A Multimedia Workflow-Based Collaborative Engineering Environment for Oil & Gas Industry

Ismael H. F. Santos[a,b,c], Martin Göbel[a], Alberto B. Raposo[b] and Marcelo Gattass[b]

[a]Fraunhofer Gesellschaft, FhG, Schloss Birlinghoven 53754, Sankt Augustin, Germany
[b]Computer Graphics Technology Group, PUC-Rio, R. Marquês de São Vicente 225, 22453-900, Rio de Janeiro, Brazil
[c]CENPES, Petrobras Research Center, Ilha do Fundão, 21949-900, Rio de Janeiro, Brazil
ismael@tecgraf.puc-rio.br

ABSTRACT

In this paper we discuss the scenario of Petroleum Engineering projects of Petrobras, a large Brazilian governmental oil & gas company. Based on this scenario, we propose a set of application requirements and system architecture to guide the construction of a Collaborative Engineering Environment (*CEE*) for assisting the control and execution of large and complex industrial projects in oil and gas industry. The environment is composed by the integration of three different technologies of distributed group work: Workflow Management System (*WfMS*), Multimedia Collaborative System (*MMCS*) and Collaborative Virtual Environments (*CVE*).

Keywords: Collaborative Engineering, Collaborative Virtual Environments, Workflow Systems.

1. INTRODUCTION

The present work is motivated by the necessity of finding effective solutions for collaboration of team works during the execution of large and complex engineering projects at Petrobras, a large Brazilian governmental oil & gas company. The necessity of collaboration is especially acute in the field of computer graphics, whose techniques such as three-dimensional geometric modeling, scientific visualization, immersive virtual environments (*VEs)* equipped with large displays walls, stereographic projection systems, head mounted displays, haptic peripherals, videoconference tools (*VC)* and auditory display systems are pushing the limits of teamwork activities in oil & gas industry especially in GeoScience, Reservoir and Petroleum Engineering. The possibility to visualize and manipulate virtual models in the computer has completely changed the professional's way of working, notably for the geologists and engineers.

In this paper we introduce a set of application requirements and based on that we define the system architecture of a Collaborative Engineering Environment (*CEE*). The proposed *CEE* is composed by the integration of three different technologies of distributed group work: Workflow Management System (*WfMS)*, Multimedia Collaborative System (*MMCS)* and Collaborative Virtual Environment (*CVE*). It is intended to control the execution of large and complex engineering projects involving many geographically distributed teams. It also allows an easy integration of different applications providing the teamworkers with means of information exchange, aiming to reduce the barriers imposed by applications with limited or no collaboration support. This environment needs to be *extensible*, *flexible* and *platform-independent*, allowing a transparent flow of information among the teams involved in the project.

The difficulties in building an effective *CEE* can be analyzed in four domains: *cooperative work*, *distributed execution*, *project management* and *system interoperability*. In the first domain there is the necessity of providing effective human-to-human interaction and communication for solving conflicts and enhancing group's productivity. In the second resides the necessity of involving specialists in different areas located in different places and using distributed resources, requiring that the solution has the ability to be easily and seamlessly distributed. The third domain points to the necessity of reducing costs and time-to-market of new products, which further requires a computerized solution capable of controlling time scheduling and costs. Finally, there is the software diversity that specialists are forced to use to accomplish their tasks in a reasonable time, which implies the necessity of interoperability among the components of the solution.

We believe that our solution, comprising the combination of a process-oriented collaboration tool (*WfMS*), a synchronous communication tool (*MMCS*), and a collaborative virtual environment (*CVE*), would tackle well with the problems in the first three domains mentioned earlier. For the last domain we propose the creation of three more specialized components, the first one to manage shared data, Shared Data Management System (*SDMS*), the second to control all the documents and data generated during project's life-cycle, Document Management System (*DMS*), and the third one to interface our system with the execution of external applications, the Engineering Application System (*EAS*). In this way the *CEE* consists of a flexible and effective environment that improves the productivity of teams involved in large and complex engineering projects. In this way the CEE consists of a flexible and effective environment that should improve the productivity of the teams involved in large and complex engineering projects.

