Pontifícia Universidade Católica
do Rio de Janeiro

Ismael Humberto Ferreira dos Santos

A Collaborative Environment for Offshore Engineering Simulations based on Visualization and Workflow

TESE DE DOUTORADO

Thesis presented to the Postgraduate Program in Informatics of the Departamento de Informática, PUC-Rio as partial fulfillment of the requirements for the degree of Doutor em Informática.

Advisor: Prof. Marcelo Gattass
Co-advisor: Prof. Alberto Barbosa Raposo

Rio de Janeiro

April 2010

Pontifícia Universidade Católica do Rio de Janeiro

Ismael Humberto Ferreira dos Santos

A Collaborative Environment for Offshore Engineering Simulations based on Visualization and Workflow

Thesis presented to the Postgraduate Program in Informatics, of the Departamento de Informática do Centro Técnico Científico da PUC-Rio, as partial fulfillment of the requirements for the degree of Doutor.

Prof. Marcelo Gattass
Advisor
Departamento de Informática – PUC-Rio

Prof. Alberto Barbosa Raposo
Co-Advisor
Departamento de Informática – PUC-Rio

Prof. Bruno Feijó
Departamento de Informática – PUC-Rio

Prof. Marco Antonio Casanova
Departamento de Informática – PUC-Rio

Profa. Judith Kelner
Centro de Informática – UFPE

Profa. Marta Lima de Queirós Mattoso
Departamento Engenharia de Sistemas e Computação – COPPE-UFRJ

Profa. Veronica Teichrieb
Centro de Informática – UFPE

Prof. José Eugenio Leal
Coordinator of the Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 9 de abril de 2010

Ismael Humberto Ferreira dos Santos

Graduated in Applied Mathematics from Universidade Federal do Rio de Janeiro – UFRJ in 1997. He also has a master's degree in Applied Mathematics from the Federal University of Rio de Janeiro. He works for Petrobras (Brazilian Oil company) since 1987.

Bibliographic data

Santos, Ismael Humberto Ferreira dos

A Collaborative Environment for Offshore Engineering Simulations based on Visualization and Workflow / Ismael Humberto Ferreira dos Santos; advisor: Marcelo Gattass ; co-advisor: Alberto Barbosa Raposo. – 2010.

145 f. : il. (color.) ; 30 cm

Tese (doutorado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2010.

Inclui bibliografia

1. Informática – Teses. 2. Trabalho colaborativo auxiliado por computador. 3. Ambientes colaborativos para a solução de problemas. 4. Sistemas gerenciadores de experimentos científicos (workflows científicos). 5. Visualização colaborativa. 6. Ambientes virtuais colaborativos. 7. Engenharia offhore. 8. Óleo e gás. I. Gattass, Marcelo. II. Raposo, Alberto Barbosa. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

Aos meus pais, Ubirajara e Dalva,
aos meus queridos filhos
João Guilherme e Pedro Henrique e a
minha amada esposa Yone.

Acknowledgments

A meus pais, Ubirajara e Dalva, por tudo que sou hoje.

A minha esposa, Yone, pelo seu amor e incentivo, e pela abnegação com que abdicou de muitas comodidades e assumiu tarefas que me cabiam.

Aos meus filhos, João Guilherme e Pedro Henrique, que são a razão da minha existência.

Aos meus orientadores Marcelo Gattass e Alberto Barbosa Raposo, pelo invulgar empenho que dedicaram a esta empreitada, pelos conselhos oportunos em momentos decisivos da consecução da pesquisa e pela confiança que depositaram em mim desde o início.

Ao meu ex-Gerente e amigo Álvaro Maia pela oportunidade que me deu confiando na minha capacidade de realização desse doutorado.

Ao meu Gerente e também amigo Augustu Petrus Levy por depositar uma confiança infindável no meu trabalho.

A Petróleo Brasileiro S. A. e ao CENPES, por todo o apoio e suporte financeiro que foram fundamentais por terem proporcionado a oportunidade e os meios para realizar esta pesquisa.

Aos amigos da Petrobrás, Heitor Araújo, Luciano Pereira dos Reis, Rodrigo Toledo pelo incentivo e motivação para continuar nessa jornada.

Aos meus amigos Rogério Pinheiro, Paulo Gallotti e a equipe do Environ, que tanto me ajudaram na implementação desta tese.

Aos amigos do TecGraf e da Puc-Rio, por terem proporcionado não só um ambiente propício para a pesquisa mas também um ambiente de extrema camaradagem e companheirismo.

Aos componentes da banca, por suas sugestões.

Abstract

Santos, Ismael Humberto Ferreira dos; Gattass, Marcelo; Raposo, Alberto Barbosa. **A Collaborative Environment for Offshore Engineering Simulations based on Visualization and Workflow.** Rio de Janeiro, 2010. 145p. DSc. Thesis - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Deep-water production systems, including floating production units (platforms or ships) and all the equipments playing a part in the production process, are currently designed by means of complex computational modeling systems. Those systems involve the areas of structural calculus, meteo-oceanography (currents, waves and wind forces), hydrodynamics, risers (rigid or flexible steel pipes for carrying oil from the well in subsurface up to the production unit), mooring systems, submarine equipment, seabed foundations and Geologic/Geotechnical risk assessment. The project of a new production unit is a lengthy and expensive process, that can last many years and consume hundreds of million of dollars, depending on the complexity of the unit and how mature is the technology developed to make the project technically and economically feasible. Projects are conducted by diverse specialists, sometimes geographically distributed, yielding independent but highly interrelated artifacts and results. The need for collaboration is an inherent characteristic of deep-water floating production unit projects. The possibility to share information among users, control the execution of different modeling tools, visualize and manipulate virtual 3D models in immersive Virtual Reality (VR) environments is pushing the limits of teamwork activities in oil & gas industry especially in Offshore Engineering. The objective of this thesis is to establish the fundamental principles and address the main issues in the development of a Collaborative Environment for Engineering, named CEE (Collaborative Engineering Environment), in order to allow the collaborative visualization and interpretation of simulation results produced in engineering projects, which in general also involve different specialties. Due to the multi-disciplinary characteristic of those projects, collaborative visualization becomes a key component during the life cycle of engineering projects, especially those in Offshore Engineering, used in this work as case of study. We propose an integrated collaborative environment to be used by project engineers' teams during the execution and control of complex engineering projects, as is the case of the projects of deep-water floating production units. The system requirements were carefully compiled aiming to enable an effective collaboration among the participants, creating a suitable environment for discussing, validating, interpreting and documenting the results of the simulations executed during the different phases of an engineering project. To further improve the interpretation capacity and a better comprehension of results the support for immersive 3D visualization is also available in the visualization tool, especially tailored for the Offshore Engineering domain. In order to meet these goals, we devise a Service-Oriented Architecture (SOA) for CEE. This architecture is composed of the integration of different technologies of Computer Supported Collaborative Work (CSCW), Virtual Reality (VR) and Grid Computing (GC). We use a Scientific Workflow Management System (ScWfMS), based on BPEL (Business Process Execution Language), a Grid-enabled software infrastructure for executing engineering simulations, and a Video Conferencing system (VCS) to furnish audio and video collaboration. For visualizing the results, a VR visualization tool, specialized for Offshore Engineering, ENVIRON, has also been developed in conjunction with the PUC-Rio/TecGraf team.

Keywords

Computer-Supported Cooperative Work; Scientific Workflow Management Systems; Collaborative Problem Solving Environments; Collaborative Visualization; Collaborative Virtual Environments; Offshore Engineering; Oil & Gas.

Resumo

Santos, Ismael Humberto Ferreira dos; Gattass, Marcelo; Raposo, Alberto Barbosa. **Um Ambiente Colaborativo para Simulações em Engenharia Offshore baseado em Visualização e Workflow.** Rio de Janeiro, 2010. 145p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Os sistemas de produção de petróleo em águas profundas, incluindo as unidades flutuantes de produção (plataformas ou navios) e todos os equipamentos que participam da produção são atualmente projetados por complexos sistemas de modelagem computacional. Tais sistemas envolvem as áreas de cálculo estrutural, meteo-oceanografia (forças de correntes, ondas e ventos), hidrodinâmica, risers (tubos de aço rígidos ou flexíveis para levar o óleo do poço em sub-superfície até a unidade de produção), sistemas de ancoragem, equipamentos submarinos, fundações e avaliação de risco geológico-geotécnico. O projeto de uma nova unidade de produção é um processo longo e custoso, podendo durar anos e consumir centenas de milhões de dólares, dependendo da complexidade da unidade e da maturidade da tecnologia desenvolvida para tornar o projeto econômica e tecnicamente viável. Os projetos são conduzidos por diversos especialistas, por vezes geograficamente dispersos, gerando artefatos e resultados independentes, porém altamente inter-relacionados. A necessidade de colaboração é uma característica inerente aos projetos de unidades flutuantes de produção para águas profundas. A possibilidade de compartilhar informações entre usuários, controlar a execução de diferentes ferramentas de modelagem, visualizar e manipular modelos 3D virtuais em ambientes imersivos de Realidade Virtual vem empurrando os limites das atividades dos times na indústria do petróleo especialmente em Engenharia de Petróleo. O objetivo desta tese é o de fundamentar os princípios e equacionar os principais problemas para o desenvolvimento de um Ambiente Colaborativo para Engenharia, denominado CEE (Collaborative Engineering Environment), de forma a permitir a visualização colaborativa e interpretação dos resultados de simulações criadas nos projetos de engenharia, que em geral envolvem também

diferentes especialidades. Devido à característica multidisciplinar dos projetos, a visualização colaborativa torna-se um componente de fundamental importância durante o ciclo de vida de projetos de engenharia, especialmente os da área de Engenharia Offshore, utilizada neste trabalho como caso de estudo. Propomos um ambiente integrado para visualização colaborativa a ser usado pelas equipes de engenheiros projetistas durante a execução e controle de projetos de engenharia complexos como é o caso dos projetos de unidades flutuantes de produção para águas profundas. Os requisitos do sistema foram levantados com o objetivo de permitir uma colaboração efetiva entre os participantes, criando um ambiente propício para discussão, validação, interpretação e documentação dos resultados das simulações executadas durante as fases de um projeto de engenharia. Para aumentar ainda mais a capacidade de interpretação e uma melhor compreensão dos resultados o suporte a visualização em ambientes imersivos 3D também esta disponibilizado na ferramenta de visualização utilizada, que foi especialmente adaptada para a área de Engenharia Offhore.
Para atingir estes objetivos, propomos uma Arquitetura Orientada a Serviços para o CEE. Esta arquitetura é composta pela integração de diferentes tecnologias de Trabalho Colaborativo Auxiliado por Computador (CSCW), Realidade Virtual e Computação em Grade. Utiliza-se um sistema de Gerência de Workflows de Experimentos Científicos (ScWfMS), baseado em BPEL (Business Process Execution Language), para execução de simulações de engenharia em uma infra-estrutura de computação em grade subjacente e um sistema de Videoconferência (VCS) para suporte a colaboração de áudio e vídeo. Para a visualização dos resultados um sistema de visualização, especializado para Engenharia Offshore, ENVIRON, foi desenvolvido em conjunto com a equipe da PUC-Rio/TecGraf.

Palavras-chave

Trabalho Colaborativo Auxiliado por Computador; Ambientes Colaborativos para a Solução de Problemas; Sistemas Gerenciadores de Experimentos Científicos (Workflows Científicos); Visualização Colaborativa; Ambientes Virtuais Colaborativos; Engenharia Offhore; Óleo & Gas.

Summary

1	Introduction	19
1.1	Motivation: Large Scale Engineering Projects (LSEP)	19
1.1.1	The Role of Visualization, Remote Collaboration and High Performance Computing	20
1.2	Offshore Engineering Projects	23
1.3	Problem Solving Environments	26
1.3.1	Collaborative Problem Solving Environments	27
1.4	CEE - Collaborative Problem Solving Environment for Offshore Engineering	28
1.5	Thesis Main Contributions	30
1.6	Thesis Outline	30
2	Related Work	32
2.1	Workflow Management Systems and Grid Computing	32
2.1.1	Data Driven Multiphysics Simulation Framework (DDMSF)	33
2.1.2	Wind Tunnel	35
2.1.3	Vistrails	37
2.1.4	Discussion	38
2.2	Virtual Reality Visualization Technology	39
2.2.1	Immersive Well Path Planning	40
2.2.2	VRGeo Demonstrator	40
2.2.3	Geological-Mapping and Displacement Analysis (GMDA)	42
2.2.4	Discussion	43
2.3	Service-Oriented Architecture	43
2.3.1	Real-Time Architecture Project (RTAP)	44
2.3.2	Integrated Asset Management framework (IAM)	45
2.3.3	Discussion	47
2.4	CEE Main Ideas	47
3	CEE Conceptual Model	50
3.1	Collaborative Visualization Environment	52
3.1.1	Video Conferencing	55

3.1.2 VR Visualization 56
3.1.3 Collaboration Manager and Collaboration Bus 57
3.2 Scientific Workflow Environment 59
3.2.1 Scientific Workflow Service 60
3.2.2 Data Access and Engineering Simulations Execution Interoperability 62
3.2.3 Grid Computing Infrastructure (GCI) 63
3.3 CEE SOA Architecture 63
3.4 CEE Usage Scenario Overview 65

4 CEE SOA Architecture 70
4.1 CEE Architecture Layers 70
4.1.1 Technology Layer 70
4.1.2 Collaborative Engineering Layer 72
4.2 Application Layer 72
4.2.1 CEE Client Services 74
4.2.2 CEE Server Services and Components 74
4.2.3 Registration of Services 80
4.2.4 Project Workflow Service 81
4.2.5 Scientific Workflow Service 82
4.2.6 Videoconference Service 84
4.2.7 Collaborative Visualization Service 87

5 CEE Application Scenarios 92
5.1 Collaborative Risers Analysis Workflow 92
5.1.1 BPEL Scientific Workflow 92
5.1.2 Video Conferencing 95
5.1.3 3D Annotations 96
5.1.4 3D Measurements 98
5.2 Design Review Workflow 101
5.2.1 Virtual Guided Tour 104

6 Conclusions and Future Work 106
6.1 Conclusions 106
6.2 Future Work 108
6.2.1 Project Workflow Environment 108
6.2.2 Scientific Workflow Environment 108
6.2.3 Collaborative Visualization Environment 109

7 Bibliography 111

8 Appendix A 124
8.1 Business Process Management and Business Process Engineering 124
8.2 Service Oriented Architecture 125
8.2.1 Enterprise Service Bus 128
8.2.2 Web Services 129
8.3 Workflow Management System 130
8.3.1 Workflow Components 132
8.3.2 Process Definition Language 134
8.3.3 Workflow Integration with other technologies 135
8.4 Scientific Workflow Management Systems 136
8.4.1 Scientific Workflows Tools 137
8.4.2 Condor 138
8.4.3 InfoGrid 138
8.4.4 Grid Job Submission and Monitoring System 140
8.5 Virtual Environments 140

9 Appendix B 143
9.1 List of Publications 143

List of Figures

1.1	Collaboration of different simulations in an aircraft project.	20
1.2	Engineers in a collaborative section in oil & gas.	21
1.3	Geologists, Geophysicists in a collaborative section	22
1.4	Deep-water production system layout.	23
2.1	DDMSF and its components. Architecture (left) and interaction scenario (right).	34
2.2	Sequential workflow using customized wind tunnel grid activities.	36
2.3	The Vistrails Builder (left) and Vistrails Spreadsheet (right)	38
2.4	IDP Desktop interface and an IDP user interacting with the virtual world.	40
2.5	Multi-viewpoint image rendering.	41
2.6	Multiple users interacting with multiple workspaces.	42
2.7	A 3D fold surface calculated from the virtually mapped data (left). Isosurface showing aliasing in the simulation viscosity field (right).	43
2.8	Automated monitoring system (left). RTAP web services layer (right).	44
2.9	IAM graphical modeling tool (left) and its architecture (right).	45
3.1	CEE Conceptual Model.	51
3.2	Basic Service of the CEE SOA Architecture.	64
3.3	CEE Project Workflow.	66
3.4	Overview of the user interaction with CEE.	67
3.5	Creating the collaborative visualization session	68
3.6	Selecting the coordinator, the kind of session and the users that will participate.	68
3.7	Sending commands to load a simulation to visualize in the session	69
3.8	Riser simulation visualized in collaborative session.	69
4.1	CEE Architecture Layers.	71
4.2	CEE Detailed SOA Architecture.	73
4.3	Registering available services in *Service Registry*.	80
4.4	Importing the Project Workflow from a ERP system.	81
4.5	Modeling and Executing a Scientific Workflow.	83
4.6	Executing a Scientific Workflow instance in CEE.	84
4.7	Starting a VC session in CEE.	85
4.8	Environ and its RemoteCommandManager.	88

4.9 Starting a Visualization session in CEE. 90
5.1 Constructing the Riser Analysis workflow on BPEL Designer. 93
5.2 CEE SOA state of execution of a Riser Analysis Workflow. 94
5.3 Riser Analysis in CEE (Environ + CSV Tool). 95
5.4 Closer look on an element of the riser 97
5.5 Two users in a CEE collaborative visualization session. 98
5.6 Measurments in a visualization session 99
5.7 Users monitoring the behavior of marine buoyant 100
5.8 Maintenance plan enriched with annotations. 102
5.9 Measurements in a CAD. 102
5.10 Engineering information. 103
5.11 Virtual Guided Tour 105
6.1 Towards tele-imersive CPSEs. 109
7.1 Business Process Engineering for CEE 125
7.2 Service-oriented terminology. (IBM RedBooks) 126
7.3 Collaboration in SOA (IBM RedBooks) 127
7.4 ESB Conceptual model (IBM RedBooks) 128
7.5 WfMC reference model. 133
7.6 Workflow pattern Sequence in XPDL. 134
7.7 WfMC reference model. 134
7.8 InfoGrid architecture. 139

List of Tables

2.1 Feature comparisons between CEE and related solutions 49

Shortening

API – Application Program Interface
AWT – Abstract Windowing Toolkit
BPM – Business Process Management
CEE – Collaborative Engineering Environment
CFD – Computational Fluid Dynamics
CFF – Component Framework Framework
CLOS – Common Lisp Object System
CLX – Component Library for Cross Platform
CMS – Content Management System
COCA – Collaborative Objects Coordination Architecture
COM – Component Objetc Model
COPSE – Collaborative Project Support Environment
CORBA – Common Object Request Broker Architecture
CoWfMS – Collaborative Workflow Management Systems
CPSE – Collaborative Problem Solving Environment
CRIWG – International Workshop on Groupware
CSCA – Computer Supported Collaborative Argumentation
CSCW – Computer Supported Cooperative Work
CVEs – Collaborative Virtual Environments
DACIA – Dynamic Adjustment of Component InterActions
DISCIPLINE – DIstributed System for Collaborative Information Processing and LEarning
DAO – Data Access Objects
DTO – Data Transfer Object
EJB – Enterprise Java Beans
EBR – First Seminar on Advanced Research in Electronic Business
ERP – Enterprise Resource Planning
FAQ – Frequently Asked Question
FPSO – Floating production, storage, and offloading production unit
FSO – Floating, storage, and offloading production unit
FTP – File Transfer Protocol
GC – Grid Computing
GPL – GNU General Public License

GRAM – GRID Resource Allocation Management
HTML – Hyper Text Transfer Protocol
HTC – High Throughput Computing
IBIS – Issue Based Information Systems
IDE – Integrated Development Environment
IDL – Interface Definition Language
IIOP – Internet Inter-ORB Protocol
IJCIS – International Journal of Cooperative Information Systems
JAAS - Java Authentication and Authorization Service
JAMM – Java Applets Made Multiuser
JCP – Java Community Process
JMF – Java Media Framework
JSF – Java Server Faces
JSP – Java Server Pages
LSEP – Large Scale Engineering Project
MoCA – Mobile Collaboration Architecture
MOM - Message-Oriented Middleware
MVC – Model, View, Controller
OEP – Offshore Engineering Project
OLE – Object Linking and Embedding
OMG – Object Management Group
PDA – Personal Digital Assistant
PME - Project Management Environment
POJO – Plain Old Java Object
RAD – Rapid Application Development
RIA – Rich Internet Application
RPCs – Remote Procedure Calls
SAP - Business Management Software Solutions Applications and Services
ScWfMS – Scientific Workflow Management Systems
SBA – Space-Based Architecture
SOA – Service-Oriented Architecture
SDG – Single Display Groupware
SDK – Software Development Kit
SGBD – Sistema Gerenciador de Banco de Dados
SOAP – Simple Object Access Protocol
SWT – Standard Widget Toolkit
UML – Unified Modeling Language

VC – Videoconference

VCS – Videoconference System

VNC – Virtual Networking Computing

VE – Virtual Environments

VR – Virtual Reality

VRCs – Virtual Reality centers

VRGeo – Virtual Reality for Geosciences

VRML – Virtual Reality Modeling Language

WYSIWIS – What You See Is What I See

WfMC – Workflow Management Coalitio

WfMS – Workflow Management Systems

WYSIWYG – What You See Is What You Get

XML – Extensible Markup Language

XOOPS – eXtensible Object Oriented Portal System

XSL – Extensible Style Language

1
Introduction

The main objective of this thesis is to find effective solutions for collaboration of team workers during the execution of Large Scale Engineering Projects (LSEP). The research is based on actual operational needs of Petrobras, a large Brazilian governmental oil & gas company. For this thesis we have focused on Offshore Engineering Projects as our case studies.

We have developed a prototype of the proposed architecture, called CEE (Collaborative Engineering Environment), considering requirements such as collaboration, workflow coordination, and visualization. CEE allows team workers to concentrate in the task of solving a problem using all the resources available, from the execution of large engineering simulations on a Grid to the collaborative visualization of results in an immersive or desktop environment.

1.1.Motivation: Large Scale Engineering Projects (LSEP)

Contemporary Science and Engineering projects, specially the large scale ones, have the following common characteristics:

- They are highly data intensive and computational demanding.
- They are highly multidisciplinary, requiring the cooperation of different specialists.
- They often involve large distributed teams of researchers working together on a single complex problem.
- Each team of specialists has its own model of the engineering artifacts to be designed, simulated or analyzed, and may use several different models or partial models for different purposes during the project life cycle.
- Specialists have limited ability to understand each other's models. They communicate using a shared vocabulary, but not necessarily shared technical knowledge [BR92].

- They proceed by successive refinement of the models, which are coordinated and updated together. Design decisions are negotiated by specialists among themselves [BR92].

Due to their huge complexity, LSEP are divided into smaller interrelated subprojects where each one has a complementary representation of the models. Figure 1.1 schematizes an aircraft project, and many of its distinct subprojects, that might be executed by different teams of specialists. Any system to support LSEP must stress a *coupling solution* to these diverse simulations and models.

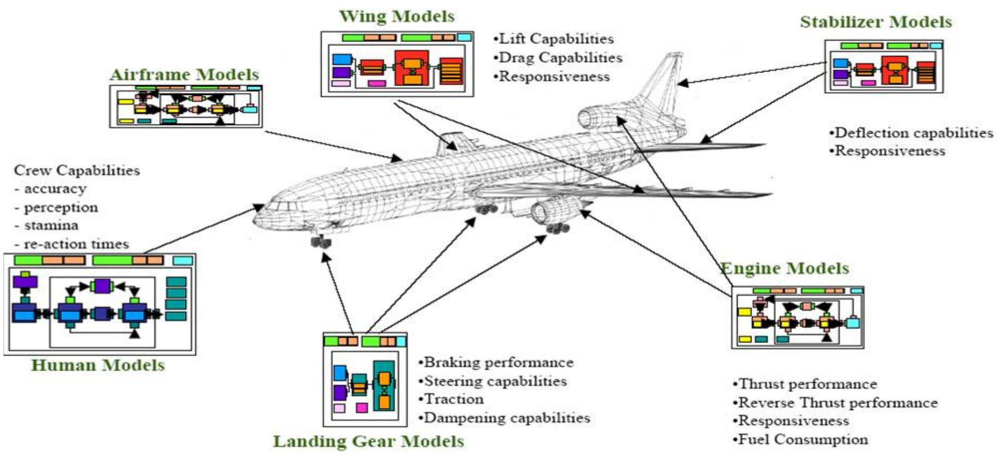


Figure 1.1: Collaboration of different simulations in an aircraft project.

LSEP also involve the interaction of people where information and data are distributed and knowledge is shared at request. Moreover, LSEP demand lengthy and complex processes involving multidisciplinary teams, usually geographically distributed with multiple information and storage systems and also using distributed and heterogeneous resources. Therefore, an integrated computer-supported solution to LSEP must include *support for human collaboration* and *distributed resource management*.

Finally, LSEP have a very dynamic nature, i.e., they cannot be completely planned in advance and are under change during their execution. For these reason, *adaptability* is also an essential issue.

1.1.1. The Role of Visualization, Remote Collaboration and High Performance Computing

The Oil & Gas industry has seen increasing costs of finding and extracting hydrocarbons, especially in remote locations, ultra-deep water reservoirs (400 m or deeper) or in hostile environments. The development of deep-water oil & gas reserves constantly faces the challenge of reducing costs of its components and activities in the selected exploitation scheme. Therefore, High Performance Computing (HPC), Visualization and Remote Collaboration technologies are being heavily used to improve productivity, leading to better cost-performance ratios.

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 scientists and engineers consists in first detecting features, then measuring them, and finally generating a model that supposedly tries 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 [Tufte83] 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 at pictures of those numbers".

Figure 1.2: Engineers in a collaborative section in oil & gas.

Visualization and Remote Collaboration technologies help us to bridge the cost-productivity problem. High-end visualization systems are commonplace in the oil & gas industry, especially in the Exploration & Production (E&P) segment, also called Upstream. In former times, the aerospace and automobile industries have shown sensitive gains in efficiency and effectiveness when carrying out Enterprise projects using Virtual Reality technologies. In the nineties oil companies were among the first to make industrial use of the so-called virtual reality centers (VRCs), equipped with immersive projection systems with large display walls (e.g., cave, cave-like, curved-panel, and powerwall), videoconference tools, among other equipments. Techniques such as three-dimensional geometric modeling, scientific visualization, immersive virtual environments (VEs), commonly used in VRCs, pushed the limits of teamwork activities in Geosciences and Petroleum Engineering, especially in Reservoir and Offshore Engineering.

The configuration of VRCs greatly improved visual communication and group collaboration in technical work sessions and decision-making meetings. The possibility of visualizing and manipulating virtual models in the VRCs has completely changed the way of working, notably for geologists and engineers (Figure 1.2 and Figure 1.3).

Collaboration is an inherent demand of Offshore Engineering projects, which require multidisciplinary teamwork, with a high degree of integration among different disciplines. The possibility to share information among several users; to control the execution of many modeling tools; and collaborative visualize and manipulate virtual 3D models in immersive Virtual Environment (VE) are interesting features that add great value to Petroleum Engineering projects.

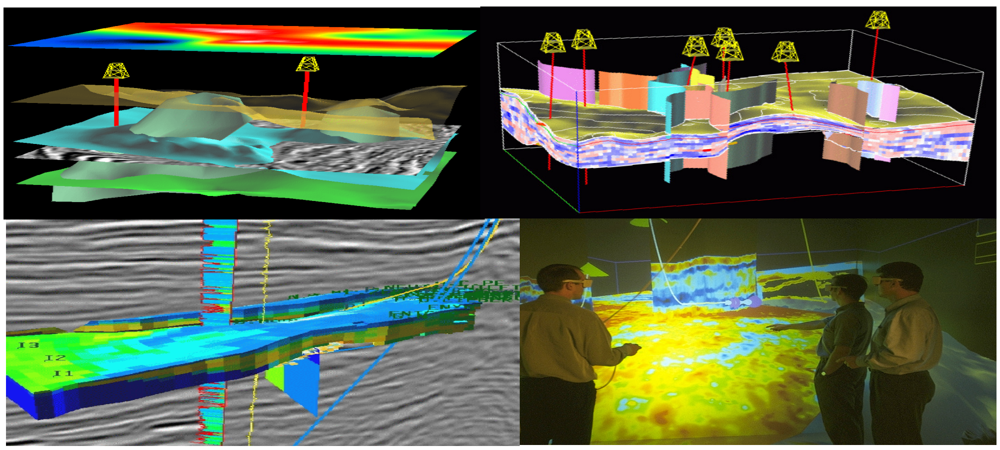


Figure 1.3: Geologists, Geophysicists in a collaborative section

To summarize we notice that VR visualization technologies enhance the content knowledge within any engineering design activity. Used in conjunction with collaboration, VR visualization provides valuable insights for better Decision Support with risk mitigation. Dodd [Dodd 04] 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 as Dodd we emphasize that the combination of Collaborative tools and VR visualization constitutes a powerful component for any software solution for Large Scale Engineering Projects.

High Performance Computing (HPC) has also become vital to oil & gas exploration and production activities due to, among other reasons, the increasingly difficult tasks of locating productive energy supplies and maximizing their extraction. This fact dictates the use of powerful compute resources to handle critical applications, like seismic processing and interpretation, reservoir simulation and offshore engineering simulations.

In the next section we discuss the main characteristics of Offshore Engineering Projects, the main targets of this thesis.

1.2. Offshore Engineering Projects

The research development in Offshore Engineering (OE) projects is conducted to design oil production units, such as platforms, or to adapt old ships to work as Floating Production Storage and Offshore Loading (FPSO) units, for operating in ultra deep water [Moan03].

Floating production systems have been utilized in remote offshore areas without a pipeline infrastructure for many years. However, they have become even more important with the push by the offshore industry into deeper waters. Floating production, storage, and offloading (FPSO/FSO) systems have now become one of the most commercially viable concept for remote or deep-water oilfield developments.

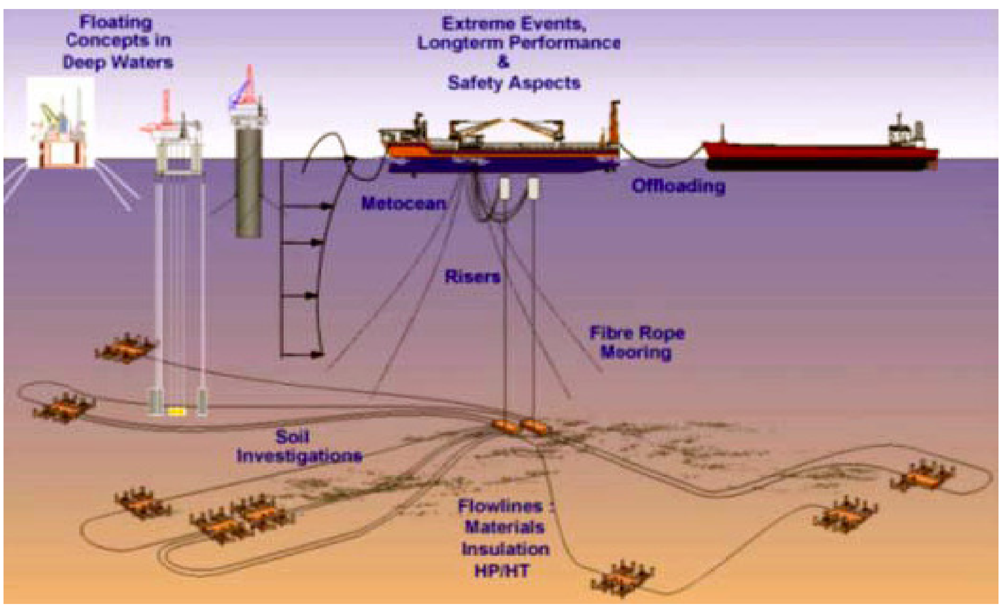


Figure 1.4: Deep-water production system layout.

Deep-water production systems (Figure 1.4), including FPSO/FSO and all the equipments playing a part in the production process, are currently designed by means of complex computational modeling systems. Those systems comprise the areas of structural analysis, meteo-oceanography (related to environmental conditions such as currents, waves and wind forces), hydrodynamics, risers (rigid or flexible steel pipes for carrying oil from the well in subsurface up to the production unit), mooring systems, subsea equipment (manifolds, “christmas-trees”, flowlines, etc), seabed foundations and Geologic/Geotechnical risk assessment.

The project of a new production unit is a lengthy and expensive process; it can last many years and consume hundreds of millions of dollars, depending on the complexity of the unit and how mature is the available technology to make the project technically and economically feasible. Depending on the maturity of the technology, further development should be devised, usually in research centers.

OE projects share all the typical characteristics of LSEP, mentioned above. They have a very dynamic nature, i.e., they cannot be completely planned in advance especially because sometimes the technology has to be developed during the project lifecycle. They involve not only geographically distributed teams but also teams of specialists in different areas using different software tools, both commercial and internally developed. The interoperability of those tools is still an issue in the industry and is a mandatory requisite for any viable collaborative solution [SVR+04].

Finally, due to their huge complexity, OE projects are divided into smaller interrelated subprojects where each one deals with an abstract representation of the others. Because decisions are interdependent collaboration is a key point in this area. Each team activity or new decision can affect other activities. During the conceptual design phase of the project, the work is carried out basically, but not only, by the following teams:

- ***Naval engineers:*** project the hull of the ship, define the optimal positioning of the array of tanks, the mooring system, and study the dynamic stability of the unit based on meteo-oceanographic information about the wind, tide and water currents;
- ***Meteo-Oceanographers:*** provide the current, wave and wind forces profile used during the stability studies;
- ***Structural engineers***: defines the internal structure of the unit and its load capacity;
- ***Production and equipment engineers:*** project the production system, encompassing risers and flow-lines, and plan the installation of deep-water production equipments, such as manifolds and “christmas trees”;
- ***Chemical and process engineers:*** project the process plant based on the characteristics and expected volume of oil and gas that will be produced;

- ***Geotechnical engineers:*** determine the position for anchoring the production unit based on studies of the behavior of the soil-structure interaction.

It can be seen by each team's duties that the need for collaboration is crucial once decisions are very interdependent. Each team activity or new decision can affect other activities. For example, during the design of an FPSO changing structural characteristics of the unit (placement of a new pressure vessel or storage tank on the platform) influences the mooring system, risers and can compromise the stability of the production unit. As a consequence, an inadequate mooring system design can possibly lead to an increase in the geologic and geotechnical risks.

Changes in environmental conditions, as the direction of wind and currents, as well as changes in the height and frequency of waves, induce movements in the mooring system, in the production risers and also in the ship, which generates second order movements that propagates to the whole system backwards. All those movements should be carefully analyzed to guarantee compatibility with the structural balance of the production unit and with the recommended operational conditions of the production risers. On the one hand, if the mooring system allows large displacements of the production unit, it can simply damage the production risers; on the other hand, the presence of the risers itself helps to weaken the movements of the production unit which positively contributes to the equilibrium of the system. This exhibits an intrinsic coupling among the solutions of the different subprojects, which requires a lot of interactions and discussions among the teams involved. On the one hand, if the mooring system allows large displacements of the production unit, it can simply damage the production risers; on the other hand, the presence of the risers itself helps to weaken the movements of the production unit which positively contributes to the equilibrium of the system. In order to achieve collaboration and interoperability between those subprojects a software-based interface is required.

Another challenge present in OE projects is related to the visualization of large-scale engineering simulations. During the conceptual design phase of an industrial plant, several simulations should be applied to confirm the robustness and feasibility of the project. Some of these simulations may require huge computational efforts to be processed, even for powerful computational grid clusters. Visualization should be as precise as possible in order to provide the user a full understanding of the results of the simulation.

1.3. Problem Solving Environments

Scientists and engineers in many application domains commonly use modeling and simulation codes developed in-house, badly documented, and often with a poor user interface. The code is usually tied to a particular computing environment, and typically only the developers of the code can make effective use of it, reducing the productivity of the team involved in a project or research activity. Another big issue is the lack of integration among those different programs, it is often necessary to convert data back and forth to pass them from one program to another in order to complete several steps of the simulation. This creates an interoperability problem, since in most of the cases, data conversion steps are needed every time a different program is to be run.

