A CORBA-based System for the Monitoring of Petroleum Fields

Diogo Salim[1,2], Thais Batista[1], and Adelardo Medeiros[2]

[1] Informatics Departament, Federal University of Rio Grande do Norte, Campus Universitário Lagoa Nova, 59072-970, Natal, RN, Brazil
{diogo@interjato.com.br, thais@ufrnet.br}
[2] Department of Computing Engineering and Automation, Federal University of Rio Grande do Norte, Campus Universitário Lagoa Nova, 59072-970, Natal, RN, Brazil
adelardo@dca.ufrn.br

Abstract. This paper presents the definition and implementation of a CORBA-based system for the distributed monitoring of automated oil wells located in on-shore petroleum fields. Conducted by the automation of industrial processes in the oil and gas area, the development of computational distributed systems, based on emerging middleware technologies and designed to monitor and actuate in such particular domain, has been strongly worldwide stimulated. The monitoring of the oil wells' behavior can serve as a powerful tool for the oil production analysis and for the verification of the decisions' correctness made throughout their existence. Since the monitoring system takes part of a larger oil production system with heterogeneous and distributed modules, CORBA and its services provide support for transparent interaction among such modules. The system implemented in this work has the following main goals: to register the individual characteristics associated with the operational process of oil extraction in each automated well of a oil field, and based on this collected and stored information, to provide descriptive and behavioral results in order to make possible an efficient monitoring process and to also supply control procedures. With the purpose of supporting multiple clients' invocations, the system provides concurrent monitoring, control tasks and data distribution.

1 Introduction

Distributed applications are steadily growing in various computational areas. With the advent of the Internet and its respective network communication protocols, the occurrence of a relevant computational progress in this specific field has been observed. Several middleware standards supporting the development of distributed applications are now defined and therefore, the implementation of a wide set of tools applied to this particular use occurs in an accelerated technological pace.

Nowadays, there is a strong tendency to integrate aspects, such as distribution, dynamicity and heterogeneity, not only in new computational systems but also to those, which have attained certain market maturity, e.g. the legacy systems, and have not incorporated such aspects. They face the necessity to be updated with this new evolution of software engineering by exploring the facility of remote access provided by the Internet.

In order to monitor oil fields automated with the most worldwide employed on-shore elevation process – the Mechanical Rod Pump, a computational distributed support is crucial for an efficient and optimized remote monitoring process. This oil pumping system is characterized by the fact of involving a great number of instrumented wells and consequently a large amount of information to be supervised, and to also present a specific and individual behavior due to each well's peculiarities.

Based on this scenario and the remark that the technological evolution, conducted by the automation of industrial processes in the oil and gas field, has stimulated production of computational systems designed to monitor and actuate in the behavior of such processes, in this work we describe a distributed system used in the acquisition and manipulation of oil reservoirs' data. The on-line monitoring of the oil wells' behavior is one of the major critical concerns for the oil companies since it serves as a tool for the production analysis and for the verification of the decisions' correctness made throughout their existence. Another important factor is that the behavior's analysis provides relevant data in order to predict how each well and its respective mechanical rod pump's system can behave in the future.

This work consists of the major step in the implementation of a distributed and dynamic system for monitoring of oil reservoirs. Such step involves the development of the computational architecture of the correspondent system: the definition and implementation of its parts. The implementation uses a platform that offers support to the development of heterogeneous distributed systems: the CORBA architecture. Thanks to its well-known features (support for transparent communication between heterogeneous and distributed components) and enhanced services, the middleware CORBA can be very effective for this respective monitoring activity. The reason for choosing this platform consists in the fact that CORBA is an open source specification that, de facto, facilitates the development of flexible distributed applications and which has also provided to our software framework other relevant features, such as reusability, multiplatform interoperability and the capacity to integrate new enhancements (e.g. CORBA services) and include complementary modules.

This paper is organized as follows. Section 2 presents the basic concepts regarding to the monitoring process of oil wells and to the middleware platform used as the infrastructure for the monitoring process. Section 3 presents the architecture of the monitoring system. Section 4 presents the CORBA-based monitoring system developed in this work. Section 5 comments about related works. Section 6 presents the final remarks and future works.

