

A Software Architecture for an Engineering Collaborative Problem Solving Environment

Ismael H. F. dos Santos
CENPES – Petrobras Research Center
Rio de Janeiro, Brazil
ismaelh@petrobras.com.br

Alberto B. Raposo, Marcelo Gattass
Tecgraf, Dept. of Informatics
Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro, Brazil
{abraposo, mgattass}@tecgraf.puc-rio.br

Abstract— In this paper we present a Service Oriented Architecture for executing engineering simulations and visualizing results in a Virtual Environment. The architecture composes a Collaborative Problem Solving Environment that enables engineers to setup computations in an integrated environment. The work is motivated by the necessity of finding effective solutions for collaboration of team workers during the execution of complex Petroleum Engineering projects.

Keywords- *Scientific Workflows, Virtual Environments, SOA*

I. INTRODUCTION

A Problem Solving Environment (PSE) is a specialized software system that provides all the computational facilities needed to solve a target class of problems. These features include advanced solution methods, automatic and semiautomatic selection of solution methods, and ways to easily incorporate novel solution methods. Moreover, PSEs use the language of the target class of problems, so users can run them without specialized knowledge of the underlying computer hardware and software technology [1].

Collaborative Problem Solving Environments (CPSE) focus on the development of a PSE coupled with collaborative environments to support the modeling and simulation of complex scientific and engineering problems. These capabilities enable engineers to easily setup computations in an integrated environment that supports the storage, retrieval, and analysis of the rapidly growing volumes of data produced by computational studies.

In this paper we present a Service-Oriented Architecture (SOA) for a Collaborative Problem Solving Environment for Petroleum Engineering (PE), tailored for assisting the control and execution of engineering projects in the oil & gas industry. This work is motivated by the necessity of finding effective solutions for collaboration of team workers during the execution of large and complex Petroleum Engineering projects. Those projects usually require the execution of a large number of engineering simulations, encapsulated as engineering services, combined in different orders and rearranged in different subsets according to project requirements. By means of a Workflow Management System

(WfMS) users are able to orchestrate the execution of engineering simulations as workflow tasks that can be arranged in many different ways. Within a workflow, as its last step, the most interesting cases can also be selected for visualization in a distributed collaborative session.

The presented solution, called Collaborative Engineering Environment (CEE) is intended to create a useful workspace for collaboration through the composition of different tools for distributed group work and a grid computing middleware:

1. Scientific Workflow Management System (ScWfMS) – a BPEL-workflow implementation is used as a process-oriented tool to control the execution of PE projects;
2. Videoconference System (VCS) – CSVTool is a custom VCS developed for supporting human communication, providing integrated audio and video channels, subject to defined control policies;
3. Virtual Reality Visualization tool (VRV) – EnViron, a custom VRV adapted for collaborative visualization of PE simulations in an immersive or desktop Virtual Environment (VE);
4. Grid Job Submission (GJS) – GridSAM, an open-source job submission and monitoring service integrated to CEE.

Thus, CEE provides an integrated environment for controlling the execution of Petroleum Engineering projects involving geographically distributed teams. It also allows an easy integration of different engineering applications providing team workers with means of information exchange, aiming to reduce the barriers imposed by applications with limited or no collaboration support.

This paper is organized as follows. In the following section we present related works that inspired the development of the CEE. In section 3, we discuss its main characteristics and requirements. In section 4 we present the CEE architecture and its implementation. A case study is detailed in section 5 and conclusions are presented in section 6.

II. RELATED WORK

Problem Solving Environments have been built for a number of scientific domains. For example, Parker et al. [2] describe SCIRun, a PSE that allows users to interactively compose, execute, and control a large-scale computer simulation by visually “steering” a dataflow network model. SCIRun supports parallel computing and output visualization, but originally has no mechanisms for experiment managing and archiving, optimization, real-time collaboration, or modifying the simulation models themselves.

Vistrails [3] is a visualization management system that provides a Scientific Workflow infrastructure which can be combined with existing visualization systems and libraries. A key feature of Vistrails is the support for data exploration. It separates the notion of dataflow specification from its instances. A dataflow instance consists of a sequence of operations used to generate a specific visualization. By maintaining detailed data provenance infrastructure of the exploration process, in a structured way, the system allows the visualization experiments to be queried and mined. Vistrails approach inspired our CEE strategy but some of the differences of the CEE are the use of a BPEL (Business Process Execution Language) ScWfMS, the focus on immersive and realistic visualization and, for the current version, the absence of a data provenance mechanism.

Paventhan et al. [4] proposed the creation of WindTunnel, a Scientific Workflow for wind tunnel applications. They observed that scientific and engineering experiments often produce large volumes of data that should ideally be processed and visualized in near real-time. The difficulty to achieve this goal is that the overall turnaround time from data acquisition, movement to a data processor and visualization of the results is frequently inhibited by factors such as manual data movement, system interoperability issues, manual resource discovery for job scheduling and disparate physical locality between the experiment and the user workstation. They argued that customized application specific workflows can reduce the time taken to accomplish a job by automating data flow driven activities, supplementing or replacing manual user-driven tasks. WindTunnel provides a series of workflow activities allowing the users to compose sequential workflows and seamlessly access Grid services. WindTunnel approach also inspired the development of our CEE by combining the ScWfMS with the execution of engineering applications in a Grid infrastructure.

In the Geology field, Kreylos et al [5] presented an approach for turning immersive visualization software into a scientific tool. They created immersive visualization software, with measurement and analysis tools that allow scientists to use their real-world skills and methods inside a virtual environment. They emphasized that VR visualization alone is not sufficient to enable an effective work environment. This observation has stimulated us to create additional tools for the CEE VR visualization component (CEE-VRV).

