification information is passed along in InforSense KDE data tables between the nodes in a workflow.

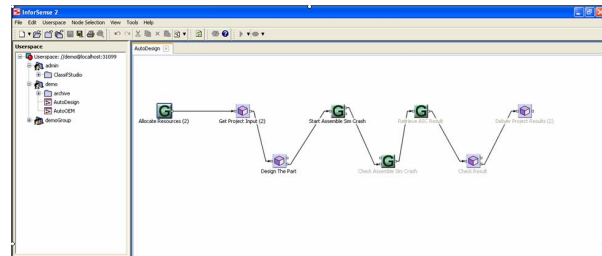


Figure 6: Automotive supplier workflow consisting of GRIA calls implemented in InforSense's KDE

4.2. Aerospace design application

The aerospace scenario under consideration provides another example of cross-organizational collaboration within a Grid setting. The scenario is based the design of a Regional Jet high lift system to reduce the acoustic footprint of the configuration during landing phase. The engineering challenge of the design service is to optimize the design of the wing within a set of constraints by varying a set of parameters that describe the wing's geometry.

The scenario includes three different classes of organization; Prime Contractor, Design Service Provider and Analysis Service Provider. The Prime Contractor initiates the design process by creating a business partnership with a Design Service Provider for the design of the high lift system. The Design Service Provider provides an optimization function for low noise and high-lift by engaging a series of Analysis Service Providers on behalf of the Prime Contractor to provide the required engineering analysis (aero-acoustics, aerodynamics, structures).

The design service provides a model of a wing with many variable parameters that, when set, fully define the wing's geometry. For a given set of geometric parameters and additional parameters such as the angle of attack, the lift, drag and noise of the wing may be

calculated. The design service must algorithmically choose sets of geometric parameters to find a wing geometry that achieves some optimum design.

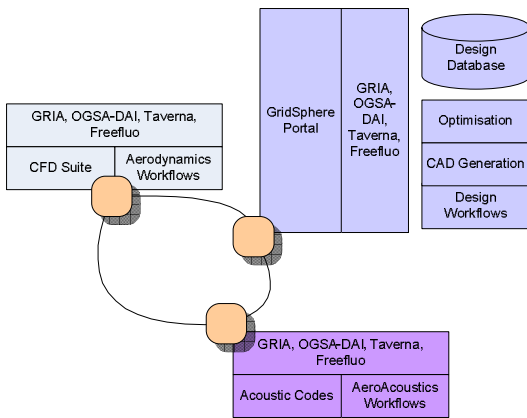


Figure 7: Overview of aerospace scenario deployment at three different sites

Implementation: The deployment of the aerospace prototype demonstrates another example of how Grid technologies can support inter-enterprise collaborative development of complex products. Each organization within the aerospace deployment operates as a GRIA service provider offering specialized engineering services such as optimization, parameterized CAD generation, aerodynamics and aero-acoustics. GRIA’s explicit business process support for dynamic, bi-lateral QoS agreements allows project managers at aerospace companies to create inter-Enterprise multidisciplinary design teams in a secure, managed, auditable and accountable environment (Figure 7).

Aerospace engineers integrate legacy applications as Grid services using GRIA and compose these applications into workflows using Taverna. The workflows can then be published to GRIA’s workflow enactment service, allowing clients to compose hierarchical distributed workflows that cross organization boundaries. At the lowest level, the aerospace workflows consist of a simple computational sequence of meshing, solving and post-processing. For example, the aerodynamics workflow performs a CFD (Computational Fluid Dynamics) analysis on the aircraft geometry and the corresponding mesh files in order to extract the flow solution that contains useful information about the forces on the aircraft. For the implementation of the aerodynamics service, the SOLAR CFD system is used. At higher-levels, the workflows are much more complex. For example, the optimization explores the design space by creating a design of experiments (DoE) from a given input specification and iteratively calculating the results for each design point in the DoE.

The first stage of the optimization process for calculating and storing the DoE is based on the design parameters read from the OGSA-DAI design database [11]. The inputs to the DOE calculation are: The choice of DOE algorithm, the number of design points required, the upper and lower bounds of the parameters, and a “current best set” of parameters (optional). The output of the DOE calculation is a set of design points that provide input to the compute workflow for calculation of the geometry, lift, drag and noise (Figure 8). The “DOE” tasks are all based on the OPTIONS package [12], and the Geodise toolkit [14] has been extended to allow Matlab engineering design workflows to be integrated within the framework.

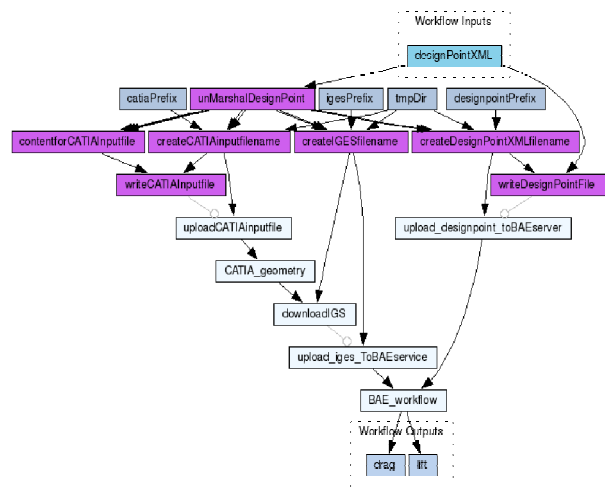


Figure 8: Aerospace compute workflow implemented using Freefluo/Taverna

The Prime Contractor accesses the design service using an engineering portal. The portal provides a simple interface to: create, edit and delete design jobs, provide parameterization of input specification, view job state during execution and view the final results.