2 Background

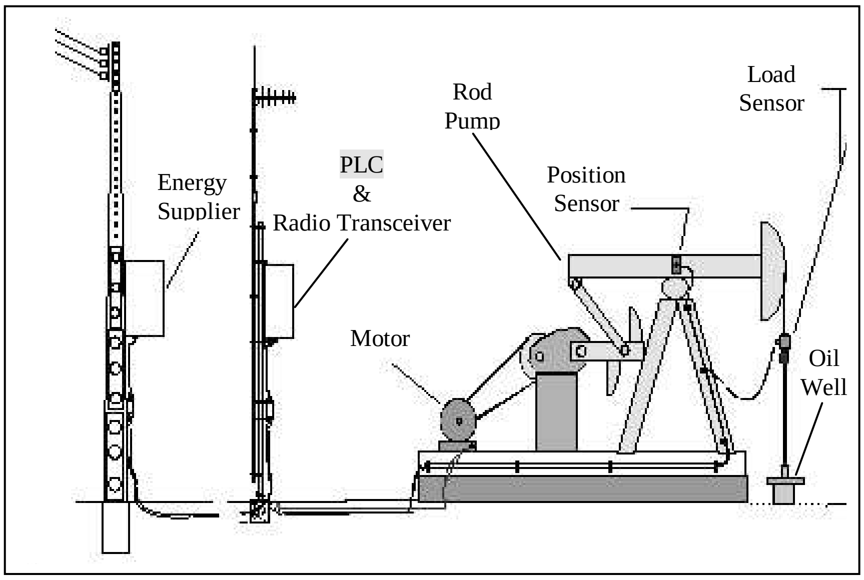


Fig. 1. Hardware structure installed in instrumented oil well

2.1 Monitoring Process of Oil Wells

The monitoring process actuates in wells perforated into large onshore oil reservoirs, which utilize a standardized mechanical system – *the Rod Pump* system – as the artificial elevation method for oil and gas extraction. Such onshore elevation method is worldwide the most employed in the oil industry. Since it is characterized by the fact of involving a great number of instrumented wells and consequently a large amount of information to be supervised, and to also present a specific and individual behavior due to each well's peculiarities, a rigorous operational distributed monitoring is necessary for an appropriate and optimized oil pumping.

The system presented in this work has been designed to monitor and control the rod pump wells located in the onshore oil production fields of the Brazilian petroleum company – Petrobras. The number of oil wells that are operated with such artificial elevation method can reach up to a thousand units. Therefore, because of the enormous quantity of units to be monitored and the previous existence of part of the wells' instrumentation, this project has been implemented in a way to incorporate some of the previously installed *hardware* structure. Figure 1 illustrates the hardware structure installed in instrumented oil well and its main components: rod pump, sensors, micro-controller, radio transceiver and energy supplier.