The necessity of integrating a myriad of different applications to solve common PE problems motivated us to use an Enterprise Application Integration (EAI) approach for CEE. Recently, EAI has been greatly simplified by the adoption of an SOA integrated with an Enterprise Service Bus [6].

III. CEE ELEMENTS

Earth Sciences and Engineering are challenged to manage and interpret increasing amounts of data coming from the field or generated by computer simulations. The typical work of a scientist and engineer consists in first detecting features, then measuring them, and finally generating a model that supposedly try to explain those observed features. This visual approach to science and engineering is powerful, as the human brain excels at visually identifying patterns. As Edward Tufte [7] wrote more than two decades ago: “At their best, graphics are instruments for reasoning about quantitative information. Often the most effective way to describe, explore and summarize a set of numbers – even a very large set – is to look pictures of those numbers”.

The importance of three-dimensional modeling and visualization has led engineering companies to increasingly adopt the use of Virtual Reality Centers in order to favor visual communication in technical work sessions and decision-making meetings. In this kind of environment, collaboration is greatly improved, as compared to the use of desktop displays, mainly due to the fact that people share the same physical space, with their attention dedicated to large-size representation of their models, facilitating the communication of concepts and reducing misunderstandings.

VR visualization technologies enhance the knowledge content within the engineering disciplines. In conjunction with collaboration both provide valuable insights for better decision making with risk mitigation. Dodd [8] has mentioned that the next big management push is the empowerment of interdisciplinary teams with collaboration tools that include remote and immersive visualization on the desktop. Sharing the same opinion of Dodd’s we emphasize that the combination of Collaborative tools and VR visualization constitutes a powerful component to our CEE architecture.

The development of a custom videoconferencing system allowed us to automatically establish videoconferencing channels among the participants of a conference which greatly simplify and improve the communication.

According to Ellis [9], Workflow Management Systems emphasize coordinated communication allowing groups of people execute, monitor, and coordinate the flow of “work cases”, in our context engineering simulations, within a distributed environment. A combination of a CPSE with a Workflow Management System and VR visualization, which is not addressed by the systems presented in the previous section, constitutes a strategic enabler for a successful data exploration and knowledge dissemination among workers in engineering enterprises. Data exploration through visualization requires scientists and engineers to go through several steps. In essence, they need to assemble complex

workflows that consist of dataset selection, specification of series of operations that need to be applied to the data, and creation of appropriate visual representations, before they can finally view and analyze the results. Usually, insight comes from comparing the results of multiple visualizations that are created during the data exploration process. Unfortunately, today this exploratory process is far from interactive and contains many error-prone and time-consuming tasks.

The developed CEE, as a specialized CPSE, allows users to collaboratively solve their problems through the use of predefined workflows or assembling new ones. Each workflow comprises a sequence of simulations, in the form of workflow tasks, which usually ends with a collaborative visualization task. This task creates a collaborative session supported by the CEE-VR Visualization component. Further details are presented later in section 4.

To achieve its goals CEE needs to be extensible, flexible and platform-independent, allowing a transparent flow of information among different teams and their models. The difficulties in building an effective CEE should be scrutinized in five domains:

1. Collaborative work - in this domain there is the necessity of providing effective human-to-human interaction and communication for solving conflicts and enhancing group productivity. Also there is the necessity of some support for coordinating the execution of tasks.

2. Project management - this domain points to the necessity of reducing costs and time-to-market of new products, which further requires a computerized solution capable of managing projects, controlling time scheduling and costs. Multiple and different visions of the on-going project must be provided by the CEE while users have different background (e.g. managers, engineers) and need different types of information to accomplish their duties. CEE should also guarantee model and data persistency, levels of access control according to different user roles.

3. Data provenance - Data provenance is the capacity of maintaining information of how a given data product was generated [10], and has many uses, from purely informational to enabling the representation of the data product. It is a very important feature for any CPSE once scientists and engineers often create several variations of a workflow in a trial-and-error process when solving a particular problem.

4. Distributed execution - in this domain resides the necessity of involving specialists in different areas located in different places and using distributed resources, requiring the solution to have the ability to be easily and seamlessly distributed.

5. System interoperability - usually there is a myriad of software that specialists normally use to accomplish their tasks, which implies the necessity of interoperability among the components of the solution.

Based on a thoroughly analysis of the domain of PE projects used as our prototype scenario, we defined a set of collaborative and visualization features to deal with the difficulties described above. Now, we present the major CEE functionalities towards this direction.

A. Collaboration-Bus and Collaborative Applications

The applications available for Computer-Supported Collaborative Work (CSCW) can be classified depending on how the support for collaboration is related to the application implementation. They can be seen as collaboration-aware or collaboration-unaware applications [11]. Collaboration-unaware applications are originally developed to be single user applications, but may be used collaboratively by means of an external support system. This external support system may be an application-sharing system, such as Microsoft NetMeetingTM, or a GUI event multiplexing system. In both cases the applications do not explicitly support collaboration; they are implemented as single user applications.

Collaboration-aware applications, on the other hand, are especially developed or adapted to support collaboration. They typically constitute distributed systems, with centralized or replicated data sharing, where each user has access to a locally executed application instance. All running applications are connected to a server process, in a client/server architecture, or interconnected, peer-server or peer-peer, and exchange information over designated communication channels. All the peers are aware of the communication channels shared with its peer applications; which information is exchanged among them; the number of connected peers and their role in the collaboration; and the coordination policies adopted by the group.