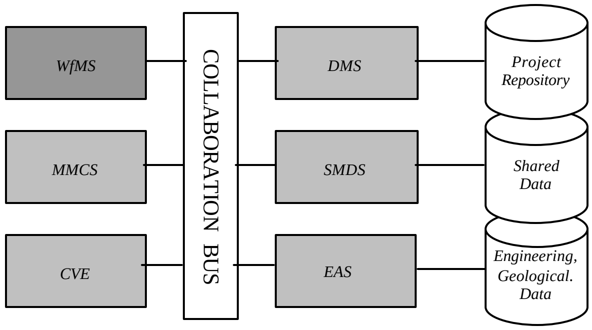

Figure 1. *CEE* Collaboration Bus.

The integration of the *WfMS* with the other components is done in a seamless way through the Collaboration Bus (*CB*us) (Figure 1) in a way that the user always interacts with the same interface independent of the application he 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. The *CB*us represents the collaborative infrastructure provided by the *CEE* core functions to fulfill the requirements discussed throughout the text.

In the following section we present some motivational aspects for our solution. In section 3 we discuss related works. Then, in section 4, we present the formal description of the problem, showing the requirements that the solution must satisfy. In section 5 our system architecture is discussed and conclusions finish the paper.

2. MOTIVATION

Workflow Management Systems (*WfMS*) assist in the specification, modeling, and enactment of structured work processes within organizations. These systems are a special type of collaboration technology which can be described as *"organizationally aware groupware"* [3]. According to the Workflow Management Coalition[25] a *WfMS* contains two basic components: the workflow modeling component, which enables administrators and analysts to define processes (or procedures) and activities, analyze and simulate them, and assign them to people. This component is sometimes called "specification module" or "build time system". The second is the workflow execution (or enactment) component, sometimes 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.

There are different types of workflows, which suits different organizational problems. Production workflows, ad-hoc workflows and scientific workflows are some examples. The type of workflow used in this work is following the definition of *"adaptive workflow"*. 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. Adaptive workflows[22] 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 our purposes 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.

Multimedia Collaborative Systems (*MMCS*) such as Videoconferencing Systems (*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[17]. The combination of *MMCS/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.

Collaborative Virtual Environments (*CVEs*) are a special case of Virtual Reality[21] environment systems, 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[6]. They can be seen as the result of a convergence of research interests within the Virtual Reality (*VR*) 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[1]. 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[14]. *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[4,15]. Those characteristics of *TE*s are very important for virtual prototyping as in projects of oil production units explained in section 4.

The development of *CVE* technology has been driven mainly by the challenge of overcoming technological problems such as photorealistic 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[13].

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.

QoS aspects such as, *Bandwidth, Reliability of the network, Latency and Delay of the communication links* are also important aspects for designing an effective *VE.*

Security[18] is becoming a serious concern as VEs proliferate. Many of those systems are being used in contexts in which there are incentives for malicious users to misuse such systems for their own gain. Besides this, in the virtual design of an engineering artifact (plant, equipment, etc), safety is a very important issue too because an improperly conceived project can provoke serious environmental accidents when released to be operational. Consequently, previously ignored aspects of security, must be transformed into primary concerns at the outset of designing a new VE system.

3. RELATED WORK

The integration of *MMCS/VCS* systems into a *WfMS* is not new, Weber[23] proposed the integration of a Videoconference 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 in his proposal is the time dimension. Conferences that are already planned at the time of the creation of the workflow are called *pre-scheduled conferences*, 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 kind of conference 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 manually updated by the users in the system.

In the literature there are a lot of proposals concerning the integration of a *WfMS* and other technologies. Joeris[10] proposes the combination with a *Document Management System (DMS)*. 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[24] proposes the junction with a *Geographic Information System (GIS)* 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. This integration is very suitable for our solution because many applications in engineering projects deals with geo-referenced data.

Sevy[20] proposes the creation of a *CEE* called *Collaborative Design Studio (CDS)* to enhance the design engineering process through the integration of a Computer-Aided Design and engineering tools (CAD/CAE), a *MMCS*, and archiving functions.

DDRIVE[2] is a system developed at HRL Laboratories at General Motors Research & Development Center, for Distributed Design Review in *VEs*. At the center of the software architecture is the *Human Integrating Virtual Environment (HIVE)*, a collaboration infrastructure and toolset to support research and development of multi-user, geographically distributed, 2D/3D shared applications.

