Enterprise Information Systems III

Edited by Joaquim Filipe, Bernadette Sharp and Paula Miranda

Kluwer Academic Publishers
Enterprise Information Systems III

edited by
Joaquim Filipe
School of Technology,
Polytechnical Institute, Setúbal, Portugal

Bernadette Sharp
School of Computing,
Staffordshire University,
Stafford, England, U.K.

and

Paula Miranda
School of Technology,
Polytechnical Institute, Setúbal, Portugal

KLUIWER ACADEMIC PUBLISHERS
DORDRECHT / BOSTON / LONDON
# TABLE OF CONTENTS

Preface................................................................................................................. ix

Conference Committee....................................................................................... xi

## INVITED PAPERS

<table>
<thead>
<tr>
<th>Title</th>
<th>Author(s)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obtaining Precision When Integrating Information</td>
<td>Gio Wiederhold</td>
<td>1</td>
</tr>
<tr>
<td>From Object-Oriented Programming to Ontology-Driven Modeling</td>
<td>Jean Bézivin</td>
<td>15</td>
</tr>
<tr>
<td>Agents and Electronic Commerce, Hype and Reality</td>
<td>Frank Dignum</td>
<td>22</td>
</tr>
<tr>
<td>Must a Computer Have a User?</td>
<td>Anatol Holt</td>
<td>36</td>
</tr>
<tr>
<td>Resource Allocation in Large Scale Systems, A Decoupling Approach</td>
<td>Nuno C. Martins, Saligramma Venkatesh and Munther A. Dahleh</td>
<td>41</td>
</tr>
</tbody>
</table>

## PART 1 – DATABASES AND INFORMATION SYSTEMS INTEGRATION

<table>
<thead>
<tr>
<th>Title</th>
<th>Author(s)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>IODBCon: An Integrated Object-Oriented Database System for Integrating Information About Architecture Design and Construction Processes</td>
<td>Farhi Marir and Yau Jim Yip</td>
<td>51</td>
</tr>
</tbody>
</table>
OBJECT EVOLUTION MECHANISMS IN OBJECT DATABASES
Slimane Hammoudi ........................................................................................................ 59

MODELLING DATA INTEGRATION IN AN OBJECT BASED GEOGRAPHICAL INFORMATION SYSTEM
Maria Luisa Damiani .................................................................................................... 68

DATA WAREHOUSE STRIPING: IMPROVED QUERY RESPONSE TIME
Jorge Bernardino and Henrique Madeira ........................................................................ 75

EVALUATION OF A VISUALISATION DESIGN FOR KNOWLEDGE SHARING AND INFORMATION DISCOVERY
Luis Borges Gouveia and Feliz Ribeiro Gouveia ............................................................ 83

NOVEL DATA VISUALISATION AND EXPLORATION IN MULTIDIMENSIONAL DATASETS
Nikolaos Kotsis, George R. S. Weir, John D. Ferguson and Douglas R. MacGregor ............................................. 90

SUPPORTING DECENTRALISED SOFTWARE-INTENSIVE PROCESSES USING ZETA COMPONENT-BASED ARCHITECTURE DESCRIPTION LANGUAGE
Iliam Alouì and Flavio Oquendo ...................................................................................... 97

THE DECOR TOOLBOX FOR WORKFLOW-EMBEDDED ORGANIZATIONAL MEMORY ACCESS
Andreas Abecker, Ansgar Bernardi, Spyridon Ntioudis, Gregory Mentzas, Rudi Herterich, Christian Honty, Stephan Müller and Maria Legal ......................................................... 107

PART 2 – ARTIFICIAL INTELLIGENCE AND DECISION SUPPORT SYSTEMS

MULTI-AGENT DYNAMIC SCHEDULING AND RE-SCHEDULING WITH GLOBAL TEMPORAL CONSTRAINTS
Joaquim Reis and Nuno Mamede .................................................................................. 117

AN INTELLIGENT AGENT-BASED ORDER PLANNING FOR DYNAMIC NETWORKED ENTERPRISES
Américo L. Azevedo, César Toscano and João Bastos .................................................... 124