EnViron, our CEE-VRV, was adapted to be transformed into a collaboration-aware application with the support provided by the CEE collaborative infrastructure. The CEE-Collaboration Bus is created by the combination of an Enterprise Service Bus and a Message Oriented Middleware, with a Java Messaging ServiceTM (JMS) compliant implementation provided, in our case, by Apache-ActiveMQ [12]. Two types of channels are available, a public-subscribe channel for group communications and a single channel per peer, available for peer to peer communications. The CEE-Service Coordinator is responsible for orchestrating the creation and controlling of the Environ Collaborative Visualization Session connecting EnViron with any other engineering simulations that want to have their results collaboratively visualized by its CEE users.

B. EnViron – CEE VR Visualization Tool

EnViron (ENvironment for VIRtual Objects Navigation) [13], is a tool developed to facilitate the use of CAD models in VR applications. It is a system composed of a 3D environment for real time model visualization, and exportation plugins, which import model data from other applications, allowing EnViron to visualize and interact with different kinds of 3D data.

EnViron is integrated into CEE offering resources for real-time 3D visualization and interaction in CAD models with enough realism and performance to be used for collaborative virtual prototyping, design review, change management systems, training, and visualization of engineering simulations among other activities.

C. CSVTool – CEE Videoconference System

Videoconferencing systems contain no knowledge of the work processes, and therefore are not “organizationally aware”. These systems are best suited for unstructured group activities once that audiovisual connectivity and shared documents enable flexible group processes.

In CEE, a videoconference is started by the creator of the workflow when he wants to share the construction of the workflow or collaboratively analyze the selected simulation results. Each conference is attached to a CEE Environ Collaborative Session that is registered in the CEE-Server.

We have developed a multiplatform videoconferencing tool, CSVTool [14] (Collaboration Supported by Video). The use of a custom tool allows a tight integration of this service into the CEE, with no duplication of session-management functionalities, and the direct control of audio and video streams according to the coordination policies defined. It was designed with the goal of providing integrated multimedia communication to collaborative applications, but can also be used as an independent videoconferencing tool.

In addition to the transmission of audio and video on multi-participant, multi-platform sessions, the video stream sent by each participant can be switched from the image captured by the camera to the captured screen, to allow the use of video for remote display of the interface operation or for the presentation of other contents on the screen and for consistency checks.

D. CEE Scientific Workflow

Ellis [9] presents Workflow Management Systems (WfMS) as a tool to assist in the specification, modeling, and enactment of structured work processes within organizations. These systems are a special type of collaboration technology which can be described as “organizationally aware groupware”.

There are remarkable differences between Business (BWfMS) and Scientific Workflows (ScWfMS). In [15] the authors identified that in a scientific environment scientists typically specify their workflows themselves, while in a business environment, a system administrator is commonly responsible for this task. Another characteristic of ScWfMS is the need to trace workflow executions. An engineer may need to reuse a workflow in order to reproduce results. The operations a user performs on a given data must be recorded in order to provide engineers with the benefits of successful and unsuccessful workflows.

Scientific workflows often begin as research workflows and end up as production workflows. Early in the lifecycle, they require considerable human intervention and collaboration; later they begin to be executed increasingly automatically. Thus in the production mode, there is typically less room for collaboration at the scientific level and the computations are more lingering. During the research phase, scientific workflows need to be enacted and animated (fake enactment) far more intensively than business workflows. In this phase, which is more extensive than the corresponding phase for business workflows, the emphasis is on execution with a view to design, and thus naturally includes iterative execution. The corresponding activity can

be viewed as a “Business Process Engineering” (BPE). For this reason, the approaches for constructing, managing, and coordinating process models are useful also in scientific settings. In this way, Scientific Workflows are to Problem Solving Environments what Business Workflows are to Enterprise Integration (EI).

ScWfMS are more data-flow oriented while BWfMS are more control-flow oriented. BWfMS require the coordination of a number of small messages and document exchanges. In ScWfMS usually no documents undergo modifications. Instead, often a dataset is obtained via analysis and transformation of another dataset. BWfMS need complex control flow, but they are not data-intensive pipelines. On the other hand, ScWfMS must deal with the heterogeneity, complexity, volume, and physical distribution of scientific data. In addition to these data problems, ScWfMS often deal with legacy or third-party programs, which can also be heterogeneous, and possibly with no source code available.

The OASIS standards organization has defined the Business Process Execution Language (BPEL) as a standard-based way of orchestrating a business process composed of services [16]. As an execution language, BPEL defines how to represent the activities in a business process, along with flow control logic, data, message correlation, exception handling, and more.

There are a number of advantages from adopting BPEL for the orchestration of scientific workflows. There are industrial-strength enactment environments and middleware technologies available that exhibit a level of scalability and reliability that a research prototype could not match. The multitude of providers supporting BPEL creates a market, which means that it is a live standard with ongoing efforts to develop new features. Furthermore, BPEL could serve as a standard representation for scientific workflows and hence aid reproducibility. Finally, as a programming language that focuses on high-level state transitions, it could enable computational scientists to compose scientific workflows themselves, relieving them of a dependence on software engineers.

E. CEE Grid Job Submission and Monitoring System

GridSAM is a Grid Job Submission and Monitoring Web service for submitting and monitoring jobs managed by a variety of Distributed Resource Managers. GridSAM implements the Job Submission Description Language (JSDL) defined by the Global Grid Forum (GGF) [17]. Transparency of the underlying Grid scheduler being used to execute jobs on a Grid (in our case, Condor) is achieved by using GridSAM. Scientists only need to define the JSDL for their jobs once and not worry about which scheduler is used now or at any point in the future.

