Distributed Workflows for Multi-physics Applications in Aeronautics

Toàn Nguyên [1], Jean-Antoine Désidéri [1] and Jacques Périaux [2]

[1] *INRIA, 655, Av. de l'Europe, FR-38334 Saint Ismier, France*

Toan.Nguyen@inrialpes.fr, Jean-Antoine.Desideri@sophia.inria.fr

[2] *Dept. Mathematical Information Technology, Agora, PO Box 35, FI-40014,*
University of Jyväskyä, Finland
jperiaux@gmail.com

Abstract

The industry requires innovative technologies to support the numeric design and simulation of manufactured products in order to reduce time to market delays and improve the performance of the products and the efficiency of the industries in the global competitive market. Innovation also requires advanced tools to support the design of new products. For example, remote teams are working collaboratively on the preliminary design of future aircraft that will be "safer, quieter, cleaner", and environmentally friendly by 2020. The automotive industry has similar concerns. The telecom industries (e.g., mobile phones design) and nuclear power plant design face large-scale multi-physics simulation and optimization challenges. This paper suggests that distributed workflows running on computational grids are adequate to support their application needs.

Keywords: Workflows, Multi-physics Design, Grids.

1. Introduction

The aircraft industry aims at virtual flight tests [9] for new commercial aircraft and at their virtual certification in the near future [11]. This means that reduced in flight prototype testing will occur. This means also that detailed design, numeric simulation and optimization will be achieved, including optimization of the aircraft flight dynamics and engine efficiency.

In order to achieve such goals, various disciplines must interact for the aircraft design and simulation, including structural, aerodynamics, acoustics, electromagnetics, flight command systems, etc (Fig. 1).

Such expertise is usually available in various teams distributed among the various partners of the projects.

It is therefore important that the project management includes a global protocol for the team interactions. It entails that various experts using different specific tools interact in a common collaborative environment [10].

Workflow techniques have long been used in the industry and service sectors [1]. However, the control techniques used are usually dedicated to documents and project management in the business sector, involving a control flow approach. In contrast, the e-science sector has extensively used a dataflow approach for the processing of large numeric data sets.

In order to support efficiently the industrial projects to come, the workflow techniques that are necessary must include:

- distributed support to collaborative teams
- deployment, management and monitoring of distributed workflows
- hierarchical composition of distributed workflows
- distributed execution of workflows on wide-area grid computing infrastructures
- immersive visualization techniques
- fast transfers of petabytes volumes of data
- secure and reliable access and execution of large application codes that invoke remote software and data

The paper is organized as follows. Section 2 deals with existing workflow approaches in the business and engineering design sectors. Section 3 deals with distributed workflow approaches. Section 4 gives details of a grid-based Web services approach for multidiscipline design. Section 5 is a conclusion.
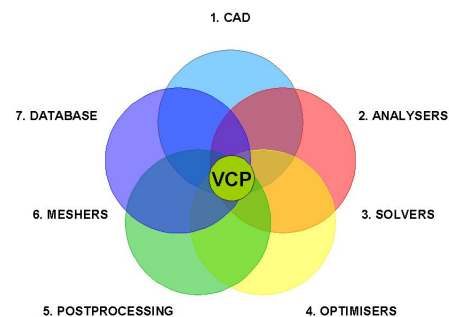

Figure 1. Tool interactions.

2. Workflow Approaches in Science, Business and Industry

There are basically two categories of workflow systems based on their control approach. Historically, they have been used extensively in business for administrative processing of documents throughout industry and commerce. They allow exhibiting the document processing protocols and thus improved traceability and improving the efficiency of administrative services. Well know business process languages have then appeared on the market. A "control flow" approach has usually been implemented in these systems: procedure cascading has focused on synchronization and serialization issues of the processes involved (Fig. 2)
.

2.1 Dataflow Approach

Eventually, workflows have been used in science applications for the processing of large sets of data. Here, new factors have been taken into account such as performance and parallelization of sub-processes. Related to threading approaches, they have focused on "on the fly" synchronization of processes. Here, processed data are transferred immediately to subsequent sub-processes to speed-up the production of the result data. Such approaches are qualified "dataflow". They are widely used in e-science applications e.g., YAWL (www.yawl-system.com).