The recently proposed concept of Problem Solving Environment (PSE) promises to provide scientists and engineers with integrated environments for problem solving in their domain, increasing their productivity by allowing them to focus on the problem at hand rather than on general computational issues.

A PSE is a specialized software system that provides all the computational facilities needed to solve a target class of problems [GHR94]. 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 operating system, computer hardware or software technology [HGB+97]. PSEs allow users to define and modify problems, choose solution strategies, interact with and manage appropriate hardware and software resources, visualize and analyze results, record and coordinate extended problem solving tasks.

In principle, PSEs can solve simple or complex problems, support both rapid prototyping and detailed analysis, and can be used both in introductory education and at the frontiers of science and engineering [DB06]. In complex problem domains, a PSE may provide intelligent and expert assistance in selecting solution strategies, e.g., algorithms, software components, hardware resources, data, etc.

1.3.1. Collaborative Problem Solving Environments

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. For LSEP, a CPSE should focus on the development and integration of scientific tools and technologies, coupled with visualization capabilities and collaborative environments to support the modeling and simulation of complex scientific and engineering problems in a collaborative way. Such 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.

Experience in dealing with LSEP design and analysis problems has indicated the critical need for a CPSE with six distinguishing characteristics:

- interoperability facilities to integrate different applications;
- support for human collaboration;
- computing power for numerical simulations;
- visualization capabilities for 3D real-time rendering of massive models;
- transparency for the use of distributed resources;
- advisory support to the user.

One of the CPSE goals is to provide an environment in which visualization and computation are combined. The designer is encouraged to think in terms of the overall task of solving a problem, not simply using the visualization to view the results of the computation [BBB+93].

A combination of CPSE and VR visualization constitutes strategic enablers for a successful data exploration and knowledge dissemination among workers in engineering enterprises. The effective integration of “smart” graphical user interfaces, with some kind of Advisory Support, Scientific Visualization, Virtual Reality techniques, Engineering Analysis and Modeling Tools aid in the automation of modeling analysis and data management for Large Scale Engineering projects. To enhance engineers’ ability to share information and resources with colleagues at remote locations, collaborative and real-time technologies integrated into CPSE provide a unified approach to the scientific and engineering discovery and analysis process.

1.4. CEE - Collaborative Problem Solving Environment for Offshore Engineering

According to the above challenges presented for OE projects and based on our previous works on the related area [SRG06, SRG08 and SVR+04] we selected different technologies to compose the CEE, a Collaborative Engineering Environment specialized for OE projects.

The CEE was conceived as a CPSE especially tailored for assisting the control and execution of shared engineering projects involving geographically distributed teams. It should also allow 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.

In order to achieve its goals the CEE architecture is a composition of different Computer Supported Collaborative Work (CSCW) technologies to create a useful Collaborative Visualization Environment based on a Virtual Reality Visualization tool, and a Videoconference System; a Scientific Workflow Environment with Grid Computing infrastructure support for executing large engineering simulations; and a Project Management Environment responsible for controlling the overall execution of the project and keeping track of all the information and different artifacts generated during the project entire life cycle.

This integration furnishes to the CEE a collaborative shared workspace [DB92] composed of the following components:

1. ***Collaborative Visualization Environment***
   a. ***Virtual Reality Visualization (VRV) tool*** – a high-performance 3D visualization tool, adapted for collaborative visualization of engineering simulations and massive CAD models. *Awareness* support is also an important feature to make users aware of others activities improving the efficiency of collaboration;
   b. ***Videoconference System (VCS)*** – a VCS to support human communication, providing integrated audio and video channels, chat conversations and desktop sharing, subject to defined control policies;
2. ***Scientific Workflow Environment***
   a. ***Scientific Workflow Management System (ScWfMS)*** – a process-oriented tool to control the execution of engineering

simulations; the collaboration among users takes place while assembling Engineering Workflows, with the help of a workflow graphical modeling tool. The ScWfMS is used by the engineers to orchestrate the execution of different experiments, visualize and validate results. Engineering applications can run on a single machine or on top of a grid-enabled system integrated into the CEE.

b. Grid Computing Infrastructure (GCI)

- ✓ ***Grid Job Submission (GJS) System*** – a job submission and monitoring service to execute engineering simulations;
- ✓ ***Grid Resource Management System –*** a distributed job scheduler and resource management system, used to manage compute intensive batch jobs in heterogeneous environments;

c. Data Access Service – a distributed data access service allowing the retrieval of raw data or aggregate data (time-based raw data e.g.) from different data sources.

3. ***Project Management Environment***
   a. ***Workflow Management System (WfMS) –*** a process-oriented tool to control the execution of the project during its entire lifecycle (workflow project);
   b. ***Document Management System (DMS) –*** a document system to allow the storage of all the documents and artifacts related to the project;

By means of a ScWfMS users are able to orchestrate the execution of engineering simulations as workflow tasks controlled by the workflow engine and executed in the Grid System. Within such a workflow, as its last step, the most important results, according to any specific design criteria, can be selected for visualization in a Collaborative session provided by CEE.

In this thesis and in the current implementation of the CEE we mainly addressed the first two components, giving special attention to the contributions that the Collaborative Visualization can provide to CPSEs. The Project Management Environment was not further elaborated.

1.5. Thesis Main Contributions

The main contribution of this work is to bring together approaches and technologies from different areas, such as offshore engineering, virtual reality, CSCW, and service oriented architecture, to build a collaborative problem-solving environment to help offshore engineers to tackle their LSEP problems.

More specifically, we can distinguish the following contribution for the different areas:

- From the offshore engineering viewpoint, the proposal of using a Scientific Workflow in their project life cycle allows them to have a more structured way to solve their problems.
- Also from the offshore engineering viewpoint, the idea of a CPSE conduces to the creation of engineering tools that can be used by a wider group of users.
- From the visualization viewpoint, the integration of a VRV and Remote Collaboration in the PSE facilitates the information exchange and common understanding of complex problems. Users are compelled to think the overall solution of an engineering problem, using the visualization as a first class tool.
- From the CPSE viewpoint, the approach used for CEE created a real world scenario for an innovative collaborative virtual reality visualization system. This scenario demanded the solution of several integration issues not commonly addressed in similar applications.

1.6. Thesis Outline

The sequence of chapters of the present thesis is organized as follows.

Chapter 2 presents the related works that inspired the concepts of CEE. It also serves as a motivation chapter once it allows us to make some reasoning about the components' purposes for the CEE, presented in Chapter 1. At the end we present a comparison of CEE features and the other systems discussed.

In Chapter 3 we further elaborate the CEE Conceptual Model and the sketch of CEE basic components, which roughly address the problems faced by engineers in LSEP. We also refine the CEE Conceptual model furnishing a more detailed and formal rationale for choosing each one of the involved technologies in order to obtain the CEE SOA (Service Oriented Architecture) Model. We finish the chapter presenting a high-level usage scenario of the CEE from the construction of an Engineering Workflow (possibly in a collaborative way) to the visualization of results in an immersive environment.

In Chapter 4, we present the CEE Detailed SOA architecture. We begin presenting the details of how SOA is used as gluing technology for connecting the CEE components and some of the most important services used by CEE. Then we discuss the usage and integration of Scientific Workflow Management Systems with the VR Visualization Tool, ENVIRON [RCW+06, SRS+08, RCW+09], and also the interconnection with CSVTool [PRS+03, LKR+07], our Videoconference System, used for improving the collaboration capabilities of the CEE. The support of the Grid Computing infrastructure is also discussed.

In Appendix A, which is closely related to Chapter 4, we present the principles of SOA architectures and their basic components, especially the Enterprise Service Bus, a middleware used to seamlessly interconnect applications. The main characteristics of CEE components: WfMS, ScWfMS, VCS and VR visualization system as a subset of a Collaborative Virtual Environment (CVE), and some existing Collaborative Environments is also presented in the Appendix.

In Chapter 5, we present different application scenarios addressed with CEE. As a proof of concept we developed a prototype for each of those scenarios in order to validate the CEE. The results are also discussed.

Finally we present the conclusions and discuss possible future works in Chapter 6. In the Appendix B, we include the list of papers already published related to this thesis.

2
Related Work

In this chapter we present a few major works that motivated us to conduct this research towards the conception of the CEE Conceptual model, presented formally in Chapter 3. In what follows we justify the choice of the selected technologies used in CEE answering the following questions:

Why use Workflow Management Systems (WfMS) and Grid Computing (GC) infrastructure.

Why use Virtual Reality Visualization technology.

Why use a Service Oriented Architecture (SOA) for the implementation of CEE.

2.1.
Workflow Management Systems and Grid Computing

Since the last decade, several industries have been improving their operations through the adoption of Workflow Management Systems (WfMS). Those systems allowed them to improve the management of activities and the flow of information in the organization through the restructuring of their business processes known as Business Process Management (BPM) [BPMI]. Initially the WfMS were associated to the automation of business processes, during which, documents, information and/or tasks are passed from one participant (human being or machine) to another for the accomplishment of an action, in agreement with a set of defined rules. Such systems have enabled productivity enhancements in tasks such as processing of customers purchase orders, invoice processing, authority-for-expenditure management, etc.

For the oil & gas industry, especially in production operations, there are innumerous advantages for adopting Workflow Management Systems. The automation of engineering processes not only requires fewer workers to manage the same assets but also allows knowledge to be transferred between workforce generations in the form of well documented, previously tested and standardized workflows. Thus new employees will be able to accomplish the same work with less experience and knowledge. Moreover the increasing complexity of production operations requires the management of larger data sets, more precise decision making, and creates opportunities for optimization through more sophisticated control mechanisms. Therefore, Engineering Workflows constitute an adequate tool to embrace all these challenges.

Recently, several industries have begun focusing on Scientific and Engineering Workflows (ScWfMS) that differ in many ways from Business Workflows. Scientific Workflows gained wide acceptance in the field of bioinformatics in the early 2000s [VEG07]. While Business Workflows tend to deal with discrete transactions, Scientific and Engineering Workflows tend to deal with large data quantities, multiple data sources in multiple formats, and multiple interconnected tools. New software tools and architecture can be created to standardize Engineering Workflows by bringing together data from heterogeneous systems and consolidating separate engineering capabilities within a single platform.

2.1.1. Data Driven Multiphysics Simulation Framework (DDMSF)

In Reservoir Engineering, the need to perform extensive reservoir studies for either uncertainty assessment or optimal exploitation plans brings up demands of computing power and data management in a more extended way. Klie et al. [KBG+06] proposed and integrated framework called DDMSF, Data Driven Multiphysics Simulation Framework (Figure 2.1). DDMSF is composed of a suite of high performance numerical tools and a grid-enabled middleware system for scalable and data-driven computations for multiphysics simulation. DDMSF also includes a decision-making software system used for running integrated multiphase flow applications during subsurface characterization and oil reservoir management.

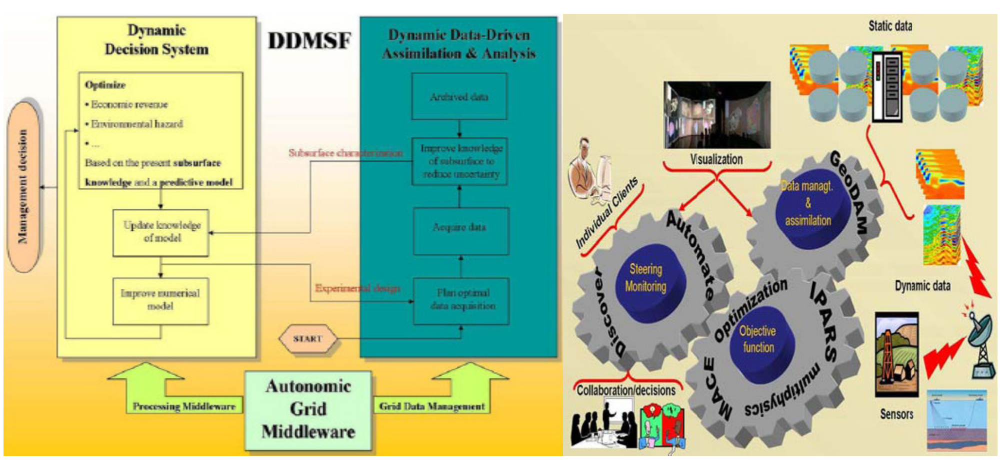


Figure 2.1: DDMSF and its components. Architecture (left) and interaction scenario (right).

The proposed suite of tools and systems consists of

- *IPARS* – a scalable and integrated multi-physics/multi-block reservoir simulator (encompassing flow, geomechanics, petrophysics and seismic);
- *Seine/MACE* (multiblock adaptive computational engine), *SPSA* (simultaneous perturbation stochastic approximation) and the *VFSA* (very fast simulated annealing), very efficient stochastic optimization algorithms (global, local and hybrid approaches) executing on distributed computing systems on the grid;
- *GeoDAM* – a geosystem data access and management software component for storing, querying, and retrieving distributed data archives of historical, experimental (e.g., data from field sensors) and simulated data;
- Discover - a decentralized grid middleware service that provide secure and coordinated access to the resources and information required by the simulations;
- External services that provide data, such as current oil market prices, relevant to the optimization of oil production or the economic profit.

The aforementioned components offer enormous potential for performing data-driven studies and efficient execution of complex, large-scale reservoir models in a collaborative environment. In Figure 2.1 the right side illustrates the interaction scenario of all these components for the optimal reservoir management carried on with DDMSF.

Dynamic data-driven approaches are increasingly becoming more feasible because of the confluence of several technologies. First, advanced sensor technologies have improved the ability to capture data faster and at higher resolution. Second, Grid Computing (GC) is making possible to realize large-scale, complex numerical models [FKN+01, FKN+02a, FKN+02b]. GC infrastructure aims to dynamically and seamlessly link powerful and remote resources to support the execution of large scale and disparate processes characterizing a particular problem. In order to harness wide-area network of resources into a distributed system, many researchers have been focused on developing grid middleware frameworks, protocols, programming and runtime environments. These efforts have led to the development of middleware tools and infrastructures such as Globus [FK99], Condor-G [FTF+01], Storage Resource Broker [RWM+02] and others.

Among all DDMSF components, the Discover Computational Collaboratory [MP03] strongly inspired the solution proposed here. Its overall objective is to realize a CPSE that enables geographically distributed scientists and engineers to collaboratively monitor, interact with, and control high performance applications in a truly pervasive manner, transforming high-performance simulations into modalities for research and instruction. Key features of Discover include a collaborative portal for interaction and control, mechanisms for web-based runtime visualization, scalable interaction and collaboration servers that reliably provide uniform access to remote distributed applications, and also security, authentication and access control mechanisms that guarantee authorized access to applications.

2.1.2. Wind Tunnel

Paventhan et al. [PTC+06] proposed the creation of 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 he 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 scientist or engineer 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.

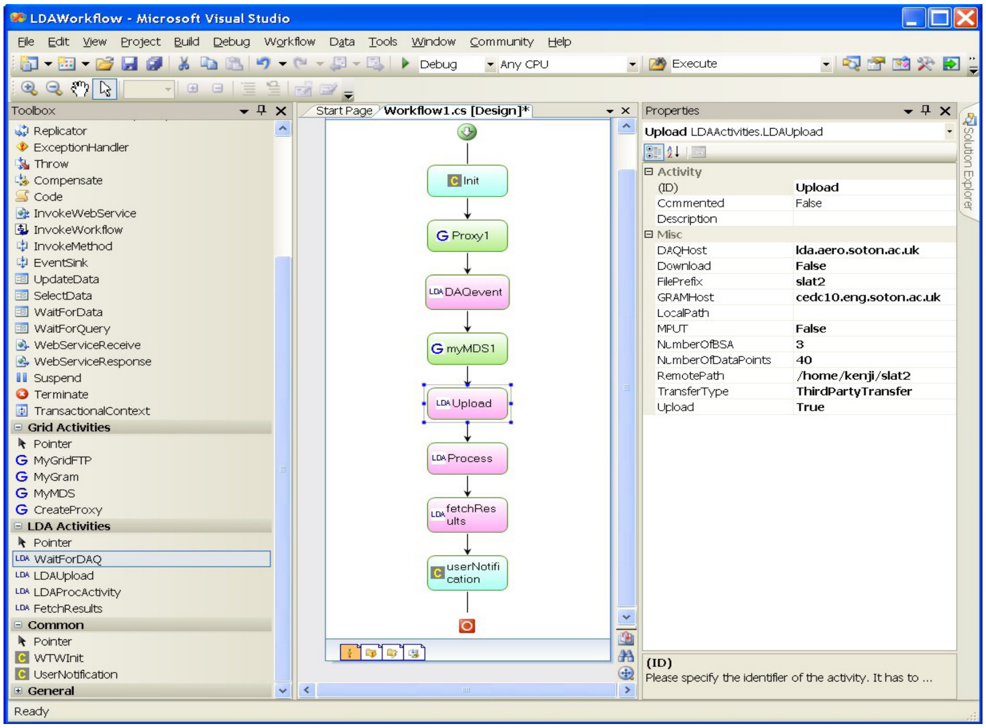


Figure 2.2: Sequential workflow using customized wind tunnel grid activities.

Two different approaches based on Windows Workflow Foundation (WWF), an extensible framework for developing workflow solutions were implemented. The WWF is a component of Microsoft WINFX [WINFX07]. It has a predefined set of activities (if-else, while, parallel, invoke WebService and so on), and allows the creation of user-defined custom activities. The first approach consisted of the extension of WINFX workflow activities to deliver a set of application-specific wind tunnel activities allowing the users to compose sequential workflows and seamlessly access Globus Grid services using a .NET-based Commodity Grid Toolkit created by the authors, MyCoG.NET [PT05]. Figure 2.2 shows a sequential workflow designed using customized wind tunnel grid workflow activities. The DAQevent is a customized event driven activity, that upon completion of the data acquisition, verifies the raw data files for completeness and in case of success enables workflow transition to next activity. The MyGridFTP, MyGRAM and MyMDS activities use MyCoG.NET Commodity Toolkit to access Globus resources. These Grid service access activities are further customized for individual experiments, as is the case of Upload and FetchResults which are activities derived from MyGridFTP, for respectively automatic uploading of raw data files from Data Acquisition host to GRAM-server (Grid Resource Allocation Management) and transfer of the results from GRAM-server to WWF server and to the user's desktop.

In the second approach, they presented a database-centric architecture for wind tunnel experimental workflow that hosts both data and processing. The strategy is to run the data parallel code on a database cluster that hosts both experimental data and user algorithms. The customized database activity set will allow the user to compose workflows based on this approach. With the rapidly evolving capabilities of Database Management Systems (DBMS) such as high-level language stored procedures (Java, C#, etc.), native support for XML, XML Web Services and Transactional Messaging are changing the role of DBMS in Scientific Workflows.

2.1.3. Vistrails

Vistrails [CFS+06] is a visualization management system developed at the University of Utah. It provides a Scientific Workflow infrastructure which can be combined with existing visualization systems and libraries. A key feature that sets Vistrails apart from other Visualization Systems as well as Scientific Workflow Systems 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.

Data provenance, i.e., the capacity of maintaining information of how a given data product was generated [SPG05], has many uses, from purely informational to enabling the representation of the data product. By maintaining a detailed data provenance infrastructure of the exploration process, in a structured way, with a flexible XML schema to represent different kinds of dataflows, the system allows the visualization experiments to be queried and mined. Users can query a set of saved dataflows to locate a suitable one for the current task; query saved dataflow instance to locate anomalies documented in annotations of previously generated visualizations; locate data products and visualizations based on the operations applied in a dataflow; cluster dataflows based on different criteria; etc. With Vistrails, users have the ability to steer their own simulations.

Data provenance is a very important feature for any CPSE because scientists and engineers often create several variations of a workflow in a trial-and-error process when solving a particular problem. 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.

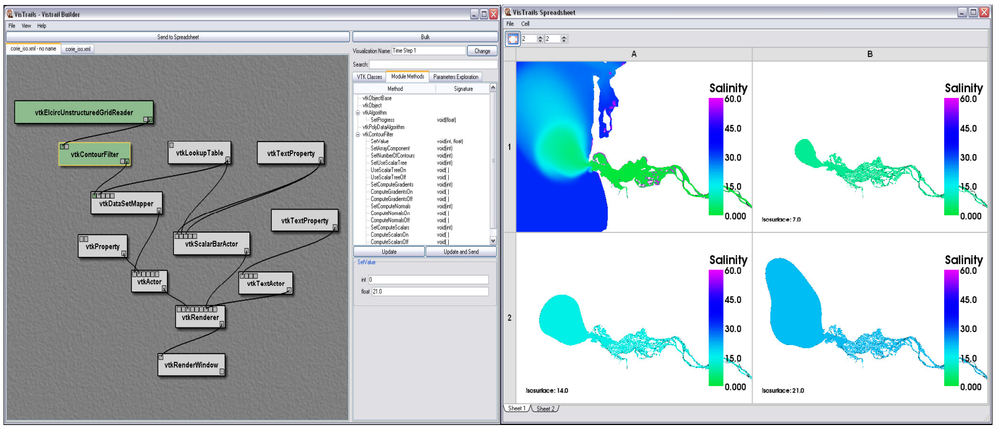


Figure 2.3: The Vistrails Builder (left) and Vistrails Spreadsheet (right)

Vistrails manage both the data and metadata associated with visualization products. Users create and edit data flows using the Vistrails Builder user interface. The dataflow specifications are saved in the Vistrails Repository. Users may also interact with saved dataflows by invoking them through the Vistrails Server, through a web-based interface, or by importing them into the Vistrails Visualization Spreadsheet. Each cell in the spreadsheet represents a view that corresponds to a dataflow instance; users can modify the parameters of a dataflow as well as synchronize parameters across cells (Figure 2.3). Dataflow execution is controlled by the Vistrails Cache Manager, which keeps track of operations that are invoked and their respective parameters. Vistrails Cache Manager infrastructure was implemented using Kepler [LAB+06, Kepler 07].

2.1.4. Discussion

Scientific Workflows and Grid Computing enable the development of complex engineering simulations. The ability to compose, design and execute rapid prototyping of experiments, provided by ScWfMS together with the grid philosophy of "on-demand" availability of computational resources are valuable features for LSEP.

The capacity of sharing resources across organizational boundaries provided by a grid computing infrastructure gives a lot of flexibility for LSEP, allowing the execution of engineering simulations “transparently everywhere”.

2.2. Virtual Reality Visualization Technology

Visualization is an important component for many PSEs. For example, Parker et al. [PMH+98] describe SCIRun [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.

Paraview [Paraview] is a kind of PSE for visualization that allows the interactive creation and manipulation of complex visualizations. Paraview is also based on the notion of dataflow, and provides visual interfaces to produce visualizations by assembling pipelines out of modules that are connected in a network. However, both SCIRun and Paraview have important limitations which hamper their ability to support the data exploration process. First, there is no separation between the definition of a dataflow and its instances. In order to execute a given dataflow with different parameters (e.g., different input files), users need to manually set these parameters through a GUI — clearly this process does not scale to more than a few visualizations. Second, modifications to parameters or to the definition of a dataflow are destructive — no change history is maintained. This places the burden on the scientist to first construct the visualization and then to remember the values and the exact dataflow configuration that led to a particular image.

Despite their limitations, SCIRun and Paraview show the importance of combining visualization with PSE. As we pointed out before in Chapter 1, the importance of three-dimensional modeling and visualization has led engineering companies to increasingly adopt the use of VRCs 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 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.

2.2.1. Immersive Well Path Planning

In the Upstream segment of the oil & gas industry, the determination of optimal well locations is a challenging problem for Reservoir engineers since it depends on geological and fluid properties as well as on economic parameters [KBW+04].

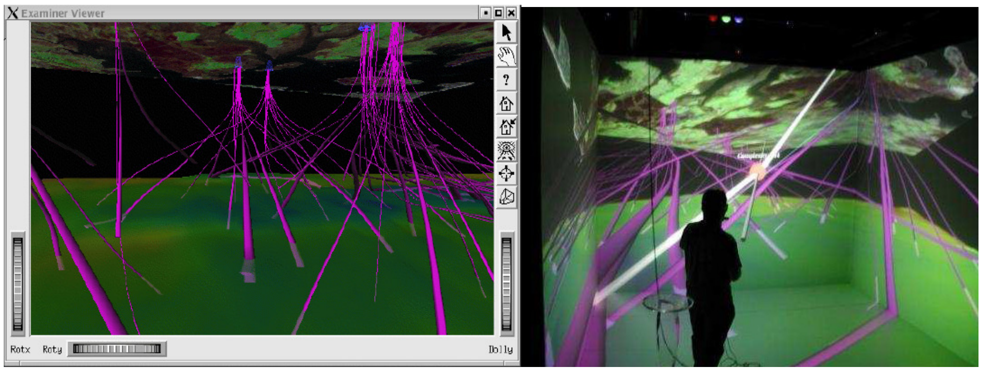


Figure 2.4 : IDP Desktop interface and an IDP user interacting with the virtual world.

Gruchalla [Gru04] investigated the benefits of immersive VR for well-path editing. He reported speed and accuracy improvements of immersive systems over desktop system, based in a study with 16 participants who planned the paths of four oil wells. Each participant planned two well paths on a desktop workstation with a stereoscopic display and two well paths in a CAVE-like [CS+92] Immersive Virtual Environment (IVE) (Figure 2.4). Fifteen of the participants completed well path editing tasks faster in the IVE than in the desktop environment. The increased speed in the IVE was complimented by a statistically significant increase in correct solutions. The results suggest that an IVE allows for faster and more accurate problem solving in a complex interactive three dimensional domain. The Immersive Drilling Planner is a long-term project to explore the impact of immersive visualization for drilling, in an effort to reduce drilling costs, risks, and time spent [DVRC].

2.2.2. VRGeo Demonstrator

The VRGeo Consortium [VRGeo] is an oil and gas international consortium for developing visualization technology for Geosciences and Engineering applications in Virtual Environments (VEs), conducted by Fraunhofer Gesellschaft (FhG, Germany)[1].

VRGeo has been presenting many significant contributions for the use of VR technology, specially in the area of Collaborative Work in Virtual Environments. Simon et al [SS+05] presented a qualitative and quantitative study comparing usability and interaction performance for multi-viewpoint images, where a large screen projection-based stereoscopic display system is shared by a small group of people, each of them with its own viewpoint (Figure 2.5).

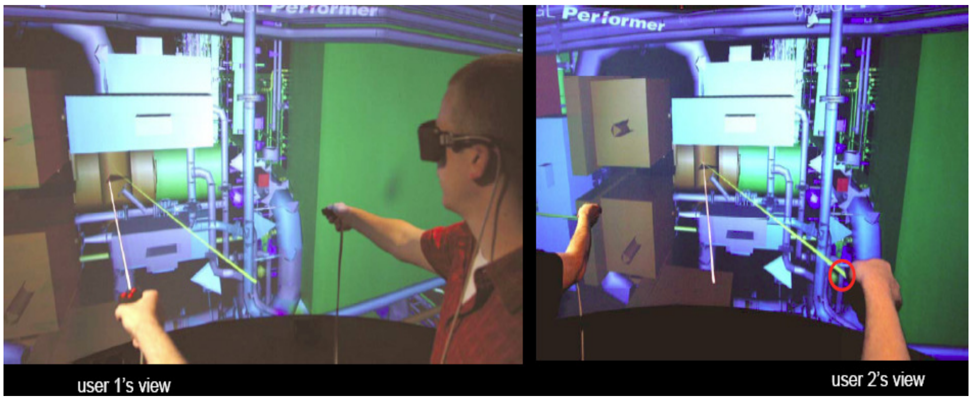
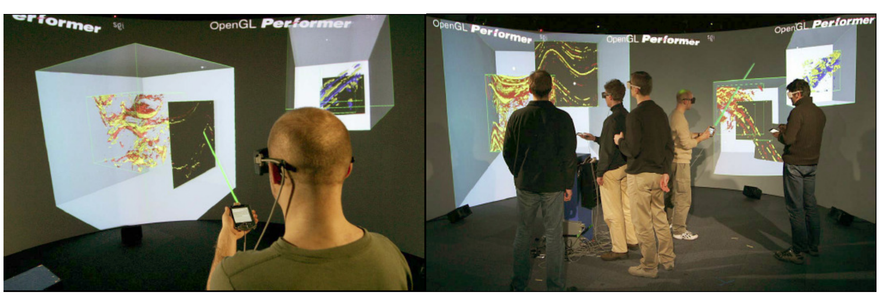


Figure 2.5 : Multi-viewpoint image rendering.

Another interesting work was the VRGeo Demonstrator Project for Co-located Collaboration interactive analysis of complex geological surfaces and volumes in an immersive VR system [Simon05]. In their paper they showed a new interaction paradigm allowing multiple users to share a virtual space in a conventional single-view stereoscopic projection-based display system, with each of the users handling the same interface and having a full first-person experience in the environment. Multi-viewpoint images allow the use of spatial interaction techniques for multiple users in a conventional projection-based display (Figure 2.6).



[1] The author of this thesis worked as a guest research scientist from 2003 to 2004 in this group.

Figure 2.6 : Multiple users interacting with multiple workspaces.

2.2.3. Geological-Mapping and Displacement Analysis (GMDA)

In the Geology field, Kreylos et al [KBB+06] presented an approach for turning immersive visualization software into scientific tool. They created immersive visualization measurement and analysis tools that allow scientists to use real word skills and methods inside Virtual Environments. They emphasized that VR visualization alone is not sufficient to enable an effective work environment. They have also conducted some informal studies to determine the impact of using VR methods on some geosciences tasks such as Geological-Mapping (identification of structures; facets, folded layers of rock and geomorphic features) and Displacement Analysis (measure the deformation of the Earth's surface and of natural or man-made structures due of geological events such as landslides, floods or earthquakes). Although not being a quantitative study, due to the small numbers of participants, they observed that VR visualization enabled scientists to make more accurate observations in less time, and to be more confident about their observations.

Another very important result, that has caught our attention, was the usage of their system as a debugging tool for Finite Element Method (FEM) simulations. Through the coupling of their VR visualization system and a FEM simulator they could solve a convergence failure in their simulation of a Plate Subduction analysis in the Aleutian chain region. For such a problem scientists use Computational Fluid Dynamics (CFD) to investigate the fact of tectonic plates entering the Earth's mantle in the vicinity of subduction zones. After having failed to find the cause of the problem using conventional tools, only by exploring their data in the VR application they could find the reason. There were several regions where one component of the simulation input exhibited severe aliasing that resulted in numerical convergence and stability problems as can be seen in the Figure 2.7-right.

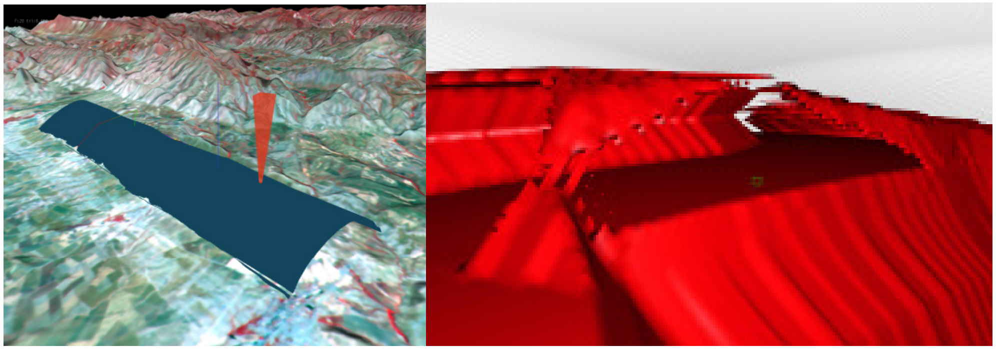


Figure 2.7 : A 3D fold surface calculated from the virtually mapped data (left). Isosurface showing aliasing in the simulation viscosity field (right).

2.2.4. Discussion

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. The ability to provide an interactive Data Exploration tool using VR visualization is a very valuable component for any CPSE constructed for LSEP.

2.3. Service-Oriented Architecture

Nowadays, businesses are dealing with two fundamental issues:

- Reduce costs and maximize the utilization of existing technology;
- The ability to change quickly.

Most enterprises today contain a range of different systems, applications and architectures of different ages and technologies. Integrating products from multiple vendors and across different platforms constitutes a real nightmare. To remain competitive, businesses must adapt quickly to internal factors such as acquisitions and restructuring, or external factors like competitive forces and customer requirements. They must have a more flexible and responsive environment, capable of dealing with the ever changing business requirements.

Service Oriented Architecture (SOA) [HKG+05, Ort05] is an alternative to alleviate the problems of heterogeneity, interoperability and changing requirements. SOA provides a platform for building application services with the following characteristics: loose coupling, location transparency and protocol independence. Based on SOA, a service consumer does not even have to care about a particular service it is communicating with, because the underlying infrastructure, or service “bus”, will make an appropriate choice on behalf of the consumer. The infrastructure hides as many technicalities as possible from a requestor. Wrapping a well defined service invocation interface around a functional module hides implementation details from other service requestors. Thus, particularly, technical specificities from different implementation technologies do not affect SOA participants. It is also possible to reconsider and substitute a service implementation for another one with an improved implementation, or with better quality of service characteristics.

2.3.1. Real-Time Architecture Project (RTAP)

The vision of intelligent or digital oilfields is roughly an interplay of several technologies that provides resources for gathering raw data (well or facilities operations) through electronic meters or gauges, transmitting this information via satellite, microwave or fiber optics to remote servers and data historians, and transforming it into knowledge for decision making (Figure 2.8).

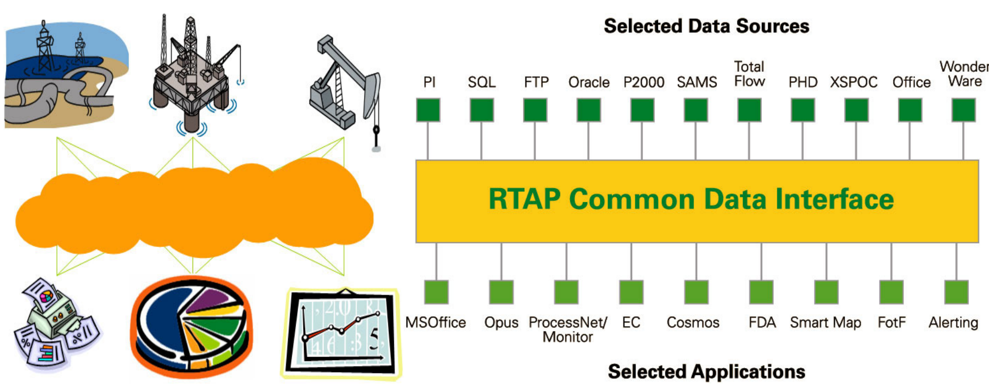


Figure 2.8 : Automated monitoring system (left). RTAP web services layer (right).

Real-Time Architecture Project (RTAP) is an initiative of the British Petroleum company (BP) to provide a common approach for all its assets to access real-time production operations data [GFF+05]. RTAP utilizes Web Service technologies, which create highly flexible interfaces based on established and emerging Internet standards. It integrates a wide range of tools such as Production Reporting, Real-time Visualization, and Active Alerting with new or existing data sources of many kinds. BP has already implemented this solution in many locations and in a number of business units, providing many common applications with access to a dozen commercial and proprietary data sources (Figure 2.8).

The ultimate goal of RTAP is to implement a common, standards-based architecture for data access and integration, replacing the large number of custom, proprietary interfaces currently in use. Since RTAP launching some years ago, significant progress has been made toward this goal, with the intent of expanding the current implementation to a next generation SOA in the near future.

2.3.2. Integrated Asset Management framework (IAM)

Another SOA application that influenced this research is the Integrated Asset Management framework (IAM). IAM provides to its users a front-end modeling environment for specifying and executing a variety of workflows from reservoir simulations to economic evaluation [SBO+06]. The IAM framework is intended to facilitate seamless interaction of diverse and independently developed applications that accomplish various sub-tasks in the overall workflow. For instance, with IAM a user can pipe the output of a reservoir simulator running on one machine to a forecasting and optimization toolkit running on another node and in turn piping its output to a third piece of software that can convert the information into a set of reports in a specified format (Figure 2.9).