SAVE[8], Safety Virtual Environment, is a *VR* based safety training system for dangerous and hazardous facilities. It is a multi purpose virtual reality software system that is mainly intended for employee training. It provides a framework and software system for a variety of training scenarios using VR technology. Each virtual training scenario comprises a scene in which the trainee can move freely and interact with objects like pumps, valves, and other control devices. The *SAVE* comprises four major parts: visual simulation, *SAVEsim*; instructor control and supervision, *SAVEdesk*; the motion simulator, *SAVEbase*; and the scenario builder, *SAVEace*.

The following *VE* projects provided us very important insights that we adopted in our solution. Next follows a brief review of the most important aspects of them.

ATLAS[13] studied *scalability* in *networked VEs* in terms of four scalability issues: *communication architecture*, *interest management*, *data replication* and *concurrency control*. It adopts a peer-server model with multicast support. A client-server model is also supported for flexibility reasons, although it may suffer from problems like server flooding. The interest management mechanism is based on user interests and spatial distance. Users with the same interests dynamically form a multicast group when they get close. Each user in the group multicasts update messages to the rest of the group whenever he/she moves or interact with the world. The concurrency control is based on an entity-centric prediction-based concurrency control scheme where only the users surrounding a target entity multicast the ownership request by using the multicast group address assigned to the entity. For data replication, *ATLAS* uses partial replication with on-demand transmission plus user-based caching and prefetching exploiting the object's access priority, which is defined based on spatial distance and individual user's interest in objects on the virtual world.

In the NPSNET-V[12] the authors propose the use of a component-based dynamic extensibility framework implemented in *JAVA™* to allow one to construct applications as components hierarchies rooted at an invariant microkernel. Applications could be implemented as a federation of dynamically loadable modules, loosely coupled by a minimal set of well defined relationships

MOVE is a *CVE* constructed on top of a component groupware framework[5], where users (avatars) can interact with each other or with shared artifacts. The proposal of the authors is to provide an extensible infrastructure offering a set of collaborative services in a seamless way. At the conceptual level, it provides essential collaborative services: shared sessions, support for synchronous and asynchronous components, security, coordination and a server-sided awareness infrastructure. At the architectural level, the framework is constructed on top of a middleware integration platform and uses high performance publish/subscribe notification services.

4. PROBLEM DEFINITION

In this work we focus on the main characteristics of Offshore Engineering projects as a case study. In this field research is being conducted to design oil production units, such as platforms, or adapt old ships to work as floating production storage offloading, for operating in ultra deep water, 400 m or deeper. The project of a new production unit is a very lengthy and expensive process; it can last more than a year and consume a few hundred million dollars, depending on the complexity of the unit and the availability of an adequate technology that makes the project technically and economically feasible.

Usually Offshore Engineering projects involve not only geographically distributed teams but also teams of specialists in different areas using different software tools, both commercial and homemade. Interoperability of those tools is still an issue in the industry and is a mandatory requisite for any viable collaborative solution. Due to their huge complexity, projects in this field are segmented, divided into smaller interrelated subprojects where each one deals with an equivalent representation of the others. During the conceptual design phase of the project the work is carried out basically, but not only, by the following teams:

1. *Naval engineers*: project the hull of the ship, defines 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.
2. *Structural engineers*: defines the internal structure of the unit and its load capacity.

3. *Production and equipment engineers*: project the production system, encompassing risers, flowlines, and plan the installation of production deep water equipments, such as manifolds and christmas trees.

4. *Chemical and process engineers*: projects the process plant based on the characteristics and expected volume of oil and gas that will be produced.

5. *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 activity that the necessity for collaboration is a must. Each team activity or new decision can affect others activities. For example changing the position of large and heavy equipment in the unit can compromise the stability of the ship. In some cases there is also an intrinsic coupling among different teams as in the case of the mooring system and the risers. In one hand if the mooring system allows great fluctuations of the ship, these can simply damage the production risers; on the other hand the presence of the risers helps to weaker the movements of the ship. In order to help users identify and solve conflicts like the one described earlier, we propose the creation of an *Agent-based Awareness mechanism.*

An additional difficulty presented in those projects is that, although the specialists deal with the same artifacts (platforms, production risers, mooring systems, etc.) they usually have different data representations for those objects according to the needs of each application, requiring that the solution provides some support for *multi-resolution representation of the data*. For example, in structural and naval engineering the models usually have dense polygonal meshes, with a few objects representing the outline of the artifacts, suitable for static and dynamic stability studies with some numerical methods such as finite and boundary element methods. In CAD/CAE the models usually have objects with coarse grid meshes suitable for giving a reasonable visual representation, but the problem is that all the objects that comprise the artifact should be represented yielding huge models. For real time visualization those models are almost intractable and, even today, represents a research challenge for computer graphics[9].