Condor is a specialized workload management system for compute-intensive jobs [18]. Condor provides a job queueing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management.

IV. CEE IMPLEMENTATION

Service-Oriented Architecture (SOA) is a style of architecting software systems by packaging functionalities as services that can be invoked by any service requester [19], [20]. An SOA typically implies a loose coupling between modules. Wrapping a well-defined service invocation interface around a functional module hides the details of the module implementation from other service requesters. This enables software reuse and also means that changes to a module's implementation are localized and do not affect other modules as long as the service interface is unchanged. Once services in SOA are loosely coupled, applications that use these services tend to scale easily because there are few dependencies between the requesting application and the services it uses.

The adoption of an SOA for CEE produced a reduction of technology development costs by leveraging functions already built into legacy systems. SOA architectures are indeed becoming a popular and useful means of leveraging internet technologies to improve business processes in the oil & gas industry nowadays [21].

An Enterprise Service Bus (ESB) is a pattern of middleware that unifies and connects services, applications and resources within an enterprise. ESB is a platform built on the principles of SOA and other open standards to help applications integrate seamlessly. Put into another way, it is the framework within which the capabilities of a business application are made available for reuse by other applications throughout the organization and beyond. The ESB is not a new software product, it is just a new way of looking at how

to integrate applications, coordinate distributed resources and manipulate information. Unlike previous approaches for connecting distributed applications, such as RPC or distributed objects, the ESB pattern enables the connection of software running in parallel on different platforms, written in different languages and using different programming models.

A basic ESB provides a messaging infrastructure along with basic transformations and routing. It mainly uses open standards like web services enabling application to talk. ESB is a centralized, scalable, fault-tolerant, service-messaging framework that:

1. Provides a transparent means for communicating with heterogeneous services over a diverse set of message protocols.
2. Provides a shared messaging layer by which enterprise engineering applications, services, and components can connect and communicate.
3. Can transmit messages synchronously or asynchronously to service endpoints and intelligently transform and secure the message content to meet the requirements of each service endpoint.
4. Provides sophisticated error recovery, allowing for failed message delivery, scalability problems, duplicate messages, network failure, etc.

The main objective of the ESB is to provide virtualization of the enterprise resources, allowing the business logic of the enterprise to be developed and managed independently of the infrastructure, network, and provision of those business services. Resources in the ESB are

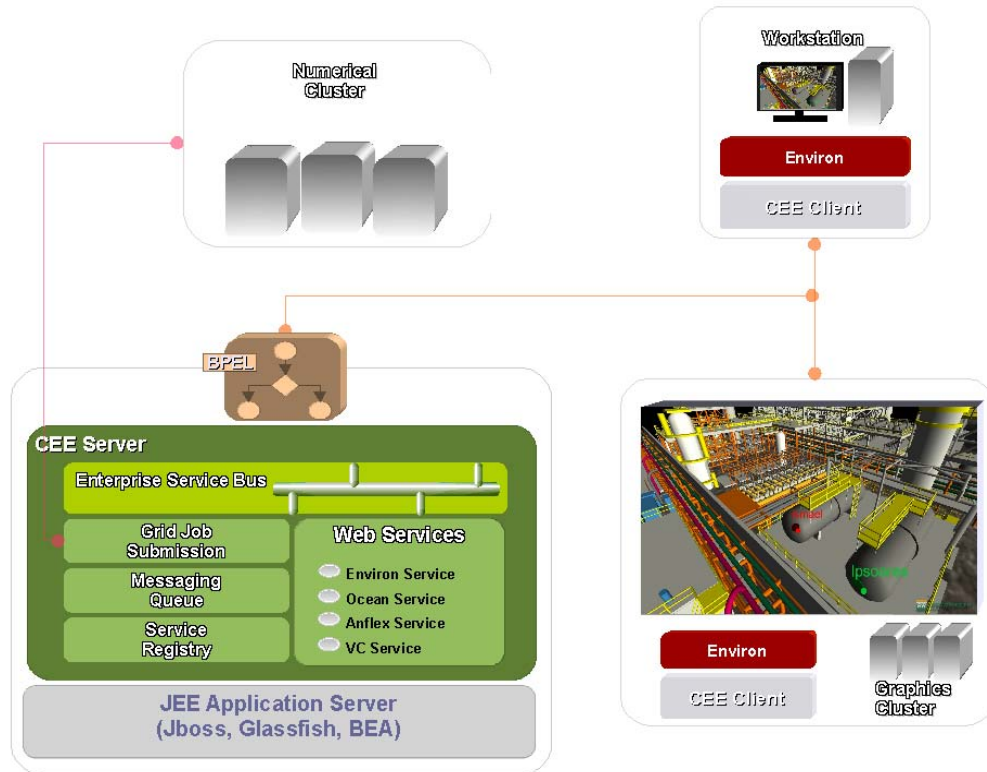


Figure 1. CEE Architecture

modeled as services that offer one or more business operations.

Concerning the difficulties in building an effective CEE, mentioned previously, the use of an ESB will furnish system infrastructure to enable distributed execution of applications as well as system interoperability.

A. CEE Architecture

Our proposed CEE has component-based architecture in order to facilitate the reuse of elements. The architecture of the CEE uses a BPEL ScWfMS as its kernel while the CSVTool, Environ, and the other components are seamlessly accessed through the ESB according to the collaborative necessities of the teamworkers.