The aerospace application portal was developed using GridSphere [13]. The portal provides a data-driven design interface where the design data is accessed using an OGSA-DAI service running at the design service provider. When a user creates a new job the job specification page is displayed. The user can enter design parameters and save these parameters to the OGSA-DAI data service. The design optimisation is then executed and workflows orchestrated reading and writing to the OGSA-DAI design database that can then be retrieved by the portal and viewed by the user.

5. Summary and future work

In this paper we discussed the requirements for developing and using distributed inter-domain Grid-based workflows for engineering applications and product design. We presented a generic approach for extending existing workflow systems to support the requirements for business partnerships based on the GRIA middleware which provides a fully decentralized management approach. We have also presented a generic approach for achieving run-time interoperability between different workflow execution engines including Taverna and InforSense KDE based on using GRIA services. The current implementations are based on GRIA 4.1 but these will be updated to use GRIA 5 providing a more flexible security and management infrastructure based on Service Level Agreements.

We are currently investigating other forms and approaches for workflow system interoperability, including interoperability with BPEL workflow engines. The BPEL language is primarily concerned with the automation of business processes. The use of use of Web Service protocols enables such processes to span different organizations. However, the focus of business process workflows is mainly on enabling automated interactions based on a set of predefined business rules. In contrast, the focus of scientific and engineering workflows is on the manipulation of larger data sets and coordinating the execution of long-running compute intensive tasks. They are adapted and changed by the scientists and engineers based on the problem they are solving.

In this paper we have used compute intensive examples from the automotive and aerospace industries. We are currently investigating how our approach to Grid-enabled workflows can be used in other application areas, including the pharmaceutical sector, where applications are mainly data intensive rather than compute intensive.

References

- [1] InforSense KDE (2006). www.inforSense.com
- [2] S. AlSairafi et al (2003). The Design of Discovery Net: Towards Open Grid Services for Knowledge Discovery. *International Journal on High Performance Computing Applications*, Vol. 17 No 3: 297-315.
- [3] T. Oinn, et al (2004). Taverna: A Tool for the Composition and Enactment of Bioinformatics Workflows. *Bioinformatics Journal*, Vol. 20 No. 17:3045-3054.
- [4] M. Surridge et al. (2005). Experiences with GRIA — Industrial Applications on a Web Services Grid. In

Proceedings of the *First International Conference on e-Science and Grid Computing*, pp. 98-105. IEEE Press.

- [5] SIMDAT (2006) <http://www.simdat.org>
- [6] S. Tai et al (2004). Composition of Coordinated Web Services. In Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware, pp 294 – 310. Springer-Verlag.
- [7] J. Cao et al (2003). GridFlow: Workflow Management for Grid Computing. In Proceedings of the 3rd International Symposium on Cluster Computing and the Grid, pp 198. IEEE Computer Society.
- [8] K. Amin et al (2004). GridAnt: A Client-Controllable Grid Workflow System. In Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 7 - Volume 7. IEEE Computer Society.
- [9] R. Stevens et al (2003) myGrid: Personalised Bioinformatics on the Information Grid. In Proceedings of 11th International Conference on Intelligent Systems in Molecular Biology. *Bioinformatics* Vol. 19: i302-i304.
- [10] M. Addis et al (2003). Experiences with e-Science workflow specification and enactment in bioinformatics, Proceedings of UK e-Science All Hands Meeting 2003, pp 459-467.
- [11] M. Antonioletti et al (2005). The Design and Implementation of Grid Database Services in OGSA-DAI. *Concurrency and Computation: Practice and Experience*, Volume 17, Issue 2-4: 357-376.
- [12] A. Keane et al (2005). *Computational Approaches for Aerospace Design: The Pursuit of Excellence*. Wiley.
- [13] J. Novotny et al (2004). GridSphere: A Portal Framework for Building Collaborations. *Concurrency and Computation: Practice and Experience*, Vol. 16 No. 5.
- [14] S. Cox et al. (2002) Grid Enabled Optimisation and Design Search (Geodise). In Proceedings of UK e-Science All Hands Meeting, pp. 54-55, Sheffield, UK.

Acknowledgements

The SIMDAT project has received research funding from the European Commission under the Information Society Technologies Programme (IST), contract number IST-2004-511438. The project is a consortium of more than 26 partners from industry and academia as well as non-profit organizations. The authors wish to thank their colleagues at InforSense Ltd and the University of Southampton as well as their collaborators in the SIMDAT consortium for their help and support.