There are other important activities that will benefit with the existence of a *CEE* as defined in this paper, such as training and security simulations; design, planning and optimization of marine installations and sub-sea layout arrangement of production equipments; remote teleoperation and interventions on submarine equipments; preparing maintenance and inspections plans in production units; planning pipeline installations.

In order to better define our solution and its corresponding requirements, it is necessary to define the adequate collaboration needs of the Offshore Engineering projects scenario and which kind of collaborative application should be used.

4.1. Collaboration Levels

The model presented in Santos[19], defines hierarchical levels for collaboration scenarios (Figure 2) that serves as a guideline to incrementally develop collaborative applications. At each level, different collaboration degrees are supposed.

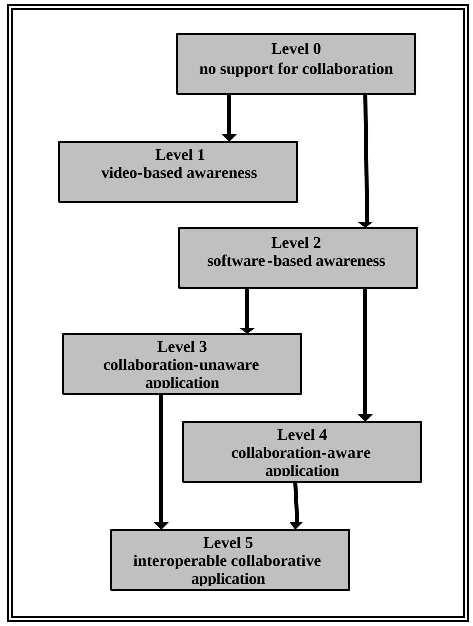

Figure 2. Hierarchy of collaborative scenarios.

At level 0, *no support for collaboration* is defined. At level 1, called *video-based awareness*, a higher degree of communication is achieved with integrated audio and videoconferencing system to the solution. At this level, the collaboration scenario is not complete, since the peer users are not able to interact with each other's application. At level 2, a degree of cooperation and coordination is possible, given the user capabilities to interact with the remote workspace. Level 3 gives collaboration support to applications that were originally developed to be single user, and do not provide explicit support for that. Applications at level 4 are similar to level 3, but the support for collaboration is provided by distributed applications especially developed for that purpose. At level 5, a framework for interoperability among different applications should be supported.

Example of *collaborative-unaware* applications is *Microsoft NetMeeting™* that provides an application-sharing support for the MS Windows. *Collaborative-aware* applications are distributed applications 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. The data sharing can be based on a centralized or replicated architecture. *CVEs* are examples of *collaborative aware* applications.

It is important to mention that the lower levels, 1 and 2, though having poorer collaborative resources, are easier to implement and, in some cases, are the only feasible solutions due to the available infrastructure and/or budget constraints. Moreover, in some cases where the most important tools used in the environment are commercial software with non-extensible functionalities it is not possible to reach higher collaboration levels, which require intrusive interventions in the software. Based on the description of the problem we can say that our *CEE* should constitute an application between levels 4 and 5 depending on the degree of interoperability that will be supplied to the users by the Shared Data Management System (*SDMS)* and Engineering Application System (*EAS)* components.

4.2. *CEE* Requirements

Based on previous works in the related area of *Adaptive Workflows*[22,11], *CVE*[7], *MMCS*[24] and *CSCW*[17], and on an analysis of the domain of *Offshore Engineering* in our scenario, we define a set of requirements for a *CEE*.