THE EFFECTS OF DIFFERENT FEATURE SETS ON THE WEB PAGE CATEGORIZATION PROBLEM USING THE ITERATIVE CROSS-TRAINING ALGORITHM
Nuanwan Soonthornpitisaj and Boonserm Kijsirikul ......................................................... 132

DATA REDUCTION TO IMPROVE KNOWLEDGE EXTRACTION
Maria de Fátima Rodrigues and Pedro R. Henriques .................................................... 139

DESIGNING INTELLIGENT TUTORING SYSTEMS: A BAYESIAN APPROACH
Hugo Gamboa and Ana Fred ......................................................................................... 146
QUALITATIVE REASONING FOR SOFTWARE DEVELOPMENT PROJECT
BY CONSTRAINT PROGRAMMING
Antonio J. Suárez, Pedro J. Abad, Rafael M. Gasca and J. A. Ortega ........................................... 153

TOWARDS SCALABLE MULTI-AGENT SYSTEMS
Ralph Deters ................................................................................................................................. 161

PART 3 – INFORMATION SYSTEMS ANALYSIS AND SPECIFICATION

DOMAIN ORIENTED FRAMEWORK CONSTRUCTION
Ivan Mathias Filho, Toacy Cavalcante de Oliveira and Carlos J. P. de Lucena ..................................... 171

MANAGING PROCESSES THROUGH A BASE OF REUSABLE COMPONENTS
Bernard Coulette, Xavier Cregut, Dong Thi Bich Thuy and Tran Dan Thu ......................................... 180

ENTERPRISE INFORMATION SYSTEMS: SPECIFYING THE LINKS AMONG
PROJECT DATA MODELS USING CATEGORY THEORY
Michael Johnson and C. N. G. Dampney ........................................................................................... 191

A FRAMEWORK FOR EVALUATING WIS DESIGN METHODOLOGIES
Luca Cernuzzi and González Magalí .................................................................................................. 198

AN INFORMATION SYSTEM VIEW OF CONSISTENCY AND INTEGRITY
IN ENTERPRISE OPERATIONS
Yoshiyuki Shinkawa and Masao J. Matsumoto .................................................................................. 208

INTEGRATING ORGANIZATIONAL SEMIOTIC APPROACH WITH
THE TEMPORAL ASPECTS OF PETRI NETS FOR BUSINESS
PROCESS MODELING
Joseph Barjis and Samuel Chong ..................................................................................................... 215

THE USE OF FORMAL AND INFORMAL MODELS IN OBJECT-ORIENTED
REQUIREMENTS ENGINEERING
Linda Dawson ..................................................................................................................................... 223

ENTERPRISE MODELLING FOR AN EDUCATIONAL INFORMATION
INFRASTRUCTURE
Ing Widya, Cees Volman, Stanislav Pokraev, Italo De Diana and Eddie Michiels ................................. 231

TOWARDS E-MANAGEMENT AS ENABLER FOR ACCELERATED CHANGE
Hugo Lérias, João Luiz, Pedro Moura, Ana Mendes, Isabel Teixeira and J. Paulo Teixeira .................... 240

EXPERIENCES IN THE DEVELOPMENT OF INFORMATION SYSTEMS IN AN
INDUSTRIAL INNOVATION CONTEXT
António Lucas Soares, Waldemar Gaida and Christian Schmidt ......................................................... 248
PART 4 – INTERNET COMPUTING AND ELECTRONIC COMMERCE

AN XML-BASED MULTIMEDIA MIDDLEWARE FOR MOBILE ONLINE AUCTIONS
Matthias Wagner, Wolf-Tilo Balke and Werner Kießling ................................................................. 259

AN ANALYSIS OF B2B CATALOGUE INTEGRATION PROBLEMS
Borys Onielayenko and Dieter Fensel .............................................................................................. 270

A CORBA AND WEB TECHNOLOGY BASED FRAMEWORK FOR THE ANALYSIS
AND OPTIMAL DESIGN OF COMPLEX SYSTEMS IN THE OIL INDUSTRY
Carlos Arévalo, Juan Colmenares, Nestor Queipo, Nelson Arapé and Jorge Villalobos .................. 278