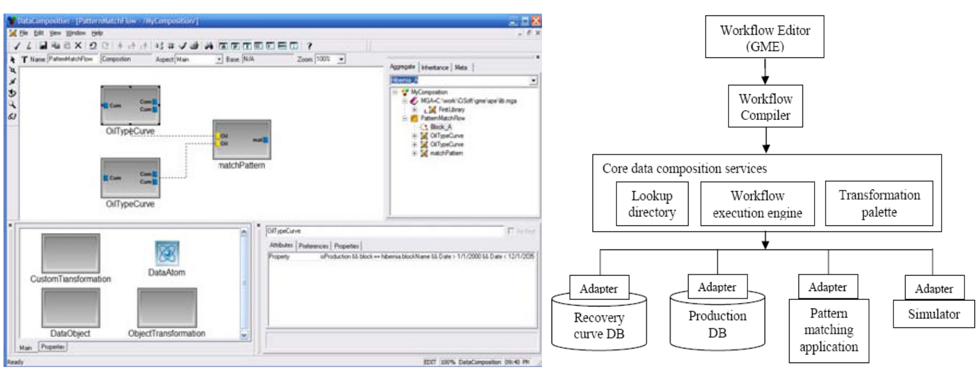


Figure 2.9: IAM graphical modeling tool (left) and its architecture (right).

IAM adopted a service-oriented approach where 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 framework. 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.). Workflows can be composed by coupling service interfaces in the desired order, through a graphical modeling or textual front-end and the actual service calls can be generated automatically.

Data composition is one of the key components of the IAM framework. It refers to a general process of applying a variety of intermediate transformations to data as it flows from one service to another as part of a larger workflow. Automating data flow among multiple information consumers will greatly expedite many workflows by eliminating the typically laborious tasks involved in manual preparation of data for input to various tools. In order to enable a Data composition perspective they created the following components and services:

- ***Data sources.*** The production data and the recovery curve catalog are the sources of 'raw' data that could be stored in a standard database. Access to the database could be through a web service that provides a query interface for data retrieval and update;
- ***Aggregation service.*** A software module aggregates time-based raw data (from production as well as simulation), and generates type curves along the desired dimensions - e.g., cumulative oil production vs. reservoir pressure;
- ***Pattern matching service.*** This software module accepts a set of reference curves from the catalog and a type curve derived from the production data, and performs pattern matching to estimate the best fit.

Figure 2.9-left illustrates the use of their graphical modeling tool for building a highly simplified real-time reservoir management workflow. In this workflow, a catalog of type curves is available from a series of a priori reservoir simulation runs. The curves in the catalog correspond to a set of differing models of the reservoir. As real world production data from the reservoir becomes available, it is periodically compared to the type curves in the catalog to estimate the best fit. The type curve(s) that best matches the production data at a given time could then be used as input to other disjoint workflows such as oil production forecasting.

2.3.3. Discussion

We argue that SOA offer to Large Scale Engineering Projects a number of compelling benefits for allowing the development of a flexible and stable architecture. Through the use of its three main concepts - loose coupling, location transparency and protocol independence – a Problem Solving Environment developed for a LSEP using an SOA will be able to:

- respond efficiently to changes in the business and competitive landscape,
- reuse of legacy system while enhancing integration;
- reduce overall technology development costs by:
  - leveraging functions already built into legacy system services;
  - reusing services developed for other process;
  - simplifying maintenance and support through elimination of redundant and siloed applications.

2.4. CEE Main Ideas

The work of the Data Driven Multiphysics Simulation Framework (DDMSF) and the former OE characteristics discussed in chapter 1, pointed out to the necessity of integrating a myriad of different applications to solve common OE problems (Figure 2.1). This motivated us to the pursuit of an Enterprise Application Integration (EAI) for CEE. Recently, EAI has been greatly simplified by the adoption of an SOA integrated with an Enterprise Service Bus [HKG+05]. In Chapter 4 we provide more detailed information about the usage of ESB in the CEE SOA architecture.

Wind Tunnel provides a series of workflow activities allowing the users to compose sequential workflows and seamlessly access Grid services (Figure 2.2). The Wind Tunnel approach also inspired the development of our CEE by combining the ScWfMS with the execution of engineering applications in a Grid infrastructure computing environment through the use of Grid Resource Allocation & Management (GRAM) job submission.

The 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 the absence of data provenance support.

The VRGeo Demonstrator's collaboration capabilities showed the benefits of collaboration in a Virtual Environment for interpreting geological data (Figure 2.6) or investigating platform 3D CAD models (Figure 2.5). This is a very important feature for our CEE which has the Offshore Engineering field as its main target (see Chapter 1).

As shown by GMDA, the usage of a VR Visualization system to debug engineering simulations is a very powerful tool for Large Scale Engineering Projects. Their observation that VR visualization enabled scientists to make more accurate observations in less time and with more confidence has also motivated to include a VR Visualization system as an important component of the CEE architecture. The fact that VR visualization alone is not sufficient to enable an effective work environment has stimulated us to create additional tools for the VR Visualization component of CEE (CEE-VRV). Some of those tools are Annotations and Measurements and are discussed further in the Collaborative Tools section in Chapter 3.

The IAM project has inspired very much the CEE architecture. The adoption of an SOA with services encapsulated as components motivated us to use a Service Component Architecture [SCA] in the development of the CEE.

To finalize this chapter we present a comparison of the features provided by CEE and the features presented by the related solutions. It can be seen from this comparison that CEE has a wider spectra addressing the most important requirements of Large Scale Engineering Projects.

	CEE	DDMSF	Wind Tunnel	Vistrails	GMDA	IAM
ScWfMS						
Scripting Language	✓	✗	✓	✓	–	✓
Visual Tool for Composition	✓	✗	✓	✓	–	✓
Data Provenance	✗	✗	✗	✓	–	✓
Grid Computing Infrastructure						
Job Submission	✓	✓	✓	✓	–	✓
Job Monitoring	✓	✓	✓	✓	–	✓
Collaboration						
Collaborative Portal	✓	✓	✗	✗	–	✗
Videoconference	✓	✗	✗	✗	–	✗
Visualization						
Virtual Reality	✓	✗	–	–	✓	–
Collaborative Visualization	✓	✗	✗	✗	–	–
Visualization Tools (3D Annotations, Measurements, Virtual Tours)	✓	✗	–	–	✓	–
Scientific Visualization	✓	✓	–	✓	✓	–
Computational Steering	✗	✗	–	✓	✗	–
Data Access Service						
Data Stage In/Out	✗	✓	✓	✓	✗	✓
Querying and Retrieval Mechanisms	✗	✓	✓	✗	✗	✗

✗	Does not provide
✓	Provide
–	Not Applicable

Table 2.1 Feature comparisons between CEE and related solutions

3
CEE Conceptual Model

The developed CEE, as a specialized CPSE, allows users to collaboratively solve their problems through the use of predefined scientific 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 VR Visualization component.

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

- ***Collaborative Visualization Environment*** – this domain encompasses very different challenges from the areas of CSCW and Virtual Reality.
  - ***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 need of some support for coordinating the execution of tasks.
  - ***Virtual Reality Visualization*** – high performance and scalability are important aspects of virtual environment architectures intended to support execution of large shared virtual worlds over long periods of time.
- ***Scientific Workflow Environment*** – this domain includes challenges related to the control of the execution of engineering simulations
  - ***Interoperability and Distributed Execution*** - in this domain there is a myriad of software that specialists, potentially geographically distributed and using distributed resources, are forced to use in order to accomplish their tasks in a reasonable time. This requires the solution to

have the ability to be easily and seamlessly distributed and demands a high level interoperability among its components.

- ○ ***Data provenance -*** Data provenance is the capacity of maintaining information of how a given data product was generated [SPG05] 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.

- ➢ ***Project Management Environment -*** this domain points to the necessity of keeping track of all the documents and artifacts generated during project's life-cycle. Multiple and different visions of the on-going project must be provided while users have different background (e.g. managers, engineers) and need different types of information to accomplish their duties.

The conceptual model of CEE (Figure 3.1) handles some of the challenges, creating specific services for them. In the conceptual model we depict the services related to each CEE specific environment.

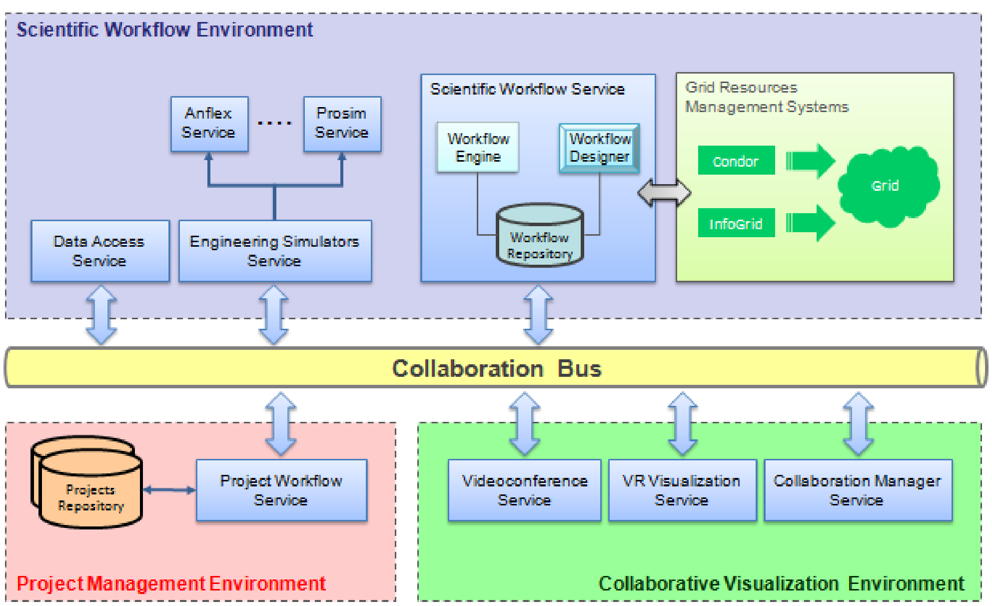


Figure 3.1 : CEE Conceptual Model.

For the ***Collaborative Visualization Environment***, we created the *Collaboration Manager Service* who is responsible for managing the user interaction with the CEE. The *Videoconference Service* and the *VR Visualization Service* work closely coupled with the *Collaboration Manager Service* to enable the creation of collaborative visualization sessions inside the CEE.

For the ***Scientific Workflow Environment****,* we created the *Scientific Workflow Service* to help the users build engineering workflows and seamlessly execute them in a Grid Computing Infrastructure (GCI). More generally for *Distributed Execution,* we use the interoperability characteristics of the ScWfMS and the distributed execution support provided both by the GCI of the CEE and by the SOA backbone infrastructure furnished by the Enterprise Service Bus (ESB). For *Interoperability* among applications it was developed a common format for data exchange among engineering applications in the Offshore Engineering field (see Section 3.2.2).

The *Engineering Simulations Service* provides a Webservices interface [LMN+04] for remotely execute an engineering simulation program. In the Offshore Engineering, some of those simulators are, among others, Anflex [MGJ95] a Finite Element riser analysis software, and Prosim [JE94] a coupled analysis software for the design of floating production systems.

For the ***Project Management Environment***, the introduction of a *Project Management System* is a valuable resource. Although very important for a real CEE implementation, the use of a Project Management System is out of the scope of this thesis.

Based on a thoroughly analysis of the domain of OE projects used as our target scenario, we present more detailed information about the adopted visualization and collaborative features to cope with the difficulties described above. In what follows it is presented the major CEE functionalities towards this direction.

3.1. Collaborative Visualization Environment

Collaborative systems should not only allow multiple users to interact with shared objects but also to communicate and to coordinate their actions. Collaboration may be seen as the combination of communication, coordination and cooperation [FRG+05]. Communication is related to the exchange of messages and information among people. Coordination is related to the management of people, their activities interdependencies and the used resources. Cooperation is the production of common artifacts taking place on a shared space through the operations available to the group. This model, called 3C model, was originally proposed by Ellis et al. [EGR91] and identifies the three essential high-level components of conceptual collaboration models.

In CEE, those three aspects of the conceptual collaboration model are enacted by the use of a Scientific Workflow Management System (ScWfMS), Videoconference System (VCS) and a Collaboration-Bus, a collaborative infrastructure support for integrating the execution of engineering applications with our VRV component allowing the users to collaboratively visualize their results and optionally create virtual Annotations, to share knowledge about the engineering artifacts being visualized.

The ability to collaboratively create persistent annotation on the model improves its usefulness. An Annotation, in our context, is any textual information that users want to add to their projects to enrich the content or just for documentation purposes. It can have a private or public (shared) scope. Annotations can be associated to any engineering artifact manipulated during a collaborative visualization session. An Annotation could, for example, represent a kind of instructional information denoting a sequence of operations that should be undertaken during an equipment maintenance intervention in a production unit. It can also be any textual information used to highlight interesting or anomalous events observed on the simulation results. Examples of such events could be unexpected values for an engineering quantity, violations of integrity, etc. Annotations can also have a more dynamic behavior, while they can represent distance measures between distinct objects that should be monitored during a simulation, e.g. the distance between two different elements in two different ascending risers to a floating production system. Some of the discussed examples can be seen in Chapter 5.

Another important aspect in the collaborative visualization session is the possibility of having Virtual Guided tours, where the coordinator of a session can create virtual paths through critical results in the simulations in order to demonstrate or discuss possible anomalies in the results of the simulation with other users. When following these paths the camera movements of the coordinator are retransmitted to other participants in the session allowing them to share the same view. This is also a very valuable tool for improving the knowledge of the simulation.

The collaboration features of CEE can be summarized according to each aspect of the 3C Model [FRG+05] as follows:

- ***Communication support*** **–** is a fundamental feature in our scenario. The VCS offers different communication support to CEE, synchronous or asynchronous, enabled in various media types (audio, video and text based communication). It is seamlessly integrated into CEE so that users can start a videoconference communication at anytime, while modeling their workflows or during visualization of results. They also should be able to plan a certain time lag for a specific communication interaction, especially when long-lived processes are enacted in the execution of the workflow. The communication support should be integrated to the other tools in the CEE and provide means of recording conversation and retrieving old ones. This feature helps the user to solve their project's problems in critical situations, with fast interaction and negotiation, and it allows the recovery of useful pieces of communication used to solve similar problems in the past.
- ***Cooperation and flexibility support*** – there should exist flexible process modeling support, like dynamic change of process instances during run-time to support dynamically evolving processes, possibility of executing rollback of processes (reset, redo, undo, recover, ignore, etc), reuse of process fragments and component libraries. The cooperation support must provide different levels of data access: local and distributed, shared, public and private access, versioning control of engineering models and related data, concurrency control and synchronization. It is also necessary to provide support for different types of data interchange, concurrent work on shared copies, change propagation, and physically shared data access. Different types of model visualization should also be available at the CEE, as well as some data management infrastructure related to these models, like real-time simulation and visualization of 3D models, possibilities of walkthroughs in the models, object interaction and manipulation, edition and planning and also access to organizational work history.

Cooperation also occurs during the assembling of useful engineering workflows that will be used to orchestrate the execution of engineering applications, and also during visualization of results when the users can collaborate to better understand the model. Users can also share persistent annotations about interesting facts, as previously discussed.

- ***Coordination and Awareness –*** there are different types of awareness support provided in CEE. In our scenario, the most important ones are:
    - ***event monitoring –*** observes what is going on in the VRV, in all separate parts and provide active notification to the right person, at the right time and the right sub-system;
    - ***workspace awareness in the virtual environment*** – provides control of collaborative interaction and changing of the user location;
    - ***mutual awareness*** – allows users see each other's identity and observe each other's actions;
    - ***group awareness*** – facilitates the perception of groups of interest connecting people who need to collaborate more intensely. Group awareness enables the user to build his own work context and to coordinate his activities with those of others'. Informal communication enhances team awareness, even with no support to cooperation and with restricted coordination functionalities for controlling the simultaneous use of communication channels [Mack99]. User awareness is a very important subject for CEE once it is a desirable feature to provide mutual awareness during the collaborative visualization session allowing users see each other's identity and observe each other's actions specially when creating Annotations in the model.

3.1.1. Video Conferencing

Audio and video communications are fundamental components of collaborative systems [IT94]. Audio is an essential channel for supporting synchronous work, and video is important to provide a sense of co-presence facilitating the negotiation of tasks.

Multimedia Collaborative Systems such as VCS, 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. The drawback is that all coordination tasks are left to the conference participants [RSV+94].

The development of a custom videoconferencing system, CSVTool [PRS+03, LKR+07], allowed us to automatically establish videoconferencing channels among the participants of a conference which greatly simplify and improve the communication. We can also tightly control the multiple audio and video streams among participants implementing different scenarios of usage, described in section 3.1.3.

Besides the transmission of audio and video to multi-participants, with different operating systems platform, CSVTool provides extra interesting features for CEE:

- 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;
- a textual chat tool, which is providential in some situations (for instance, when somebody is having problems with capture devices);
- snapshots, useful for documenting the work session.

In CEE, a videoconference session 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 Collaborative Session that is registered in the CEE Server.

3.1.2. VR Visualization

Modern floating production units' construction projects are carried out with the creation of a so-called Virtual Prototype. The aim of virtual prototyping is to build a full virtual model in such a way that design and manufacturing problems are anticipated and discussed within a cooperative and distributed work environment [BFD+03].

The applications available for 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

[RSV+94, PMH+98]. 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 or a GUI event multiplexing system. In both cases the applications do not explicitly support collaboration; they are implemented as single user applications [Tietze01]. This is important since, in our case, the applications developed for OE projects at Petrobras fits in this type.

Collaboration-aware applications, on the other hand, are specially 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 (ENvironment for VIRtual Objects Navigation) [RCW+06, RSS+09] is a tool designed to allow visualization of massive CAD models and engineering simulations in immersive environments (VR and Desktop). It is a system composed of a 3D environment for real-time visualization and plug-ins to import models from other applications, allowing users to view and interact with different types of 3D data, such as refineries, oil platforms, risers, pipelines and terrain data.

In order to serve as the CEE's VRV, Environ was adapted to be transformed into a collaboration-aware application with the support provided by the CEE collaborative infrastructure.

3.1.3. Collaboration Manager and Collaboration Bus

The Collaboration Manager is responsible for managing the users' participation in a collaborative session and also integrates the resources of VRV and VCS. There are three kinds of sessions available:

- Informal – where each participant uses its individual telepointers all the time. There is no mediation of camera movements and the users are free to move around the scene propagating the camera movements to others. In this model, once a collaborative session is created, audio and video can be used at any time by all users. The only mediation mechanism supported is furnished by the social protocol available whenever a videoconference is started.

- Classroom – where one specific participant, the instructor, acts as a coordinator of the session which means that all camera movements he performs are followed by other users, while the other participants have their telepointers disabled. The instructor also controls the audio and video channels of the participants, and he is also allowed to pass control of collaboration resources (telepointers, camera control, etc.) among participants, taking it back at any time.

Users can request the coordination role to the current coordinator who can accept or reject the request, generating a visual feedback to the requestor. Upon the occurrence of a *"change coordinator"* event in a CEE collaborative session, all users are notified by the CEE Awareness mechanism.

- Lecture – where one specific participant, the speaker, acts as a coordinator of the session, with the same characteristics as in the Classroom session. In this type of session there is no exchange control between the coordinator and participants and the participants can only receive audio and video stream from the coordinator. This model of session is used for example when the speaker wants to create a virtual guided tour showing important details of the results of a simulation to other participants.

At any time the user can disconnect from the session, for doing some private work, and reconnect to session in later time, when its state is synchronized with the sate of the session, that is controlled by the *Collaboration Manager Service*.

The *Collaboration Bus (CBUS)* is a key component of the overall architecture and provides synchronous and asynchronous communication for the CEE components. The CBUS is an infrastructure for communication based on the JMS Service Provider, the Message Oriented Middleware (MOM) used for giving the public/subscribe and point-to-point paradigms, and the Enterprise Service Bus (ESB). The CEE *Awareness Service* is an awareness service providing appropriate actuators for events received from the CBUS. It is responsible for signaling distributed events to the users participating in a collaborative session. In one side all components trigger events to this distributed bus, and in the other side awareness components listen to the bus for information about what is happening in the system. For example, when users leave a collaborative session or when there is a change in its state from offline to online and vice-versa, *"update user"* events are triggered to the CBUS and the CEE Awareness mechanism send messages to *VRV Visualization Service* and *VC Service* notifying the event. By their turn, those services signal those events in their user interfaces making the user conscious of what have happened. Examples of such kind o signaling could be: using different icons in the GUI for showing the state of the users; windows messages warning the entrance of a new user in the session or the leaving of the user from the session, etc.

The integration of the VRV and the VCS with the other components is done in a seamless way through the Collaboration Bus, in a way that the user always interacts with the same interface independent of the application he/she is currently using. This is a very important aspect of the solution to keep the user conscious of what he/she is doing and what should be the next steps of the current task being executed.

3.2.
Scientific Workflow Environment

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

All the consistency, adequacy and compatibility of the shared data among its users should be done by the kernel of the CEE, in order to avoid, or at least to diminish, non useful iterations during the project's life cycle. The ability of reusing partial workflows, which were previously stored in the system with some guidelines, provides an optimized usage of the available computational resources and also a better control of the costs and time scheduling.

Similarly to the VRV and VCS, the integration of the ScWfMS with the other components is also done through the Collaboration Bus.

3.2.1. Scientific Workflow Service

According to Ellis [Ellis99] 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.

Although the above definitions make reference to “Business Process”, WfMS is not only employed by business applications. In recent years, several industries have improved their operations through WfMS, improvement of data management and better coordination of activities through specific Business and Scientific and Engineering Process. However, there are remarkable differences between Business (BWfMS) and Scientific Workflows (ScWfMS). In [MVW95] the authors identified that in a scientific environment scientists will typically specify their workflows themselves, while in a business environment, a system administrator is commonly responsible for this task. Another characteristic of ScWfMS mentioned in their work 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 (ScWfMS) describe series of structured activities and computations that arise in scientific problem-solving. In many science and engineering areas, the use of computation is not only heavily demanding, but also complex and structured with intricate dependencies. Graph-based notations, e.g., Generalized Activity Networks (GAN), are a natural way of representing numerical and human processing. These structured activities are often termed studies or experiments. However, they bear the following similarities to what the databases research community calls workflows:

- Scientific problem-solving usually involves the invocation of a number and variety of analysis tools. However, these are typically invoked in a routine manner. For example, the computations involve much detail (e.g., sequences of format translations that ensure that the tools can process each other's outputs), and often routine verification and validation of the data and the outputs. As data sets are consumed and generated by the pre and post processors and simulation programs, the intermediate results are checked for consistency and validated to ensure that the computation as a whole remains on track.

- Semantic mismatches among the databases and the analysis tools must be handled. Some of the tools are designed for performing simulations under different circumstances or assumptions, which must be accommodated to prevent spurious results. Heterogeneous databases are extensively accessed; they also provide repositories for intermediate results. When the computation runs into trouble, semantic roll forward must be attempted; just as for business workflows, rollback is often not an option.
- Many large-scale scientific computations of interest are long-term, easily lasting weeks if not months. They can also involve much human intervention. This is especially so during the early stages of process (workflow) design. However, as they are debugged, the exceptions that arise are handled automatically. Thus, in the end, the production runs frequently require no more than semiskilled human support. The roles of the participating humans involved must be explicitly represented to enable effective intervention by the right person.
- The computing environments are heterogeneous. They include supercomputers as well as networks of workstations (clusters). This puts additional stress on the run-time support and management. Also, users typically want some kind of a predictability of the time it would take for a given computation to complete. Making estimates of this kind is extremely complex and requires performance modeling of both computational units and interconnecting networks.

Consequently, it is appropriate to view these coarse-granularity, long-lived, complexes, heterogeneous, scientific computations as workflows. By describing these activities as workflows, we bring to bear on them the advanced techniques that have been developed in workflows research. These include sophisticated notions of workflow specification and toolkits besides environments for describing and managing 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 long-lived. 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.

Business Process Execution Language (BPEL) was chosen for the ScWfMS in the CEE implementation. BPEL is further explained in the Appendix A.

3.2.2. Data Access and Engineering Simulations Execution Interoperability

Engineering simulations are computer and data intensive. In a typical scenario, data is usually passed from one program to another in order to complete several steps of the simulation. In the CEE, the sequence of operations to perform an engineering simulation are modeled as scientific workflows [DGS+09]. This creates an interoperability problem, since in most of the cases, data conversion steps are needed every time a different program needs to be run over the data. To solve the data interoperability problem, allowing applications to share engineering data in the context of such scientific workflows, a unified data format have been defined and developed. This format is called GXML, Galileo

XML [SBM+09]. As the name says, it is based on XML, which can be easily handled by applications using standard XML APIs.

For scalability purposes GXML classifies data as light and heavy, according to the amount of information it represents. In this sense, light data is allowed to be stored in the GXML file's body, while heavy data is stored in HDF5 (Hierarchical Data Format) [HDF] in an internally compressed format and described in the GXML file. HDF5 provides efficient ways for reading and writing huge volume of data which is very important for engineering and scientific data.

3.2.3. Grid Computing Infrastructure (GCI)

Compared to other models of computing, systems designed and implemented in the grid style deliver a higher quality of service, at a lower cost, with greater flexibility. Higher quality of service results from having no single point of failure, a powerful security infrastructure, and centralized, policy-driven management. Lower costs derive from increasing the utilization of resources and dramatically reducing management and maintenance costs [GridOracle05].

SOA has emerged as a superior model for building applications, and SOA concepts align exactly with the core tenets of grid computing.

3.3. CEE SOA Architecture

A Service Oriented Architecture (SOA) is a very attractive architecture for allowing independence between service providers and consumers. ESB represents the next generation of integration middleware, which establishes an enterprise-class messaging bus that combines a messaging infrastructure with message transformation and content-based routing in a layer of integration between service consumers and providers. The use of an ESB in the CEE architecture allows a seamlessly integration of distributed applications modeled as SOA services. For each external engineering application that might be invoked by the Scientific Workflow during the execution of a user job, we built a service interface (*Engineering Simulation Service*) that allows the application to be called from inside the workflow or from any other application connected to the ESB.

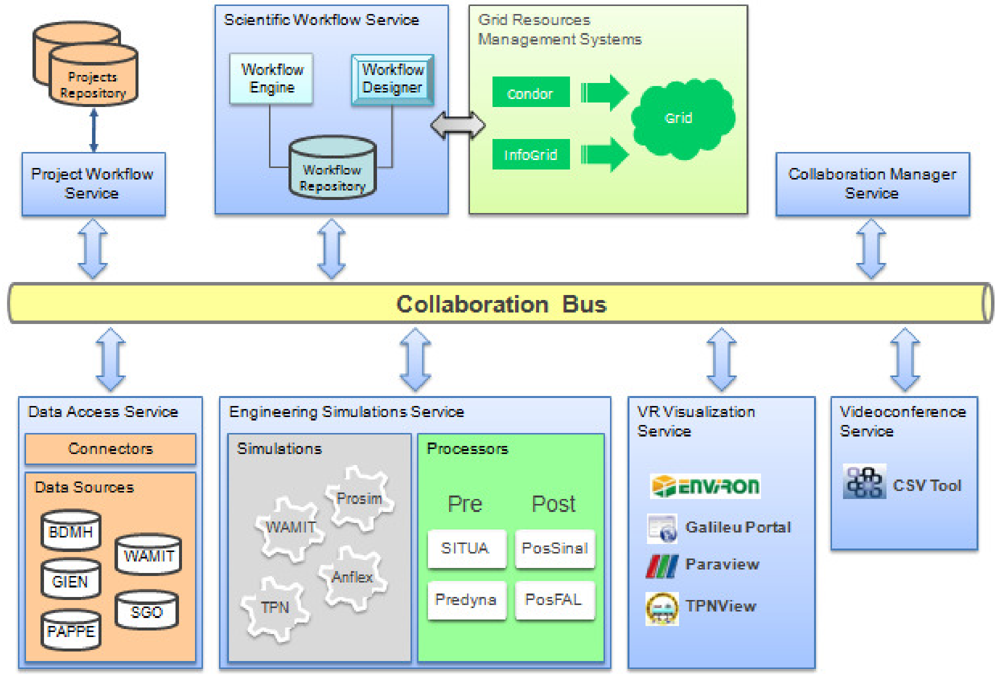


Figure 3.2 : Basic Service of the CEE SOA Architecture.

The 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 (VCS), Environ (VR Visualization tool) 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.).

Figure 3.2 illustrates the CEE SOA architecture, which is an instantiation of the CEE conceptual model for the OE field. In this figure, engineering simulators, data sources, VRV tools and VCS are specialized for the target scenario chosen for this thesis.

3.4. CEE Usage Scenario Overview

This section describes how the user interacts with the system. The user accesses the system through the CEE Portal and is able to create engineering workflows, execute them and visualize the results in a visualization session.

The diagram presented in Figure 3.3 shows the macro-processes, managed by CEE, that are interconnected with the Enterprise Service Bus in which messages are transmitted using GXML [SBM+09].

The first step in the simulator is the creation of the *Project Workflow,* whose main activities will be imported from a projects database, such as SAP [SAP]. As shown in detail, certain steps in the Project Workflow will consist of the execution of Scientific Workflows available in CEE for viewing an engineering simulation / analysis, such as the design of a mooring system or a fatigue analysis of a set of risers in a production unit. These Scientific Workflows are modeled as Abstract Workflows, previously created by teams of analysts. The Abstract Workflows will be converted into Concrete Workflows, and later executed in a Scientific Workflow Management System, in our case a BPEL Workflow engine.

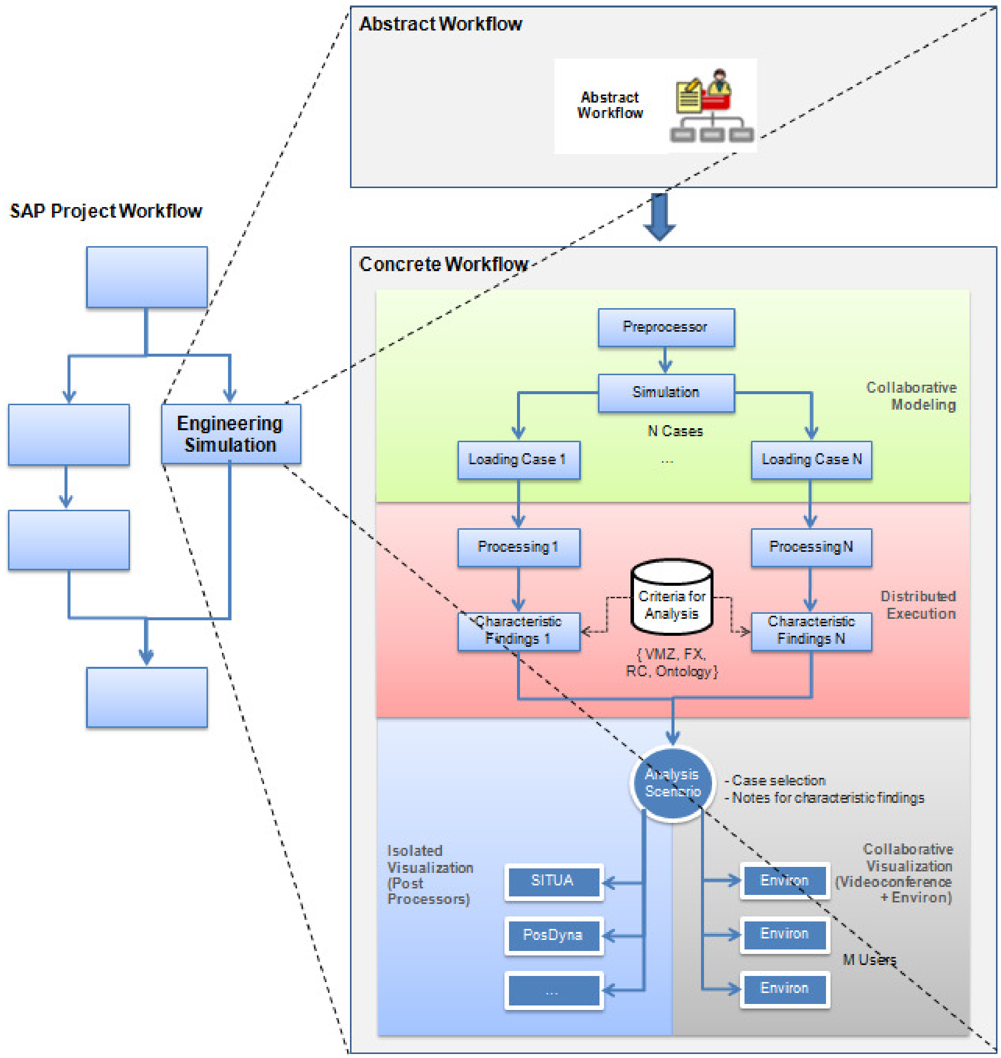


Figure 3.3: CEE Project Workflow.

In the concrete workflow, the user has to model the simulation input data through the corresponding Pre Processors (AnflexGUI for Anflex, SITUA for Prosim, etc.). Videoconference support allows the collaborative modeling of the engineering workflow, before its execution on the Resource Management System attached to CEE. After the execution, users can analyze the results separately in their Post Processors (AnflexGUI, SITUA, etc.), or in a Collaborative Visualization Session supported by Environ and the CEE infrastructure.

Figure 3.4 shows the main components of the CEE interaction. Initially after the user is logged in the system, the CEE *User Service* on the client machine registers the user in the *Collaboration Manager Service* on the CEE server, all services that the user's machine is able to support (*Environ Service, CSVTool Service,* etc) is also registered on the CEE server *Service Registry*.

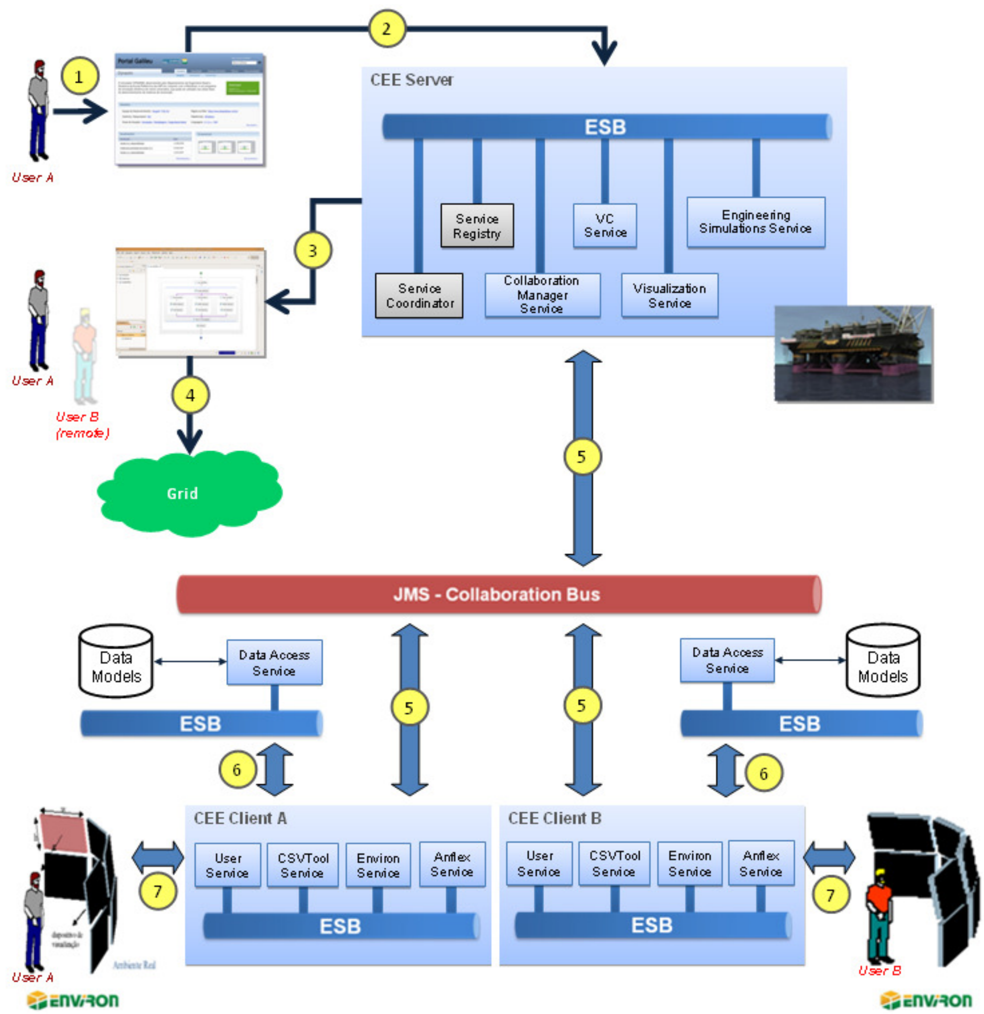


Figure 3.4 : Overview of the user interaction with CEE.