When the service-oriented approach is adopted for designing the CEE, every component, regardless of its functionality, resource requirements, language of implementation, etc., provides a well-defined service interface that can be used by any other component in the environment. The service abstraction provides a uniform way to mask a variety of underlying data sources (real-time production data, historical data, model parameters, reports, etc.) and functionalities (simulators, optimizers, sensors,

actuators, etc.). A Workflow, actually, in our context, a Scientific Workflow, is composed by coupling service interfaces in the desired order. These workflows specifications are created through a graphical front end, workflow designer, and the actual service calls are generated automatically and have their execution controlled by the workflow engine.

CEE has a client-server architecture (Figure 1), where the CEE-server is deployed in a JEE (Java Enterprise Edition) Application Server (JBoss, Glassfish, etc) which allows better scalability and automatic transaction control. The CEE main services reside in the AppServer where a Service Registry is used to record all available services existent in the CEE-clients, for example the CEE VR Visualization tool, Environ, should be available on a CEE-client machine for allowing the user to participate in an Environ Collaborative Session, controlled by the CEE Service Coordinator.

The Environ Collaborative Session is implemented with the help of a Java Messaging Service (JMS) infrastructure available in the AppServer, providing a communication channel between peers participating in the Session. The channels are implemented as JMS topics, global channels, and queues, individual channels for each CEE-client (Figure

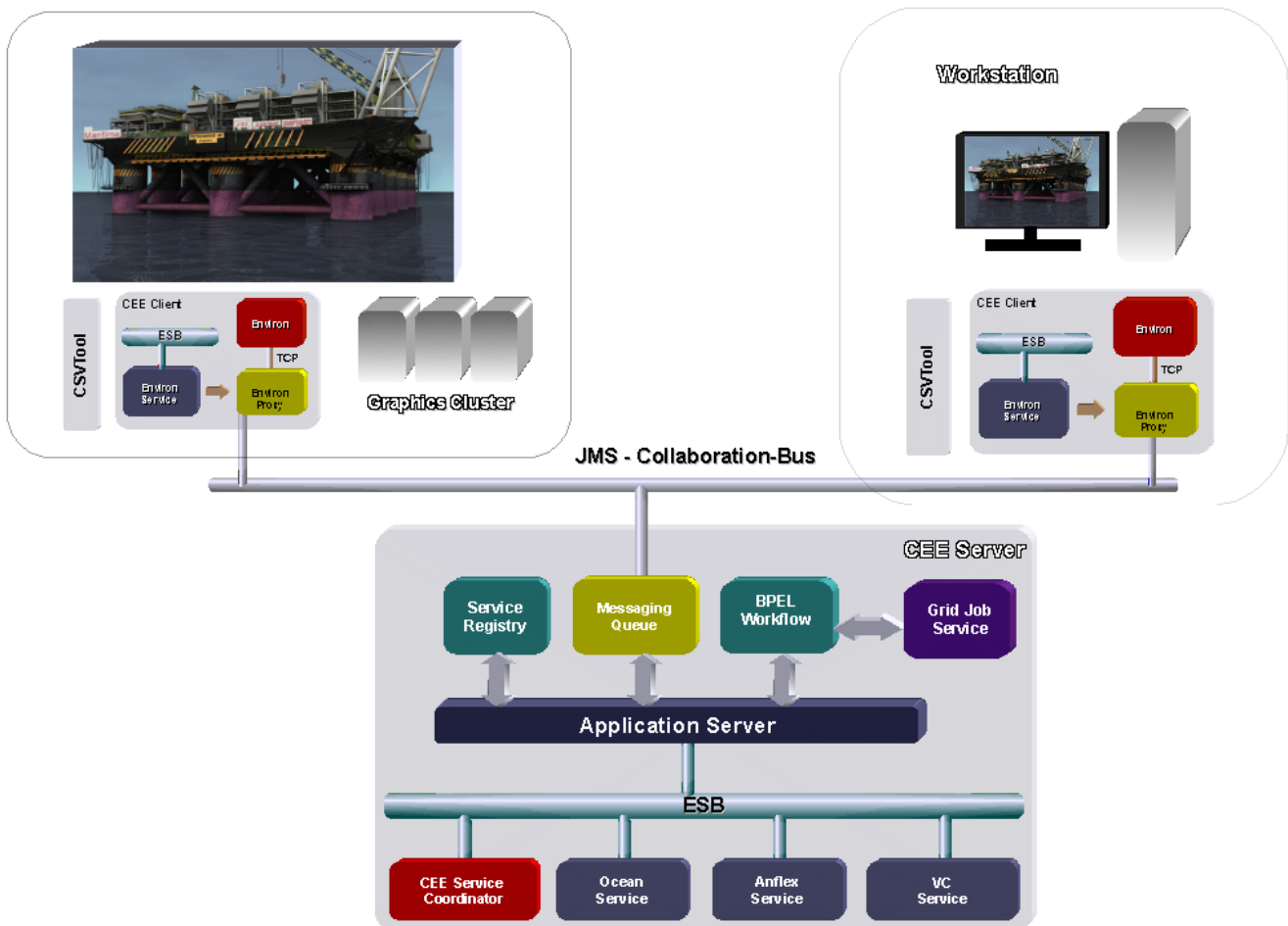


Figure 2. Collaboration-Bus using JMS messaging architecture

2). The Environ Collaborative Session is controlled by the CEE Service Coordinator in conjunction with every EnvironProxy which is started on demand by the Environ Service on all CEE-client machines participating in the session. Environ Service is a service daemon registered at the Service Registry in CEE-server indicating the availability of Environ in the CEE-client.