E-BUSINESS CHANGE AND ORGANISATIONAL PERFORMANCE
Colin Ash ........................................................................................................................................ 287

QOS NEGOTIATION BASED ON MANAGEMENT DELEGATES
José Luís Oliveira and Rui L. Aguiar ................................................................................................. 295

MAY YOUR INFORMATION SERVICE LIVE IN INTERESTING TIMES...
Steven Willmott, Bernard Burg and Paul O’Brien ........................................................................ 303

AGENT-BASED PERSONALIZED SERVICES FOR MOBILE USERS OVER A VPN
Hamid Harroul, Mohamed Ahmed and Ahmed Karmouch .......................................................... 312

AN AGENT-BASED KNOWLEDGE SHARING MODEL FOR INFORMATION
RETRIEVAL ON THE INTERNET
Bin Ling, Colin Allison and Kecheng Liu ...................................................................................... 320

AUTHOR INDEX .......................................................................................................................... 329
A CORBA AND WEB TECHNOLOGY BASED FRAMEWORK FOR THE ANALYSIS AND OPTIMAL DESIGN OF COMPLEX SYSTEMS IN THE OIL INDUSTRY

Carlos Arévalo, Juan Colmenares, Nestor Queipo, Nelson Arapé, Jorge Villalobos.
Applied Computing Institute, Faculty of Engineering, University of Zulia, Maracaibo, Zulia, Venezuela
Email: carevalo@luz.ve, juancol@luz.ve, nqueipo@luz.ve, narape@luz.ve, jorge@ica.luz.ve

Key words: CORBA, WEB-based Software Engineering, Modeling and Optimization, Reservoir Characterization

Abstract: This paper discusses the design and implementation of a CORBA and WEB technology-based framework for the analysis and optimal design of complex engineering systems. The framework provides an environment for the coupled execution over a network of data analysis, modeling and optimization heterogeneous software using a WEB browser. A framework application is illustrated solving a reservoir characterization problem, critical for devising an optimal strategy for the development of oil and gas fields. The results suggest that the framework can be effectively and efficiently used for solving reservoir characterization problems and holds promise to be useful not only in other areas of petroleum engineering (e.g. hydraulic fracturing design, enhanced oil recovery processes) but also in the solution of complex system design problems found in industries such as electronics, automotive and aerospace.

1. INTRODUCTION

Many problems of interest for the oil industry require the coupled execution of different applications. For example, reservoir characterization involves the use of a reservoir simulator and an optimization program, hydraulic fracturing design implies the cooperative execution of different applications to address issues such as fracture geometry, pre and post fracture production, profitability and optimization, whereas solving enhanced oil recovery process optimization problems requires the use of applications for simulating the process at hand, evaluating the economic benefits and selecting the optimal set of control parameters.

In a typical scenario, an engineer or researcher has several applications for modeling and optimization, which have been written in different languages and run in different hardware/software platforms. The coupled execution of these programs usually requires either to rewrite/recompile them in different platforms or to move data among computers through files or the network. These are time consuming tasks not related to the user’s main interest, which result in delays and additional costs in achieving the expected outcome.

During the last decade, technologies such as CORBA (OMG 2001, Vinoski 1997, Henning 1999) and DCOM (Microsoft 1998, Eddon 1998) allow software integration across dissimilar hardware/software platforms. However, the direct use of these technologies for software integration is a time consuming task and may be limited to people with expertise in software development, object oriented programming, and distributed computing. In addition, regarding using interfaces, Web browsers are ubiquitous being the primary point of access to Internet. As a result, it is of primary interest the developing of frameworks that assist the users in the integration of heterogeneous applications for the analysis and optimal design of complex systems across a network with Web browsers as user interface.