2.2 CORBA

Bla ……

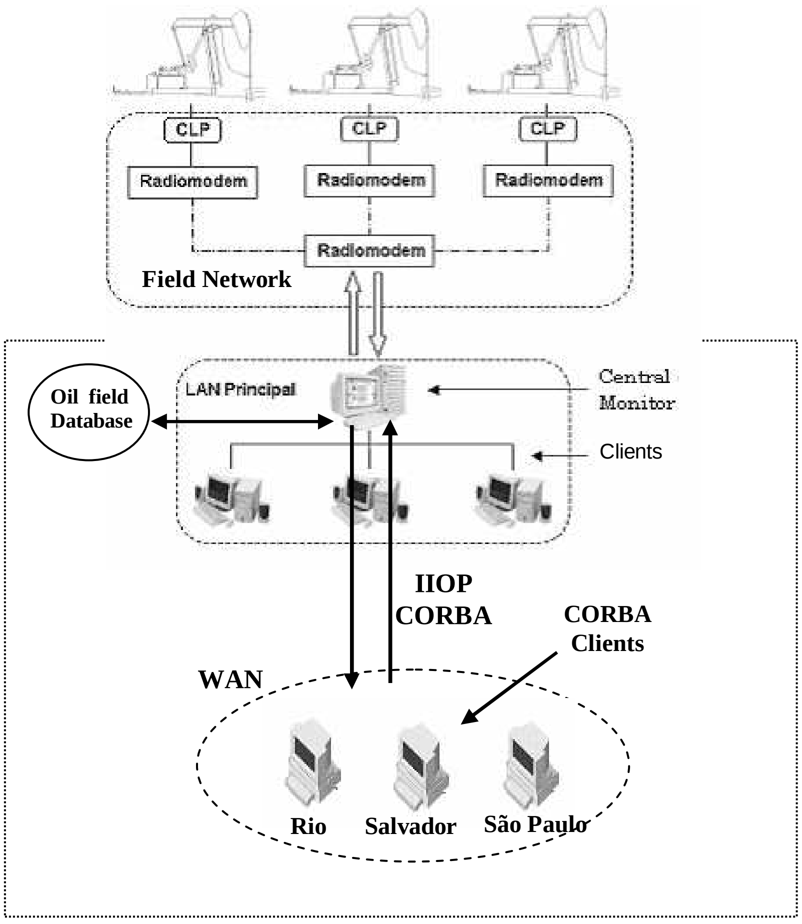


Fig. 2. Architecture of the Monitoring System

3 The Architecture of the Monitoring System

Building such application is a very demanding effort since it must satisfy a wide set of requirements deriving from several research areas, arising from the concepts and applications in automation and instrumentation fields, development and implementation of wireless communication, and also the implementation of a distributed system that can be composed of heterogeneous modules. We emphasize that the main focus of the present paper is to describe the study of data distribution and the integration of the parts of the computational framework supplied by the CORBA pattern.

The physical parts of the system are divided into four major sub-areas, as illustrates Fig. 2: the hardware and instrumentation of the rod pumps, the field communication net, the monitoring central, and the client.

The main parts of the **hardware and instrumentation of the rod pumps** are the position and load sensors, a Programmable Logic Controller and a radio modem with baud rate of 9600 bps. The power actuators of the motor are controlled by the PLC to maintain a constant rotation velocity: the set-points of the controlled variables (velocity, idle time, etc.) are received through the radio link.

The **field communication network infrastructure** was developed in order to provide a master-slave communication between the Programmable Logic Controllers (PLC) installed in every rod pump well and the Central Monitor. The implementation of such wide net was done during the process of automation of the oil wells and partially incorporated into this project. Since most of the rod pumps are usually located in remote and uninhabited areas, information between hosts is transported via a wireless communication by means of radio modems installed and connected to the Programmable Logic Controllers of every instrumented rod pump.

The **Central Monitor** (CM) is one of the main components of this distributed system. It is defined as the link between the universe of the monitored oil reservoirs and the clients present either in a local network or even connected in a wider net, such as the Internet. In one side, its major task is to communicate with all the wells registered in a given oil field, to continuously store the collected data into a database, and to conduct the remote monitoring of the entire process of oil extraction done by artificial elevation. On the other hand, this CM supports also distributed and heterogeneous aspects in order to make possible an IIOP communication between remote Clients and Servers based on CORBA.

The following tasks consist in the major events performed either by CM or via a teleoperation control accomplished by a remote client: (1) **Automatic Data Sampling**: a periodic task that accesses the monitoring data collected and stored in every well pertaining to a specific oil field and transmits it to CM; (2) **System Monitoring**: it is a task directly associated with the *Automatic Data Sampling*. After updating the received data, the system processes and tests the integrity and status conditions of the corresponding information. Several procedures are implemented in order to check and analyze the operational pumping of each well. Based on the collected information, automatic or human-assisted decisions can be taken depending on the encountered conditions. Such decisions are mainly the adjustment of variables associated with the rod pumping process. The major goal is to obtain an optimal condition for the functioning of all wells being monitored by CM. Some efficient algorithms have been developed and are capable to simultaneously optimize the oil extraction and to reduce energy consumption spent by the electrical components (motor, controllers, etc.) of the rod pumps.

The **client** is the module of the system by which the users can access the oil monitoring application and remotely operate it. Since its implementation is based on the CORBA standard, the communication and interaction with the CM is carried out in a complete distributed and transparent mode, performing thus remote data requisitions and control procedures made by the users.

4 A Framework for Monitoring based on CORBA

Once the parts of the monitoring system, such as the hardware, the wells' instrumentation and the field communication network infrastructure, have been previously defined, this section describes the development and implementation steps of a CORBA based model capable to provide dynamicity, data distribution and heterogeneity to the monitoring system. The designed model was conceived in a way that the system has the potential not only to monitor the different parameters and behaviors associated with onshore oil extraction process but also to execute control tasks performed by remote users.