Extensions and deployment to parallel environments are often implemented because this approach neatly matches thread control and processing on parallel architectures, e.g., PC-clusters.

Their extension to distributed computing systems is however questionable because they generate heavy communication loads between remote processing units.

2.2 Control-flow Approach

It appears that dataflow approaches are amenable to tight coupling between the processes involved, while control flow approaches are amenable to loose coupling.

While the first one are well-suited to parallel processing, it is clear that the second ones are adapted to distributed processing deployed on remotely located computing systems, e.g., grids infrastructures. It is our opinion that a combination of both approaches is particularly well-suited for the deployment of distributed workflows involving parallel components running on remotely connected parallel architectures, e.g., wide -area grids of PC-clusters [5].

3. Distributed Workflows

A range of issues appear when combining distributed and parallel computing with workflows. The rationale behind this approach is the complexity of innovative multi-physics applications where multiple codes are invoked to contribute to the optimization of the design goals.

Expertise from remote teams having to cooperate can be supported by collaborative environments [2]. However, deploying, managing and monitoring the applications running in these environments are still a challenge. Their definition is also fundamental in order to simplify their implementation and access to engineers [3].

High-level graphic interface, including immersive systems, are a must, but the construction of application workflows that may include legacy software is of paramount importance. One aspect is the support of composite workflow, i.e., workflows incrementally constructed that invoke remotely existing workflows.

3.1 Workflow composition

3.1.1 Composite workflows. Composite workflows are used to build complex applications requiring a number of distributed codes or services which interact in a controlled way. This includes legacy application software that are running on specific computing hardware and cannot be moved and also immersive visualization systems. It also includes large volumes of data which are of interest to the applications and cannot be transferred. This is also the case for large simulation models: 3D aircraft models, etc.

3.1.2 Hierarchical workflows. The simplest approach is to consider hierarchical workflows which can be built incrementally using existing workflows. This approach can easily be extended to remote workflow to implement distributed computing environments. It also complies with Virtual Environments and Collaborative Environments because distributed teams can cooperate by publishing their workflows to other remote collaborating teams [6].
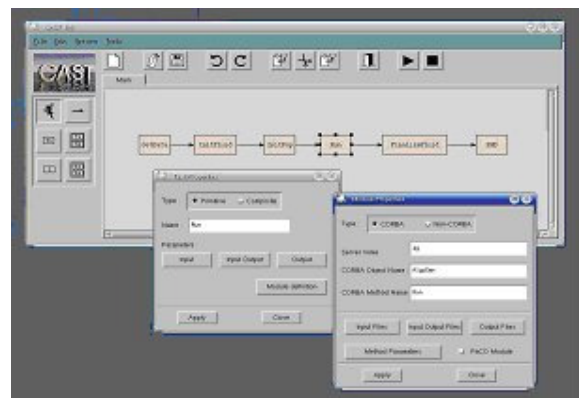

Figure 2. A workflow interface.

3.1.3 Embedded workflows.
A more sophisticated approach is to build embedded workflows, i.e., workflows that are not limited to hierarchical approaches, but include also interactions among sub-workflows whatever their level in the hierarchy. This builds workflow graphs. They are very useful for complex applications that involve several iterations among sub-workflows.

3.1.4 Nested workflows.
Nested workflows are useful for controlling remote workflows that interact at run-time. The control flow therefore might need to jump from inner sub-workflows to outer workflows and vice-versa. This situation is not compatible with the strict hierarchical and embedded approaches.

3.2 Distributed code and data access

Except for the use of specific techniques, such as dedicated port assignment, cross-domain access to data and software can be a very complex task. This is due to necessary security policies. It requires special authorization mechanisms to be implemented. Single signature granting access to multiple domains requires also specific access management tools. Some are implemented for application deployment on grid computing environments [3].