There have been limited work in the development of frameworks for modeling and optimization of complex systems and CORBA/Web integration that satisfy the above-referenced requirements (Michelena 1999, Papalambros 1997 and Merle 1996). Michelena et al. and Papalambros et. al. developed a CORBA based framework for the design of large complex systems, which they have used in the design of various mechanical systems, including a pressure vessel, an automotive hybrid power train and a tracked vehicle; this framework have a limitations in the user interface and in
supporting services such as session and user management. Merle et al. developed CorbaWeb, a gateway between the World Wide Web and CORBA based on the CorbaScript scripting language (OMG 1996).

This paper presents a framework based on CORBA and Web technology for the integration of software tools typically used for the analysis and design of complex systems and illustrates its application in reservoir characterization, a critical task for the oil industry. The framework has been designed using object-oriented techniques, and includes basic support services (e.g. session, user and project management), a CORBA-WEB gateway, and several modeling and optimization applications. Reservoir characterization addresses the problem of estimating the distributions of permeability and porosity in heterogeneous and multiphase petroleum reservoirs by matching the static and dynamic data available. The remaining sections of the paper discuss the framework overview (Section 2), architecture (Section 3), and its application to the solution of a critical problem in the oil industry (Section 4); conclusions and recommendations are the subject of Section 5.

2. FRAMEWORK OVERVIEW

The objective is to create an environment for the coupled execution of applications (modeling and optimization) deployed over a network using a WEB browser. A single application is executed through the following steps:

- The user enters the framework URL. A welcome page is presented with fields to login to the framework. Once the user has been validated, an HTML page is displayed where he can select an application to work with.
- When the user selects an application, a request is sent to the latter to return its user interface (an HTML page to be displayed in the browser).
- The user fills in the information required by the application and clicks a button to execute it. An execution request is sent to the application including all the necessary information.
- The application performs the requested task and returns an HTML page that displays the results on the browser.

A similar procedure is followed in the case of the execution of several applications. Applications interact dynamically by creating CORBA objects and making them available to other applications through basic support services provided by the framework.

The requests that come from the WEB browser are HTTP requests. On the other hand, the application modules are CORBA objects that use the IIOP protocol. Therefore, the requests must be converted from one protocol to the other. This is achieved by a CORBA-WEB Gateway. Details of the framework components are provided in the following section.

3. FRAMEWORK ARCHITECTURE

With reference to Figure 1, the framework has the following components:

- A CORBA-WEB Gateway module that serves as a bridge between the WEB and CORBA.
- A portal module that is the entry point to the framework.
- A session manager module that provides mechanisms for preserving the state of applications and for accessing objects created by other applications.
- A user manager module that handles user access and permissions.
- A project manager module that handles user access and permissions.
- A naming service that manages a directory of all framework resources and applications.
- Application modules that perform specific tasks such as data analysis, modeling and optimization.

![Figure 1. Framework architecture.](image-url)
3.1. The CORBA-WEB Gateway

A bridge between the WEB and CORBA. It receives HTTP requests from a WEB browser, translates them to CORBA messages, and sends them to the target application module. If necessary, it translates the response into HTML format and ends it back to the client for display.

The Gateway consists of a core object and several Java servlets (Figure 2). The servlets receive requests coming from the WEB client, gather the information needed to execute the request and invoke the core object for execution. The core object performs the following tasks:

- Invoke the naming service to retrieve a CORBA reference for the target module.
- Call the target module to obtain the list of parameters required to make the request.
- Create a dynamic request by means of the CORBA dynamic invocation interface (DII).
- Perform the request, wait for the target application to finish and return the output data to the calling servlet, which in turn returns the data to the client for display.

For each request made by the WEB client, a new thread is created. Since the CORBA request is always made synchronously, the thread will be blocked until the application returns. However the gateway will continue to process requests due to its multithreaded nature.

The gateway module includes the following objects (Figure 3):

- The dispatcher servlets that receive the requests made by the WEB client.
- A core gateway object. This object receives the request from the servlets and coordinates the construction and execution of the dynamic CORBA requests.

3.2. Portal module

The entry point to the framework. It consists of a CORBA object that returns the HTML login page and the framework main page that contains links to available applications and framework services.