4.1 CORBA Client

One of its major tasks is to initialize the communication between the CORBA Client process and users in order to start and bind a remote connection. This Client process receives different types of data sent by the users via an interface, e.g., all the information and requisitions permutated between the CORBA Client and the connected users can be executed and performed by means of standard well known Internet protocols, such as HTTP, or either by some specific CORBA implementations which support Web services (VER referencia do TAO com Web).

When an user starts a connection with the CORBA Client and requires data related to the monitoring process of oil extraction, an IIOP communication is established between the CORBA Client and the corresponding Server being executed in the Central Monitor. The CORBA client is, de facto, in charge to provide to the users all the information required. The data associated with the process of wells' monitoring and those related with control commands are analyzed and transmitted through the network by means of specific procedures defined and implemented in this monitoring system. Such procedures are responsible to establish and to maintain the communication between the CORBA Client and Server and to provide methods that support users requests of data and tasks.

The tasks and information that compose the data requisition process associated with the rod pumping system consist in typical monitoring methods and the execution of control procedures as well. Among these tasks defined in this project, the main ones are listed as follow.

- Request for acquisition and storing of specific information (Dynamometric card, operational and status data, etc.) associated with either an individual instrumented well or an entire set of wells operating in a oil reservoir;

- Control procedures associated with the mechanical pumping system;
- Adjustment of variables responsible for timing, potency control and energy consume associated with the oil pumping operation of each automated well.

4.2 CORBA Server

All tasks performed by the CORBA Server are directly related to data supervision and distribution context. According to the implemented model of the system, the Server is in charge to provide efficiently each data proceeding from the users' solicitations. However, certain informations (mostly the real-time operational characteristics of oil extraction) are not necessarily always present in the Central Monitor. Such information are subjected to a continuous process of storage and monitoring, but both real-time services usually take place in their respective field locations of oil exploration and elevation by means of the hardware (Programmable Logic Controllers) installed in every rod pumping well. Therefore, one of the major tasks of the CORBA Server is described as the free-error capture of such information and an efficient database storing, all in order to provide to the CORBA Clients the most possible realistic information. Once an entire collection of data associated to a set of rod pump wells is available, the CORBA Server is now ready to filter, accept and process users' solicitations – monitoring and control tasks or packages of messages consisting of handshake, acknowledge, requests of ending the connection, and many others. Table 1 shows the variables used in the monitoring process.

For the proposed model, the parameters defined in the IDL interface consist of informative and functional data, e.g., part of those parameters are associated directly with variables of the monitoring process; others are used to assist some steps inherent to the communication process between the CORBA Server and Clients, for instance: CORBA Clients have an unique identifier that must necessarily be furnished to the Server every time a task requisition has been made.

Table 1. Variables used by the Monitoring Process

VARIABLE	DESCRIPTION
Run_time	Operating time of a single rod pump well
Idle_time	Idle time of a single rod pump well
Last_update	Last recorded time of data updating from a specific well
Dynam_card	Dynamometric card represent by a set of 128 pairs of position samples
Pump_status	The operational status of the pumping process of a specific well
Field	Identification number of the monitored oil exploration field
Well	Identification number of the monitored oil rod pump well
Field_names	Array identifier of the field´names supervised by the Central Monitor
Task	Identifies the types of monitoring and control solicitations
Tag_	Auxiliary variable in the process of message exchange

ID_sender	Unique identifier number of a connected CORBA client

The IDL interface defined for this CORBA application can be divided into two parts: the first one consists of a set of **structs** in which parameters associated with the variables of the remote monitoring system are defined; the second part is the interfaces definition of the server and client objects. For each interface, a group of methods have been defined and implemented.

Figure 3 shows….

Figure 4 illustrates….

```
#if !defined (_EVENT_COMM_IDL)
#define _EVENT_COMM_IDL
module Event_Comm
  {//  CORBA IDL Module for distributed event notification
  struct data{ // well data
      long idle_time;          // idle time set for well
      long Last_update;        // time of last updated data
      float carta_dinam [256]; // dinam card: 128 pairs
      long pump_status;        //pumping conditions
   };
  struct attribute{
      long task;        //task identifier
      long field;       // field ID
      long well;        //well ID
      data data_well;   //struct containing well's data
   };
  struct Event {// Define interface for an event <Event>.
      string tag_;        // monitoring auxiliary variable
      object object_ref_; // callbacks Object reference.
      long ID_sender;     //identifies the client
      attribute well_attribute;//struct of well's attributes
      string field_names[20];  //field names of a CM
   };
```