Whenever the user wants to create an Environ Collaborative Session he should select this activity in his workflow after the execution of the simulations, and define to whom he wants to collaborate. Upon having executed all simulations the workflow engine uses the ESB infrastructure to ask Environ Service to start Environ and its representative EnvironProxy in the collaborative session. EnvironProxy communicates with Environ by a TCP socket connection for sending and receiving commands. Once started the JMS messaging infrastructure is used to exchange commands between WfMS and EnvironProxy.

V. CASE STUDY - RISER ANALYSIS WORKFLOW

Floating production units (oil platforms) use ascending pipes, called risers, to bring the oil from the wellhead on the sea floor to the oil platform's separator system tanks. The risers are connected to the platform using special connections called "joints". To certify the operation of the risers for their entire life cycle (30 years or so), simulations of the stress applied to the riser system are conducted based on

meteo-oceanographic data about wind, tide and water currents. In order to avoid operational problems, simulations are made under extreme environment conditions to test against stress resistance. The riser analysis software used is Anflex [22], an internally developed Finite-Element-based structural analysis package.

For automating the process of validation and certification of riser analysis we have defined an Anflex-based riser analysis workflow controlled by the BPEL engine (Figure 3). Other services were also created for doing part of workflow activities. The workflow started with an Anflex base case supplied by the user, where the basic configuration of the experiment is defined such as a production unit, riser's geometry, soil batimetry, etc. Anflex service receives user input parameters from BPEL engine and is responsible for creating different analysis cases, with different meteo-oceanographic conditions provided by the Ocean service. After that, BPEL instructs CEE Grid Job Submission service to communicate with Condor to submit jobs for executing Anflex FE program for every generated analysis cases on available nodes of the Numerical Grid. Upon finishing the execution of Condor jobs, Anflex service is called again to select the worst cases that will be analyzed in an Environ Collaborative Session created by the CEE Service Coordinator involving the participant users, as defined in the workflow. The VCS Service is available at anytime for providing human to human interaction for solving conflicts.