3.3. Session manager module

It consists of two CORBA object types: Session and SessionManager. The former includes a public container where applications store object references that become available to other applications. It also contains a private container that stores CORBA data types and object references representing the application state. In addition, the SessionManager creates and manages user sessions. Hence, this module serves two purposes: i) to provide a way for applications to access objects created by other applications and ii) to provide a mechanism for preserving the state of applications.

Sessions are created by the gateway (through a request made to the session manager) and passed to applications in every request made by a WEB client. The Java session tracking mechanism is used to ensure that all requests coming from the same user during some period of time get the same session object.

3.4. User manager module

A CORBA based user administration tool implemented in Java. It allows an administrator to create and delete framework users, change their properties through a WEB interface, and offers an UNIX-like user authentication service.
3.5. Project manager module

A basic framework service that allows users to create, rename and delete experiments and projects through a WEB interface. An experiment is a persistent entity that stores data related with configuration parameters and results of applications. Experiments are grouped in projects, which are stored in a repository handled by the Project Manager and mapped on a file system, with projects and experiments being directories where application files are stored. In addition, through this module, applications may store results associated with experiments and retrieve configuration data for their execution. The Project Manager Module is implemented in Java using multithreaded programming.

3.6. Naming Service

It manages a directory of all the framework resources with a hierarchical structure similar to the corresponding to the application interface (described in the next section). This service was implemented directly with the CORBA naming service.

3.7. Application modules

These modules are the focus of the user interests and perform not only those tasks related to modeling and optimization but also framework related operations such as returning the HTML pages that make up the module’s user interface, initializing user sessions and registering themselves with the framework name service.

An application module may have objects that implement different interfaces. For example, a typical modeling application module consists of a model builder (creational object) that constructs models (core objects) which implements the interface Model and an object that implements the Application interface if the module has a WEB based user interface. Below it is shown the IDL code of the Model and Application interfaces.

```
typedef sequence<double> DoubleVector;
typedef sequence<DoubleVector> DoubleMatrix;

interface Application {
    string initWorkSession(in Session session);
};

interface Function {
    DoubleVector evaluate(in DoubleVector inputData);
    DoubleMatrix multiEvaluate(in DoubleMatrix inputData);
};

interface Model : Function {};
```

Usually, the first message received by an application module is a request to retrieve its user interface. This is an HTML page that will be displayed by the WEB browser. This page has at least a button to request the execution of framework servlets, and includes the target application and operation names. The operation is executed using the mechanism outlined in Section 3.1. The page may also have several execution buttons; an example of this situation is the case of a neural network application user interface that has a button to train the network and a button to execute it; alternatively, a genetic algorithm application may have a button to create and set the parameters of a GA and a button to run it.

Considering that the applications are responsible for creating dynamically their web pages, an object that facilitates this task have been designed. This object, implemented in Java, is named UserInterfaceManager. An application module can create an UserInterfaceManager (using a factory object), configure it, and use it to create dynamic HTML pages.

At present the framework has available the following application modules:
- Genetic algorithm. It has a CORBA object that implements the GeneticAlgorithm interface. The implementation is based on the

![Figure 4. Application Module Interface Hierarchies.](image-url)
The identification of the permeability and porosity parameters that best match the data (static and dynamic) available for a given reservoir is critical for devising an optimal strategy for the development of oil and gas fields. The static data makes reference to those originated from geology, electrical logs, core analysis, fluid properties, seismic and geostatistics; while the dynamic data is represented by field measurements such as, production history, bottom hole pressures from permanent gauges, water-cut, and gas-oil ratio.

Estimating permeability and porosity parameters from available data is difficult because of the following reasons: i) in general, the number of parameters to be estimated are very high, since data is scarce, and the reservoirs are heterogeneous (permeability and porosity have spatial variability), ii) the available data may have very different scope and nature, and, iii) the numerical simulation of the reservoir, necessary to assess how well given permeability and porosity parameters match the available data are computationally expensive.