Communication support – electronic communication support is a fundamental requirement in our scenario. The *CEE* must provide different communication support possibilities: synchronous or asynchronous, enabled in various media types (audio, video and text based communication). This support should be provided in a seamless way, so that our users can start a communication of one or of another type while they are interacting with the *CEE*, or they should be able to plan certain time for a specific communication interaction. The communication support should be integrated to the other tools in the *CEE* and provide means of recording conversation and retrieving old ones. This requirement helps user solve their project's problems in critical situation, with fast interaction and negotiation, and it allows the recovery of useful pieces of communication used to solve similar problems in the past.

Coordination support – at the project management level, multiple and different visions of the on-going project must be provided by the *CEE*. Users have different background (managers, engineers) and need different types of information to execute their duties. Project management should also be feasible in a *CEE*.

Cooperation and flexibility support – there should be process model flexibility support, like dynamic change of process instances during run-time to support dynamically evolving processes, possibility of executing rollback of processes (e.g., 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 modeling 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 managed models, object interaction and manipulation, edition and planning and lately, access to organizational work history.

Awareness – there are different types of awareness that can be foreseen in a *CEE*. In our scenario, the most important ones are: *event monitoring infrastructure* - to observe what is going on 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* – to provide control of collaborative interaction and changing of the user location; *mutual awareness* – to allow users see each other's identity and observe each other's actions; *group awareness* – to facilitate the perception of groups of interest connecting people who need to collaborate more intensely.

Integration Management Infrastructure – at this level, several smaller services should be available in order to guarantee the data and modeling persistency, and the different levels of access control to different user roles in our scenario. Here we include the shared workspace and results service, access control service, user management service, data synchronization service, security service and software mediators.

CVE specific requirements – *VEs* architectures intended to support large shared virtual worlds over long periods of time requires: *high performance*; *scalability* to deal with virtual worlds which varies widely in size and number of participants; *a persistence mechanism* to save and restore world state between activations; *version-safe* updating mechanisms, because large and long-lived virtual worlds tend to incorporate different versions of the same components; *composability*, so that one may easily and effectively combine worlds and world components developed by different organizations; *dynamic extensibility*, i.e., to as large an extent as possible, the architecture must permit the seamless run time extension and replacement of any part of its hosted application.

5. SYSTEM ARCHITECTURE

Our proposed *CEE* has component based architecture (Figure 3) in order to facilitate the reuse of the elements. The architecture of the *CEE* uses a *WfMS* as its kernel while the *MMCS*, *CVE* and the other components are seamlessly accessed according to the collaborative necessities of the team workers.

The integration of the *WfMS* and *MMCS* follows the same approach suggested by Weber[23]. We have developed a Java/JMF based *VC* tool, *CSVTool*[16], that will be integrated into our *CEE*. It is important to say that although the proposed *CEE* will use the *CSVTool* the proposed architecture should keep it free to use any other possible *MMCS* system that might offer more resources in the future. The same is valid for the used *CVE* and the others components that take part in the solution. The *CVE* is being constructed on top of *Avango*[21], an object-oriented framework for distributed, interactive Virtual Reality applications.

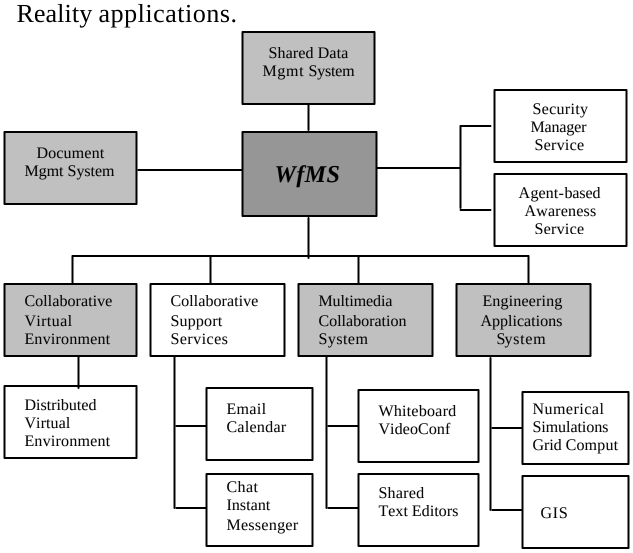

Figure 3. Components of the *CEE*.