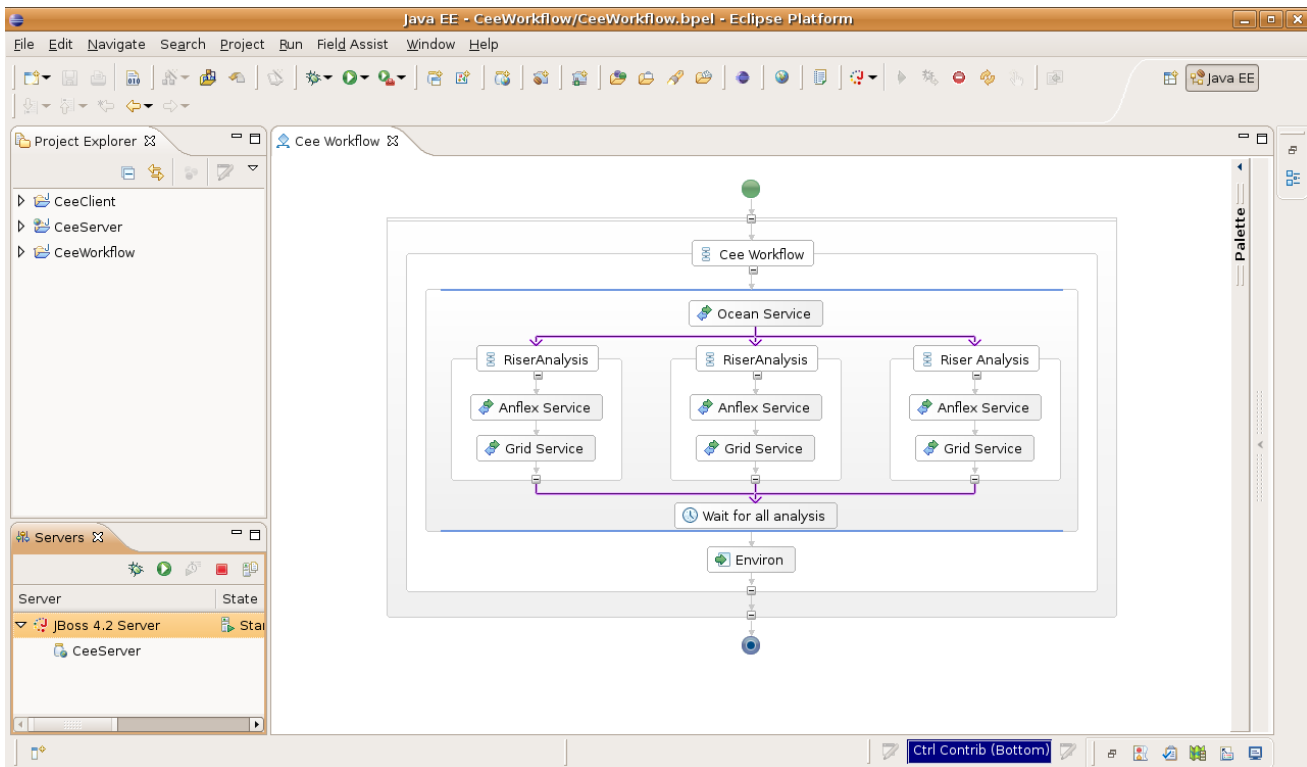


Figure 3. Riser analysis workflow – BPEL Engine

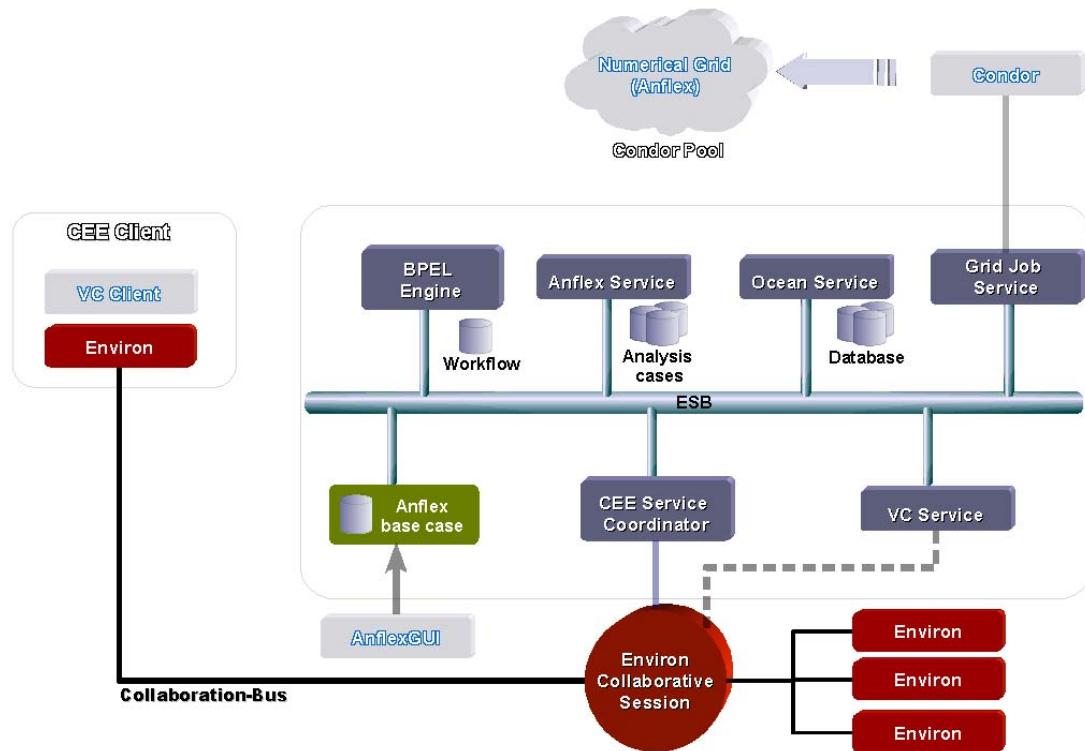


Figure 4. Riser analysis workflow – architectural diagram

In the Environ Collaborative Visualization Session (Figure 4) results of the simulations can be analyzed by users at a desktop or in an immersive VE. Among other resources, it is possible to playback the simulation, examine pipes, sea waves and ship movements, and track elements in the risers that are subjected to extreme conditions (e.g., high stress values). It is also possible to select any element in a riser and examine it carefully; especially those elements in places subject to great stress, such as the joint connections and the TDP (touch down point). Environ has special capabilities to show the user extreme values and where are they localized. Annotations, private or public (shared) can also be created by the users. Observe, in Figure 5, the presence of two users, represented by two distinct 3D-cursors, collaborating in a Environ Session were one of the users has created a private annotation that could be, for example, about an anomalous observed value.

VI. CONCLUSION

This paper presented an SOA of the CEE that we are developing. CEE is proving to be an effective Collaborative Problem Solving environment, allowing users to mitigate their problems during the execution of large and complex PE projects.

Scientific Workflows and Grid Computing enable the development of large Petroleum engineering applications. The ability to compose, design and execute rapid prototyping of experiments, provided by ScWfMS, and the grid philosophy of "one-demand" availability of computational

resources together with the capacity of sharing resources across organizational boundaries are valuable features for Petroleum Engineering projects.

In addition, we argue that SOA offer engineering companies a number of compelling benefits. It allows organizations to be able to respond efficiently to changes in the business and competitive landscape. It also allows seamless integration of applications and reduces technology development costs by leveraging functions already built into legacy systems, by re-using services developed for other process, and by simplifying maintenance and support through elimination of redundant, siloed applications.

As future works, we intend to pursuit some usability tests with specialized workflows developed for executing specific engineering applications in pre-defined sequences. We also plan to implement a data provenance approach for our ScWfMS.

This work has been focused on a solution for Petroleum Engineering projects; we believe that the proposed CEE could also be used in other areas.

ACKNOWLEDGMENT

The authors would like to thank Petrobras and Tecgraf/PUC-Rio for the expressive support. Alberto Raposo is partially sponsored by CNPq, project number 472967/2007-0.