After registration of its services on the server, **User A** accesses the **CEE Portal** (**1**) through a web browser to request the execution services on the CEE server on his behalf (**2**). As an example, **User A** can model collaboratively with **User B** a *Concrete Scientific Workflow* (**3**). When the model is assembled and all input parameters of the concrete workflow is informed, **User A** can submit the workflow as a simulation on *a Grid* integrated into the CEE infrastructure (**4**). Upon finishing its execution, the results of the concrete workflow may be visualized in a *Collaborative Visualization Session* with **User B** (**5**).

During the collaborative visualization session, the users can require the execution of alternative simulations and have its results exhibited automatically (**6 and 7**).

In the following we present an overall solution for developing a complete workflow, including a sequence of screenshots describing the creation of a collaborative visualization session between two users on the CEE through the portal. Chapter 5 will give more scenarios tested with the CEE.

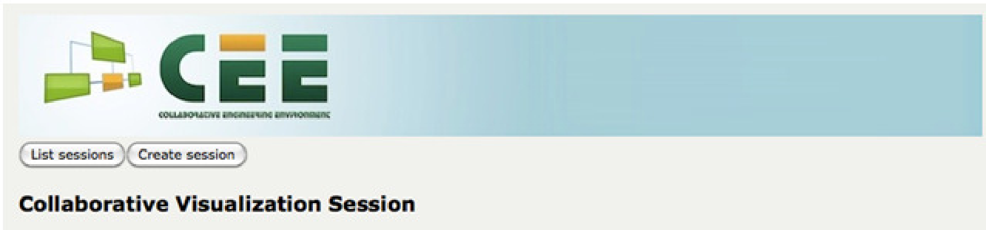


Figure 3.5 : Creating the collaborative visualization session

First the user inside a web browser accesses the CEE-Portal and then selects the option to create a new visualization session (Figure 3.5). The user has to select which users, already registered in the CEE server, will participate in order to create a session. The session name, its type (Informal, Classroom or Lecture) and the role of each participant (Coordinator or Participant) should be also selected (Figure 3.6).

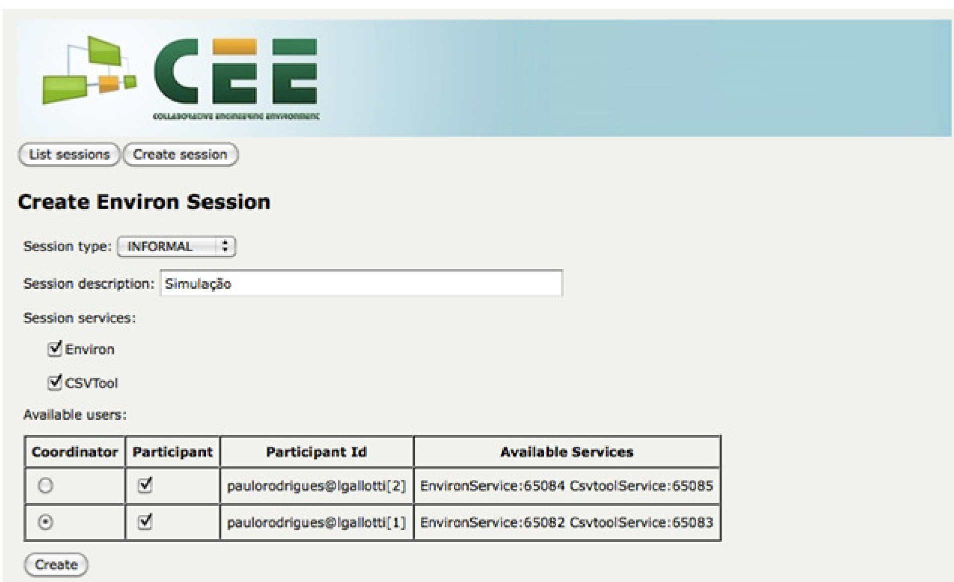


Figure 3.6 : Selecting the coordinator, the kind of session and the users that will participate.

Once the session is created the user can send commands (Figure 3.7 : Sending commands to load a simulation to visualize in the session to all or individual participants. One such command could be to load the simulation that could be analyzed in conjunction with another specialist (Figure 3.8).

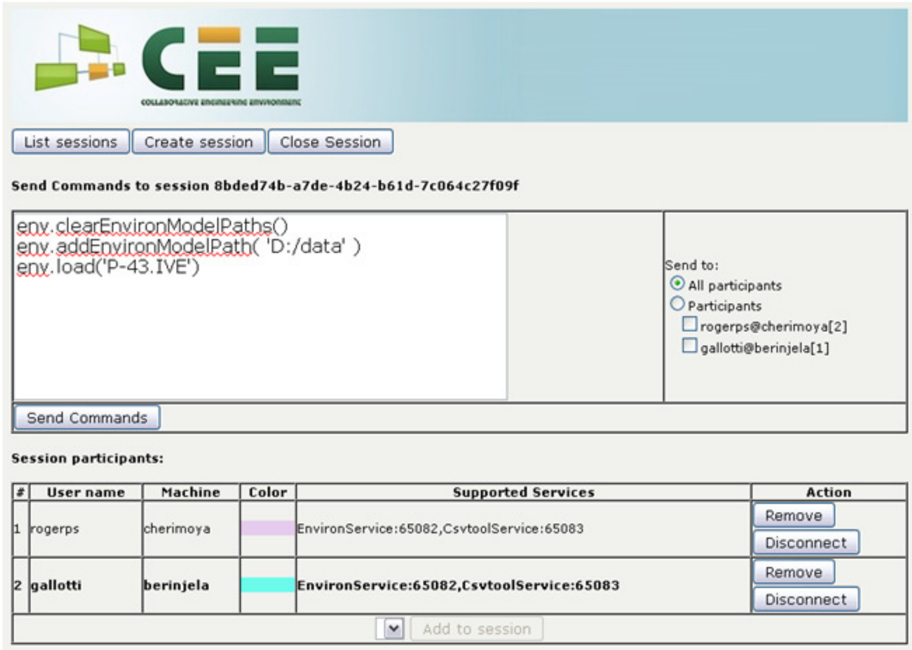


Figure 3.7 : Sending commands to load a simulation to visualize in the session

An example of command that might be used can be load a simulation to be analyzed in conjunction with another specialist.

Notice that the awareness mechanism indicates the status of each user (on-line or offline, i.e. momentarily disconnected from session), its role (coordinator or participant) and the user system ID.

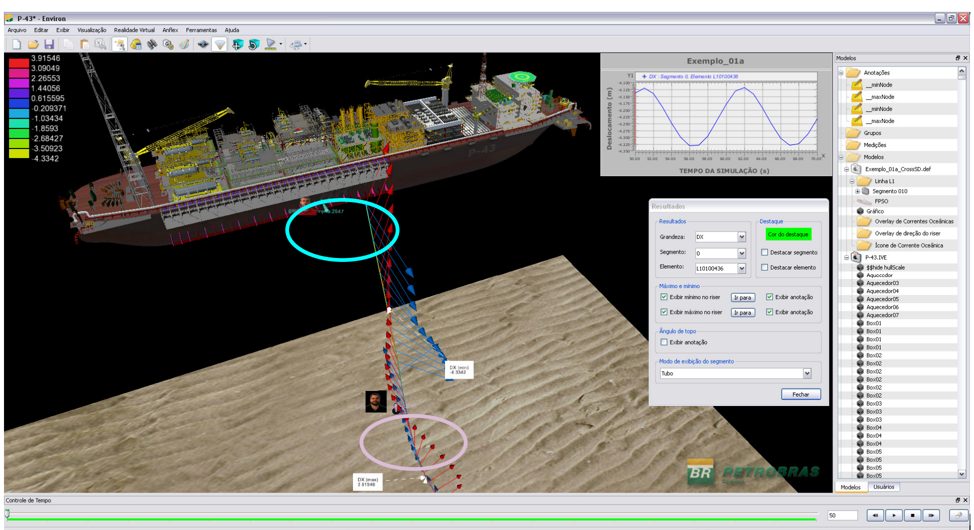


Figure 3.8: Riser simulation visualized in collaborative session.

4
CEE SOA Architecture

This chapter will present the CEE Architecture and its main components. The approach of this chapter starts with the description of the architectural layers of the solution, followed by the description of the main services and components description that implement the SOA model presented in the previous chapter. Further details about the architecture, its services and components are presented in Appendix A.

4.1.
CEE Architecture Layers

CEE aims to provide a multi-user collaborative environment for the execution, control and visualization of engineering simulations. Thus, it is necessary to base on solid distributed technologies as well as provide some CSCW services to its users. We can distinguish three main layers in the overall architecture (Figure 4.1): a technology layer, a collaborative engineering layer and an application layer. The technology layer refers to the basic Information Technology (IT) infrastructure selected for implementing the basic services of CEE. The collaborative engineering layer comprises all the necessary components to enable the execution of an engineering simulation and the visualization of its results in a collaborative session. The application layer comprises all the end-user applications that will benefit from the CEE collaborative resources.

4.1.1.
Technology Layer

CEE requires a solid infrastructure to provide security, persistence, transactions support, scalability and performance. We have chosen the JEE (Java Enterprise Edition) standard [JEESun] as the technology infrastructure for this research project. It saves us from implementing infrastructure and system-specific code, besides allowing us to base on open specifications and components. This technology makes our system vendor-independent, and consequently any JEE-compliant application server can be used. The JEE middleware is responsible for the basic infrastructure such as security, performance, server federation among others. Concerning the database management, the system is also generic, implementing a Data Access object layer based on a standardized object to relational mapping (ORM) provided by the Java Persistent API (JPA). As our Message Oriented Middleware (MOM), we have used ActiveMQ [ActiveMQ], an open source Java Messaging Service Provider. The overall architecture uses pervasively XML for data interchange among the Engineering Simulations (Anflex, Prosim, etc), Pre and Post Processors and the VR Visualization Tool (Environ).

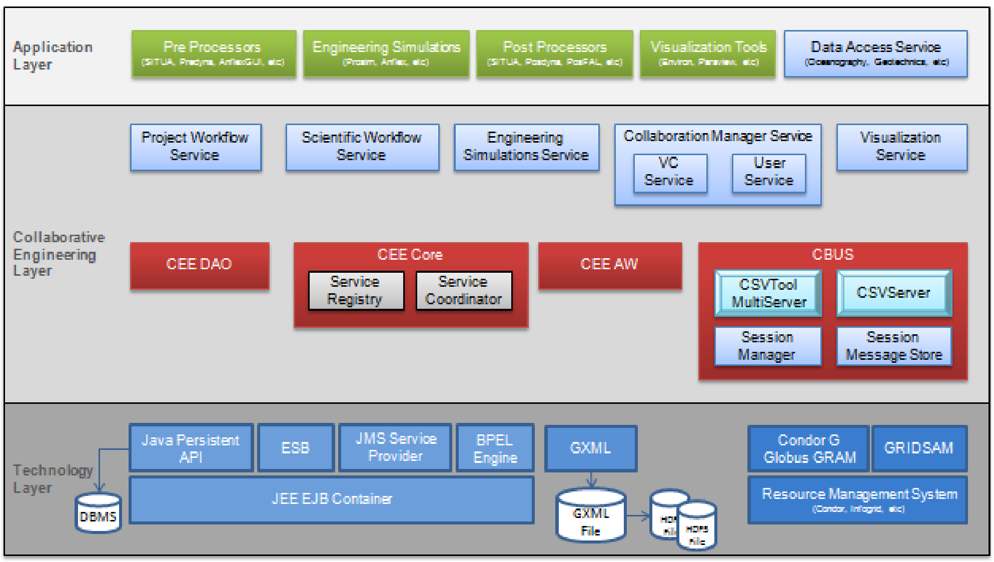


Figure 4.1: CEE Architecture Layers.

The OASIS (Organization for the Advancement of Structured Information Standards) has defined the Business Process Execution Language (BPEL) as a standard-based way of orchestrating a business process composed of services [OASIS]. 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. This capability is very important for having a flexible environment for the execution of Scientific Workflows; therefore we have chosen the BPEL Engine as our Scientific Workflow.

For the Grid subsystem we have choose Condor [Condor] and GridSAM [GridSAM]. GridSAM is a Grid Job Submission and Monitoring WebService 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) [LMN+04].

Using GridSAM to execute jobs on a Grid (in our case, Condor) gives us transparency of the underlying Grid scheduler. 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.

4.1.2. Collaborative Engineering Layer

The collaborative engineering layer is the most important part of the overall system, and has been designed taking into account the CEE main components presented in Chapter 1. The system is divided into several modules, seamlessly integrated. CEE Core is composed by a collection of collaboration tools, providing services like shared spaces, access control, floor management, and integration for both synchronous and asynchronous communication through the use of a *Collaboration Bus (CBUS). CBus* is an infrastructure for communication based on the Java Message Service (JMS) Provider and the Enterprise Service Bus (ESB) available on the technology layer. The *CEE Awareness Service* (AWS) is service mechanism providing group awareness for the CEE components.

There are a lot of services in this layer providing collaboration support to CEE applications, further details are presented later in this chapter. The *VR Visualization Service* and the *Collaboration Manager Service* are the most important components. They use the *CEE Core, CEE AWS* and *CEE CBUS* components to create a collaborative visualization tool to allow the users to collaboratively visualize the results of an engineering simulation in an immersive or desktop environment.

4.2. Application Layer

The engineering applications supported by the CEE, are in the Application Layer. It can be generically divided in four different components: Pre and Post Processors, Engineering Simulators (Anflex [MGJ95], Prosim [JE94]), *Data Access Services* and VR Visualization Tools. One example of VR Visualization tool is Environ [RCW+09], which was developed to visualize massive CAD models and engineering simulations in immersive environments (VR and Desktop)

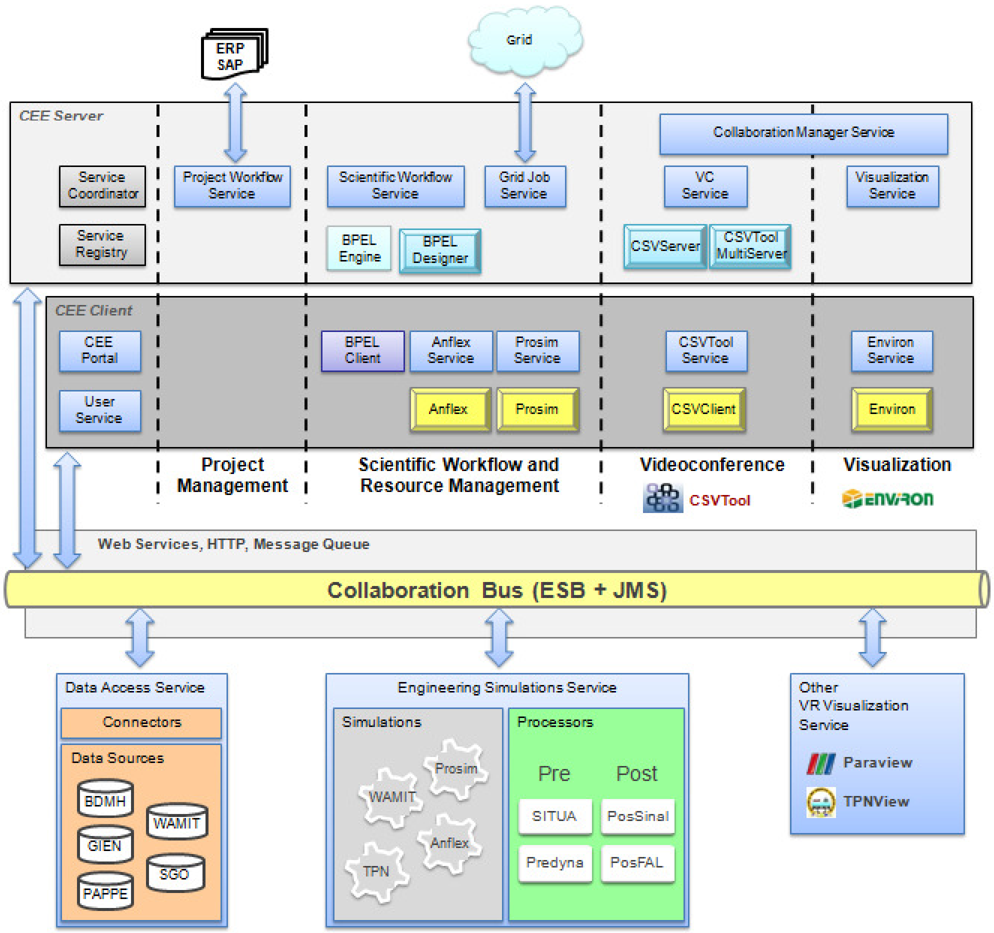


Figure 4.2: CEE Detailed SOA Architecture.

CEE has a client-server architecture (Figure 4.2), where the CEE-server is deployed in a JEE Application Server (Glassfish in our case) which allows better scalability and automatic transaction control. The CEE main services reside in the Application Server where a *Service Registry* is used to record all available services present in the CEE clients. For example the CEE VR Visualization tool, Environ, should be available on a CEE client machine allowing users to participate in a *Collaborative Visualization Session*, controlled by the *Service Coordinator.*

The features of JAX-WS [JEESun] platform enable the publication of a WebService interface for each Session Bean, so that the remote clients are able to request services through the interface. On the server, that's the case of *Service Coordinator, Service Registry* and *Collaboration Manager Service*. On the client, *Environ Service, CSVTool Service,* and *Engineering Simulation Services (Anflex Service, TPN Service, Prosim Service, etc).* Those services will be described in the next section.

4.2.1. CEE Client Services

The services on the client machine are designed basically to manage and control the execution of the target application locally, which usually does not have any collaboration support. In order to make them collaboration-unaware applications, a software infrastructure is provided by the *CEE Server* and its associated services running on server and client machines.

The *User Service* is responsible to registry user information and the availability of the other services on the client machine using the *Service Registry* WebServices interface.

Environ Service, for example, is the service responsible for creating a mechanism that allows the communication of the VR Visualization tool, in our case *Environ,* running on a client machine and the *Collaboration Manager Service* that runs on the server. *Environ Service* run as a daemon process and whenever it receives a command to start a session on the client it creates a proxy, *Environ Proxy*, which acts as a mediator between the *Environ* and the *Collaboration Manager Service.* Following this pattern *Environ* can receive and send commands to other users through *Environ Proxy*, transforming Environ into a collaboration-unaware application, which is essential for the CEE as a CPSE. Further details are given in section **Error! Reference source not found.**.

Analogously the same happens with the *CSVTool Service* and the *Engineering Simulations Services (Anflex Service, Prosim Service, etc)*, which are detailed in the next sections.

Communication between users and the CEE server will be accomplished through the CEE Portal, which accesses the *Service Coordinator* through its WebServices interface and calls for some service, such as the creation of a Collaborative Visualization session for visualizing results. More details can be seen on the section about creating a *VC* and *VRV session* (sections **Error! Reference source not found.** and **Error! Reference source not found.**).

4.2.2. CEE Server Services and Components

Some of the services executing on the server are implemented as EJB (Enterprise Java Beans) components and exported as WebServices for servicing remote communications from the clients. The main components and services on the server are: *Service Coordinator*, implemented as an EJB with an exported

WebService interface; *Service Registry,* like *Service Coordinator* is also implemented as an EJB and have a WebService interface. The other services are implemented as WebServices.

4.2.2.1.Service Coordinator

The *Service Coordinator (SC)* is a centralized service bridging clients to servers and providing deployment-wide services within a cluster-deployed CEE instance. The *Service Coordinator* is a singleton within a CEE deployment, and thus manages information and services that are relevant to a CEE deployment as a whole. Its primary responsibilities include:

- acting as a 'well known' CEE access point for clients;
- acting as a gatekeeper handling authentication and authorization of clients accessing the CEE system. The SC will use an LDAP (Lightweight Directory Access Protocol) [LDAP] service to manage client-related authentication and authorization information;
- acting as an environment manager for the CEE deployment as a whole.

Service Coordinator is responsible for creating and controlling a *CEE Session*, which is a composition of different kinds of sessions available: a project workflow session, a scientific workflow session, a visualization session and a videoconference session. Each client's interaction with the server initiates with the creation of a specific session where the interactions take place. Depending on the course of actions in the client-server interaction, the *CEE Session* will be composed of those different sessions.

The CEE client can start a *Project workflow session* to import project information from an Enterprise Resource Planning system, such as SAP [SAP] to an internal project representation properly adapted to CEE functionalities. To run an engineering simulation the client starts a scientific workflow session, assembles an engineering workflow using scientific workflow designer, and submit the workflow as a job for execution on a Grid Computing infrastructure provided by the CEE server. At the end of the execution of a scientific workflow session, a visualization session can be created to allow the client to inspect the results of the simulation. During the whole process the client can start a videoconference session at any time to add support for audio/video communication with other clients. This sequence of actions is explained at the end of Chapter 3 and will be seen in the case studies of Chapter 5.

The *Service Coordinator* can be invoked remotely to send commands to a *CEE Session.* This is what happens when the user access the *CEE portal* to create a new *CEE Session* or to join an existing one. All the commands sent from the *CEE client* to *Service Coordinator* are redirected to *Session Manager* who manages the *CEE Session* and coordinates the aggregate functionality added to it depending on which session was started by the client. For example, the client can initially start *CEE Session* requesting the creation of a *VC Session* to discuss project details with a remote client. After some discussions they can decide to start a scientific workflow session to collaboratively create an engineering workflow that will be scheduled to run on the Grid Computing infrastructure provided by the CEE Server. In this way, the *CEE Session* will be a composition of a *ScientificWorkflow Session* and a *VC Session*. If they decide to continue they can start a *VR-Visualization session* and see the results.

It is important to mention that a *ScientificWorkflow Session,* a *VC Session*, and a *Visualization Session* can coexist at the same time in the server for the client that are participating in the collaboration. On the other hand, the *Project Workflow Session* is required to run first, possibly with the support of a *VC Session*, to allow the user to create its own project environment to store data generated during the execution of the project.

4.2.2.2.Collaboration Manager Service

Collaboration Manager Service handles information about logged users in a collaborative *CEE Session*. It is responsible for the collaborative session management, access control policies, and behavior of each participant. One of the main components of the *Collaboration Manager Service* is the *Session Manager*, which is responsible to manage registered clients on the server and also coordinate the execution of the *VR Visualization Service* and the *VC Service* when both are selected services to be used in a CEE session. *Collaboration Manager Service* is also responsible for initiating the JMS Service Provider, in order to start the Collaboration Bus, the communication mechanism used by the collaborative session users.

VR Visualization Service is responsible for giving support for the Visualization Session execution and for supporting the creation of the *Collaborative Visualization Session* together with Collaboration Manager Service. The *VR Visualization Service* integrates Environ [RCW+06, RSS+09] as the CEE VR Visualization tool furnishing the necessary support.

The *VC Service* is a videoconference service that integrates CSVTool (see section **Error! Reference source not found.**). *VC Service* responsibility is to control the evolution of a videoconference session which could occur simultaneously with a *Visualization Session,* characterizing a *Collaborative Visualization Session*.

The *Service Coordinator* in conjunction with the *Session Manager* is responsible for the creation and controlling of the *Collaborative Visualization Session* connecting Environ with any other engineering simulations that wants to have its results collaboratively visualized by its users.

4.2.2.3.
Collaboration Bus Implementation

The CEE-Collaboration Bus is created by the combination of an Enterprise Service Bus and a Message Oriented Middleware (MOM), with a Java Messaging ServiceTM (JMS) compliant implementation provided, in our case, by Apache-ActiveMQ [ActiveMQ].

Remote procedure call (RPC) systems, including Java RMI, are synchronous – the caller must block and wait until the called method completes execution, and thus offer no potential for developing loosely coupled enterprise applications without the use of multiple threads. In other words, RPC systems require the client and the server to be available at the same time. However, such tight coupling may not be possible or desired in some applications. MOM systems provide solutions to such problems. They are based on the asynchronous interaction model, and provide the abstraction of a message queue that can be accessed across a network. More generally, MOM is a category of software for communication in a loosely-coupled, reliable, scalable and secure manner amongst distributed applications or system. The overall idea with a MOM is that it acts as message mediator between message senders and message receivers.

JMS was defined to allow Java application to use enterprise messaging systems. It provides a common way for Java applications to access such enterprise messaging systems. Two types of channels are available, a Point-to-Point (i.e. a single channel per peer), available for peer to peer communications, and a public-subscribe channel for group communications.

- *Point-to-Point (Queue destination):* In this model, a message is delivered from a producer to one consumer. The messages are delivered to the destination, which is a queue, and then delivered to one of the consumers registered for the queue. While any number of producers can send messages to the queue, each message is guaranteed to be delivered, and consumed by one consumer. If no consumers are registered to consume the messages, the queue holds them until a consumer registers to consume them.

- *Publish/Subscribe (Topic destination):* In this model, a message is delivered from a producer to any number of consumers. Messages are delivered to the topic destination, and then to all active consumers who have subscribed to the topic. In addition, any number of producers can send messages to a topic destination, and each message can be delivered to any number of subscribers. If there are no consumers registered, the topic destination does not hold messages unless it has durable subscription for inactive consumers. A durable subscription represents a consumer registered with the topic destination that can be inactive at the time the messages are sent to the topic.

In our implementation, the *Collaboration B*us is created by the *Collaboration Manager Service* every time a new CEE Session is created. For that session a topic, with the name of the CEE session is created, to allow group communication. A queue for every participant in the CEE session is also created allowing peer to peer communications.

In Section **Error! Reference source not found.**, we give more details about the usage of the collaboration bus for collaborative visualization with our VRV, Environ.

4.2.2.4.Scientific Workflow Service

The Scientific Workflow Service is used to create engineering workflows (comprised of the engineering simulations) and orchestrate the execution of those workflows using BPEL to describe the workflow and schedules it for execution on a GCI through the use of the Grid Job Service of the CEE.

4.2.2.5.Project Workflow Service

This service is responsible to help the user import project definition from an external Enterprise Resource Planning (ERP) system and represent it internally in the CEE. It is also responsible to make all the necessary synchronization to update the external ERP system improving the ERP's capabilities having a better estimate of the efforts and costs involved in the actual course of a Large Scale Engineering Project.

4.2.2.6.Service Registry

Service Registry records the available services in every client machine connected to the CEE server. The registry stores the URL of each service available in the client machine, the VR *Visualization Service (Environ Service),* the *Videoconference Client Service (CSVTool Service)* and any *Engineering Simulation Services (Anflex Service, Prosim Service, e.g.).*

The *Service Registry* is consulted by the *Service Coordinator* to obtain information about the services that could be called on the client machine whenever a client requests the creation of a *CEE Session.* The *Service Registry* gives transparency of location for the services that will be invoked by the *Service Coordinator* on the client machine whenever a *CEE Session* is created.

In the next sections more details will be presented about the execution of the different types of section that the clients can invoke on the *CEE server*.

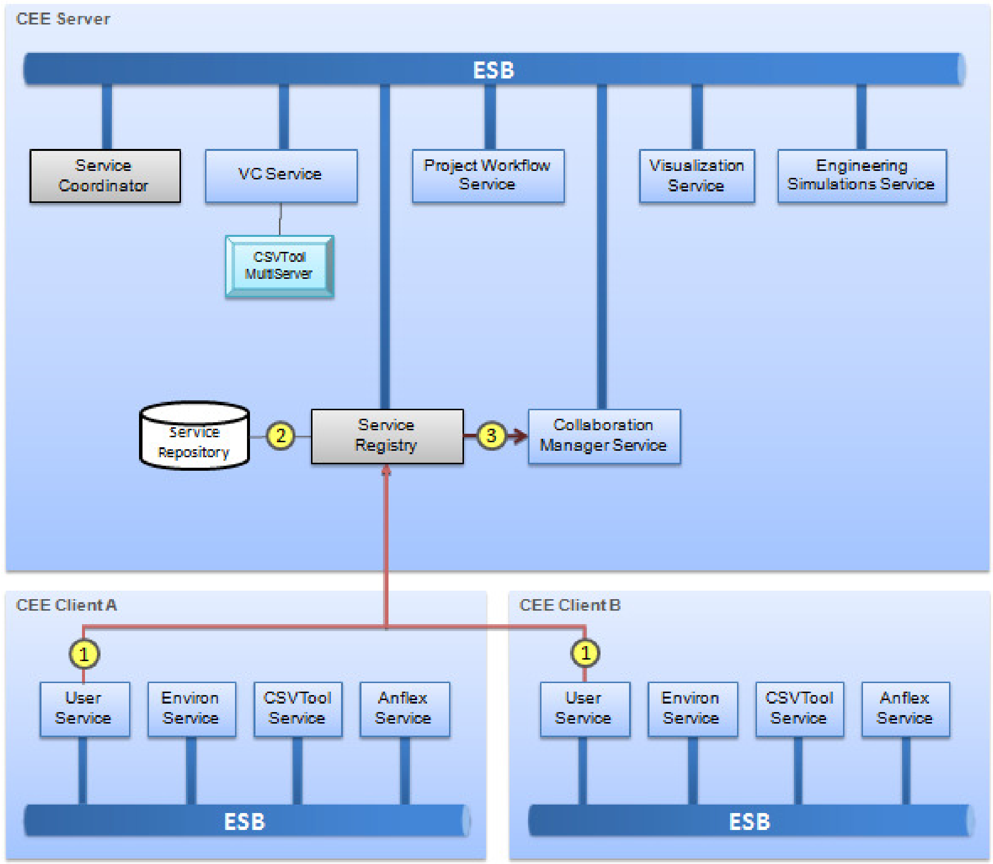


Figure 4.3: Registering available services in *Service Registry*.

4.2.3. Registration of Services

Before using the CEE server, the CEE client must authenticate on the CEE Server. The authentication process is not addressed in this thesis, once there are many standard ways of doing this. Particularly for JEE Application Server there is a standardized process available using the JAAS (Java Authentication and Authorization Service) [NT06].

Once authenticated in the server, the CEE client should register the services available on its client machine (Figure 4.3). The components responsible for this registration are the *User Service*, *Service Registry* and *Collaboration Manager Service*. The *User Service* is the service utilized by the CEE Client application. The sequence of actions for the registration process is described as follows:

1. The *Client,* through the *User Service,* invokes *Service Registry* WebServices interface to register user information, and the available local

services: *Environ Service*, *CSVTool Service* and E*ngineering Simulation Services*. This information is static and dependent on each machine configuration, so it can be saved on a local CEE-configuration file and defined during the installation of the CEE client on the client machine;

2. Upon receiving the user information and the available services the *Service Registry* store this information on the Service Repository;
3. In the last step, *Service Registry* sends the user information to the *Collaboration Manager Service*, enabling the user to start a new session or joining an already started session.

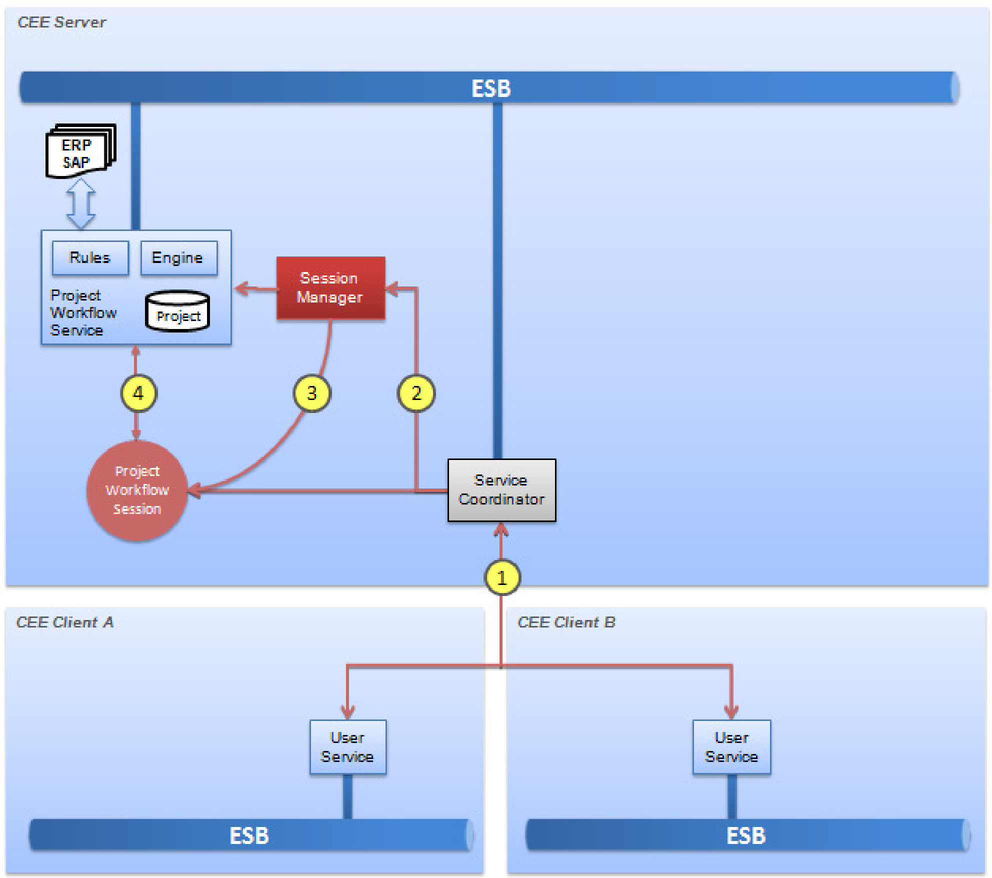


Figure 4.4: Importing the Project Workflow from a ERP system.

4.2.4. Project Workflow Service

Whenever a user wants to use the CEE server, he must create a Project Management Environment (PME) area on the CEE server or use an old one. The PME allows users to manage, in collaboration with others, the execution and control of engineering projects, imported from an ERP system such as SAP, and referred as Project Workflow.

The components responsible for importing a project workflow to a PME are the *User Service*, and the *Service Coordinator*. The *User Service* is the service utilized by the CEE Client application. The sequence of actions for importing the project workflow are described as follows (Figure 4.4):

1. *The Client,* through the *User Service,* invokes *Service Coordinator* WebServices interface to request the importation of a Project Workflow to a PME;
2. *Service Coordinator* sends a request to *Session Manager* to create a *Project Workflow Session* and passes all the information about the project that will be imported from the ERP system;
3. *Session Manager* asks the *Project Workflow Service* to create the *Project Workflow Session* and then send the ID of the session to the *Service Coordinator;*
4. *Project Workflow Session* imports the project workflow into the CEE Project Management Environment and the *Service Coordinator* notifies the client about the creation of the project.

4.2.5. Scientific Workflow Service

In the Scientific Workflow Environment, the services representing the Engineering Simulators and Grid Computing Infrastructure have WebServices interface which allows them to be directly executed on a BPEL engine. To properly define the input parameters of the simulators the users will possibly have to execute the Pre-processors for assembling the input data for each simulation case (Figure 4.6).