All the consistency, adequacy and compatibility of the shared data among its users should be done by the kernel in conjunction with the *SDMS*, 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, will provide an optimized usage of the available computational resources and also a better control of the costs and the time scheduling.

We are using OpenORB to implement the architecture of the system with the following CORBA™ services: *Persistence, Life Cycle, Trading, Event* e *RelationShip* (Figure 4). The user interacts with the system through its GUI and from there the *WfMS* guides the execution accessing the other components accordingly.

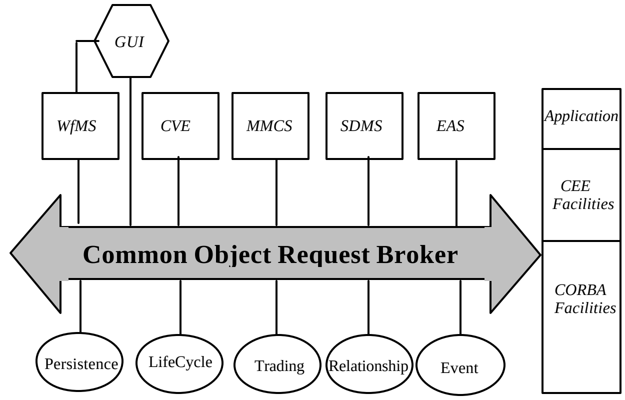

Figure 4. CORBA architecture of the *CEE*.

In this architecture the requirements are fulfilled by different association of the components, Communication: *MMCS and Collaborative Support Service;* Coordination: *Adaptive WfMS;* Cooperation: *Adaptive WfMS, DMS, SDMS; Collaboration: CVE; Awareness: Agent-based Awareness Service.*

6. CONCLUSION

This paper presented a set of requirements and system architecture of the *CEE* that we are currently undertaking. There are many open doors yet, especially concerning the better way on how to connect all the components. As next steps of this work, we plan to continue refining the architecture of the *CEE*, and as a proof of concept we intend to develop a prototype that will be used by the Offshore Engineering group at Petrobras. Usability studies will follow after that in order to check the usefulness of the system.

Through the use of the *CEE* we expect to have an effective collaborative environment that will allow users to easily mitigate their problems that usually happens during the execution of large and complex engineering projects. We also intend to improve the effectiveness of the use of VR technology once that it will be easily integrated in the work flow of the team workers. We expect that our *CEE* will constitute a very effective problem solving tool for the engineers in our company.

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 as well.

ACKNOWLEDGMENTS

The authors would like to thank Petrobras, Tecgraf/PUC-Rio and Fraunhofer Gesellschaft (*FhG)* for the expressive support. Ismael Santos works at CENPES, Petrobras's Research Center, and currently is working as a visiting researcher at *FhG*, Sankt Augustin, Germany. Alberto Raposo receives an individual grant from CNPq (Brazilian Research Council). The authors would also like to thank Carla Valle and Gernot Goebbels for the fruitful discussions that helped to conduct this work.