Fig. 3. Monitoring Parameters Definition

```
interface Consumer
{ // Define an interface for <Consumer> on events occurrence
  void push (in Event event_instance);
  void disconnect (in string reason);
};

interface Notifier
{ // Define an interface for <Notifier> on events occurrence
  exception CannotSubscribe
    string reason_;
```

```
     };
     exception CannotUnsubscribe
       string reason_;
     };

     void disconnect (in string reason);
     void push (in Event event_instance);
     void subscribe (in Consumer subscriber)
          raises (CannotSubscribe);
     void unsubscribe (in Consumer unsubscriber)
          raises (CannotUnsubscribe);
  };
};
#endif /* _EVENT_COMM_IDL */
--------------------------------------------------------------------------------------------------------------------
```

Fig. 4. Client and Server IDL interfaces

4.3 The Communication Steps between Server and Client

Step 1: Searching the Server via *Naming Service*

The first step in order to integrate the different parts of this CORBA-based system consists in defining and searching the specific server that the client needs and present among the possible centrals monitors. The Brazilian onshore oil extraction system, instrumented with rod pumps, is divided in oil fields, which are composed of groups of wells. Since there is a monitoring central in every oil field and consequently its correspondent CORBA Server, the first communication step is to provide a way that clients can determine, unequivocally, the correct searched server. This initial task of hosts' identification is performed by the CORBA *Naming Service* [Referencia ao Naming Service]. In view of the fact that distribution is one of the major features of this correspondent system, the use of the naming service is crucial for the accomplishment of the communication between processes being executed in hosts physically distinct, furnishing thus a greater transparency.

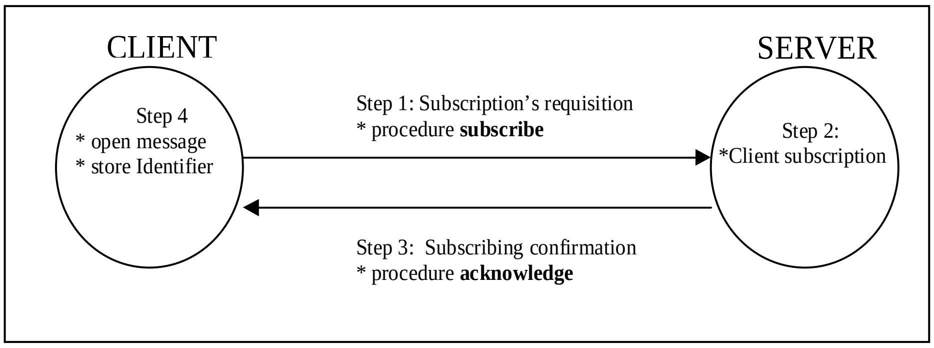
After the specified host has been successfully found, a next phase, defined here as ***Handshake***, surges in the communication process.

Step 2: Handshake

Among the architecture' features of distributed systems, this project specifies a communication with connection between client and server. For such purpose, a recognition and authentication step made by the server is executed every time a new client sends a connection's solicitation. The client must first carry out a requisition for a dynamic subscription in the server by means of a remote procedure call.

The server proceeds first by verifying the client's solicitation. Then, the corresponding new client is subscribed and added to the mapping list of online subscribed clients. At the end of the subscription's process, a final procedure, here denominated ***acknowledge***, is performed. Its function is to inform the client the

acknowledge, is performed. Its function is to inform the client the confirmation of its well-succeed connection and a collection of basic information, such as the client's identifier number, data associated with the chosen monitored oil fields and their respective oil wells and the offered operational control tasks. The generation of a unique identifier to the client is crucial for the designed system, since in the following communication steps, every task for data requisition or control procedure has to


CLIENT
SERVER
Step 4
* open message
* store Identifier
Step 1: Subscription's requisition
* procedure **subscribe**
Step 2:
*Client subscription
Step 3: Subscribing confirmation
* procedure **acknowledge**

include the identifier of the matching client.

Fig. 5. Steps in the Handshake process

Step 3: Tasks Solicitations via CORBA

Once the previous step has been completed, the new client is able to send solicitations to the server in order perform the various tasks offered by the system. The system is implemented in a way to adapt to the different variations of configuration, permitting new fields and wells to be added and monitored in execution time and to also remove those which are no longer in the operational phase without affecting the efficiency of the system.