After the creation of a Project Workflow, the user can identify certain stages that might require the execution of many engineering simulations. For that, the user will model different scientific engineering workflows (in our case, using BPEL designer). Those scientific workflows can be created from scratch, using previously used workflows or pre-defined BPEL workflows templates (Figure 4.5). After obtaining the new workflow, the user will be able to execute it on the BPEL Workflow Engine.

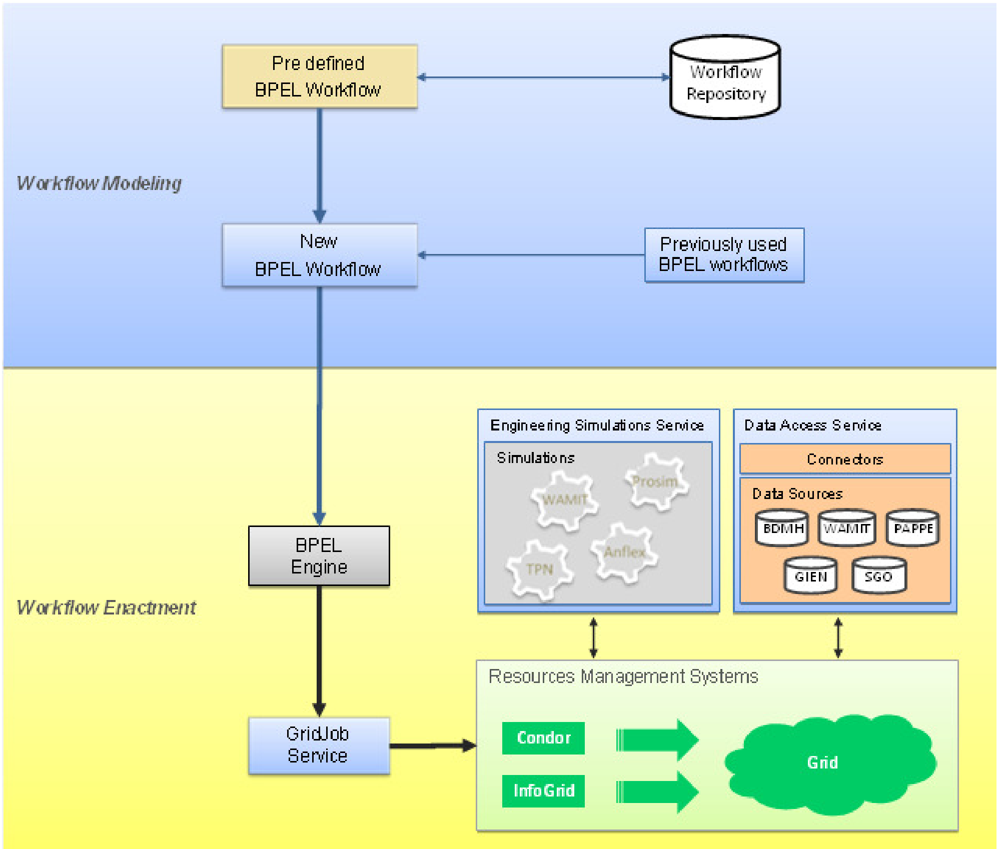


Figure 4.5: Modeling and Executing a Scientific Workflow.

Figure 4.6 presents the needed services to execute a scientific workflow for riser analysis using the *Anflex* simulator, more details are given in the Chapter 5. The workflow starts generating different simulation cases from a base-case, prepared by the *Anflex* Pre-processor, *AnflexGUI*. During the creation of the simulation cases, the *Ocean Service* informs the different environmental conditions that are combined by the *Anflex Service* with the base-case to generate different loading cases. Those loading cases will be used by the Anflex simulator to analyze different scenarios. The execution of those simulations will be orchestrated by BPEL engine in cooperation with the CEE *GridJob Service* to execute the engineering simulation on a Numerical Grid.

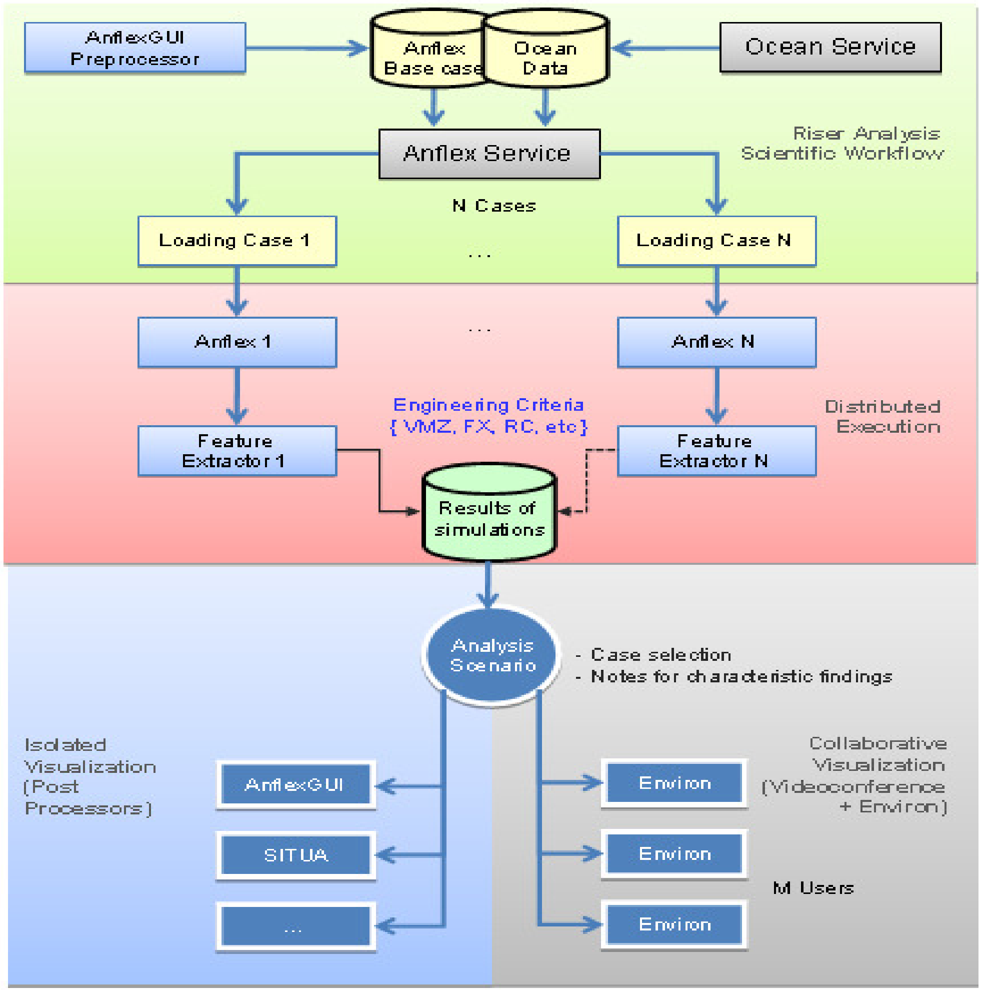


Figure 4.6: Executing a Scientific Workflow instance in CEE.

The CEE is being constructed in order to allow the management and use of high-performance computational resources. The integration of CEE and Grid Resource Management Systems such as Condor [Condor] and InfoGrid [LMC+05], can be done though the GridJob Service. We choose the GridSAM as our Grid JobService.

4.2.6. Videoconference Service

CSVTool (Collaboration Supported by Video) [PRS+03, LKR+07] was developed as a multiplatform videoconferencing tool focused on audio and video communication with support to cooperation (such as desktop image transmission) and coordination features like management of multiple VC sessions, individual control of transmitted and received audio and video streams for each participant, and the possibility of creating new custom types of video conference sessions. CSVTool is implemented with JMF (Java Media Framework) [JMFSun], which employs a high-level abstraction for codec and transmission protocol details. It was designed with the goal of providing integrated multimedia communication to collaborative applications, but can also be used as an independent videoconferencing tool.

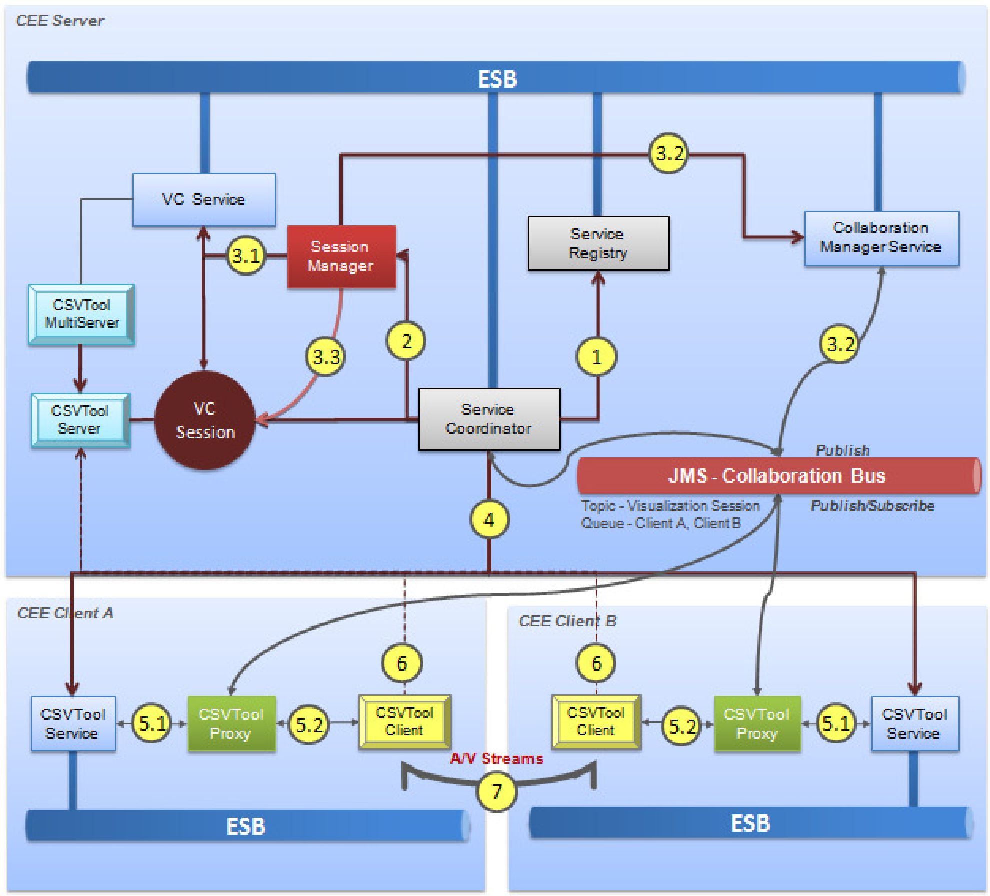


Figure 4.7: Starting a VC session in CEE.

The *VC Service* initiates the *CSVTool Multiserver* [LKR+07] in the server, which will handle information of multiple different videoconference sessions, allowing CEE users to choose which videoconference they might join, or decide to create a new one, according to each *VC session* access policies. Each *VC session* is managed by the *CSVTool Server* created in the server whenever a new *VC session* is initiated. In the client, the *VC Client Service* is responsible for initiating the *CSV Client*, which communicates to, and is controlled by, the *CSVTool Server*. The *CSVTool Client* is responsible for managing the audio and video streams between the *VC Session participants* creating a point-to-point communication among all the participants.

In CEE, a videoconference is started by the user whenever he wants to share the creation of a scientific workflow, to discuss the best alternatives and parameters of engineering simulators, or even to analyze collaboratively the results of a simulation. Each created conference is registered on the *CSV Multiserver* (name, description and type – public or private) in such a manner that enables other users to know which videoconferences are happening and eventually choose to participate in one of them. The videoconferences in CEE can be public or private, the difference between them is that in the public anyone can participate, whereas in the private, only pre-authorized users, defined by the session creator, are allowed to attend.

The components responsible for executing a videoconference in CEE are the *Service Registry, Service Coordinator, CSVTool MultiServer*, *CSVTool Server CSVTool Service, CSVTool Client* and *CSVTool Proxy,* The sequence of actions for starting a Videoconference Session in CEE is described as follows (Figure 4.7):

1. *Service Coordinator* verifies in the *Service Registry* whether the client requesting the creation of the VC Session supports the requested services or not;
2. *Service Coordinator* sends a request to *Session Manager* to create a *VC Session*;
3. *Session Manager*:
    - 3.1. The *Session Manager* requests to *VC Service* the creation of the VC Session. *VC Service* creates on the server a *CSVTool Server* responsible for the management and control of audio and video (A/V) stream of all participant. The *CSVTool Server* registers itself in *CSVTool Multiserver*, informing the creation of a new *VC Session*;
    - 3.2. The *Session Manager* requests the creation of a Collaboration Bus to the *Collaboration Manager Service*;
    - 3.3. The *Session Manager* requests to *VC Service*, the creation of a *VC Session* and then send the sessionID of this session to the *Service Coordinator;*
4. The *VC Service* invokes *CSVTool Service* on each participant, passing the sessionID;
5. *CSVTool Service*

5.1. *CSVTool Service* creates the *CSVTool Proxy* passing information of the *VC Session* and the JMS – Collaboration Bus.;

5.2. *CSVTool Service* creates the CSVTool Client ;

6. CSVTool Client connects to *CSVTool Server* to obtain A/V communication parameters among participants. *CSVTool Server* is responsible for the control of the *VC Session* participants;
7. *CSVTool Client* connects all the other *CSVTool Clients* and create the RTP A/V streams between them. A/V streams are transmitted between clients without *CSVTool Server* participation.

4.2.7. Collaborative Visualization Service

Environ, the CEE–VRV, was adapted to be transformed into a collaboration-aware application with the support provided by the CEE collaborative infrastructure. Figure 4.8 demonstrates how Environ was adapted to be able to send and receive messages from other clients.

4.2.7.1.Collaborative support for Environ

The adaptation follows the Remote Procedure Call mechanism, created in the former times, as an alternative for distributed computation. The process of adaptation was made in two levels. In the first level, one instance of Environ acting as a server connects to many other Environ running as clients. The Environ server creates a server socket and starts listening the port waiting for client connections, while the other Environ clients creates a socket and connects to the Environ server.

For every new client connection, a thread on the Environ server is created to deal with this new client, and the exchange of commands between that thread in Environ server and the Environ client can start. This synchronous communication is very limiting for collaboration purposes.

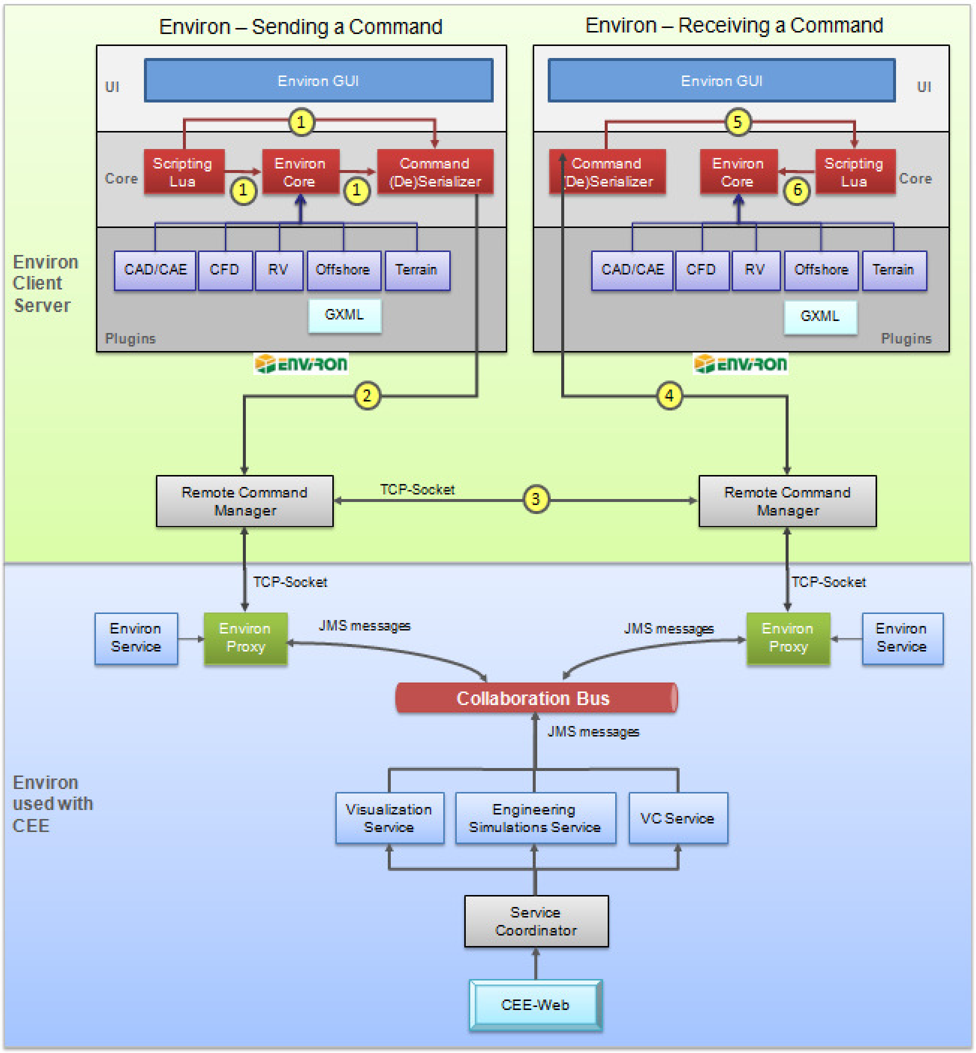


Figure 4.8: Environ and its RemoteCommandManager.

To implement the exchange of commands we created the *ComandSerializer* component that is in charge of transforming the commands within Environ into its equivalent serialized form. This serialized form used the characteristics of the Lua language [IFC96] that is fully integrated into Environ through a Lua scripting component (Figure 4.8). Another component of the solution is the *RemoteCommandManager* who is responsible to send and receive the serialized commands through a TCP/IP socket network connection. The flow of commands is depicted as follows:

1 Every command executed by the *EnvironCore,* or Lua command sent to Environ from the console, that should be retransmitted, is sent to

ComandSerializer component who translate the command into a serialized form;

2 *ComandSerializer*, upon serializing the command send it to the *RemoteCommandManager* whose job is to package the serialized command in an appropriate message format to allow peer *RemoteCommandManager* to deal with those messages in an efficient manner;

2.1 The message protocol defined consists of a header and tail demarcating the limits of the message, i.e., the serialized command. Inside the header, besides information like timestamp and sender identification, we have two distinct information characterizing the message itself the message category and the message command.

The categories are, among others: CollaborationCommands, ConsoleCommands, WebConsoleCommands, AnnotationCommands, CameraCommands, etc.

2.2 This division in categories allows an efficient treatment because messages can be discarded according to its category depending on the state of the Environ peer. For example, for a client running Environ which is in an offline state, most of the commands could be discarded except the CollaborationCommands.

3 *RemoteCommandManager* sends the serialized command package inside the message format, to all other *RemoteCommandManager* that are connected;

4 The *RemoteCommandManager* peer upon receiving the message, look into the header to check if it should process it or not;

5 *ComandSerializer* then receives the message from the socket, desserialize it and send it for execution in the Scripting Lua component;

6 In the Scripting Lua component the command is then executed locally.

Following this procedure the commands are executed back and forth in the Environ peers and we can achieve the first level of collaboration, with synchronous messages.

The second level of integration is a more powerful mechanism once it extends the types of communications that exists for Environ adding the group communication mechanism and the peer-to-pear mechanism provided by CEE Colaboration Bus.

In this second level (Figure 4.8), the *Environ Service* creates a local *Environ Proxy* that talks to the *Collaboration Bus* and acts as a server for

Environ, which connects to a server socket on *Environ Proxy*. This way, all collaboration communication is tracked by *Environ Proxy* that sends back those commands to *Environ*. On the other way, every command sent by Environ is resent by *Environ Proxy* to a topic created for the CEE Session in the Collaboration Bus or to an specific Queue.

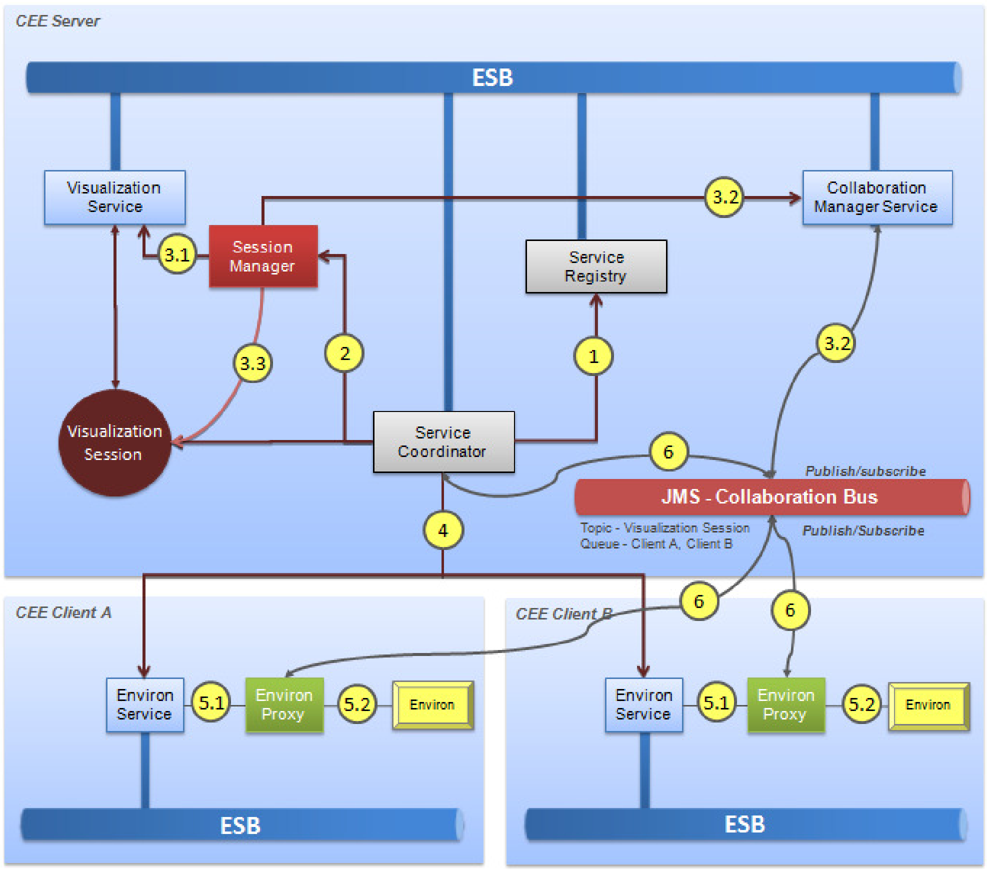


Figure 4.9: Starting a Visualization session in CEE.

4.2.7.2.Environ Service

The Environ Collaborative Session is implemented with the help of a Java Messaging Service (JMS). The Environ Collaborative Session is controlled by the CEE service coordinator in conjunction with all *Eviron Proxy* started on demand by the each *Environ Service* on a CEE-client machine that takes part in the CEE Session.

The components responsible for executing Collaborative Visualization Session in CEE are the *Service Registry, Service Coordinator*, *Environ Service, Environ* and *Environ Proxy.* The sequence of actions for the starting a *Collaborative Visualization Session* is described as follows (Figure 4.9):

1. *Service Coordinator* verifies in the *Service Registry* whether the client requesting the creation of the Collaborative Visualization Session supports the requested services or not;
2. *Service Coordinator* sends a request to *Session Manager* to create a Collaborative Visualization Session;
3. Session Manager:
    - 3.1. The *Session Manager* requests to *Visualization Service* the creation of *Collaborative Visualization Session*;
    - 3.2. The *Session Manager* requests the creation of a Collaboration Bus to the *Collaboration Manager Service*;
    - 3.3. The *Session Manager* requests the creation of *Visualization Session* to *Visualization Service* and then send the sessionID of the session to the *Service Coordinator;*
4. The *Visualization Service* invokes *Environ Service* on each participant, passing the sessionID;
5. *Environ Service*
    - 5.1. *Environ Service* creates the *Environ Proxy* passing information of the *Visualization Session* and the *JMS – Collaboration Bus*;
    - 5.2. *Environ Service* creates the Environ as client connected on *Environ Proxy*;
6. *Service Coordinator* receives commands from the *CEE Portal* and sends them to the topic or one of the queues created for the session in the *JMS Collaboration Bus*, which means that the commands is sent to all participants (topic) or to an specific one (queue). Similarly, all participants can exchange commands among themselves using the *Environ Proxies.*

In the next chapter we present some usage scenarios where we have tested the CEE prototype. In those scenarios we emphasize the contribution of the VR Visualization for the comprehension of the engineering simulation results.

5
CEE Application Scenarios

This chapter describes some scenarios for applying the CEE. First, we present the project of Collaborative Risers Analysis Workflow. Second, we describe the case of a Design Review Workflow of an engineering project where the support provided by CEE infrastructure allows the creation of a collaborative visualization session for Design Review.

5.1.
Collaborative Risers Analysis Workflow

Over the last ten years there has been a significant effort to develop offshore oil&gas reserves from ultra deeper water. One of the main challenges associated with deep-water field development is the riser system, necessary to transport the production fluids from the seabed to the floating production facilities.

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 certificate 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. In our case we have used a riser analysis software called Anflex [MGJ95], an internally developed Finite-Element-based structural analysis package.

5.1.1.
BPEL Scientific Workflow

We have defined an Anflex-based riser analysis workflow controlled by the BPEL engine for automating the validation process and certification of riser analysis [SCJ+02]. The workflow integrates the execution of the following services: *Ocean Service, Anflex Service e Grid Job Service.* This workflow was described in section 3.4, where we mentioned that before running the engineering simulations, the loading cases must be prepared using a pre-informed Anflex base-case, prepared with Anflex Pre-processor, AnflexGUI. In Figure 5.1 we show the final version of the Riser analysis workflow in a BPEL designer, in this case a plugin for the Eclipse development tool [EclipseBPEL].

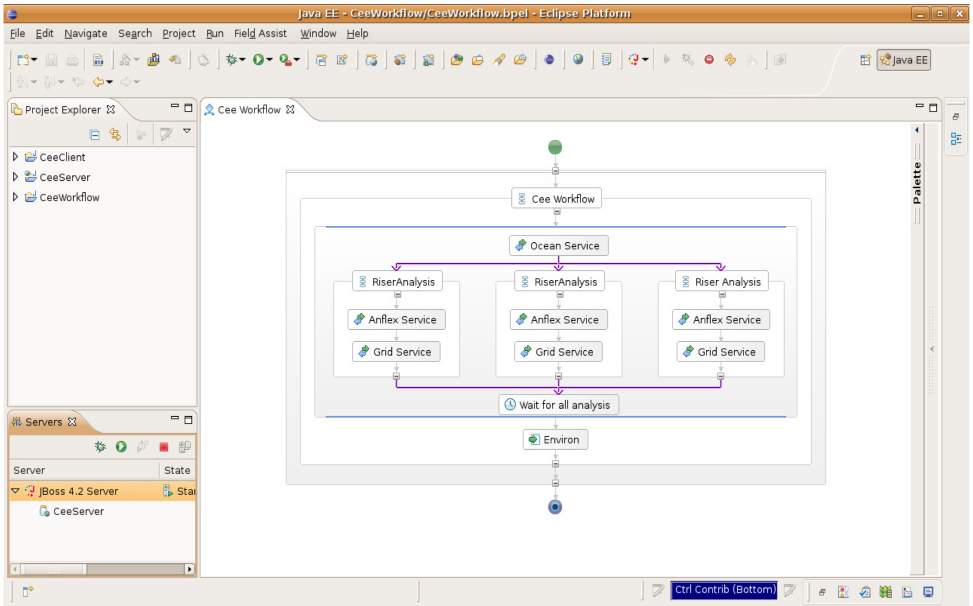


Figure 5.1: Constructing the Riser Analysis workflow on BPEL Designer.

The workflow starts with an Anflex base-case, where the basic configuration of the experiment is defined such as a production unit, riser's geometry, soil bathymetry, etc. *Anflex Service* receives user input parameters from BPEL designer and is responsible for creating different loading cases according to the different meteo-oceanographic conditions provided by the *OceanService*. After that, BPEL instructs CEE *GridJob Service* to communicate with Condor to submit jobs for executing the Anflex simulation program on the 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 a *Collaborative Visualization Session* created by the CEE *Service Coordinator*. The VC Service is available at anytime providing human to human interaction for solving conflicts. To start the *Collaborative Visualization Session,* the *Service Coordinator* is called to start the session. The *Service Coordinator* uses the ESB infrastructure to invoke the execution of the *Environ Service* to start the session. *Environ Service* then starts *Environ Proxy* and the Environ Application. *Environ Proxy* communicates with Environ by a TCP socket connection for sending and receiving commands. The

Collaboration Manager Service starts the *JMS Collaboration Bus* to allow users to exchange commands among themselves using the instance of *Environ Proxy*.

Figure 5.2 depicts the Sessions created in this situation. In this specific situation both users have the *Scientific Workflow Service, Environ Service, and the CSVTool service* initiated, which means that they will have a support for executing a Videoconference and also are able to execute the Anflex program.

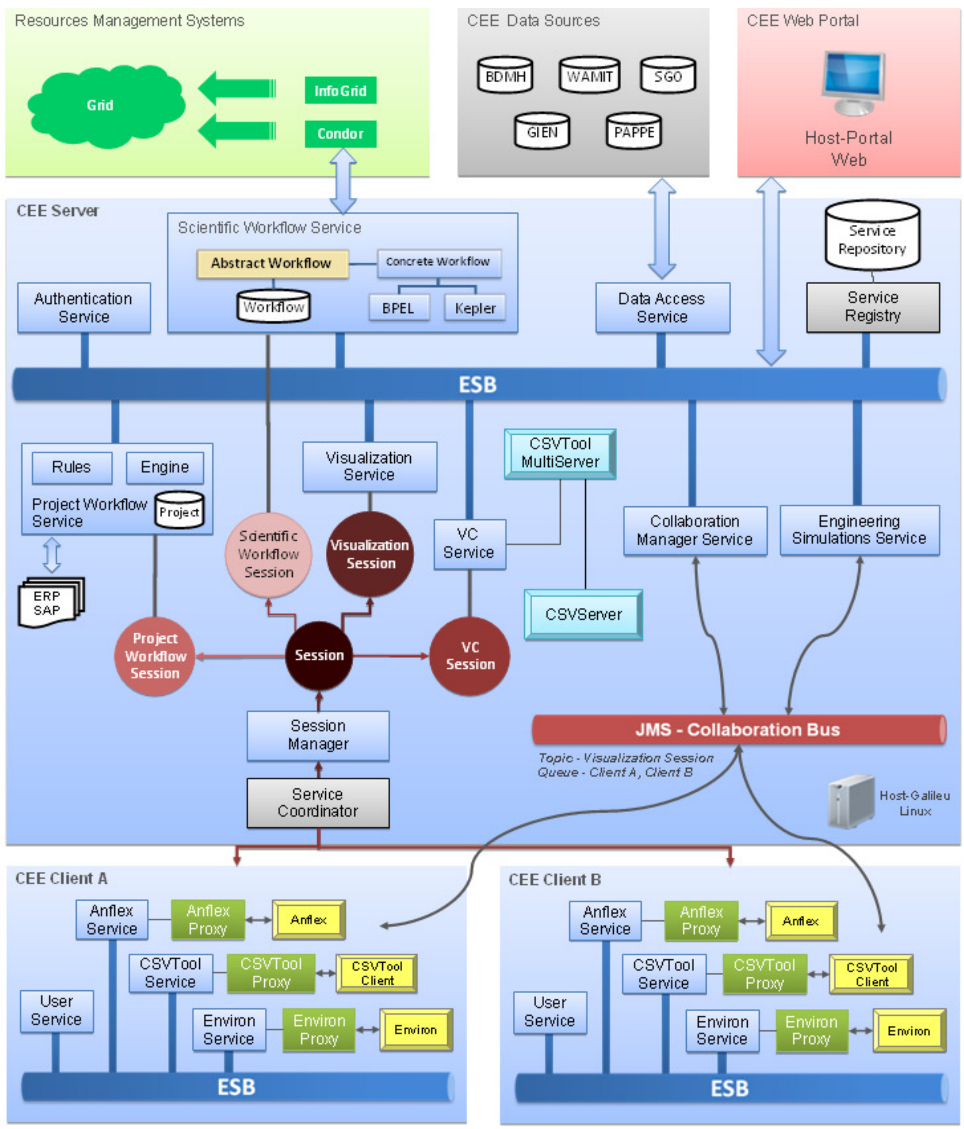


Figure 5.2: CEE SOA state of execution of a Riser Analysis Workflow.

5.1.2. Video Conferencing

Figure 5.3 shows a collaborative visualization session with the presence of two users, represented by two distinct 3D-cursors, visualizing the simulation results in their desktop with the support of a Videoconference using the *CSVTool*.

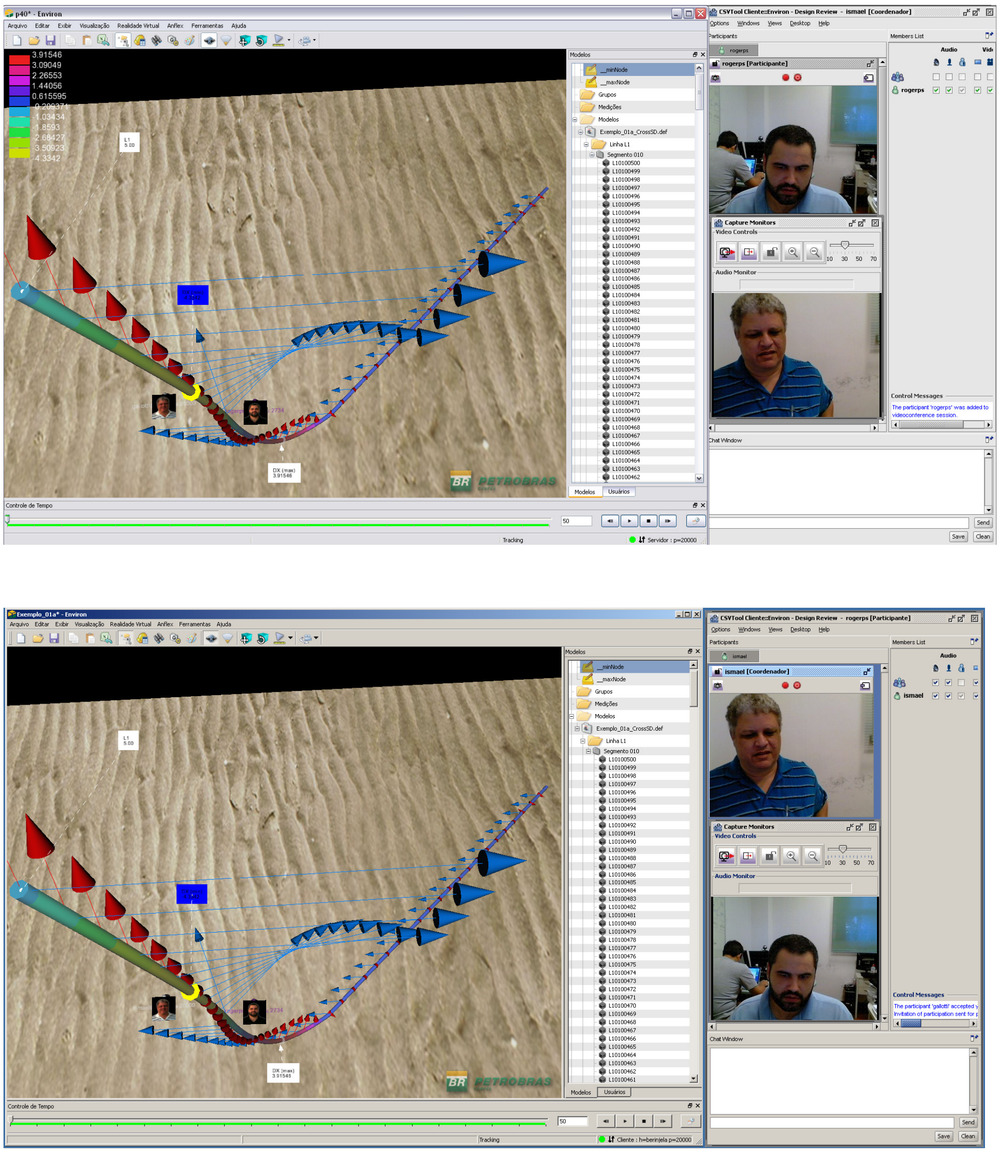


Figure 5.3: Riser Analysis in CEE (Environ + CSV Tool).