Queipo et. al. developed a methodology for solving the problem of interest which includes the construction of a “fast surrogate” of an objective function whose evaluation involves the execution of a time-consuming mathematical model (i.e. reservoir numerical simulator) based on neural networks, DACE modeling and adaptive sampling. The proposed solution approach, called NEGO (Neural-network based Efficient Global Optimization), is an improved version of the EGO algorithm (Jones 1998) for the optimization of computationally expensive black-box functions. It involves the following five steps:

1. Construct a sample of the parameter space using the latin hypercube method. The latin hypercube sampling procedure has been shown to be very effective for selecting input variables for the analysis of the output of a computer code.
2. Conduct mathematical simulations using the sample from the previous step and record the response values associated with static and dynamic data available, and the objective function values.
3. Construct a parsimonious neural network using the data obtained in the previous step. The purpose of this neural network is to capture the general trends observed in the data; no rigorous performance criteria are placed on the neural network. The input variables of the neural network are the permeability and porosity parameters and the output variable is the corresponding objective function value.
4. Construct a DACE model for the residuals, that is, the difference between the observed objective function values, and the neural network responses using the sample data. These models provide not only an estimate of the residual values but also of the respective errors. The surrogate model for the evaluation of the objective function is the sum of the neural network and DACE models.
5. Additional points are obtained balancing the exploitation of the information provided by the surrogate model (where the surface is minimized) with the need to improve the surface (where error estimates are high), until a stopping criterion has been met. This balance is achieved by sampling where a figure of merit is maximized.

As can be noticed, this problem involves the coupled execution of several optimization and modeling tools such as: sampling programs, a reservoir numerical simulator, surrogate models, and optimization algorithms. To implement this methodology using the framework it is necessary to:
1. Design and implement a framework application for optimization based on NEGO.
2. Design and implement a framework application for reservoir numerical simulation.
3. Integrate the two applications using the framework resources.

4.1. Design of application for optimization based on NEGO.

The framework includes an interface hierarchy for solving optimization problems. These interfaces, their implementations and relationships are shown in Figure 5. The general procedure to solve an optimization problem within the framework is:

1. Create a concrete optimizer object by invoking the create operation on the optimizer factory.
2. Obtain the reference to an objective function from the user session and pass it to the optimizer.
3. Invoke the optimize operation on the optimizer.
4. Obtain the optimal value and solution from the optimizer.

The NEGO algorithm falls under the Bayesian optimizer category (see Figure 6). Below it is shown a fragment of the framework CORBA interface definitions needed for Bayesian optimization applications. Note that there are interface definitions for Optimizer, GlobalOptimizer and BayesianOptimizer.

typedef sequence<ConstraintFunction> ConstraintFunctionVector;
interface Optimizer {
   // Exceptions
   struct FaultExceptionDetails {
      string info;
   };
   exception FaultException {
      FaultExceptionDetails details;
   };
   struct BadParamDetails {
      unsigned short numOfBadParam;
      string msg;
   };
   exception BadParamException {
      BadParamDetails details;
   };
   //operations
   void optimize() raises(FaultException);
   void setObjectiveFunction(in Function objFunction);
   void setMaxNumIterations(in ULONG m) raises(BadParamException);
   void setReportFrequency(in USHORT frecuency) raises(BadParamException);
   void defineSearchSpace(in USHORT dim) raises(BadParamException);
   void defineBoundedSearchSpace(
      in USHORT dim,
in DoubleVector upperBounds,  
in DoubleVector lowerBounds)  
raises(BadParamException);

void defineConstrainedSearchSpace(
  in USHORT dim,  
in DoubleVector upperBounds,  
in DoubleVector lowerBounds,  
in ConstraintFunctionVector constraints)  
raises(BadParamException);

double getOptimalValue();

DoubleVector getOptimalValues();

DoubleVector getOptimalSolution();

void abort();

};

// Global Optimizers
interface GlobalOptimizer : Optimizer {
}

interface BayesianOptimizer : GlobalOptimizer {
}

interface NEGO : BayesianOptimizer {
  void
    setNeuralNetworkTrainer(
      NeuralNetworkTrainer neuralNetTrainer);
};

The NEGO application consists of the following CORBA objects:

- NEGOApplication. This object presents the application's user interface. It implements CORBA interface Application (section 3.7).
- NEGOFactory. This object creates NEGO objects.
- NEGO. This object does the optimization. It implements the NEGO strategy outlined in the previous section.