The solicitations are carried out through calls of specific methods (push or pull) provided by the CORBA Event Service [9 da monografia]. Given that the type of transmission defined for data transfer among hosts of the monitoring system is asynchronous, the use of the event service is needed since it supports asynchronous communication between remote objects. The clients solicitations can be performed at any time. Therefore, the CORBA server, instead of keeping itself waiting indefinitely for client's solicitations, can take advantage of the event service and so perform meanwhile the other major monitoring tasks, such as collecting and storing the updated data of each individual oil well. Another advantage derived from the event service's use in such application is the low memory allocation and processing time required for the execution and operation of the CORBA server process in the monitoring central hosts.

It is important to remark that, since the interaction between clients and servers are carried out through calls of remote procedures, the data, the acknowledge messages and control tasks are defined according to the variables and methods previously declared in the CORBA IDL interfaces.

```
   void Notifier_i::push (const Event_Comm::Event &event

ACE_ENV_ARG_DECL)
      ACE_THROW_SPEC ((CORBA::SystemException))
   {
      ACE_DEBUG ((LM_DEBUG,"in Notifier_i::send_notification =
                    %s\n",(const char *) event.tag_));
      MAP_ITERATOR mi (this->map_);
      int count = 0;
      //execute the client requisition
      Event_Comm::Event event_resp;
      event_resp = treat_req (event ACE_ENV_ARG_PARAMETER);
       // server responds to client according to the req. made
       // search client in the mapping list
      for(MAP_ENTRY *me = 0; mi.next (me) != 0; mi.advance ())
      {
         Event_Comm::Consumer_ptr consumer_ref=
                     me->int_id_consumer ();
         ACE_ASSERT (consumer_ref != 0);
         //execute a push to client
         if(event.ID_sender==(long)consumer_ref){
            ACE_TRY
            {
              consumer_ref->push (event_resp
                                   ACE_ENV_ARG_PARAMETER);
              ACE_TRY_CHECK;
            }
            ACE_CATCHANY
            {
              ACE_PRINT_EXCEPTION
                 (ACE_ANY_EXCEPTION,"Unexpected exception\n");
            }
            ACE_ENDTRY;
            break;
         }
      }
   } //end of push procedure
```

Fig. 6. Naming service - Redefinition of the *push* method.

4.4 Tests and Results

In order to evaluate the correct definition and implementation of each module defined in this project and to also evaluate the different factors associated with the system's performance, several tests have been carried out. The factors' analysis consisted in verifying if the computational distributed structure was correctly designed to assure the major concern: the capacity to efficiently monitor oil fields instrumented with rod pump wells. Therefore, the simulated tests seek to identify if the development of the system's parts and their respective integration suit all the requirements defined for such dedicated remote monitoring model. For that purpose, tests have been executed through simulations of oil fields and associated rod pump wells, which were submitted to a distributed monitoring performed by remote clients. The simulations preserve the authenticity and consistence of the main characteristics defined in the petroleum engineering area and consequently in the process of oil extraction executed by means of the artificial elevation – the rod pump system.

The tests' method focused in simulations between server and clients' objects in a distributed environment. The division of the system's modules in such distributed network followed several different tests' configuration, where the server module, the naming service, and a certain n-ary amount of clients were invocated and executed in specific hosts based on the various combinations defined for every test's configuration.

The obtained results revealed not only the correct suitability and functioning of the system's modules to the required architectural features but also the effective capacity to execute the designed tasks related to the monitoring process. Another confirmed aspect consists in the efficiency of the transparency provided by CORBA and consequently present in this respective system.