The blue arrow represents the water currents that actuate over the riser, while the red arrow represents the direction of the movement of the riser (i.e. instantaneous velocity). Observe that the greater the alignment of those two groups of arrows the greater the influence of the water currents in the final movement of the riser. For that situation we can see that there is no such alignment, which means that other environmental forces (winds and waves) have a greater influence in the final movement of the riser.

In the first picture appears the coordinator's desktop, user ismael, while in the second appears the participant's desktop, user rogerps. Note that, the coordinator receives a video stream from the participant, while in the second picture the participant receives the image of the coordinator. This way the efficiency of the collaboration is dramatically improved due to the user Awareness obtained by the use of CSVTool, where the users can see each other and also receive a screen copy of the remote user desktop. Each user is represented by its own avatar associated with the position of its telepointers. The transmission of the desktop image among users in same times is very important, especially when a user wants to show input parameters of an application or wants to teach how to make an operation to the other user.

5.1.3. 3D Annotations

Environ has special capabilities to show the extreme values and where are they located. 3D Annotations can also be created by the users. In Figure 5.3 two 3D annotations were created automatically by the Environ, showing the extreme points (maximum and minimum values) of a selected force or strength in the riser. The third 3D annotation was created by one of the users to register some important observation made in this collaborative session.

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 subjected to great stress, such as the joints connection and the TDP (Touch Down Point). In Figure 5.4, the users are looking closely to the behavior of a selected element in the riser (green ring), they can also follow the movements of the element playing the simulation on the timeline bar.

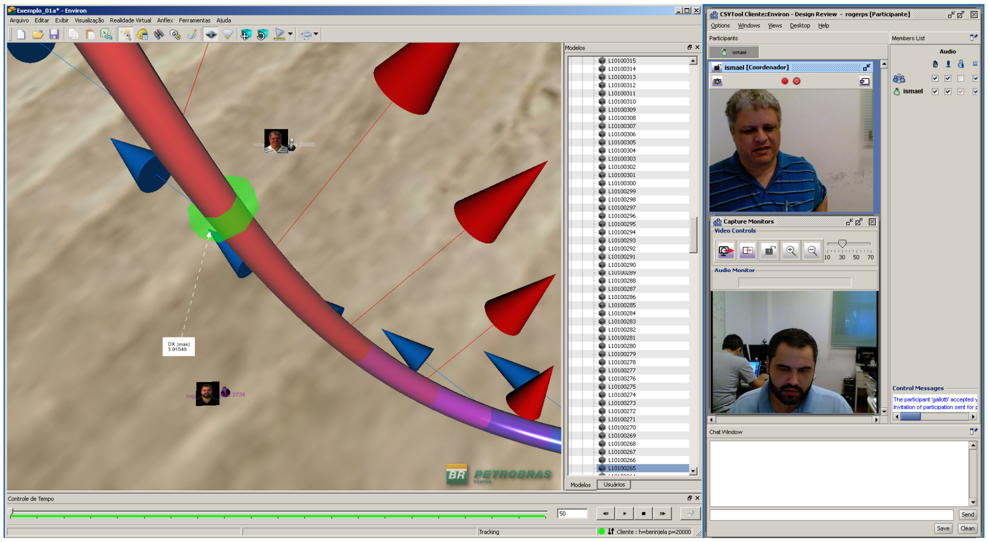
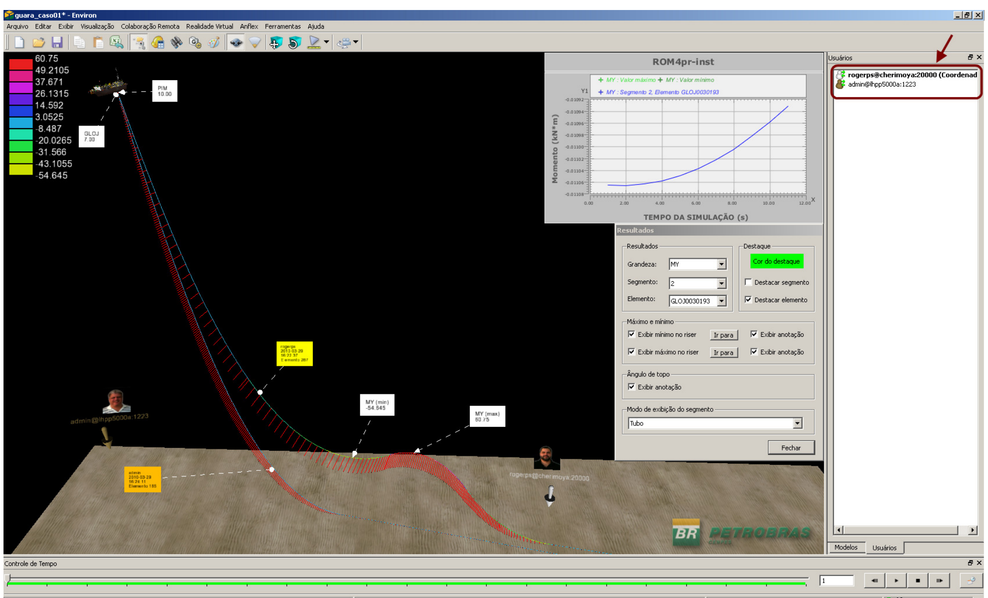


Figure 5.4: Closer look on an element of the riser

At the end of the session both users will have all the information attached to the model. This information represents the state of the collaborative visualization session and can be persisted in a file that can be loaded again in the future to reconstitute the scenario that was analyzed.



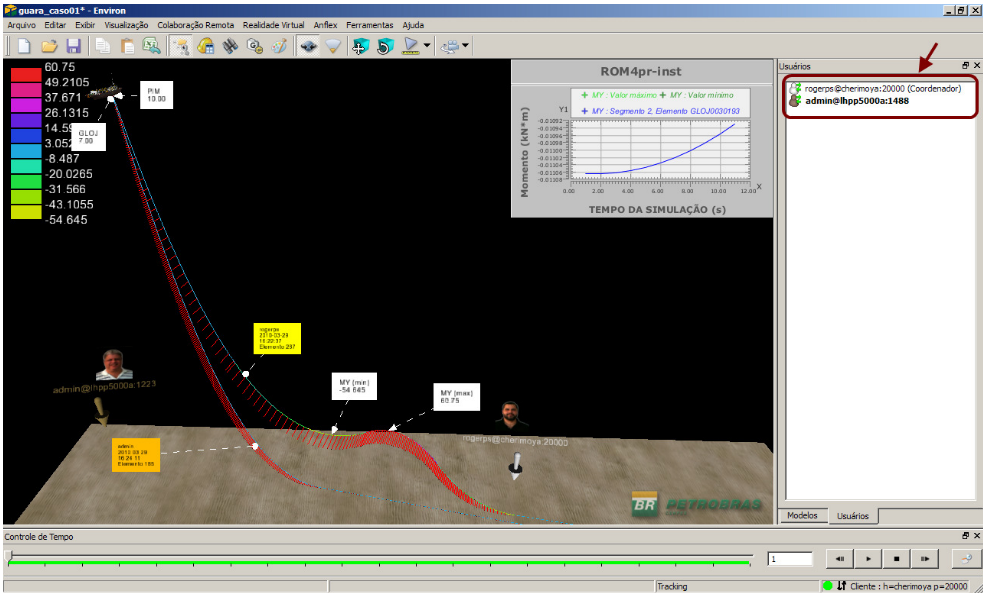


Figure 5.5: Two users in a CEE collaborative visualization session.

Figure 5.5 shows another collaborative visualization session now with another set of risers. Those pictures show the white 3D annotations created automatically by Environ, and two other annotations created by each user, making comments about different elements in those two risers.

Observe that in the users tab, we have the awareness mechanism showing information about the status of the user (online, offline) and its role in the session (coordinator or participant).

5.1.4.
3D Measurements

In Figure 5.6, we show a sequence of measurements created by the users, in this case was a 3D distance between two distinct points on the risers. The sequence of images shows the animation that can be seen by the users when they want to monitor that distance. In riser analysis this is important because the engineers have to avoid collisions between risers in order to preserve structurally the risers during their entire life (usually 20 to 30 years).

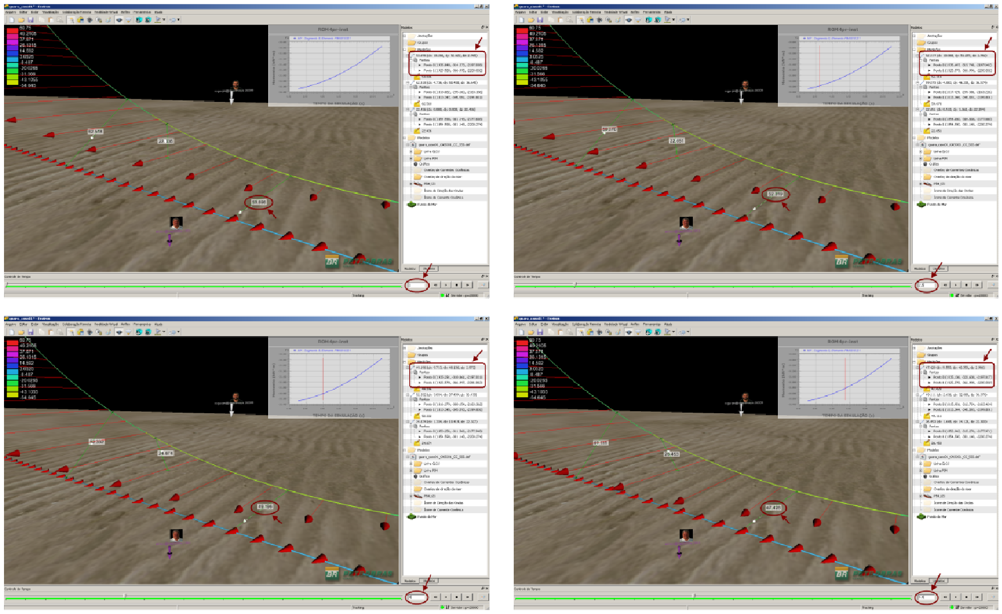
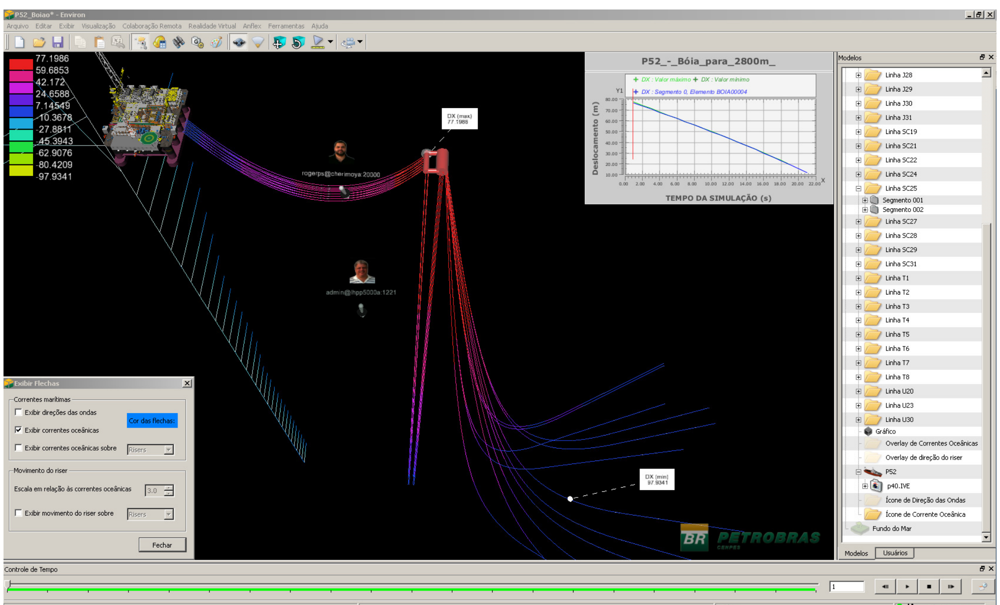


Figure 5.6: Measurments in a visualization session



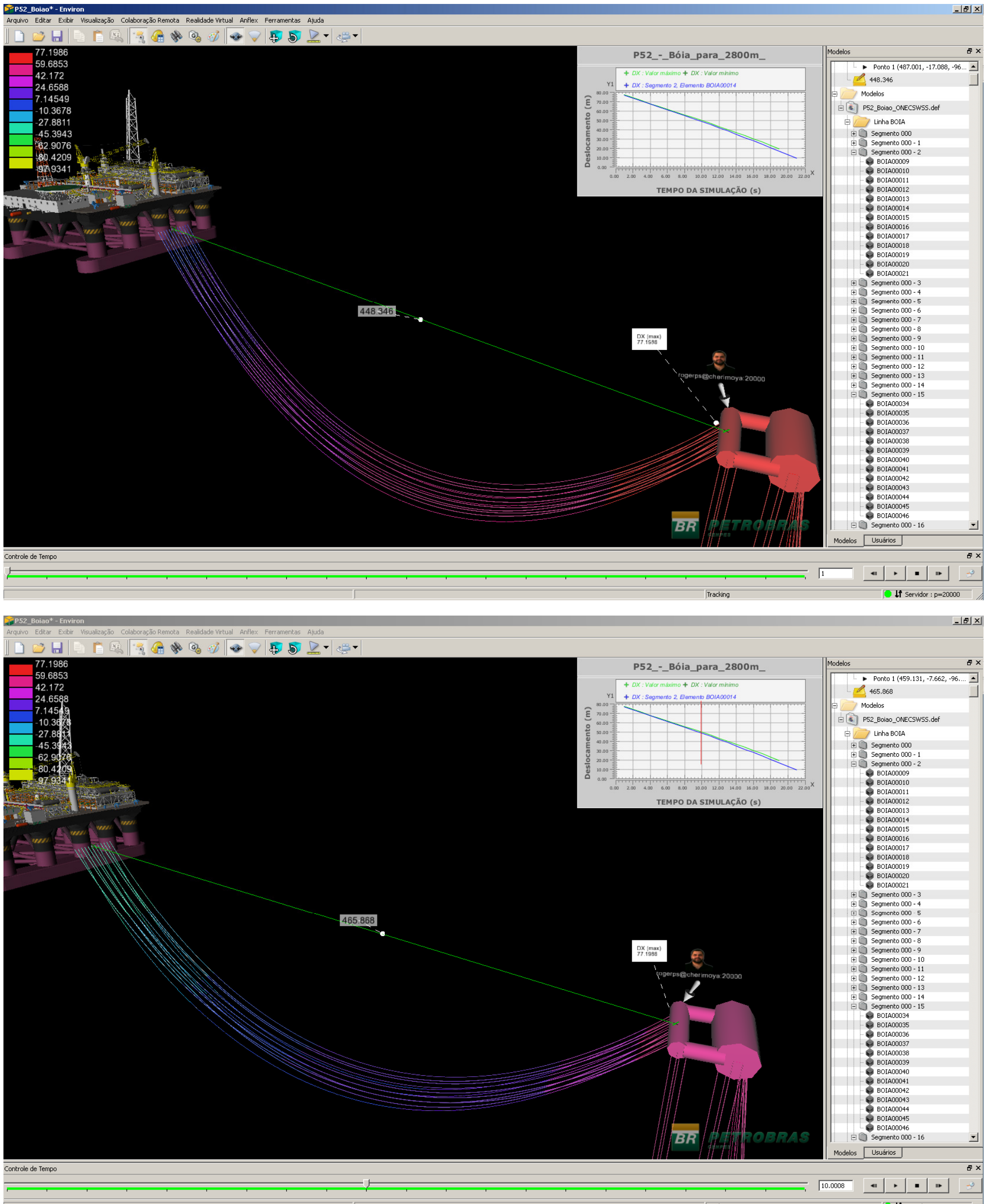


Figure 5.7: Users monitoring the behavior of marine buoyant

In Figure 5.7, we show another engineering project where the users want to study the movement of a buoyant, the usage of the buoyant is to reduce the stress that are submitted the risers especially when there is a great fluctuation in the platform movements due to strong environmental conditions (wave and winds). Through the use of the buoyant we can decouple the movement of the platform hull and the movement of the riser system. In the sequence of figures engineers is monitoring the distance between the buoyant and the platform and also are observing the behavior of some force on the risers.

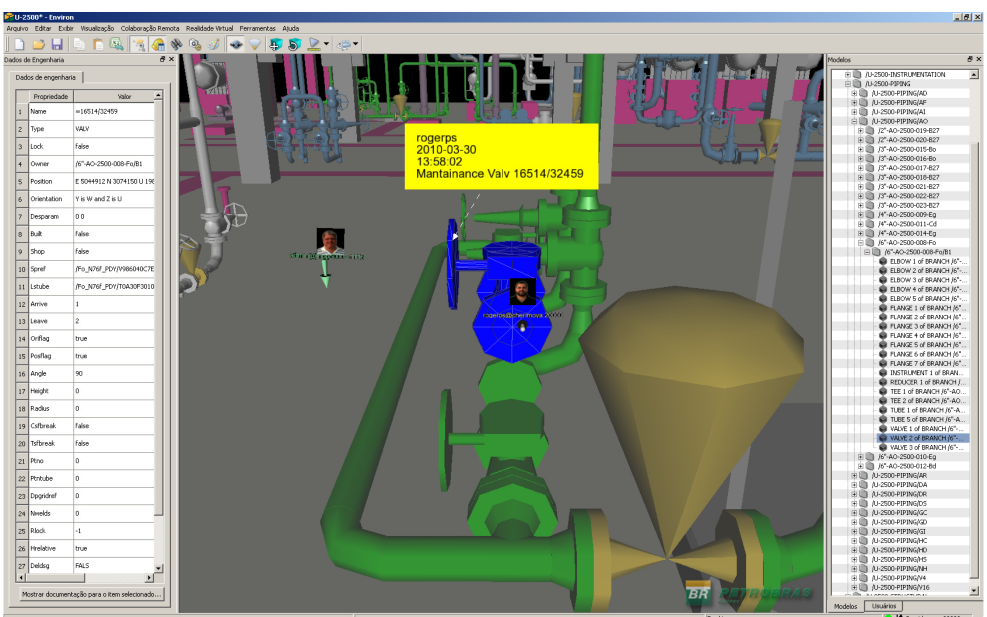
5.2. Design Review Workflow

Design review is the process of checking the correctness and consistency of an engineering project, and making the necessary corrections to it. CEE is very helpful in this process, for instance to assess the safeness of different emergency escape pathways in case of an emergency in the plant.

The Design Review workflow is a simplified version of the riser analysis workflow, where BPEL engine invokes *Service Coordinator* to create a *Collaborative Visualization Session* with the support o *VC Session* according to the user's choice. In this session the users manipulate engineering artifacts and create 3D annotations and make 3D measurements in the model.

Object manipulation is an important resource in design review. The ability of moving, rotating and scaling objects is important for various purposes such as joining different models in a scene, viewing hidden portions of the model, planning the placement of a new piece of equipment on a plant, and simulating a maintenance or intervention operation in a process plant are also valuable tools. As an example, the maintenance plan can be enriched with a detailed sequence of operations with annotations carefully chosen to be presented as an animation for the maintenance engineers during the operation (

Figure 5.8). Moreover, integration with a database is useful to allow user to create annotations on the model emphasizing critical parts. It is also possible to show comments attached to objects, which can be used, for example, as recommendations for project management.



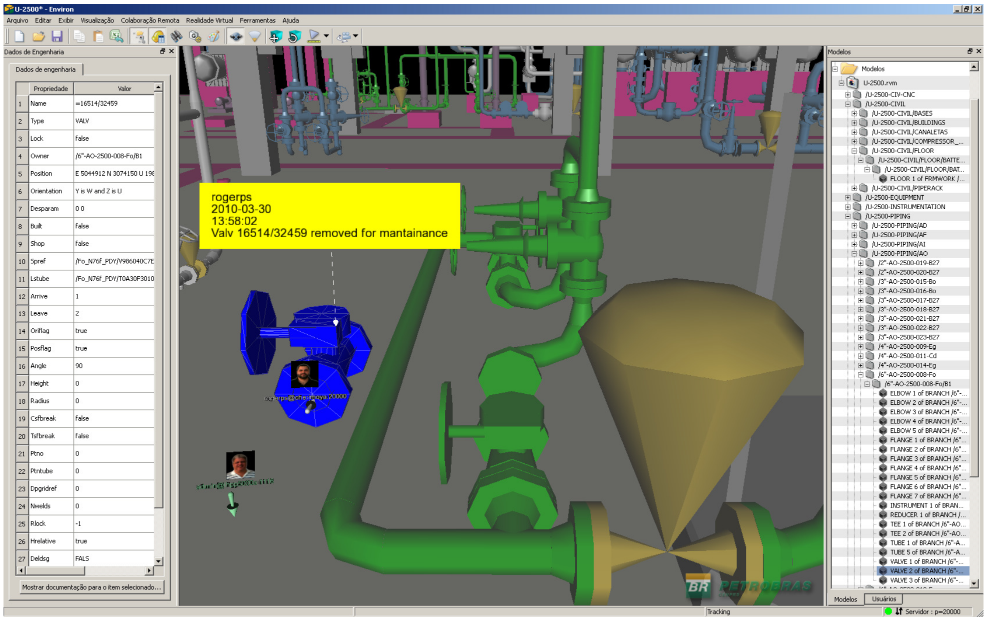


Figure 5.8: Maintenance plan enriched with annotations.

Figure 5.9 shows some measurements taken for planning the movement of a large tank on production unit. The users creates a lot of 3D annotations to guide the maintenance process.

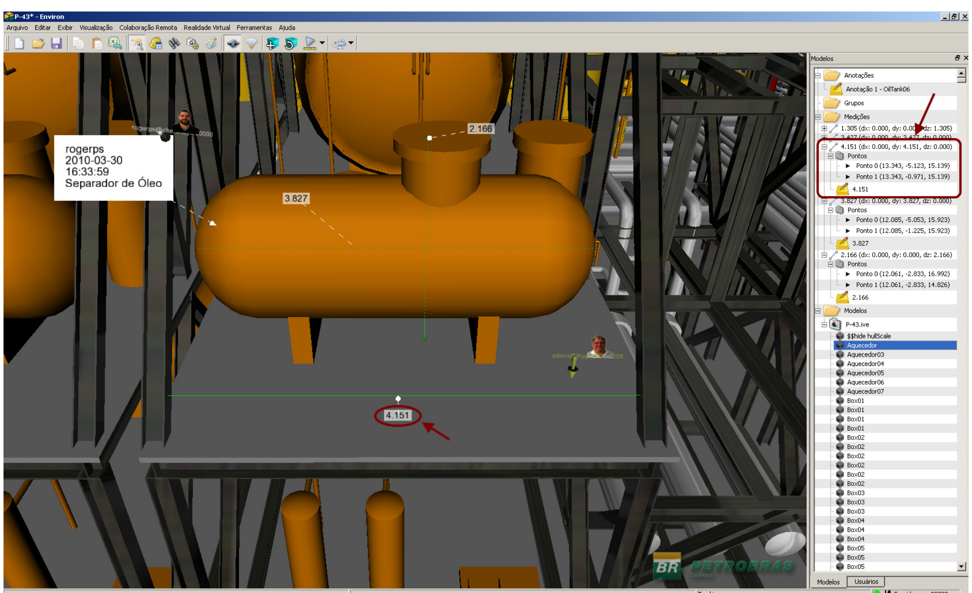


Figure 5.9: Measurements in a CAD.

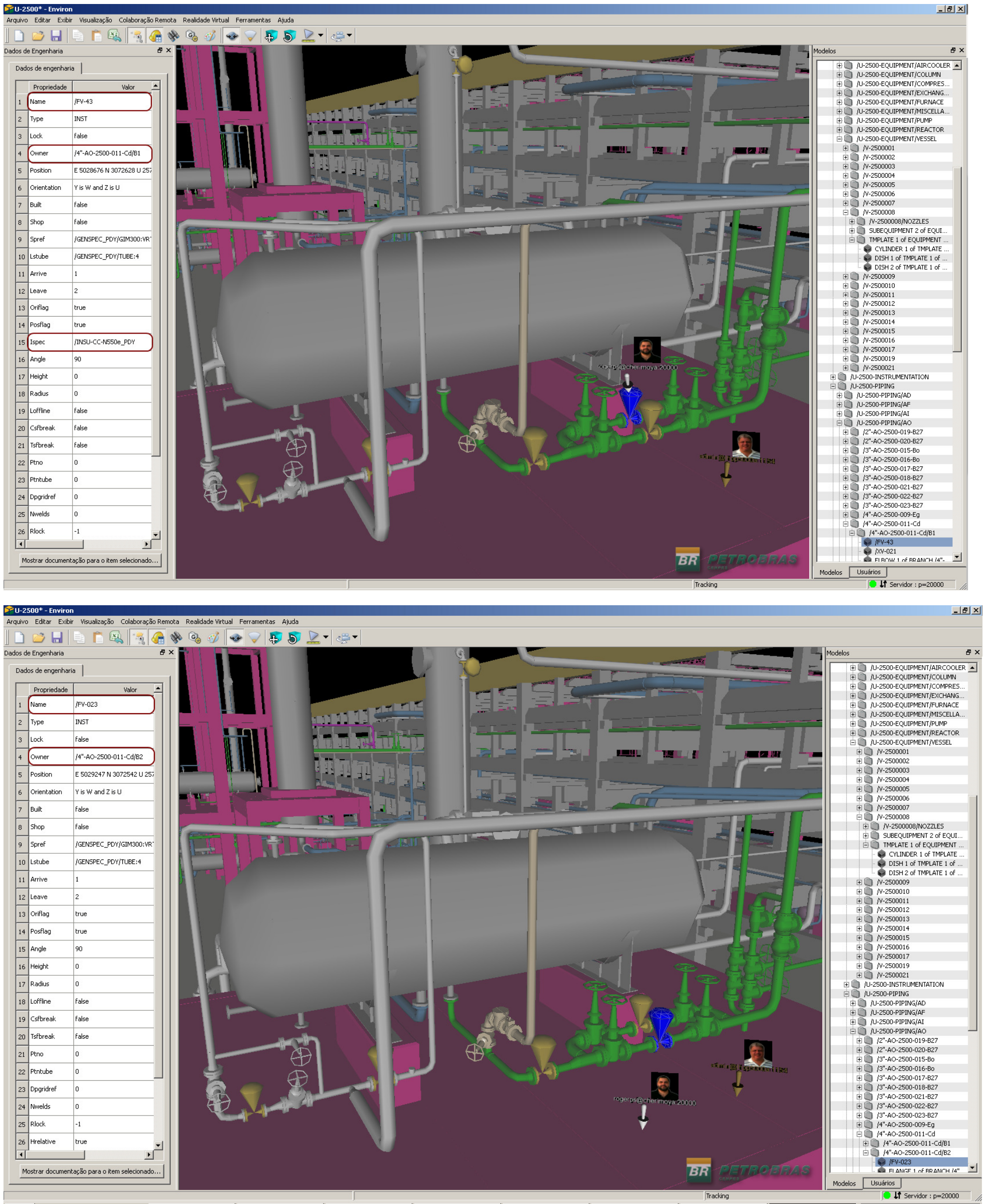


Figure 5.10: Engineering information.

Another important aspect in design review is the integration of the visualized model with project information. Several CAD models have technical information attached to each object. Using database resources it is possible to recover this information in real time and use it to help the user taking operational decisions. Figure 5.10 shows a simple window with information on a gas tank. It is possible to know what is the pressure that is actuating in a valve once there is a connection with an real time measurement system, such as ECOS-PI, very well known in the oil industry.

5.2.1. Virtual Guided Tour

The final interesting tool in Environ is the *Virtual Guided Tour*, where a user follows the movements of another user, sharing the same view of the model. Figure 5.11 shows another collaborative visualization session now on a platform. The pictures show both users following a 3D path passing through important points and at the end an annotation is created to mark some important event on the platform, maintenance or commission of new equipment could be programmed.

As we mentioned in Chapter 3, we have three types of sessions (Informal, Classroom and Lecture). So for each type of session the user has a status that is determined according to its role (coordinator or participant). In a Virtual Guided Tour the coordinator is in a state that he can only send camera movements and must ignore any camera movements from the other users (SendOnly), while all the other participants are in a state that they can only receive commands and cannot send any camera movements (ReceiveOnly).

In the first image both users are in a SendAndReceive state, because we are in an Informal collaboration session. The awareness mechanism shows the icons of each user with 2 green lights, one for input and another for output. When changing to a Classroom or Lecture collaborative session the state of the coordinator and participant changes accordingly, as shown in the picture of Figure 5.11. Observe that the awareness mechanism changes the icons of each user accordingly, the coordinator change its state to SendOnly (upper arrow green, lower arrow red) while the participant change its state to ReceiveOnly (upper arrow red, lower arrow green).

In Environ there is also a possibility of requesting the coordinator role, but only when we are in a Classroom collaborative session. When in a Lecture session this possibility is forbidden by the definition of a Lecture, see Chapter 3.

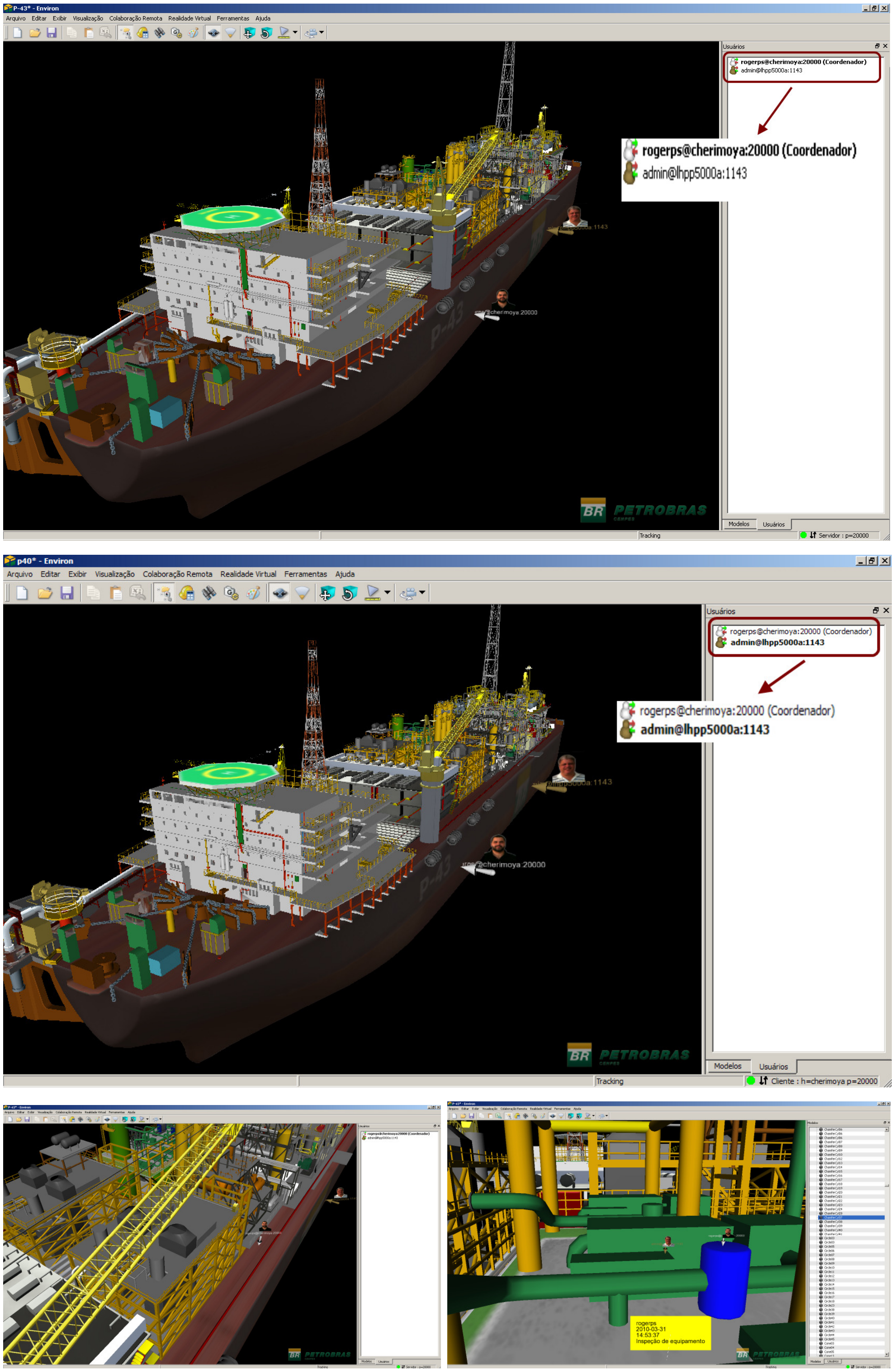


Figure 5.11: Virtual Guided Tour

6
Conclusions and Future Work

6.1.
Conclusions

This thesis presented the conceptualization (the SOA architecture) and implementation of a CPSE we devised for OE projects. As a proof of concept we have developed CEE, a collaborative environment to optimize the execution of Large Engineering Projects such as Offshore Engineering projects developed at Petrobras. Through the use of the CEE we have build an effective collaborative environment that allow users to easily mitigate their problems that usually happen during the execution of large and complex engineering projects. We also believe that the goal of creating a user-friendly workflow system operating in a collaborative environment was achieved.

Upon the integration of VR technologies into the workflow of the teamworkers we expect to improve the use of VR in OE projects, which is unfortunately just used in very few areas of Petrobras nowadays. It is clear that visualization resources improve the quality of engineering projects as we have pointed out in Chapter 2, but users do not want to spend their time preparing the content to be visualized in other system, like an immersive multi-projection environment. In this concern CEE is already showing its value, upon simplifying the daily job of the engineers, from running simulations on a Grid through visualizing its results on an immersive environment or on a desktop. Now everything can be done through the *CEE Portal* just accessing a web page and sending commands to CEE server.

In addition, we argue that SOA offers E&P companies a number of compelling benefits. It allows organizations to be able to respond efficiently to changes in the business and competitive landscape, preservation of legacy system while enhancing integration, lower technology development costs by leveraging functions already built into legacy system services, by re-using services developed for other process, and by simplifying maintenance and support through elimination of redundant applications.

We believe that the main contribution of this thesis is the junction of approaches and technologies from different areas composing a CPSE suitable for LSEP (more specifically, OE projects), with distinguishable characteristics, when compared to similar systems as was pointed out at the end of Chapter 2.

From the OE point of view, the introduction of a Scientific Workflow in the project life cycle and the use of a CPSE are important contributions in the sense of providing a more structured way to solve the problems and the creation of tools more widely used.

From the VR and Visualization point of view, CEE approach treats them as first class tools, exploring their potential for facilitating information exchange and common understanding of complex problems. It was not possible to find any other approach complete as presented here in the academic literature or in any oil & gas company in the world.

The perspectives for the future is that many other organizations are going to start to use Scientific Workflows and this will become a common solution in high complex enterprises that have several areas that must be integrated and synchronized.