The Figure 6 illustrates UML class diagram of NEGO application where the gray boxes represent objects that are already present in the framework and white boxes represent objects that must be implemented.

4.2. Application for reservoir numerical simulation

This application is developed using an existing reservoir numerical simulation program. In order to use this program within the framework a wrapper must be created. The wrapper is a framework aware object that receives the CORBA requests and executes the simulation program using the facilities provided by the operating system. Data is exchanged between the wrapper and the simulator through files.

The framework reservoir numerical simulation application consists of the following objects:

- ReservoirSimulationApplication. This object presents the application's user interface. It implements CORBA interface Application (section 3.5).
- ReservoirSimulator. This object is a wrapper of the reservoir numerical simulation program. It implements CORBA interface Function and performs the following tasks:
  - Receives the CORBA requests and writes a file formatted for the numerical simulation application.
  - It executes the simulator and waits for it to exit.
  - Finally, it reads the output data produced by the simulator from a file and produces the CORBA response.

The reservoir numerical simulation program reads input data from a file, performs the simulation, writes an output file with the simulation results and exits.
4.3. Coupled execution of created applications using the framework.

The integration of the two applications requires the creation by the user of a reservoir simulator object through the corresponding application. The reference to this object is stored in the user session (section 3.3), then the user executes the NEGO application, selects the reservoir simulator object (wrapper) as the objective function and performs the optimization.

5. CONCLUSIONS

This paper discussed the design and implementation of a CORBA and WEB technology based framework for the analysis and optimal design of complex engineering systems. The framework provides an environment for the coupled execution over a network of data analysis, modeling and optimization heterogeneous software using a WEB browser, and includes the following modules:

- A **CORBA-WEB Gateway module**, Multithreaded and based on Java servlets and CORBA Dynamic Invocation Interface (DII), it serves as a bridge between the WEB and CORBA.
- A **portal module**, It is the entry point to the framework and consists of a CORBA object that returns the HTML login page and the framework main page that contains links to available applications and framework services.
- A **session manager module**, It provides mechanisms for preserving the state of applications and for accessing objects created by other applications. Its functionality is based on the Java session tracking mechanism and CORBA session objects that hold containers where applications store object references that become available to other applications and CORBA data types and object references representing the application state.
- A **user manager module** that handles user access and permissions, and offers an UNIX-like user authentication service.
- A **project manager module** that allows users, through a WEB interface, to create, rename and delete experiments and projects stored in a repository mapped on a file system.

- A **naming service** that manages a directory of all framework resources and applications. It was implemented directly with the CORBA naming service.
- **Application modules**, They are the focus of the user interests and perform not only specific tasks such as data analysis, modeling and optimization but also framework related operations. They implement interfaces developed to standardize the call protocol that application clients must use to invoke operations on other objects. In addition, the applications are responsible for creating dynamically their web pages.

A framework application was illustrated solving a reservoir characterization problem, critical for devising an optimal strategy for the development of oil and gas fields. The results suggest that the framework can be effectively and efficiently used for solving reservoir characterization problems and holds promise to be useful not only in other areas of petroleum engineering (e.g. hydraulic fracturing design, enhanced oil recovery processes) but also in the solution of complex system design problems found in industries such as electronics, automotive and aerospace. Currently, additional applications are being developed associated with the solution of hydraulic fracturing, steam assisted gravity drainage, and water alternating gas optimization.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support provided by the Consejo Nacional de Investigaciones Científicas (CONICIT – Venezuela) through the Projects G-9700824 and G-97003899.

REFERENCES


Vinoski, S., 1997. 'Integrating Diverse Applications within Distributed Heterogeneous Environments', *IEEE Communications Magazine*.


Queipo, N., et. al., 2000, ‘Surrogate Modeling-Based Optimization for the Integration of Static and Dynamic Data Into a Reservoir Description’, *2000 SPE Annual Technical Conference and Exhibition*. 

286