Web service implementations can alleviate somewhat these constraints but require ad-hoc wrapping techniques to encapsulate software and data [7].

3.3 Dynamic distributed control

Web services can also help for the dynamic control of distributed tasks provided they are encapsulated or invoked in a such way. Depending on the sophistication of the application software, synchronization can however be a complex task.

For example, optimization software can produce results that can be processed asynchronously in parallel by subsequent tasks. This is the case of evolutionary optimization algorithms [6].

Synchronization among the subsequent tasks to gather and process their results when they are distributed can be challenging. This is because their completion is based on the termination of the feeding optimization software. Therefore, termination conditions are dependent on runtime production of the results by the optimizers and the set of all subsequent tasks.

Should this set be dynamically defined and invoked, e.g., based on the volume of intermediate results, the synchronization must take into account a varying number of runtime parameters.

3.4 Verification and validation issues

An important issue deals with verification and validation of workflows. It is out of the scope of this paper, and a first hypothesis is that local workflows are proved correct and have been certified.

Concerning distributed workflows, contingency plans are limited by the boundaries of distributed software proof of correctness. It is well known that distributed software proof is a hard task, and that runtime errors are somewhat difficult to reproduce and checked…

4 Workflow Infrastructures for Multi-physics Design

Multi-physics design in aeronautics includes several disciplines and various tools that pertain to each particular expertise involved. This includes CAD tools, meshers, solvers, analyzers and optimizers, which in turn are used to modify the meshes in iterative and incrementally optimized design processes.

Multiple solvers and analyzers are used cooperatively to solve multi-physics challenges. In turn, subsequent optimizers are used to reach global optimum under the various constraints of the disciplines involved, and possible uncertainties that are taken into account, for example uncertainties concerning the angle of attack or Mach number for the flight conditions considered [4].

4.1 Middleware support

4.1.1 Grids.
There are many options for the control and support of distributed software. Grid computing environments have been the subject of large number of software development and experiments in the past decade [7]. They are the basis of large computing environments, particularly in e-science applications, throughout the world [1]. The corresponding middleware manages resource discovery, allocation and job execution and synchronization. Also, security issues are dealt with, as well as checkpointing and restart facilities, although less frequently [3].There are a number of middleware available, most are freeware and open source, e.g., Globus, Unicore, g-Lite, etc. The main difficulty lies in the technical expertise required to deploy and use them [5]. This should change in the future, but our opinion is that today, the best answer lies in the use distributed workflows. This is because they are application oriented and tend to hide to their users the technicalities of grid and distributed computing [1].

4.1.2 Web servers.
Web servers are also another seamless solution to distributed computing. Because they can be connected to Web browsers, they are user friendly and do not have the steep learning curve required by grid computing environments.

However, they also require advanced programming skills to implement the interactions between the browsers and the application software. This makes use of various tools like Java, PHP scripts, etc.

4.2 Web services

4.2.1 Wrappers. Web services are a technique used to simplify Web and grid programming. Although they were initially not compatible with grid services, they have been merged into one unified framework [4]. The idea was to adapt the web services to context sensitive services or "stateful services" for application deployment [8]. This convergence opens great perspectives for seamless distributed application deployment on the grid. It combines the ease of use of the Internet with powerful computing environments based on collaborative hardware and software, e.g., simulation environments running on several remote PC-clusters.

4.2.2 Nested services. Similar to distributed components, services can be combined in more complex ways. This includes composite services, hierarchical or embedded services, as well as nested services. In the latter, services can invoke one another before completion, giving rise to sophisticated programming tools. This is particularly useful when deploying hierarchical, nested and embedded workflows. Workflows can then be invoked by dedicated services in charge of the attached parameters and configuration issues: data management and transfer, synchronization and event management, etc.