REFERENCES

1. Bentley, R., Hughes, J. A., Randall, D., Rodden T., Sawyer, P., Shapiro, D., Sommerville, I. Ethnographically-informed systems design for air traffic control. *In The Proceedings of the ACM Conf. on CSCW'92, Toronto, Ontario, CA, 1992.*
2. Daily, M., Howard, M., Jerald, J., Lee, C., Martin, K., McInnes, D., Tinker, P. Distributed Design Review in Virtual Environments. *In the Proceedings of the Third ACM International Conf. on CVE'00, San Francisco, CA, USA, 2000.*
3. Ellis, C. A., Nutt, G. J. Workflow: The Process Spectrum. *In NSF Workshop on Workflow and Process Autom. in Info. Systems, Athens, GE, 1996.*
4. Fröhlich, B., Barrass, S., Zehner, B., Plate, J., and Gobel, M.. Exploring GeoScience Data in Virtual Environments. *In Proc. IEEE Visualization, 1999.*
5. García, P., Montal, O., Pairot, C., Rallo R., Skarmeta, A., G. MOVE: Component Groupware Foundations for Collaborative Virtual Environments. *In The Proceedings of the 4th ACM Internat. Conf. on CVE'02, Bonn, Germany, 2002.*
6. Goebbels, G, Lalioti, V., Göbel, M. Design and Evaluation of Team Work in Distributed Collaborative Virtual Environments. *In The Proc. of the ACM Symposium on Virtual Reality Software and Technology, VRST'03, Osaka, Japan, 2003.*
7. Greenhalgh, C. M., Benford, C. D. Supporting Rich and Dynamic Communication In Large Scale Collaborative Virtual Environments. *In Presence: Teleoperators and VEs, vol 8(1), MIT Press, 1999.*
8. Haller, M., Holm, R., Volkert, J., Wagner, R. 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.*
9. Hofmam, M., Gattass, M., and Celes, W. Organization of Large Engineering Models for Real-Time Walkthrough Rendering. *In The Intern. Conf. on Comp. Engineering&Sciences,ICES, 2002.*
10. Joeris, G. Cooperative and Integrated Workflow and Document Management for Engineering Applications. *In 8th Intern. Workshop on Database and Expert Systems Applications, DEXA, 1997.*
11. Kammer, P. J., Bolcer, G. A., Taylor, R. N., Hitomi, A. S., Bergman, M.. Techniques for Supporting Dynamic and Adaptive Workflow. *In ACM CSCW Vol. 9. Kluwer Academic Publishers, 2000.*
12. Kapolka, A., McGregor, D., and Capps, M. A Unified Component Framework for Dynamically Extensible VEs. *In The Proc. of the 4th ACM Intern. Conf. on CVE'02, Bonn, Germany, 2002.*
13. Lee, D., Lim M. and Han, S. ATLAS - A Scalable Network Framework for Distributed Virtual Environments. *In The Proc. of the 4th ACM Intern. Conf. on CVE'02, Bonn, Germany, 2002.*
14. Leigh, J., Johnson, A., Brown, M., Sandin, D. and DeFanti, T. *Visualization in teleimmersive environments. IEEE Computer, vol 32(12), 1999.*
15. Nikitin, I, Nikitina, L., Frovlov, P., Goebbels G., Göbel, M. Real-time simulation of elastic object in VEs using finite element method and precomputed Green's functions. *In The Proc. of 8th Eurographics Workshop on VEs, Barcelona, Spain, 2002.*
16. Pozzer, C. T., Raposo, A. B., Santos, I. H. F., Campos, J. L. E., Reis, L. P.. CSVTool - A Tool for Video-Based Collaboration. *In The Proc. Of Webmídia, Brasil, 2003.*
17. Reinhard, W., Schweitzer, J., Völksen, G. CSCW Tools: Concepts and Architectures. *In The IEEE Computer, vol 27(5), 1994.*
18. Sallés, E. J., Michael, J. B., Capps M., McGregor D., Kapolka, A. Security of Runtime Extensible Virtual Environments. *In The Proc. of the 4th ACM Intern. Conf. on CVE'02, Bonn, Germany, 2002.*
19. Santos, I. H. F., Raposo, A. B. and Gattass, M. Finding Solutions for Effective Collaboration in a Heterogeneous Industrial Scenario. *In The 7th Inter. Conf. Comp. Supp. Coop. Work in Design, CSWID'02, Rio de Janeiro, Brazil, 2002.*
20. Sevy, J., Zaychik V., Hewett T.T., Regli, W. C. 2000. Devel. Evaluating Collab. Engin. Studios. *In The 9th Proc. of WET ICE'2000, USA, 2000.*
21. Tramberend H., 1999. Avocado: A Distributed Virtual Reality Framework. *In The Proc of the IEEE Virtual Reality, 1999, http://www.avango.org*
22. van der Aalst, W.M.P. Flexible Workflow Management Systems: An Approach Based on Generic Process. *In The Proc. of DEXA, 1999.*
23. Weber, M., Partsch, G., Hock, Schweitzer, J. Integrating Synchronous Multimedia Collaboration into Workflow Mgmt. *In The ACM SIGGROUP, Phoenix, USA, 1997.*
24. Weske, M., Vossen, G., Medeiros. C.B., Pires, F. Workflow Management in Geoprocessing Apps. *In Tech. Report No.04-98-I, Univ.Muenster, GE, 1998.*
25. Workflow Reference Model, 1995. In WfMC - Workflow Management Coalitition. *Document No. TC00-1003, Issue 1.1, 1995.*