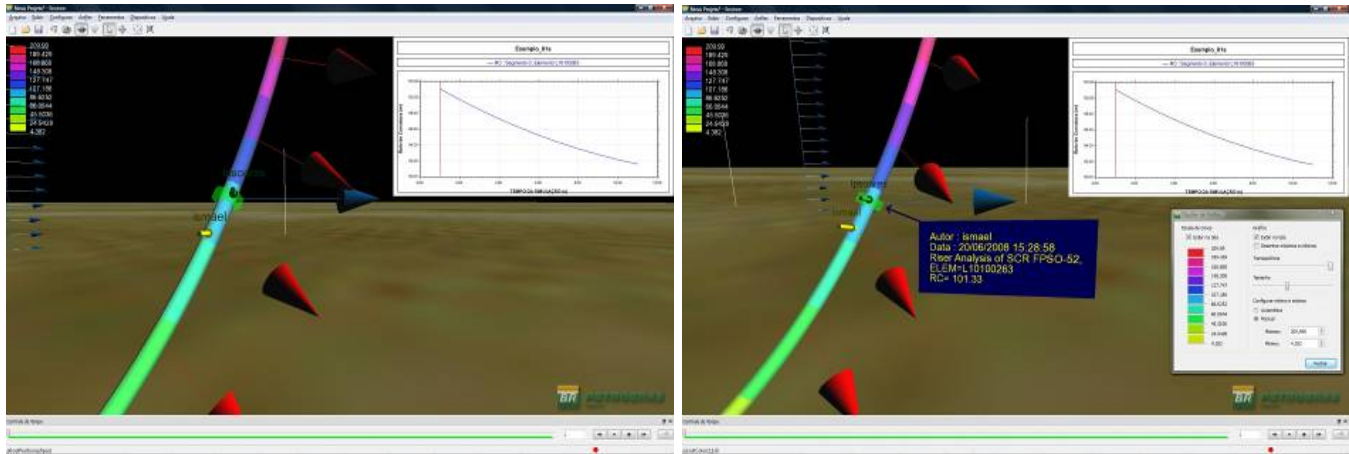


Figure 5. Two users in a CEE collaborative visualization session.

REFERENCES

- [1] E. N. Houstis, E. Gallopoulos, R. Bramley, and J. R. Rice, "Problem-Solving Environments for Computational Science", IEEE Computational Science and Engineering, vol. 4(3), 1997, pp. 18-21.
- [2] S. G. Parker, M. Miller, C. D. Hansen, and C. R. Johnson "An integrated problem solving environment: The SCIRun computational steering system", 31st Hawaii Int. Conference on System Sciences (HICSS-31), 1998, pp. 147-156.
- [3] S. P. Callaban, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo, "VisTrails: Visualization meets Data Management", Proc. ACM SIGMOD Int. Conf. on Management of Data, 2006, pp. 745-747.
- [4] A. Paventhan, K. Takeda, S. J. Cox, and D. A. Nicole, "Leveraging Windows Workflow Foundation for Scientific Workflows in Wind Tunnel Applications", Proc. 22nd International Conference on Data Engineering Workshops (ICDEW), 2006, pp. 65-75.
- [5] O. Kreylos, G. Bawden, T. Bernardin, M. I. Billen, E. S. Cowgill, R. D. Gold, B. Hamann, M. Jadamec, L. H. Kellogg, O. G. Staadt, and D. Y. Sumner, "Enabling Scientific Workflows in Virtual Reality", Proc. ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCIA), 2006, pp. 155-162.
- [6] M. Endrei, J. Ang, A. Arsanjani, S. Chua, P. Comte, P. Krogdahl, M. Luo, and T. Newling, "Patterns: Service-Oriented Architecture and Web Services", IBM RedBooks, April 2004. <http://www.ibm.com/redbooks>.
- [7] E. Tufte, The Visual Display of Quantitative Information, Graphics Press, Cheshire, Connecticut, 1983.
- [8] E. J. Dodd, "Visualization and Collaboration for the On Demand Upstream Petroleum Enterprise", IBM technical reports, 2004.
- [9] C. A. Ellis, "Workflow technology", in Computer Supported Cooperative Work, M. Beaudouin-Lafon, Ed., vol. 7 of Trends in Soft Series, John Wiley&Sons, 1999, pp 29-54.
- [10] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-science", SIGMOD Record, vol. 34(3), 2005, pp 31-36.
- [11] W. Reinhard, J. Schweitzer, G. Völksen, "CSCW Tools: Concepts and Architectures", IEEE Computer, vol. 27(5), 1994, pp 28-36.
- [12] Apache ActiveMQ website. <http://activemq.apache.org/>.
- [13] A. B. Raposo, E. T. L. Corseuil, G. N. Wagner, I. H. F. Santos and M. Gattass, "Towards the Use of CAD Models in VR Applications", Proc. ACM SIGGRAPH Int. Conf. Virtual-Reality Continuum and its Applications in Industry (VRCIA), 2006, pp. 67-74.
- [14] C. T. Pozzer, L. S. Lima, A. B. Raposo, and C. J. G. Vieira, "A Multi-user Videoconference-based Collaboration Tool: Design and Implementation Issues", Proc. 9th Int. Conference on CSCW in Design (CSCWD), 2005, pp. 547-552.
- [15] M. Weske, G. Vossen, C. B. Medeiros, and F. Pires, "Workflow Management in Geoprocessing Apps.", Tech. Report No.04-98-I, Univ.Muenster, GE, 1998.
- [16] OASIS Web Services Business Process Execution Language (WSBP). http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
- [17] W. Lee, S. McGough, S. Newhouse, and J. Darlington, "A standard based approach to job submission through web services", Proc.UK e-Science All Hands Meeting, 2004, UK EPSRC, pp. 901-905.
- [18] Condor – High Throughput Computing. <http://www.cs.wisc.edu/condor/>
- [19] R. High, S. Kindler, S. Graham, "IBM SOA Foundation: An architectural introduction and overview", IBM developer Works. <http://www-128.ibm.com/developerworks/webservices/library/ws-soa-whitepaper/>
- [20] E. Ort, "SOA Architecture and Web Services: Concepts, Technologies, and Tools", Sun Dev. Network, <http://java.sun.com/developer/technicalArticles/WebServices/soa2/>
- [21] R. Soma, A. Bakshi, A. Orangi, V. K. Prasanna, and W. D. Sie, "Service-Oriented Data Composition Architecture for Integrated Asset Management", Proc. SPE Intelligent Energy Conference and Exhibition (IECE), 2006.
- [22] M. M. Mourelle, E. C. Gonzalez, and B. P. Jacob, "ANFLEX - Computational System for Flexible and Rigid Riser Analysis", Proc. 9th International Symposium on Offshore Engineering, 1995.