Although this work is focused on a solution for Offshore Engineering projects, we believe that the proposed CEE could also be used in other areas. There are many important Petroleum Engineering activities ([SCL+01], [RRF+04]) that would benefit with the implementation of the CEE as defined in this thesis, such as:

- Collaborative real-time visualization, walkthrough and fly-over offshore facilities modeled with massive CAD models;
- Project of ultra-deep water riser and mooring systems;
- Oceanographic model visualization;
- Stability analysis of oil platforms;
- Controlling and monitoring of the construction process of large production units. Estimation of the progress based on the differences between the captured 3D “as-built” model and the "as-planned";
- Design, planning and optimization of marine installations and sub-sea layout arrangement of production equipments;
- Simulation and evaluation of the performance of remote teleoperations and interventions on submarine equipments in deep waters;

- Change Management Planning Systems - execution of interference and visibility checks to elaborate maintenance and inspection plans for production units;
- Planning oil pipeline installation and monitoring;
- Training and safety simulations applied to emergency scenarios;
- Integrated Oil Reservoir Management;
- Immersive Well-path planning and Geosteering;

6.2. Future Work

There are a lot of important future works that could be developed in many directions. Considering the CEE SOA architecture and its components we can propose future works for each environment of CEE, the *Project Management Environment,* the *Scientific Workflow Environment* and the *Collaborative Visualization Environment*. The following subsections present the current main ideas.

6.2.1. Project Workflow Environment

For this environment we mainly suggest as future works the topics that we have not addressed in this thesis for this environment.

- *Introduction of Project Management System with an Integrated Data Management System tor control all documents and artifacts generated during the project lifecycle ;*
- *Close and functional integration with ERP systems, like SAP, providing better accountability of resources when executing engineering simulations;*

6.2.2. Scientific Workflow Environment

For this environment we mainly suggest as future works the topics that we have not addressed in this thesis, which are Data Provenance and Ontology.

- *Implementation of the Open Provenance Model [OPM] for BPEL workflows;*

- *Usage of an Ontology inference engine to help the creation of new workflows in the context of Offshore Engineering;*
- *Support for data replication. once CEE supposes that the data is already replicated or centralized there is no mechanism for data distribution in CEE;*

6.2.3. Collaborative Visualization Environment

In the visualization area, there are also many additional resources that may be addressed in the context of CEE, such as:

- *Multimedia Communication System with a more powerful tools, such as instant messages, better audio / video tools, etc;*
- *Agent Based system for improving collaboration among users during a collaborative visualization session;*

We believe that CEE is a step towards a new frontier in CPSE, which is the use of a computation steering approach in tele-immersive CPSE (Figure 2.1).

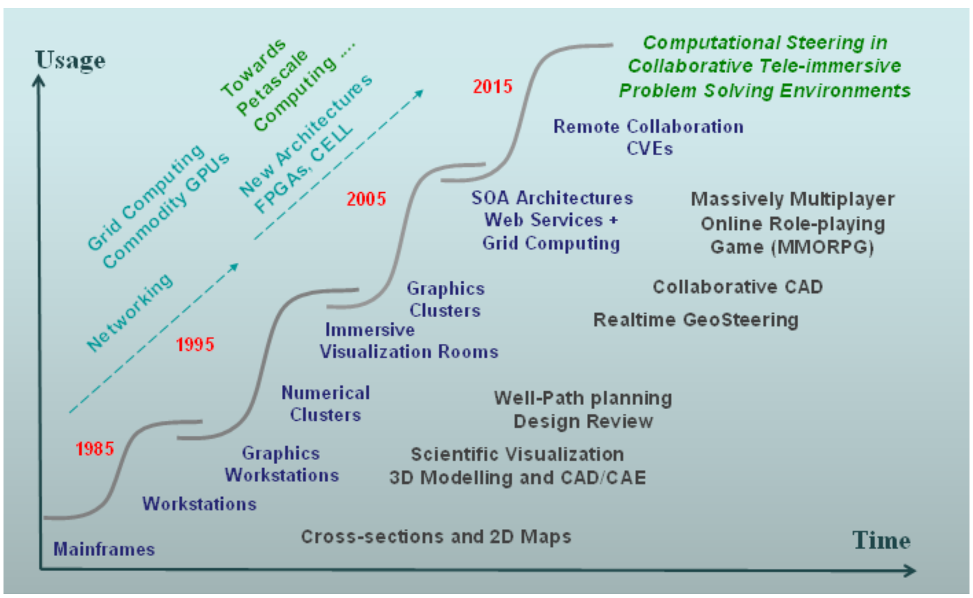


Figure 6.1: Towards tele-imersive CPSEs.

Following this line of thought, two additional lines of research can be envisioned for CEE:

- *Implementation of Computational Steering approach*

Computational steering is the practice of manually intervening with an otherwise autonomous computational process, to change its outcome. Abstractly, we can think of it as an API for Interactive Application Control, furnishing interesting tools for data exploration visualization such as:

- Modify Parameters While (Long) Running – helps to eliminate wasteful cycles of ill-posed simulation. Drive simulation to more interesting solutions; enhance convergence of Numerical Algorithms;
- Allows "What If" Explorations – closes loop of standard simulation cycle. explore non-physical effect;

The steering approach is a very valuable feature for any CPSE for Science and Engineering.

- *Tele-immersive CPSE*

Tele-immersion is a technology that will enable users in different geographic locations to come together in a simulated environment to interact. Users will feel like they are actually looking, talking, and meeting with each other face-to-face in the same room. This kind of environment will improve collaborative work, which is essential for a CPSE like the proposed CEE in this thesis.

In the oil & gas industry projects for building inhabited production plants would have a great benefit for such a tele-immersive CPSE.

7 Bibliography

[ActiveMQ] ActiveMQ, opensource JMS Provider - http://activemq.apache.org/

[BBB+93] K. Brodlie, L. Brankin, G. Banecki, A. Gay, A. Poon, and H. Wright. "Grasparc – a problem solving environment integrating computation and visualization". *In Proceedings of IEEE Visualization '93,* pages 102-109. IEEE Computer Society Press, 1993.

[EclipseBPEL] Eclipse BPEL Project. http://www.eclipse.org/bpel/

[BR92] Alan H. Bond & Richard J. Ricci. "Cooperation in aircraft design". *Research in Engineering Design*, 4(2): 115-130. 1992. Springer.

[BFD+03] C. A. M. Barbosa, B. Feijó, M. Dreux, R.N. Melo, J. Bento and S. Scheer. "An Object Model for Collaborative CAD Environments". *Journal of Integrated Design and Process Science*, V7, N2, 2003.

[BH+92] R. Bentley, J.A. Hughes, D. Randall, T. Rodden, P. Sawyer, D. Shapiro, I. Sommerville. "Ethnographically informed systems design for air traffic control". *In The Proc. of the ACM Conf. on CSCW'92*, Ontario, CA, 1992.

[BPMI] Business Process Management Initiative. http://www.bpmi.org/

[CDS02] P. Christianson, N. Dawood, K. Svidt. "Virtual Building (VB) and Tools to Manage Construction Process operations". *Internat Conference on Distributing Knowledge In Building, CIB W78*, June 2002.

[CFS+06] S. P. Callaban, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, H. T. Vo. "VisTrails: Visualization meets Data Management". *In the Proceedings of the 2006 ACM SIGMOD international conference on Management of data.* Chicago, IL, USA, 2006.

[CGF+00] L. C. G. Coelho, M. Gattass, L.H. Figueiredo. "Intersecting and trimming parametric meshes on finite element shells". *Intern Journal Num. Methods Eng,* vol 47, pp 777-800, 2000.

[Condor] Condor web site – High Throughput Computing. http://www.cs.wisc.edu/condor/

[CJO+03] L. C. G. Coelho, C. G. Jordani, M. C. Oliveira, I. Q. Masetti. "Equilibrium, Ballast Control and Free-Surface Effect Computations Using The Sstab System". *8th Int. Conf. Stability of Ships and Ocean Vehicles - Stab*, pp 377-388, 2003.

[CKM+01] L. C. G. Coelho, K. Nishimoto, I.Q. Masetti. "Dynamic Simulation of Anchoring Systems Using Computer Graphics". *OMAE Conference,* 2001.

[CS+92] C. Cruz-Neira, D. J. Sandin, T.A. DeFanti, R.V. Kenyon, J.C. Hart. "The CAVE: Audio visual experience automatic virtual environment". *Communications of ACM, 35-6, 1992.*

[CSD+93] C. Cruz-Neira, D. J. Sandin, T.A. DeFanti. "Surroundscreen projection-based virtual reality: The design and implementation of the CAVE". *In Proceedings of SIGGRAPH 93, Computer Graphics Proceedings, Annual Conference Series, 135–142*, *1993.*

[DB06] J. Dunbar, W. Brooks, NASA Problem Solving Environments: Distributed Collaborative Virtual Wind Tunnel. http://www.nas.nasa.gov/Research/Tasks/probsolvingtasks.html

[DB92] Dourish, P., Bellotti, V. "Awareness and coordination in shared workspaces". In Proceedings of the 1992 *ACM Conference on Computer-Supported Cooperative Work (CSCW '92)*, ACM Press, 107-114, New York, NY, 1992.

[Dew99] P. Dewan. "Architectures for collaborative applications". *Computer Supported Co-operative Work*, M. Beaudouin-Lafon.

[DGS+09] E. Deelman, D. Gannon, M. Shields, I. Taylor. "Workflows and e-Science: An overview of workflow system features and capabilities", *Future Generation Computer Systems,* v25, p.528-540, 2009.

[DHJ+00] M. Daily, M. Howard, J. Jerald, C. Lee, K. Martin, D. McInnes, P. Tinker. "Distributed Design Review in Virtual Environments". In the Proceedings of the 3rd ACM International Conf. CVE'00, San Francisco, CA, USA, 2000.

[Dodd04] E. J. Dodd. "Visualization and Collaboration for the On Demand Upstream Petroleum Enterprise". *IBM technical reports*, Toronto, Canada, May*,* 2004*.*

[Dus00] S. Dustdar. "Multimedia computer-supported cooperative work". *Multimedia*, *IEEE, October,* 2000*.*

[EAA+04] M. Endrei, J. Ang, A. Arsanjani, et al. "Patterns: Service-Oriented Architecture and Web Services", IBM RedBooks, April 2004. http://www.ibm.com/redbooks.

[Edw94] W. K. Edwards. "Session management for collaborative applications". *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (CSCW '94), ACM Press pp 323-330, New York, NY, 1994.

[Edw96] W. K. Edwards. "Polices and Roles in Collaborative Applications". *Proceedings of the ACM Conference on Computer Supported Cooperative Work,* 11-20, 1996.

[EG89] C.A. Ellis, S. J. Gibbs. "Concurrency Control in Groupware Systems". *SIGMOD Conference*, vol. 18, 399-407, 1989.

[EGR91] C.A. Ellis, S. J. Gibbs, G. L. Reins. "Groupware: Some Issues and Experiences". *Communications of the ACM*, vol 34, no. 1, January 1991.

[Ehr99] K. Ehrlich. "Designing Groupware Applications"*. M.Beaudouin-Lafon, (ed.): Computer Supported Co-operative Work, Vol. 7, Trends in Software. John Wiley & Sons, 1-28, 1999.*

[Ellis99] C.A. Ellis. "Workflow Technology". *Computer Supported Co-operative Work, M. Beaudouin-Lafon, Ed., vol. 7 of Trends in Software. John Wiley & Sons, 29-54, 1999.*

[EN96] C. A. Ellis, G. J. Nutt. "Workflow: The Process Spectrum"*. NSF Workshop on Workflow and Process Autom. Info. Systems, Athens, GE, 1996.*

[ESW03] e-Science Workflow Services Workshop, *Edinburgh, Scotland, 2003.* http://www.nesc.ac.uk/events/303/index.html.

[EW94] C. A. Ellis, J. Wainer: "A Conceptual Model of Groupware". *Proceedings of the Conference on Computer Supported Cooperative Work* (CSCW'94), ACM Press, 79-88, 1994.

[FB+99] B. Fröhlich, S. Barrass, B. Zehner, J. Plate, M.Goebel. "Exploring GeoScience Data in Virtual Environments". *In Proc. IEEE Visualization*, 1999.

[FK99] I. Foster, C. Kesselman (eds). "Globus: a toolkit based grid architecture". *Morgan Kaufman, San Francisco, pp 259–278,* 1999.

[FKN+01] I. Foster, C. Kesselman, J. Nick, S. Tuecke. „The Anatomy of the Grid: Enabling Scalable Virtual Organizations". *International Journal Supercomputing Applications, 2001*.

[FKN+02a] I. Foster, C. Kesselman, J. Nick, S. Tuecke. "The physiology of the Grid: An Open Grid Services Architecture for distributed systems integration". *Technical report, Open Grid Services Architecture WG, Global Grid Forum, 2002*. http://www.globus.org/research/papers/ogsa.pdf

[FKN+02b] I. Foster, C. Kasselman, JM Nick, S. Tweck. "Grid services for distributed systems integration". *IEEE Computer vol. 36 (6), pp. 37-46*, 2002.

[FRG+05] H. Fuks, A. B. Raposo, M. A. Gerosa, C. J. P. Lucena. "Applying the 3C Model to Groupware Development". *International Journal of Cooperative Information Systems (IJCIS),* v.14, n.2-3, Jun-Sep 2005, World Scientific, ISSN 0218-8430, pp. 299-328, 2005.

[FTF+01] J. Frey, T. Tannenbaum, I. Foster, M. Livny, S. Tuecke. "Condor-G: a Computation Management Agent for multi-institutional grids". In Proceedings of the 10th IEEE Symposium on High Performance Distributed Computing (HPDC10). IEEE Press, New York, 2001.

[GFF+05] R. Gregovic, R. Foreman, D. Forrester, J. Carroll. "A Common Approach to Accessing Real-time Operations Data: Introducing SOA to E&P", *SPE Annual Technical Conference and Exhibition (ATCE)*, October 2005.

[GHR94] E. Gallopoulos, E. N. Houstis, J. R. Rice. "Computer as thinker/doer: Problemsolving environments for computational science". *IEEE Computational Science and Engineering, vol* 1 pags 11-23, 1994.

[GLG03] G. Goebbels, V. Lalioti, M. Göbel. "Design and Evaluation of Team Work in Distributed Collaborative Virtual Environments". *The Proc. of the ACM Symposium on Virtual Reality Software and Technology, VRST'03, Osaka, Japan, 2003*.

[Gocad] The GOcad consortium. http://www.gocad.org/

[GR99] S. Greenberg, M. Roseman. "Groupware toolkits for synchronous work". *Computer Supported Co-operative Work*, M. Beaudouin-Lafon, Ed., vol. 7 of Trends in Software. John Wiley & Sons, 135-168, 1999.

[GridSAM] GridSAM - Grid Job Submission and Monitoring WebService http://www.omii.ac.uk/wiki/GridSAM

[GridOracle05] Grid Computing with Oracle. Thecnical Paper at http://www.oracle.com/grid

[Gru04] K. Gruchalla. "Immersive well-path editing: investigating the added value of immersion". *In Proceedings IEEE Virtual Reality, VR04, 157–164, Chicago, USA, 2004.*

[HDF] HDF Group. HDF5 File Format Specification Version 2.0. Available at http://www.hdfgroup.org/HDF5/doc/H5.format.html.

[HGC02] M. Hofmam, M. Gattass, W. Celes. "Organization of Large Engineering Models for Real-Time Walkthrough Rendering". *The Intern. Conf. on Comp. Engineering & Sciences, ICES, 2002.*

[HGB+97] E.N. Houstis, E. Gallopoulos, R. Bramley, J.R. Rice. "Problem-Solving Environments for Computational Science". *IEEE Computational Science and Engineering,* 4(3): 18-21, 1997.

[HHV+99] M. Haller, R. Holm, J. Volkert, R. Wagner. „A VR based safety training in a petroleum refinery". In The 20th Annual Conf. of the European Association of CG, Eurographics'99, Milano, Italy, 1999.

[HKG+05] R. High, S. Kindler, S. Graham. "IBM SOA Foundation: An architectural introduction and overview". IBM developer Works, Nov, 2005. http://www-128.ibm.com/developerworks/webservices/library/ws-soa-whitepaper/

[IFC96] R. Ierusalimschy, L. Figueiredo, W. Celes. "Lua – an extensible extension language". *Software: Practice and Experience*, 26(6), 1996.

[Ish99] H. Ishi. "Integration of shared workspace and interpersonal space for remote collaboration". *Computer Supported Co-operative Work*, M. Beaudouin-Lafon, Ed., vol. 7 of Trends in Software. John Wiley & Sons, 83-102, 1999.

[IT94] E. A. Isaacs, J. C. Tang. "What Video Can and Cannot Do for Collaboration: A Case Study", *Multimedia Systems*, Vol. 2, 63-73, 1994.

[JE94] B. P. Jacob, N. F. F. Ebecken. “An Optimized Implementation of the Newmark/Newton-Raphson Algorithm for the Time Integration of Nonlinear Problems”. *Communications in Numerical Methods in Engineering,* vol. 10 pp. 983-992, John Wiley & Sons, UK/USA, 1994.

[JEESun] Sun Microsystems (2009) Java Enterprise Edition Home Page. Available at: http://java.sun.com/javaee/technologies/

[JE98] G. G. Johnson, T. T. Elvins. “Introduction to collaborative visualization”. *SIGGRAPH Comput. Graph*. 32, 2, 8-11, 1998.

[Joeris97] G. Joeris. “Cooperative and Integrated Workflow and Document Management for Engineering Applications”. In 8 *th* Intern. Workshop on Database and Expert Systems Applications, DEXA, 1997.

[JMFSun] Sun Microsystems (2003) Java Media Framework Home Page. Available at: http://java.sun.com/products/java-media/jmf/

[KMC02] A. Kapolka, D. McGregor, M. Capps. “A Unified Component Framework for Dynamically Extensible VEs". *The Proc. of the 4 th ACM Intern. Conf. on CVE’02, Bonn, Germany, 2002.*

[KBB+06] O. Kreylos, T. Bernardin, M. I. Billen, et alli. “Enabling Scientific Workflows in Virtual Reality”. *ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry, VRCIA’06*, CUHK, Hong Kong, June 2006.

[KBG+06] H. Klie, W. Bangerth, X. Gai, M.F. Wheeler, P.L. Stoffa, M. Sen, M. Parashar, U. Catalyurek, J. Saltz, T. Kurc. *Models, Methods and Middleware for Grid-enabled Multiphysics Oil Reservoir Management. Engineering with Computers*, Springer Verlag, 2006.

[KBW+04] H. Klie, W. Bangerth, M. Wheler, M. Parashar, V. Matossian. “Parallel well location optimization using stochastic algorithms on the Grid computational framework”. *In Proceedings of the 9th European Conference on the Mathematics of Oil Recovery (ECMOR IX),* 2004.

[Kepler] Kepler Project. http://kepler-project.org/

[LandMark] LandMark Geological and Geophysical Technologies. http://www.lgc.com/landmark/integrated+solutions/index.htm.

[LAB+06] B. Ludscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger-Frank, M. Jones, E. Lee, J. Tao, Y. Zhao. “Scientific workflow management and the KEPLER system”. *Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows, vol 18(10),* 2006.

[LDAP] Lightweight Directory Access Protocol - LDAP http://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

[LJB+99] J. Leigh, A. Johnson, M. Brown, D. Sandin, T. DeFanti. “Visualization in Teleimmersive Environments”. *IEEE Computer, vol 32(12),* 1999.

[LKR+07] L. S. Lima, B. Karlsson, A. B., Raposo, I. H. F Santos. “Uma Ferramenta de Videoconferência para apoiar Múltiplas Sessões de Trabalho Colaborativo”. *XIII Brazilian Symposium on Multimedia and Web, WebMidia 2007,* Gramado, RS, Brazil, Oct 2007.

[LMC+05] M. J. Lima, T. Melcop, R. Cerqueira, C. Cassino, B. Silvestre, M. Nery, C. Ururahy. “CSGrid: Um Sistema para Integração de Aplicações em Grades Computacionais”. *In: SBRC'2005, Salão De Ferramentas, May 2005*, Fortaleza, Ceará, Brasil.

[LMH02] D. Lee, M. Lim M., S. Han. “ATLAS - A Scalable, Network Framework for Distributed Virtual Environments”, *In The Proc. 4th ACM Intern.* Conf. on CVE, GE, 2002.

[LMN+04] W. Lee, S. McGough, S. Newhouse, J. Darlington. “A standard based approach to job submission through web services”. *Proc. of the UK e-Science All Hands Meeting, pp. 901–905. UK EPSRC, 2004.*

[LM98] D. Li, R. Muntz. “Coca: Collaborative objects coordination architecture”. *ACM Conference on Computer Supported Cooperative Work*, CSCW’98, 179-188, 1998.

[LM99] D. Li, R. Muntz. “A collaboration specification language”. Proceedings of the 2nd Conference on Domain-Specific Languages PLAN '99. ACM Press, 149-162, 1999

[Mack99] W.E. Mackay. “Media Spaces: environments for informal multimedia interaction”. In Computer Supported Cooperative Work, M. Beaudouin-Lafon, Ed., vol. 7 of Trends in Software.

John Wiley & Sons, 55-82, 1999.

[MD96] J. Munson, P. Dewan. "A concurrency control framework for collaborative systems". *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 278-287, November 1996.

[Moan03] T. Moan. "Marine Structures for the Future". *Internal Report 2003-01*. Centre for Offshore Research & Engineering National University of Singapore, NUS, 2003.

[MP03] V. Mann, M. Parashar. "DISCOVER: a computational collaboratory for interactive grid applications". *In: F. Berman, G. Fox, T. Hey (eds). Grid computing: making the global infrastructure a reality. Wiley, New York, pp 727-744,* 2003.

[MVW95] C.B. Medeiros. G. Vossen, M. Weske. "WASA: A Workflow-Based Architecture to Support Scientific Database Applications". *DEXA,* pp 574-583, 1995.

[MSM04] G. Monacelli, F. Sessa, A. Milite. "An Integrated Approach to Evaluate Engineering Simulations and Ergonomics Aspects of a New Vehicle in a Virtual Environment: Physical and Virtual Correlation Methods", *FISITA 2004 30th World Automotive Congress, Barcelona,* Spain(2004), pp. 23-27.

[MGJ95] M. M. Mourelle, E. C. Gonzalez, B. P. Jacob, 1995, "ANFLEX - Computational System for Flexible and Rigid Riser Analysis", *Proceedings of the 9th International Symposium on Offshore Engineering, Brazil*.

[NT06] S. Nakajima, T. Tamai. "Formal specification and analysis of JAAS framework". *Proceedings of the 2006 international workshop on Software engineering for secure systems, International Conference on Software Engineering*

[NW98] T.J. Nam, D.K. Wright. "CollIDE: A Shared 3D Workspace for CAD". *Conference on Network Entities*, UK, 1998.

[OASIS] OASIS Web Services Business Process Execution Language (WSBPEL). http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel

[OPM] Open Provenance Model, website at http://openprovenance.org/

[OSG] OpenSceneGraph. http://www.openscenegraph.org/

[Ort05] E. Ort. "SOA Architecture and Web Services: Concepts, Technologies, and Tools. Sun Developer Network", http://java.sun.com/developer/technicalArticles/WebServices/soa2/

[OO95] J. Olson, G. M. Olson. "What mix of video and audio is useful for small groups doing remote real-time design work". *Proceedings of SIGCHI'95*, ACM Press, 362-368, 1995.

[Paraview] Paraview website. http://www.paraview.org

[PMH+98] S.G. Parker, M. Miller, C.D. Hansen, C.R. Johnson. "An integrated problem solving environment: The SCIRun computational steering system". *In 31st Hawaii International Conference on System Sciences* (HICSS-31), pp 147-156, 1998.

[PRS+03] C. T. Pozzer, A. B. Raposo, I. H. F. Santos, J. L. E. Campos, L. P. Reis. "CSVTool - A Tool for Video-Based Collaboration". *The Proc. Webmídia, Brasil, 2003.*

[PKL00] K. S. Park, A. Kapoor, J. Leigh. "Lessons learned from employing multiple perspectives in a collaborative virtual environment for visualizing scientific data". *Proceedings of the Third international Conference on Collaborative Virtual Environments (CVE '00)*, E. Churchill and M. Reddy, Eds.. ACM Press, 73-82, 2000.

[PKS+00] K. S. Park, Y. Cho, N. Krishnaprasad, C. Scharver, M. Lewis, J. Leigh, A. Johnson. "CAVERNsoft G2: A Toolkit for High Performance Tele-Immersive Collaboration". Proceedings of the ACM Symposium on Virtual Reality Software and Technology 2000, pp. 8-15, 2000.

[PT05] A. Paventhan, K. Takeda. "MyCoG.NET: Towards a Multilanguage CoG Toolkit". *In the 3rd International Workshop on Middleware for Grid Computing, MGC´05,* Grenoble, France, November 2005.

[PTC+06] A. Paventhan, K. Takeda, S.J.Cox, D. A. Nicole. "Leveraging Windows Workflow Foundation for Scientific Workflows in Wind Tunnel Applications". *Proceedings of the 22nd International Conference on Data Engineering Workshops, ICDEW'06*, Atlanta, Georgia, April 2006.

[PtolemyII] Ptolemy II, http://ptolemy.eecs.berkeley.edu/ptolemyII/

[RCW+06] A. B. Raposo, E.T.L. Corseuil, G.N. Wagner, I.H.F. Santos, M. Gattass. "Towards the Use of CAD Models in VR Applications".

ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry, VRCIA 2006, CUHK, Hong Kong, June 2006.

[RSS+09] A. B. Raposo, I. H. F. Santos, L. P. Soares, G. Wagner, E. Corseuil, M. Gattass "ENVIRON: Integrating VR and CAD in Engineering Projects". *IEEE Computer Graphics & Applications, v.29, n.6, p.91-95, 2009. (ISSN 0272-1716). DOI: 10.1109/ MCG.2009.118.*

[RF02] A. B. Raposo, H. Fuks. "Defining Task Interdependencies and Coordination Mechanisms for Collaborative Systems". A. M. Pinna-Dery, K. Schmidt and P. Zaraté (eds.), *Cooperative Systems Design: A Challenge of the Mobility Age* (Frontiers in Artificial Intelligence and Applications, Vol. 74), 88–103, IOS Press, Amsterdam, 2002.

[RG96a] M. Roseman, S. Greenberg. "Building real time groupware with GroupKit, a groupware toolkit". *ACM Transactions on Computer–Human Interaction*, 3(1):66–106, March 1996.

[RG96b] M. Roseman, S.Greenberg. "TeamRooms: Network places for collaboration". *Proceedings of the ACM CSCW'96 Conference on Computer Supported Cooperative Work*, 16–20, November 1996.

[RRF+04] E. E. R. Russo, A. B. Raposo, T. Fernando, M. Gattass. "Workspace Challenges for the Oil & Gas Exploration & Production Industry". In: 4 *th* Conference of Construction Applications of Virtual Reality - CONVR 2004, p. 145-150, Lisboa, Portugal. 2004.

[RSV+94] W. Reinhard, J. Schweitzer, G. Völksen. "CSCW Tools: Concepts and Architectures". *In The IEEE Computer,* vol 27(5), pp 28-36, 1994.

[RWM+02] A. Rajasekar, M. Wan, R. Moore. "MySRB & SRB - components of a data grid". *In the 11th International Symposium on High Performance Distributed Computing (HPDC-11),* 2002.

[SAP] SAP - Business Management Software Solutions Applications and Services. http://www.sap.com/index.epx

[SBM+09] I. H. F. Santos, V. Braganholo, M. Mattoso B. P. Jacob, C. Albrecht. "Integrating The Galileo Applications For Simulation of Offshore Systems Via The GXML Unified Format". *XXX*

CILAMCE - Iberian Latin American Congress on Computational Methods in Engineering - 2009 (CD-ROM), p 1-15. Buzios, RJ, Brasil.

[SBO+06] R. Soma, A. Bakshi, A. Orangi, V. K. Prasanna, W. D. Sie. "Service-Oriented Data Composition Architecture for Integrated Asset Management", *SPE Intelligent Energy Conference and Exhibition (IECE)*, April 2006.

[SCA] Service Component Architecture http://tuscany.apache.org/

[Schlumb] Schlumberger Inside Reality website. http://www.slb.com/ content/services/software/virtual/

[SCJ+02] S. F. Senra, F. N. Correa, B. P. Jacob, M. M. Mourelle, I. Q. Maseti. "Towards the Integration of Analysis and Design of Mooring Systems and Risers, Part I an II", *Proceedings of the 21st International Conference on Offshore Mechanics and Artic Engineering,* June, 23-28, OMAE 2002, Oslo, Norway.

[SCIRun] SCIRun website. http://software.sci.utah.edu/scirun.html

[SCL+01] C. L. N. Santos, G. G. Cunha, L. Landau. "Use of VRML in Collaborative Simulations for the Petroleum Industry". *Proceedings of the 34 th Annual Simulation Symposium, SS01, 2001.*

[SD] Scientific Data Management Framework Workshop, Argonne National Lab, USA. August 2003. http://sdm.lbl.gov/~arie/sdm/SDM.Framework.wshp.html

[Simon05] A. Simon. "First-person experience and usability of co-located interaction in a projection-based virtual environment". *In Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST'05, Monterey, November 7-9, 23–30*, *CA, USA, 2005.*

[SPG05] Y. L. Simmhan, B. Plale, D. Gannon. "A survey of data provenance in e-science". *SIGMOD Record, vol. 34(3), pp 31–36*, 2005.

[SRG02] I. H. F. Santos, A. B. Raposo, M. Gattass. "Finding Solutions for Effective Collaboration in a Heterogeneous Industrial Scenario". *In The 7th Inter. Conf. Comp. Supp. Coop. Work in Design, CSWID02,* Rio de Janeiro, 2002.

[SRG06] I. H. F. Santos, A. B. Raposo, M. Gattass. “A Service-Oriented Architecture for a Collaborative Engineering Environment in Petroleum Engineering”. *Virtual Concept International Conference. CanCun, Mexico, 2006. Research in Interactive Design Proceedings of Virtual Concept 2006*, Springer Verlag (ISBN-10: 2-287-48363-2, ISBN-13: 978-2-287-48363-9).

[SRS+08] I. H. F. Santos, A. B. Raposo, L. P. Soares, E. T. L. Corseuil, G. N. Wagner, P. I. N. Santos, R. Toledo, M. Gattass. “ENVIRON: An Integrated VR Tool for Engineering Projects”. *12th International Conference on CSCW in Design, CSCWD08. Xi'an, China, Abril 2008.*

[SRG08] I. H. F. Santos, A. B. Raposo, M. Gattass. “Software Architecture for an Engineering Collaborative Problem Solving Environment”. *IEEE Software Engineering Workshop, SEW 2008, p.43-51. Kasandra, Greece, 2008.*

[SS05] A. Simon, S. Scholz. “Multi-Viewpoint Images for Multi-User Interaction”. *In Proc IEEE Virtual Reality Conference, VR '05, pp. 107-113, March,* 2005.

[SVR+04] I. H. F. Santos, C. Valle, A. B. Raposo, M. Gattass. “A Multimedia Workflow-based Collaborative Engineering Environment: Integrating an Adaptive Workflow System with a Multimedia Collaboration System and a Collaborative Virtual Environment for Petroleum Engineering”. *6th International Conference on Enterprise Information Systems. ICEIS 2004,* Vol. 5, p.259-262, Porto, Pt, March 2004.

[Tietze01] D. Tietze. “A Framework for Developing Component-based Cooperative Applications”, Ph.D. Dissertation, Technischen Universitat Darmstadt, 2001.

[Triana] Triana Project. Open-source Problem Solving Environment. http://www.trianacode.org/index.html.

[TR03] F. E. H. Tay, A. Roy. “CyberCAD: A Collaborative Approach in 3D-CAD Technology in a Multimedia-Supported Environment”. *Computers in Industry*, Vol. 52, Number 2, 127-145, 2003.

[Tramb99] H. Tramberend. “Avocado: A Distributed Virtual Reality Framework”. *The Proceedings of the IEEE VR, 1999*.

[Tufte83] E. Tufte. “The Visual Display of Quantitative Information”.

Graphics Press, Cheshire, Connecticut, 1983.

[VanderAlst03] W. M. P. Van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, A.P. Barros. *Workflow Patterns. Distributed and Parallel Databases,* 14(1), pp 5–51, 2003.

[VEG07] K. K. Verdi, H. J. Ellis, M. R. Gryk. “Conceptual-level workflow modeling of scientific experiments using NMR as a case study”, *BMC Bioinformatics*, v. 8, p. 31. 2007.

[VRGeo07] The VRGeo Consortium Project. http://www.vrgeo.org

[Walkinside06] Walkinside. A visualization tool for viewing complex 3D CAD models in real time. http://www.walkinside.com/index.htm

[WfMC95] Workflow Reference Model, 1995. WfMC - Workflow Management Coalition, 1995. Document No. TC00-1003, Issue 1.1, 1995. http://www.wfmc.org

[WINFX07] WINFX Dev Center, http://msdn.microsoft.com/winfx

[WPS97] M. Weber, G. Partsch,, J. H. Schweitzer. “Integrating Synchronous Multimedia Collaboration into WfMS”, *ACM SIGGROUP, Phoenix, USA,1997.*

[WVM+98] M. Weske, G. Vossen, C.B. Medeiros, F. Pires. “Workflow Management in Geoprocessing Apps”. In Tech. Report N.04-98-I, Univ.Muenster, GE, 1998.

[YB05] J.Yu, R. Buyya. “A Taxonomy of Scientific Workflow Systems for Grid Computing”. *SIGMOD, Vol.34, No.3, 2005*.

8
Appendix A

In this appendix we provide some technical detailed about some of the technologies used by CEE. The objective here is to give a brief overview of those technologies in order to make clear how are they combined in the CEE SOA architecture.

8.1.
Business Process Management and Business Process Engineering

SOA is an approach to delivering business solutions through services (capabilities) that are linked together by business logic – this approach reflects how a business actually operates compared to conventional applications development methods. As a result, the relationship between IT and the business is transformed from consumer/supplier to collaborating partners. SOA is an architectural style that creates new business applications through the intelligent “orchestration” of discrete, reusable business functions called “services” (figure X), each of which performing a single and well-defined task.

SOA will help to react much more quickly and cost effectively to new market opportunities, changes in business climate, and new regulation. When the business wants to introduce, change or improve a process, often one can simply adapt, reconfigure and resequence the existing services. When there is a need to bring new Software, this can be taken off-the-shelf, making it more cost-effective and faster to implement. In this way SOA offers an exciting opportunity in a world where companies need to adapt quickly and costs are a constant challenge.

Business solutions, in this new paradigm, are “composite applications” consisting of standard services linked together with business logic and standard service connections. Unlike traditional monolithic software applications, which reflect current (even outdated) process, a suite of component services can be rapidly rearranged and/or extended to reflect new business strategies and evolving market conditions.

In a conceptual model of a SOA, users of a composite business application average a common interface layer, which provides access to standard business process modeling and orchestration tools, a common set of generic SOA functions (including security, management and governance of services), and a repository of specific business services they can work with – including component services provided by external vendors, and legacy internal applications "wrapped" with a standard interface to look and act like any other service.

Once a global organization has a sufficient library of services available, almost any business process can be orchestrated without having to write new code. Besides that, new and better services can be swapped out for old ones without causing a ripple in the business workflow.