4.3 Distributed workflow enactment

4.3.1 Initialization. Distributed workflow enactment requires several critical operations to succeed. This includes runtime parameters initialization, software code localization, data files localization and allocation, processors and memory allocation on remote sites. All of these have to be successfully completed and acknowledged by the remote systems implied. Distributed resource allocation systems have been designed for grid computing environments, including Web service implementations [4]. They can be very useful for distributed workflow systems [9]. Basically, they interface with local resource allocation and job scheduling services. They provide a single interface for multiple remote computing resources, e.g., GRAM for Globus. They often include basic security and fault tolerance services. For more details on GRAM, refer to: www.globus.org/toolkit/docs/3.2/gram/ws/

4.3.2 Nested transactions. Nested transactions, i.e., indivisible logical units of work, can be very useful in order to implement efficient execution strategies. This

means that results can only be usable when a specified block of control or set of web services or set of component workflows have successfully completed. This is orthogonal to parallelism because it implies a high-level or macroscopic degree of serializability in the execution order of services or workflows. It is however necessary to ensure the validity of critical results.

4.3.3 Checkpointing. An interesting side-effect of transactions is that they allow the implementation of checkpoints and restart protocols. Rollback and restart protocols are important in distributed execution environments because unexpected hardware and software failures may occur. Their impact on large-scale multi-physics applications can be devastating. Therefore, seamless, efficient and hopefully transparent rollback and restart procedures must be implemented, using checkpoint/restart protocols [3].

4.4 Reliable and secure distributed workflows

4.4.1 Authentication and Authorization. Authentication and authorization issues are the most critical aspects of distributed computing. They are the main barriers that hamper the use of grid environments by the industry. Although considerable progresses have been achieved, e.g.., certificates, PGP protocols, there remains psychological refrains due to vulnerability issues. Considerable damage can occur due to unauthorized access to industrial data, and workflow systems are no safer than any other system.
However, grid research has provided satisfactory solutions, e.g., GSI for Globus, to encourage the safe use of distributed computing infrastructures. We plan here to base the distributed workflow environment on such security systems. Authorization is planned here using X certificates.

4.4.2 Other security issues. Another facility concerning security issues is the use of virtual environments [7]. In such systems, users are isolated from one another by the virtual machine system, which protects them from undesirable intrusion and excursion from their private workspace. Even network communications, which are enabled in these environments through dedicated IP addresses, are protected by the use of specific firewalls and proxies [8].

4.5 Distributed workflow control

4.5.1 Embedded workflows. Distributed workflow control is a crucial issue in multi-physics applications due to the large volume of data involved, which can be in the order of petabytes, and the runtime duration of the application programs. Indeed, if local simulation and optimization applications run several days on PC-

clusters of a few hundreds of processors, they might involve much larger applications on multiple interconnected computing resources. This "application pull / technology push" race implies that always larger applications are developed, e.g., 3D instead of 2D models, full aircraft models instead of partial models, flight dynamics models instead of static flight conditions, etc.

It follows that a detailed synchronization scheme cannot be implemented because controlling the production of step by step terabytes of result data is impossible.

A decentralized control is necessary and involves sophisticated procedures, including standard and exception handling ones. This complexity is the main challenge from an operational point of view. It is essential that they are correct and valid, because this is the source of the reluctance for large communities adherence to distributed computing.

From a technical point of view, synchronization mechanisms can be implemented that use the grid services framework, i.e., stateful web services [9]. This approach requires that a workflow and all its component sub-workflows be wrapped by the appropriate web services, or at least that web services are used as proxies for the component workflows. This situation is easy to implement for hierarchical and embedded workflows. It is however more complicated for nested workflows.

4.5.2 Nested workflows. In the case of nested workflows, context sensitive information must be retained for each invocation of remote components. This means that either different instances of services must be created dynamically for each invocation of a component with the appropriate context, or that lists of dedicated contexts must be maintained for invocation of the remote components. The first approach seems safer and more fault-tolerant in a distributed environment.

4.5.3 Workflow execution engines. There are a wide range of workflow systems available, both freeware and on the commercial market [1]. Although very different and sometimes dedicated or developed by application expert communities, e.g., Taverna by bioinfo experts, the challenge here is the inter-operability of the various systems for a consistent and effective use in collaborative environments. The simplest approach is to use commonly agreed file formats for transferring the result data between component workflows. These files can be used as pipes for dataflow control or as full fledged storage media for intermediate results.