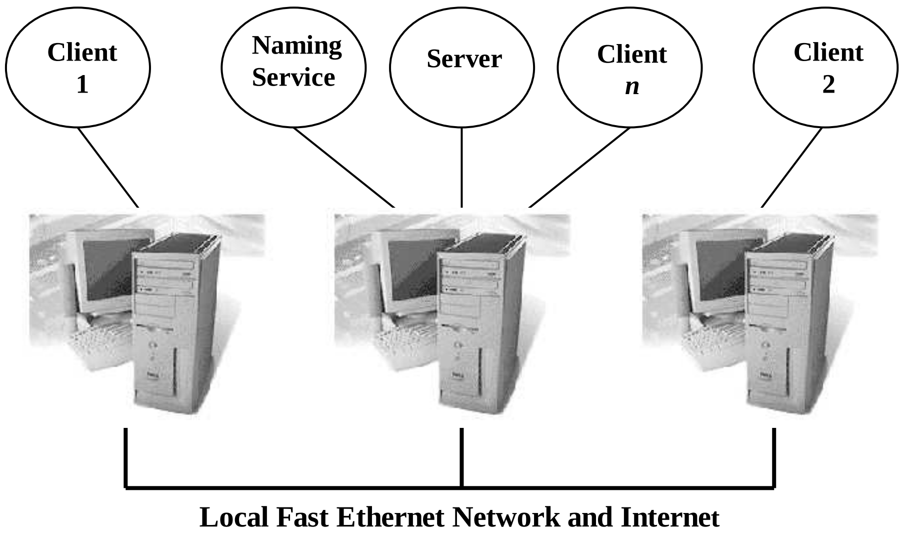


Fig. 7. Simulations of oil fields in a distributed environment.

5 Related Works

Souza and Medeiros [5,6] proposed a standard for interconnecting wells to the Central Monitor (CM). This system is already being used by Petrobras to monitor some oil wells, allowing the presented CORBA-based system to be used with real data.

6 Conclusions and Future Work

The computational system implemented in this work is an useful monitoring tool for distributed oil fields that are composed of a group of wells and are remotely controlled and monitored. It brings an economic-operational contribution to the oil industry by registering relevant information related to the oil pumping process of oil fields and providing ways to distribute such information to users remotely connected. It consists of a computational tool by which production analysis and the behavior of the pumping process can be performed. Such analysis can provide methods for the optimization of fluids extraction and energy consume, and can also assist to secure and maintain the physical structures of the instrumented mechanical rod pumps along the operational life cycle. In order to satisfy the features required by this specific teleoperated monitoring process, the designed system was implemented by means of the CORBA standard and services. The results obtained show that by implementing the monitoring system based on the CORBA standard, a number of remarkable features in this specific oil-monitoring domain have been brought, providing thus portability, reconfiguration and support for transparent communication between heterogeneous and distributed components.

We now plan to extend the system with further advanced CORBA-based computational aspects. The major one consists in turning the system into a model capable to support real-time features associated with some critical characteristics of the monitoring process. Since the CORBA implementation of the CORBA used in this project is TAO – a CORBA 3.0 version middleware specified to support real-time aspects, among others services – such computational upgrading should not be, de facto, a costly time and effort concern. Such tool is capable not only to assist but also to accelerate the development process of real-time distributed computational systems.

Based on the remark that the system is primarily designed for the monitoring of oil wells instrumented with a specific artificial elevation method (Mechanical Rod Pump), another relevant proposal is to expand the scope of the system's application in order to monitor wells instrumented with other artificial elevation methods, such as the *Gas Lift*. Motivated by the potential reusability of the designed and implemented modules defined in this project and present in the CORBA standard and services as well, the expansion of the application capacity can provide to the users an effective and more generic monitoring system.

References

1. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1 (1997) 108–121
2. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Abadi, M., Ito, T. (eds.): Theoretical Aspects of Computer Software. Lecture Notes in Computer Science, Vol. 1281. Springer-Verlag, Berlin Heidelberg New York (1997) 415–438

3. van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. Lecture Notes in Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)
4. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
5. Souza, Rodrigo B.; Medeiros, Adelardo A.D. Um Padrão para Interconexão entre Redes de Campo e de Supervisão e sua Aplicação na Indústria do Petróleo (A Standard for Interconnecting Field Networks and Supervision Networks and its Application in Oil Industry). VI IEEE Induscon - Conferência Internacional de Aplicações Industriais. Joinville, SC, Brazil – available in http://www.dca.ufrn.br/~adelardo/ (2004)
6. Souza, Rodrigo B.; Medeiros, Adelardo A.D. Um Padrão para Interconexão entre Redes de Campo e de Supervisão em Ambiente Industrial (A Standard for Interconnecting Field Networks and Supervision Networks in Industrial Environment). XV CBA - Congresso Brasileiro de Automática. Gramado, RS, Brazil – available in http://www.dca.ufrn.br/~adelardo/ (2004)