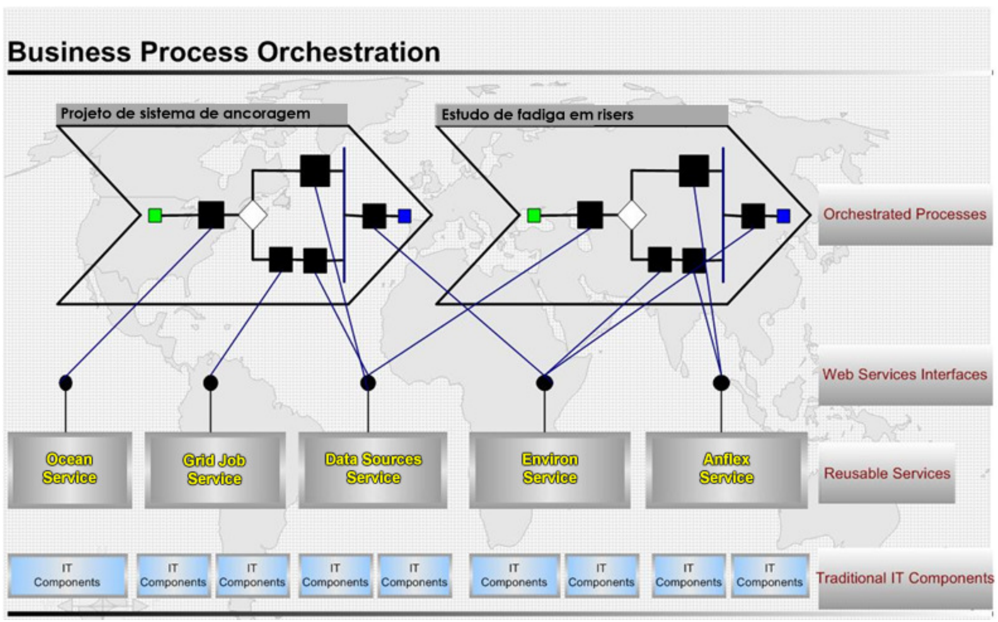


Figure 8.1: Business Process Engineering for CEE

8.2. Service Oriented Architecture

Service-Oriented Architecture (SOA) is a style of architecting software systems by packaging functionalities as services that can be invoked by any service requester [HKG+05], [Ort05]. 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 will produce a dramatic reduction of technology development costs by leveraging functions already built into legacy systems, by reusing services developed for other process, and by simplifying maintenance and support through elimination of redundant, siloed applications. Indeed SOA architectures are becoming a popular and useful means of leveraging Internet technologies to improve business processes in the oil&gas industry nowadays [GFF+05], [SBO+06]

In service-oriented design a service is generally implemented as a course-grained, discoverable software entity that exists as a single instance and interacts with applications and other services through a loosely-coupled, message-based communication model. The following definitions comprise important service-oriented terminology:

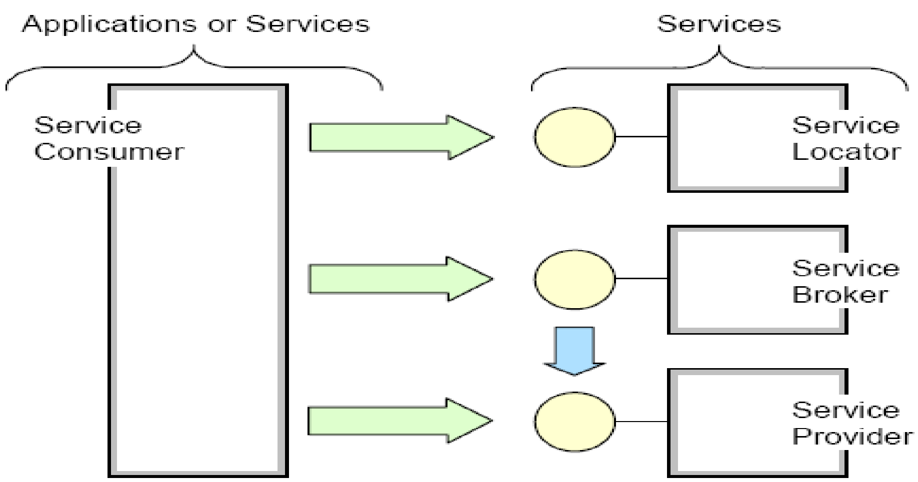


Figure 8.2 : Service-oriented terminology. (IBM RedBooks)

- ***Services***: logical entities, with contracts defined by one or more published interfaces.
- ***Service provider***: network-addressable software entity that implements a service specification. Accepts and executes requests from consumers. It publishes its services and interface contract to the service registry so that service consumer can discover and access.
- ***Service consumer (or requestor):*** an application, a software module or another service that requires a service from a service provider. It initiates the enquiry of the service in the registry, binds

to the service over a transport, and executes the service function. The service consumer executes the service according to the interface contract.

- ➢ ***Service locator:*** a specific kind of service provider that acts as a registry and allows for the lookup of service provider interfaces and service locations.
- ➢ ***Service broker:*** a specific kind of service provider that can pass on service requests to one or more additional service providers.
- ➢ ***Service registry:*** the enabler for service discovery. It contains a repository of available services and allows for the lookup of service provider interfaces to interested service consumers.

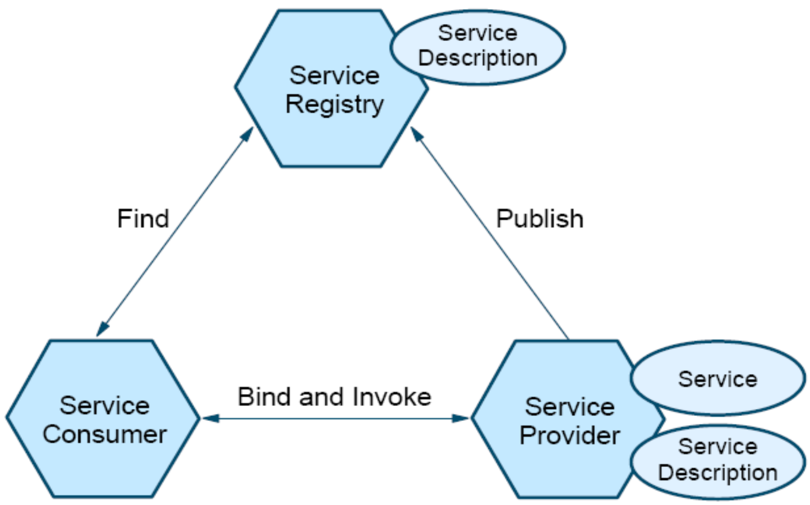


Figure 8.3 : Collaboration in SOA (IBM RedBooks)

SOA constitutes an approach for building distributed systems that deliver application functionality as services to either end-user applications or other services. The collaborations in SOA follow the “find, bind and invoke” paradigm [EAA+04], where a service consumer performs dynamic service location by querying the registry for a service that matches its criteria. If the service exists, the registry provides the consumer with the interface contract and the endpoint address for the service.

The “find, bind and invoke” paradigm presents some drawbacks. First, the point-to-point nature of interaction between services means that service consumers often need to be modified whenever the service provider interface changes. This is often not a problem on a small scale, but in large enterprises it could mean changes to many client applications. It can also become increasingly difficult to make such changes to legacy clients. Second, it can lead to a fragile and inflexible architecture when a large number of service consumers and providers communicate using point-to-point “spaghetti” style connections. Last, every new deployed service requires that each service consumer has a suitable protocol adapter for that new service provider. Having to deploy multiple protocol adapters across many client applications adds to cost and maintainability issues.

8.2.1. Enterprise Service Bus

An Enterprise Service Bus (ESB) is a pattern of middleware that unifies and connects services, applications and resources within a business [EAA+04]. ESB is a platform built on the principles of SOA and other open standards to help applications integrate seamlessly. Put 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's 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.

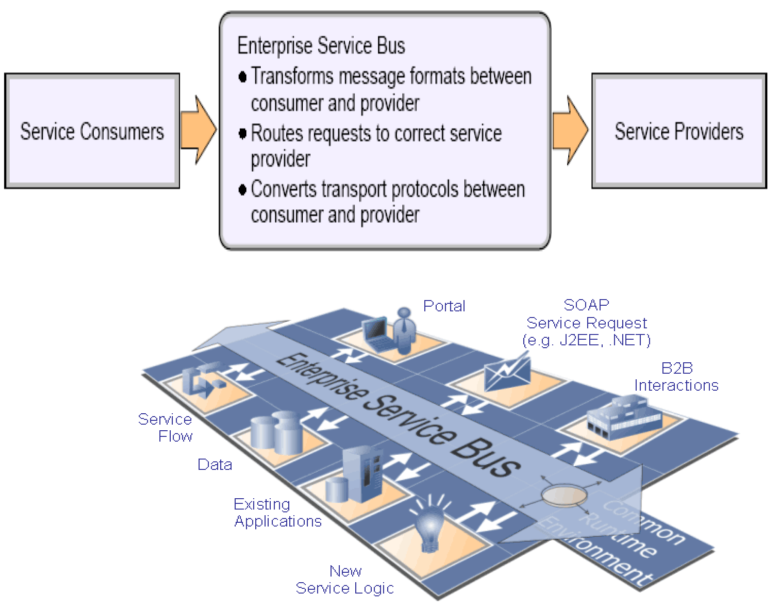


Figure 8.4 : ESB Conceptual model (IBM RedBooks)

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:

- Provides a transparent means for communicating with heterogeneous services over a diverse set of message protocols.
- Provides a shared messaging layer by which enterprise engineering applications, services, and components can connect and communicate.
- 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.
- Provides sophisticated error recovery, allowing for failed message delivery, scalability problems, duplicate messages, network failure, etc.

The main aim of the Enterprise Service Bus 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 modeled as services that offer one or more business operations. Implementing an Enterprise Service Bus requires an integrated set of middleware services that support the following architecture styles:

- ***Service-oriented architectures***, where distributed applications are composed of granular re-usable services with well-defined, published and standards-compliant interfaces.
- ***Message-driven architectures***, where applications send messages through the ESB to receiving applications.
- ***Event-driven architectures***, where applications generate and consume messages independently of one another

8.2.2. Web Services

Web services form an attractive basis for implementing service-oriented architectures for distributed systems. Web services rely on open, platform-independent protocols and standards, and allow software modules be accessible over the internet. Web services and service-oriented architectures are becoming a popular and useful means of leveraging Internet technologies to improve business processes in the oil&gas industry as we showed in the Chapter 2.

8.3. Workflow Management System

Ellis [Ellis99] presents Workflow Management Systems (*WfMS*) as a tool to assist in the specification, modeling, and enactment of structured work process within organizations. These systems are a special type of collaboration technology which can be described as *"organizationally aware groupware"* [EN96]. According to the Workflow Management Coalition (WfMC), a WfMS is "the computerized facilitation or automation of a business process, in whole or in part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" [WfMC95].

A *WfMS* contains two basic components:

- ***Workflow modeling component***, which enables administrators and analysts to define processes (or procedures) and activities, analyze and simulate them, and assign them to people, agents or processes. This component is sometimes called "specification module" or *"build time system"*.
- ***Workflow execution component (or enactment),*** sometimes also called the *"run-time system".* It consists of the execution interface seen by end-users and the *"workflow engine"*, an execution environment which assists in coordinating and performing the processes and activities. It enables the units of work to flow from one user's workstation to another as the steps of a procedure are completed. Some of these steps may be executed in parallel; some executed automatically by the computer system.

There are different types of workflows, which suit different organizational problems:

- ***Production workflow*** – the key goal is to manage large numbers of similar tasks, and to optimize productivity.

- ***Administrative workflow –*** its most important feature is the ease to define the process. Flexibility is more important than productivity, and these systems handle one or two orders of magnitude lower numbers of instances per hour than Production Workflow Systems.
- ***Collaborative Workflow –*** focuses on teams working together towards common goals. Groups can vary from small, project-oriented teams, to widely dispersed people with interests in common. Effective use of collaborative workflow to support team working is now considered a vital element in the success of enterprises of all kinds. Throughput is not an important consideration, and Process Definitions are not rigid and can be amended frequently.
- ***Ad-hoc Workflow –*** allows users to create and amend Process Definitions very quickly and easily to meet circumstances as they arise. So it is possible to have almost as many Process Definitions as there are instances of the definitions. It maximizes flexibility in areas where throughput and security are not major concerns. Whereas in Production Workflow, clearly the organization owns the process, Ad-Hoc Workflow users own their own processes.

These are workflows that enable the coordination of different types of exception, dynamic change problem and possibilities of late modeling and local adaptation of particular workflow instances [vdAalst99]. Adaptive workflows aim at providing process support like normal workflow systems do, but in such a way that the system is able to deal with certain changes. These changes may range from simple changes to ad hoc changes towards the redesign of a workflow process, as usually happens when an organization finishes a review on its business process.

The support for managing partial workflows present in an “adaptive workflow” is very attractive for Large Engineering Projects because processes in engineering domains have a very dynamic nature which means that they cannot be planned completely in advance and are under change during execution. Furthermore, in contrast to well-structured business processes, they are characterized by more cooperative forms of work whose concrete process steps cannot be prescribed.

Typically, a workflow system is implemented as a server machine which has and interprets a representation of the steps of the procedures and their precedence; along with client workstations, one per end-user, which assists the user in performing process steps. This is typically combined with a network and messaging system (or communication mechanism) to allow the server to control and/or to interact with end-user workstations. Also included is a database that stores the process representation, attributes of end-users, and other pertinent workflow information. Many of the workflow products are combined with imaging and/or Document Management Systems (DMS).

8.3.1. Workflow Components

To achieve workflow interoperability, the Workflow Management Coalition (WfMC) created The Workflow Reference Model that describes FIVE Interface definitions [WfMC95].

- ***Interface 1 (Process Definition)*** - deals with passing Process Definitions from external tools to the workflow engine where they are enacted. This is the link between the so-called "Process Definition Tools" and the "Enactment Service".
- ***Workflow APIs (Interfaces 2 & 3)*** - these interfaces have been combined and cover the WfAPIs (Workflow API's). The support of these interfaces in workflow management products allows the implementation of front-end applications that need to access workflow management engine functions (workflow services). Such implementations might be written by workflow management exploiters or workflow systems integrators (WfSI). Integration between workflow and other desktop tasks (calendar, mail, reminders, etc) is often a common target and the workflow APIs allow workflow task integration into a common desktop.

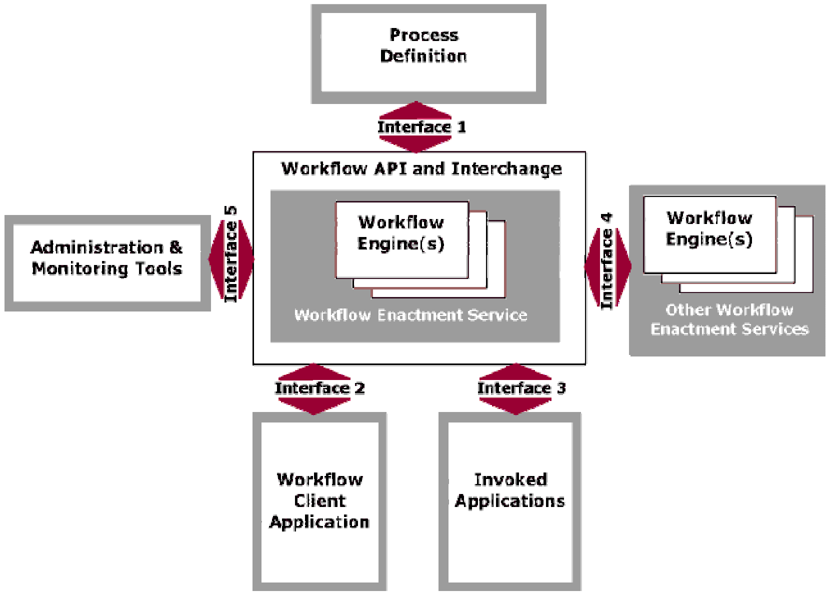


Figure 8.5: WfMC reference model.

- ***Inter-Engine Workflow (Interface 4)*** - defines the mechanisms that workflow product vendors are required to implement in order that one workflow engine may make requests of another workflow engine to effect the selection, instantiation, and enactment of known process definitions by that other engine. The requesting workflow engine is also able to pass context data (workflow relevant or application data) and receive back status information and the results of the enactment of the process definition. As far as possible, this is done in a way that is “transparent” to the user. This interface is intended for the use of WfSIs, and not users. As a side effect of facilitating communication between workflow engines, there is a requirement for audit data to be produced.
- ***Audit and Monitoring (Interface 5)*** - the support of this specification in workflow products allows analysis of consistent audit data across heterogeneous workflow products. During the initialization and execution of a process instance, multiple events occur which are of interest to a business, including WfAPI events, internal workflow management engine operations and other system and application functions. With this information, a business can determine what has occurred in the business operations managed by

8.3.2.
Process Definition Language

The WfMC defines a Process Definition as “the representation of a business process in a form which supports automated manipulation, such as modeling, or enactment by a workflow management system. The Process Definition consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated IT applications and data, etc.” [WfMC95]. This reveals the necessity for a Process Definition interchange mechanism. First, within the context of a single workflow management system there has to be a connection between the design tool and the execution/run-time environment. Second, there may be the desire to use another design tool. Third, for analysis purposes it may be desirable to link the design tool to analysis software such as simulation and verification tools. Fourth, the use of repositories with workflow processes requires a standardized language. Fifth, there may be the need to transfer a definition interchange from one engine to another.

```
<WorkflowProcess Id="Sequence">
    <ProcessHeader DurationUnit="Y"/>
    <Activities>
        <Activity Id="A">
            ...
        </Activity>
        <Activity Id="B">
            ...
        </Activity>
    </Activities>
    <Transitions>
        <Transition Id="AB" From="A" To="B"/>
    </Transitions>
</WorkflowProcess>
```

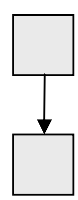


Figure 8.6: Workflow pattern Sequence in XPDL.

Figure 8.7: WfMC reference model.

The XML Process Definition Language (XPDL) is a format standardized by the WfMC to interchange Business Process definitions between different workflow products like modeling tools and workflow engines. XPDL defines a XML schema for specifying the declarative part of workflow. This language is a low level language and it can be used to model higher level business languages.

A workflow pattern is a specialized form of a design pattern as defined in the area of software engineering. Workflow patterns refer specifically to recurrent problems and proven solutions related to the development of workflow applications in particular, and more broadly, process-oriented applications. presents an example of Sequence pattern [VanderAlst03].

8.3.3. Workflow Integration with other technologies

In the literature there are a lot of proposals concerning integration of a WfMS and other technologies. [Joeris97] proposes the combination with a Document Management System. He suggests the creation of a new data-oriented perspective for the WfMS, centered on the documents and data produced during the execution of tasks, in order to improve the coordination and cooperation support for engineering processes.

Weske [WVM+98] proposes the junction with a Geographic Information System to combine a data-oriented view with a process-oriented view aiming to support the complex cycle of process and data modeling in environmental-related geoprocessing applications.

8.3.3.1. VCS and WfMS

The integration of VCS into a WfMS is not new. Weber et al. [WPS97] proposed the integration of a VC tool into a WfMS in order to furnish a synchronous collaboration work. To allow the coordination of the conference by the WfMS he suggests the creation of new entity in the workflow model, called “conference activity”. Another important aspect is the time dimension. Conferences that are already planned at the time of the creation of the workflow are called pre-scheduled, while an ad-hoc conference is the one that was not foreseeable at the time when the workflow model is specified. This implies that in the former case some of the steps can be formally prescribed in the WfMS providing a tighter control of the results and documents generated during the conference section by the workflow engine, while in the later the results of the section should be updated by the users in the system.

The combination of VCS and WfMS can support problems which cannot be well supported by each one of them isolated. Embedding synchronous teamwork as part of the workflow produces a complementary way of conducting project activities. Such integration would enable a continuous stream of tasks and activities in which fast, informal, ad hoc, and direct actions can be taken through conferences within the usual formal workflow. The use of a coordination tool, WfMS, and a communication tool, VCS, constitute a good combination which improves the collaborative capabilities of the CEE [Dus00].

Another important aspect is the time dimension. Conferences that are already planned at the time of the creation of the workflow are called pre-scheduled, while an ad-hoc conference is the one that was not foreseeable at the time when the workflow model is specified. This implies that in the former case some of the steps can be formally prescribed in the WfMS providing a tighter control of the results and documents generated during the conference section by the workflow engine, while in the later the results of the section should be updated by the users in the system.

8.4. Scientific Workflow Management Systems

Scientific Grid computing environments are increasingly adopting the Open Grid Services Architecture (OGSA) [Ort05], which is a service oriented architecture for Grids.

OGSA was developed by the Globus Alliance and based on standard XML-based web services technology. With the proliferation of OGSA, Grids effectively consist of a collection of Grid services, web services with certain extensions providing additional support for state and life cycle management. Hence, the need arises for some means of composing these basic services into larger workflows in order to, for example, express a scientific experiment.

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. WS-BPEL 2.0 was ratified as a standard in 2007. As an execution language, WS-BPEL defines how to represent the activities in a business process, along with flow control logic, data, message correlation, exception handling, and more.

BPEL is emerging as the standard XML-based workflow language for defining and executing business processes using XML Web services. Without this standardization, the environment of the commercial systems would be not unlike the current Grid workflow engine landscape.

BPEL enables the composition, orchestration and coordination of web services. A business process described in BPEL can itself be treated as an XML web service. BPEL converged from two other workflow description languages –

Microsoft's XLANG [12] and IBM's WSFL [13]. BPEL provides constructs for invoking a web service and exchanging messages with a web service, both synchronously and asynchronously. It also has other primitive constructs which include constructs for manipulating data variables, indicating faults and exceptions, terminating a process and, waiting for some time. It also supports compensation blocks for exception handling. BPEL also has control constructs, such as looping, if-then-else and switch-case activities. BPEL supports both sequential and parallel execution of activities. Since BPEL is XML-based, it is extensible, which means that we can add our own constructs and also provide our own implementation of these extensions.

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.

8.4.1. Scientific Workflows Tools

8.4.1.1.Kepler

Kepler [Kepler] is another extensible workflow system aimed at scientific workflows. The Kepler project is cross-project collaboration between SDM (Scientific Data Management) Center, SEEK (Science Environment for Ecological Knowledge), GEON (Cyber-infrastructure for the Geosciences) and RoadNet (Real-time Observatories, Applications, and Data Management Network). The aim of Kepler is to provide a framework for design, execution and deployment of scientific workflows. Kepler is built on top of Ptolemy II [PtolemyII]– an API for heterogeneous, concurrent modeling and design. Kepler currently provides the following major features [LAB+06]:

- Prototyping workflows: Kepler allows scientists to prototype scientific workflows before implementing the actual code needed for executions

- MoML – an internal XML language for specifying component-based models and composing actors into workflows
- Distributed execution (Web and Grid-Services): Kepler's Web and Grid service actors allow scientists to utilize computational resources on the network in a distributed scientific workflow.
- Database access and querying: Kepler includes database interactions.
- Other execution environments: Support for foreign language interfaces via the Java Native Interface provides the flexibility to reuse existing analysis components and to target appropriate computational tools.

8.4.2. Condor

Condor [Condor] is a specialized workload management system for compute-intensive engineering simulations. Condor provides a job queueing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. Condor is known to provide a High Throughput Computing (HTC) environment on a large size of distributed computing resources. It can manage a large size of machines and networks owned by different users. Besides controlling idle components, Condor can be configured to share resources. When a user submits a job to Condor it put it into a queue, selects when and where to run the job based on a policy, monitors the job, and informs the users about the status of the task upon completion. Condor-G is used to schedule and run jobs on heterogeneous grid resources. It uses Globus GRAM service, a uniform interface to heterogeneous batch systems. Condor-G creates an abstract view of the grid as local resource and allows the user to submit jobs to different batch systems (Condor, Load Leveler, etc.) and get updates regarding the status of the tasks.

8.4.3. InfoGrid

InfoGrid [LMC+05], is a client/server system for grid environments which, in addition to the support for usage and management of distributed computational resources, offers facilities to integrate applications and manage data and users (Figure 11). InfoGrid presents to its users, through a web browser, a workspace with all available applications and with the user's data files organized by project. A user can extend the system adding new applications. InfoGrid also provides its users with some collaborative work facilities.

Applications which are executed in the client utilise available services of the InfoGrid to have access to and to manage distributed computational resources. One of these services is the remote execution of algorithms which are in computers linked to the InfoGrid. For InfoGrid, algorithms are defined as executable programs implemented in any language which accept input parameters, generate an output and do not have any type of interaction with the user during their execution. Many computers can be incorporated to the grid environment to serve as a platform for algorithms execution. New algorithms can be easily made available in the environment and the process to execute them is turned into a transparent task for the user.

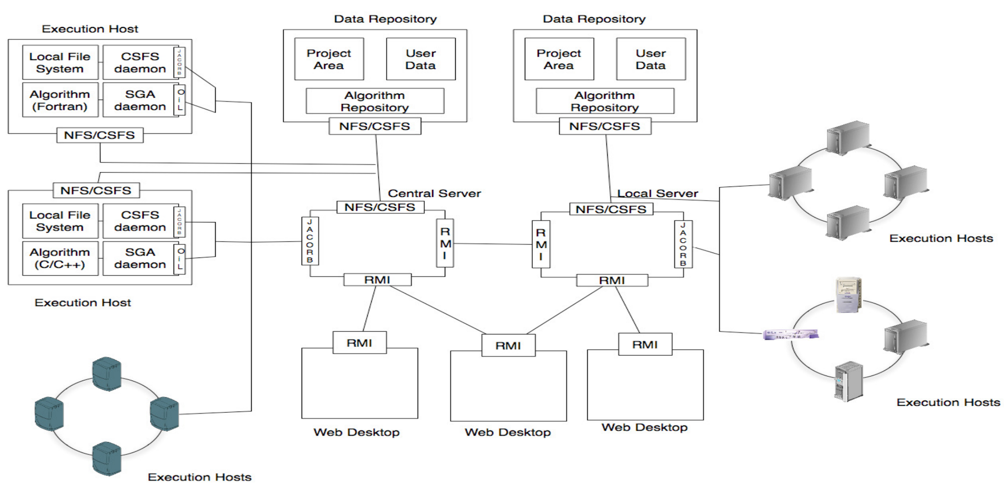


Figure 8.8: InfoGrid architecture.

Applications which are executed in the client utilize available services of the InfoGrid to have access to and to manage distributed computational resources. One of these services is the remote execution of algorithms which are in computers connected to the InfoGrid network. For InfoGrid, algorithms are defined as executable programs implemented in any language which accept input parameters, generate an output and do not have any type of interaction with the user during their execution. Many computers can be incorporated to the grid environment to serve as a platform for algorithms execution. New algorithms can be easily made available in the environment and the process to execute them is turned into a transparent task for the user.

8.4.4. 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 (DRM). GridSAM implements the Job Submission Description Language (JSDL) defined by the Global Grid Forum (GGF) [LMN+04]. Transparency of the underlying Grid scheduler being used to execute jobs on a Grid 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.

8.5. Virtual Environments

The terms Virtual Environment (VE) and Virtual Reality (VR) are often used synonymously to describe a computer-generated, artificial environment or reality that is presented to a user. A VE tries to evoke a strong sense of reality in the user. This is achieved by the generation of artificial input to the user's visual, acoustic and haptic senses.

By interfacing some of the user's articulations in the real world back into the VE, the user can consciously interact with the environment. Typically, interfaces to direct-manipulation devices are used, but nowadays more advanced interaction techniques like speech and gesture recognition have become a major research interest.

The generation of high-quality visual feedback from the virtual environment is often considered the most important aspect in generating a high degree of immersion. The desire to increase the degree of immersion led to the development of sophisticated image generators and display devices. Beginning with low-resolution monoscopic CRT displays used in early flight simulators and image generators that where capable of rendering only a few hundred polygons per second, the development progressed toward today's high-resolution stereoscopic display systems like the CAVE [CS+92] and readily available graphic cards that render hundreds of millions of polygons per second.

Parallel to the development of new display devices, image generators and input devices, various toolkits and application frameworks are developed. They provide a basic software infrastructure for the development of VE applications.

The main goal of these efforts is the maximization of software reuse in order to minimize the necessary development resources for application development. Designed for different application domains, the only common nominator of most toolkits and frameworks is a scene-graph based object model. The provided API, the supported hardware and operating systems and the set of supported display and input devices vary greatly.

Collaborative Virtual Environments (CVEs) are a special case of Virtual Reality Environments [Tramberend99], where the emphasis is to provide distributed teams with a common virtual space where they can meet as if face-to-face, co-exist and collaborate while sharing and manipulating, in real-time, the virtual artifacts of interest [GLG03]. They can be seen as the result of a convergence of research interests within the Virtual Reality and Computer Supported Cooperative Work (CSCW) communities. CVEs are becoming increasingly used due to a significant increase in cost-effective computer power, advances in networking technology and protocols, as well as database, computer graphics and display technologies. They have been used mainly by automotive and aircraft manufactures aiming to improve the overall product's quality and also aiming to reduce project's life cycle, cutting down costs and reducing the time-to-market of new products. Examples of applications are Visualization of real-time simulation of 3D Complex Phenomena, Collaborative Virtual Design and Product Development, Training and Edutainment, Telepresence and Telerobotics, Business meetings among others.

Studies of a cooperative work in real-world environments have highlighted the important role of physical space as a resource for negotiating social interaction, promoting peripheral awareness and sharing artifacts [BH+92]. The shared virtual spaces provided by CVEs may establish an equivalent resource for telecommunication. In teleimmersive environments (TE), a VCS is integrated with a CVE to provide collaborators at remote sites with a greater sense of presence in the shared space [LJB+99]. TEs may enable participants to discuss and manipulate shared 3D models and visualizations in such a way that each user can adopt their own viewpoint and can naturally indicate the others where they look and point. Scientific visualization has also been used in many application areas and has proven to be a powerful tool in understanding complex data [FB+99]. Those characteristics of TEs are very important for Virtual Prototyping as in projects of oil production units explained in section 2.

The development of CVE technology has been driven mainly by the challenge of overcoming technological problems such as photo realistic rendering and supporting multiple users in CVEs. Once those users are geographically distributed over large networks like the Internet, and the number of users has been increasing continuously, scalability turns to be a key aspect to consider for real-time interactions [LMH02].

Other important aspects are composability and extensibility or dynamic reconfigurability for assembling applications and improving adaptability of system at runtime with component-based system design, plug-ins functionality and service discovery mechanisms. In order to support the execution of CVEs with large-scale virtual worlds over long periods of time, they must be based on technologies that allow them to adapt, scale and evolve continuously. VE applications offer an almost limitless number of opportunities for the inclusion of plug-in technology. Graphical plug-ins may generate 3D models on the fly; network plug-ins may provide support for new protocols and filtering schemes; plug-ins for physical simulation may introduce previously unknown forces that improves the reality of the simulation. Persistence and portability aspects have also to be considered in order to guarantee the ability of building reusable large virtual worlds commonly needed in engineering projects.

9
Appendix B

9.1.
List of Publications

In what follows is a list of all papers related to this thesis, it can be downloaded in the following website:

http://www.tecgraf.puc-rio.br/~ismael/imk/publications/publications.html

<u>Periodicals</u>

- ***Environ: Integrating VR and CAD in Engineering Projects.*** *IEEE Computer Graphics & Applications,* v.29, n.6, p.91-95, 2009. (ISSN 0272-1716). DOI: 10.1109/MCG.2009.118. Raposo, A., Santos, I. H. F., Soares, L., Wagner, G., Corseuil, E., Gattass, M.

<u>Book Chapters</u>

- ***Collaborative Environment for Engineering Simulations with Integrated VR Visualization.*** *In: Lecture Notes in Computer Science, On the Move to Meaningful Internet Systems: OTM 2008 Workshops - Vol. 5333/2008, Springer Verlag, Berlin/Heidelberg, 2008, p. 12-13. (ISBN-978-3-540-88874-1).* Santos, I. H. F, Raposo, A. B., Gattass, M.
- ***A Service-Oriented Architecture for a Collaborative Engineering Environment in Petroleum Engineering.*** *Virtual Concept International Conference.* CanCun, Mexico, 2006. Research in Interactive Design Proceedings of Virtual Concept 2006, Springer Verlag (ISBN-10: 2-287-48363-2, ISBN-13: 978-2-287-48363-9). Santos, I. H. F, Raposo, A. B., Gattass, M.

<u>Papers in Conferences</u>

<u>2009</u>

- ***Integrating The Galileo Applications For Simulation of Offshore Systems Via The GXML Unified Format.*** *XXX CILAMCE - Iberian Latin American Congress on Computational Methods in Engineering - 2009 (CD-ROM), p 1-15. Buzios, RJ,* *Brasil.* Santos, I. H. F, Braganholo, V., Mattoso M., Jacob, B. P., Albrecht, C.

- ***Integrating VR in an Engineering Collaborative Problem Solving Environment.*** *ICEIS 2009 – 11th International Conference on Enterprise Information Systems,* Vol HCI, p. 124-129. Milan, Italy, 2009. Santos, I. H. F, Raposo, A. B., Gattass, M.
- ***Managing Information of CAD Projects in Virtual Environments..*** *XI Simposium on Virtual and Augmented Reality – SVR 2009,* p. 168-174. Porto Alegre, Brasil, 2009. Soares, L. P., Carvalho, F. G., Raposo, A. B., Santos, I. H. F.

<u>2008</u>

- ***A Software Architecture for an Engineering Collaborative Problem Solving Environment.*** *32nd Annual IEEE Software Engineering Workshop – SEW 2008, p.* 43-51. Kassandra, Greece, 2008. Santos, I. H. F, Raposo, A. B., Gattass, M.
- ***Collaborative Environment for Engineering Simulations with Integrated VR Visualization.*** *16th International Conference on Cooperative Information Systems – COOPIS 2008. Lecture Notes in Computer Science, Vol. 5333, p. 12-13. Monterrey, Mexico. Springer-Verlag, 2008.* Santos, I. H. F, Raposo, A. B., Gattass, M.
- ***EnViron: An Integrated VR Tool for Engineering Projects.*** *12th International Conference on CSCW in Design – CSCWD 2008. Xi'an, China, RJ, Abril 2008.* Santos, I. H. F, Raposo, A. B., Soares, L. P., Corseuil, e. T. L., Wagner, G. N., Santos, P. I. N., Toledo, R, Gattass, M.
- ***Environ: Uma Ferramenta de Realidade Virtual para Projetos de Engenharia.*** *XXIX CILAMCE - Iberian Latin American Congress on Computational Methods in Engineering - 2008 (CD-ROM), 17p. Maceió, AL, Brasil, 2008.* Soares, L. P., Corseuil, E. T. L., Raposo, Gattass, M., Santos, I. H. F.

<u>2007</u>

- ***Uma Ferramenta de Videoconferência para apoiar Múltiplas Sessões de Trabalho Colaborativo.*** *XIII Brazilian Symposium on Multimedia and Web, WebMidia 2007, Oct 2007, Gramado, RS, Brazil.* Lima, L. S., Karlsson, B., Raposo, A. B., Santos, I. H. F.

2006

- ***Towards the Use of CAD Models in VR Applications*** - *ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, *VRCIA 2006*, pp 67-74, June, Hong Kong, China, (ISBN 1-59593-324-7). Raposo, A. B., Corseuil, e. T. L., Wagner, G. N., Santos, I. H. F, Gattass, M.

2004 and before

- ***A Multimedia Workflow-Based Collaborative Engineering Environment for Oil & Gas Industry***. *ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry, VRCAI 2004*, pp 112-119, June 2004, Singapore, SI, (ISBN 1-58113-884-9). Santos, I. H. F, Göbel, M., Raposo, A. B., Gattass, M.
- ***A Multimedia Collaborative Engineering Environment***.- *6th International Conf. on Enterprise Information Systems, ICEIS 2004*, pp 259-262, April 2004, Porto, PT, (ISBN 972-8865-00-7). Santos, I. H. F, Valle, C., Raposo, A. B., Gattass, M.
- ***CSVTool – A Tool for Video-Based Collaboration***. *VII Symposium on Web and Multimedia Applications, WebMidia 2003*, pp 353-367, Nov 2003, Salvador, Brazil. Pozzer, C. T., Raposo, A. B., Santos, I. H. F., Campos, J. L. E., Reis, I. P.
- ***Buscando o Uso Operacional de Realidade Virtual em Grandes Modelos de Engenharia***. *VI Symposium on Virtual Reality, SVR 2003*, pp 187-198, October 2003, Ribeirão Preto, Brazil. Raposo, A. B., Corseuil, E. T. L., Santos, I. H. F, Gattass, M., Pinto, M. H. G.
- ***Finding Solutions for Effective Collaboration in a Heterogeneous Industrial Scenario.*** *The Seventh International Conference on CSCW in Design, CSCWID-2002,* p.74 - 79, September 2002, Rio de Janeiro, Brazil. Santos, I. H. F, Raposo, A. B., Gattass, M.