5 Conclusion

Distributed workflows are presented in this paper as an advanced tool to support large-scale multidiscipline projects. Examples are given in the aeronautics sector where multi-physics design and optimization are used to achieve simulation and optimization of new commercial aircraft. The aim is to implement virtual flight tests within a decade for large projects and attain virtual certification by 2020, thus avoiding costly and time consuming prototype aircraft development and testing.

Advanced technology based on distributed workflow techniques can support large distributed multidiscipline projects. They can be deployed on wide area (grid-based and broadband) networks involving remote expert teams working in collaborative environments [12, 13].

A number of points are not addressed in this paper, including workflow interoperability, knowledge sharing and ontology development and management, workflow specification languages and workflow modeling techniques. Other important issues are distributed computing items such as dynamic resource discovery and allocation, component relocation and dynamic reconfiguration which are out of the scope of this paper [3].

Augmented with workflow composition techniques, fast data transfers of petabytes files and immersive visualization environments, multidiscipline collaborative environments are a realistic goal today.

It is clear that Web-based distributed workflows running on distributed computing facilities that include large PC-clusters and supercomputers are a technical reality. Large aircraft manufacturers are testing and are currently planning the development of such environments for their daily operations to help them become an industrial reality.

References

[1] C. Goble *The Workflow Ecosystem: plumbing is not enough*. Invited Lecture. Third Grid@Asia Workshop. Seoul (Korea). June 2007.
[2] Janka A, Andreoli M. Desideri J.A. *Free form deformation for multilevel 3D parallel optimization in aerodynamics*. International Parallel CFD Conference. Universita de Las Palmas. Gran Canaria (Spain). May 2004.

[3] IEEE TCSC *Workflow Management in Scalable Computing Env*. www.swinflow.org/tcsc/wmsce.htm

[4] NESSI. Networked European Software and Services Initiative. Vision Document. Version 1.2b. May 2005

[5] T. Nguyên, V. Selmin *Collaborative Multidisciplinary Design in Virtual environments*. Proc. 10th Int'l Conf. Computer Supported Collaborative Work in Design. CSCWD'2006. Nanjing (P.R. China). May 2006.

[6] T. Nguyên, *Aeronautics multidisciplinary applications on grid computing infrasructures*. Invited lecture. Second Grid@Asia Workshop. Shanghai (P.R. China). February 2006.

[7] T. Nguyên, J. Periaux, *New Collaborative Working Environments for Multiphysics Coupled Problems*. Proc. ECCOMAS Thematic Conference "Coupled Problems 2007". Ibiza (Spain). May 2007.

[8] Next Generation Grids Expert Group Report 3. *Future for European Grids: GRIDs and Service Oriented Knowledge Utilities. Vision and Research Directions 2010 and beyond.* January 2006. http://cordis.europa.eu/ist/grids/ngg.htm

[9] Perrier P. "*Virtual flight tests*". PROMUVAL Seminar. National Technical University of Athens. November 2005.

[10] T. Nguyên, J-A Désidéri, J. Périaux. *Virtual Collaborative Platforms for Large-Scale Multiphysics Problems*. Proc. West East High-Speed Flow Field Conference. Moscow (Russia). November 2007.

[11] V. Selmin *Virtual and Physical Prototyping and Simulation in Aeronautics*. Proc. China-Europe Workshop on Aeronautics. Nanjing University of Aeronautics and Astronautics Nanjing (P.R. China). October 2007.

[12] Proc. 1st Int'l Workshop on Workflow Systems in Grid Environments (WSGE06). October 2006. Changsha (P.R. China). http://www.swinflow.org/confs/WaGe08/WaGe08.htm

[13] Proc. 2nd Int'l Workshop on Workflow Management and Application in Grid Environments (WaGe07). August 2007. Xinjiang (P.R. China). http://www.swinflow.org/confs/WaGe07/WaGe07.htm