Web Based Virtual Frameworks for Emergency Decision Support Systems

Claudio Balducelli, Giordano Vicoli
ENEA, V. Anguillarese 301, 00060 Rome, Italy
balducelli_c@casaccia.enea.it, *giordano.vicoli@casaccia.enea.it*

Keywords: Virtual toolkit, Middleware, Component based, Design for Environments

Abstract

The objective of this work is the specification and development of a set of tools and a "tools container" architecture to support the management of emergency/crisis situations: inside this set there are already available tools as well as new developing ones. The tools are named "Virtual Tools" because they will be available, through a Web Application Server, to their end-users (operators and emergency managers) and to their designers (environmental and computer engineers) independently from the respective geographical locations and from the hardware/software used resources.

The paper will give a description of the architecture of the tools Application Server. The efficacy of different applicable middleware technologies (DCOM, CORBA, CGI and ASP) as well as of several adopted object oriented languages (Java, Delfi, Visual Basic ect.) is actually under evaluation in an applicative end-user scenario relative to environmental/economical computing models aimed to reduce the dangerous chemical pollution produced by manufacturing industrial processes.

1. Introduction

The efficacy of Distributed Decision Support Systems (DDSS) depends by the co-operation of many autonomous hardware and software components and their users. They must have flexibility and scalability which cannot be reached by hierarchical and procedural software architectures.
Each software component has a distinct role and functionality. Therefore different development approaches (languages/techniques, numerical algorithms, knowledge-based problem solvers, territorial GIS data bases etc.) are used for their implementation.
The distinction of roles and competencies as well as the mutual co-ordination capacity between software components and their users promotes emerging of a global intelligent behaviour that is not produced from the separate activities of single units [5].
Actually, the availability of the Internet network and the possibility to implement distributed client/server component architectures (through Middleware layers), allows more efficient implementations of such type of DDSS [2]. They seem to be extremely

useful as co-operative and cross-organisational emergency management support. The utilisation of component-based architectures allows also the integration of software tools, languages and data bases systems that were not compatible in the past [4].

2. Emergency/crisis applicative domain description

The goal of this work is relative to the prevention and the reduction of the negative impacts coming from critical energy and manufacturing plants/infrastructure. As shown in fig. 1 two types of impacts are considered:

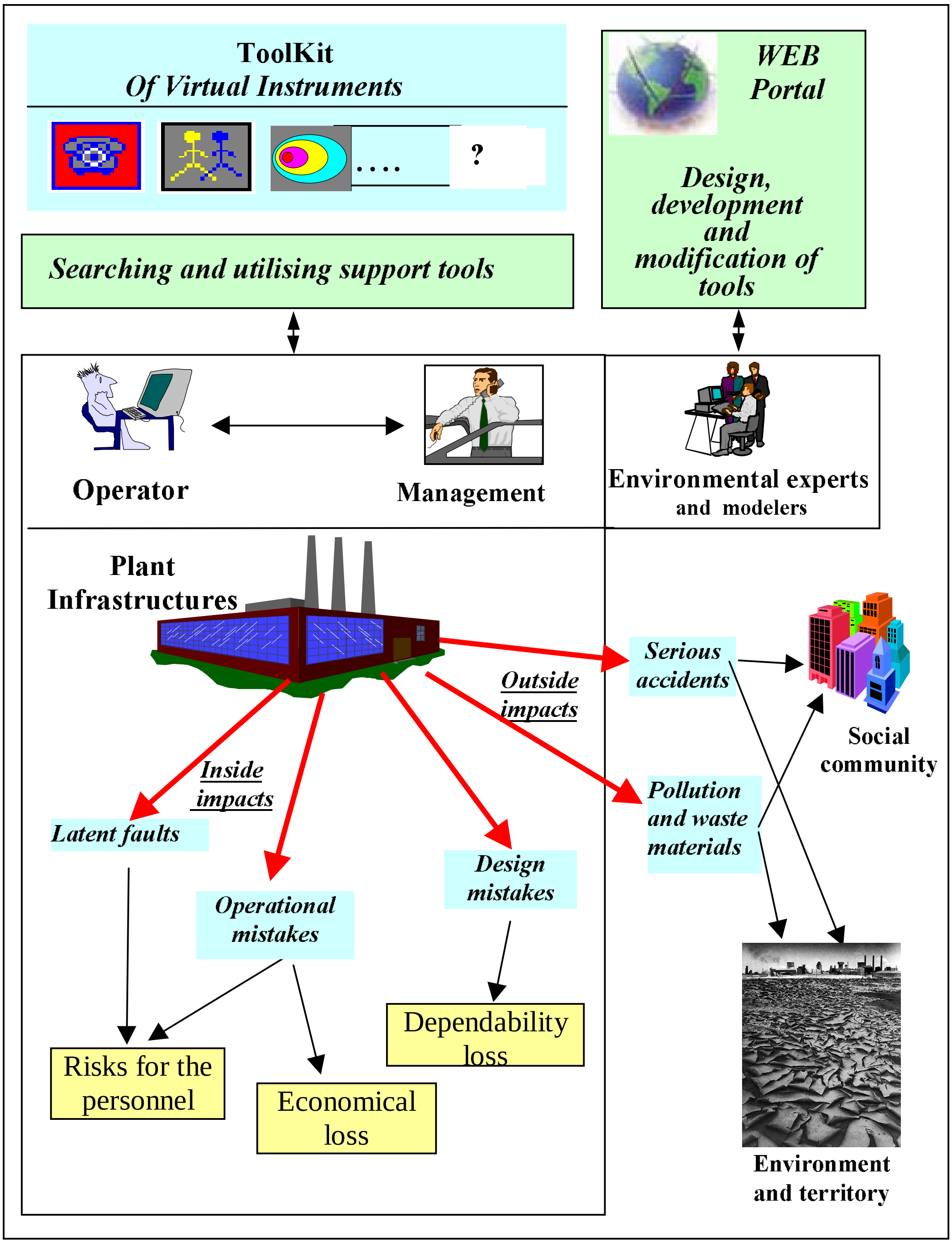

Fig 1 – Description of emergency domain, impacts, and support functions

- The first type is directed **inside** the plant/infrastructure itself and has negative effects on the plant resources and personnel, causes loss of money and reduces the dependability of the service offered by the plant itself.
- The second one is directed **outside** the plant/infrastructure, toward the social community and/or toward the external environment. It refers to serious instantaneous faults or dangerous conditions but may also be caused by pollutions and emissions that for a long time are emitted toward the external environment.

In this situations the plant operators, managers and administrators must co-operate together in the management of the industrial/technological infrastructure, avoiding as much as possible the previous conditions.
The availability of tools and instruments able to help them to evaluate, prevent and manage these dangerous conditions is a key point needed for the reduction of internal/external dangerous impacts. WEB networks is a decisive resource to be utilised for such high social goal. WEB could be utilised as a container for the set of support tools, during crisis or emergency prevention and management.
WEB could be used today not only to display static pages containing texts or pictures; it includes novel type of interfaces on which is possible utilising several computer systems as a single computer interface.
If standard middleware components are used, virtual emergency management support tools could be developed on the application server and make anywhere available. The plant or industrial infrastructure owners could use the tools independently by the proprietary computer configurations or resources.
Also the designers and developers of the tools, i.e. the environmental experts and the information technology experts could operate in a remote way, to implement or upgrade the toolkit with new characteristics and functionalities.

3. Emergency management application Server

The efficacy of emergency mitigation support tools depends about their usability in a short time inside distributed environments. A comparative analysis has been done of the different available middleware technologies able to support in different ways the tools distribution throw private/public computer networks.

3.1 Background

Several tools based on artificial intelligence technology was developed and tested in the past years, to support plant operators and managers during the execution of emergency plans and allocation/optimisation of resources and risk objects evacuation.
The best solutions during great emergencies, like the management of long-time fires involving reservoirs of fuel storing/dispatching plants or burning materials inside tunnels, could be analysed and identified by the utilisation of different type of software tools as for example: event simulation tools, information retrieval tools, optimisation tools and advanced decision support tools like case base reasoning tools.
In fact during emergencies operators must evaluate current status and dynamics of the physical events, must search useful information like resources and means locations and availability, design optimised strategies for resources utilisation and for people

and risk objects evacuation for the entire emergency duration and, finally, they must take right decisions at the right time co-ordinating their actions together.
In [3] the possibility to integrate these different types of tools inside a multi-agents hybrid architecture was illustrated and tested.

3.2 Architecture

Many types of available technologies could be applied to build distributed applications. Anyway it is possible to classify a multi-node shared applications in three different type of architectures, depending on how digital processing is executed inside the different nodes:

1) Major part of processing runs on clients side: *applets applications*
2) Part of the processing run on clients and part on servers: *component based applications*
3) Major part of processing runs on servers side: *CGI or ASP applications*

The Application Server has been implemented utilizing a Three-Tier architecture. This is the typical architecture for Web Based Applications. Three-Tier architecture is the evolution of the Client/Server architecture. Client/Server architecture consists of a Client program that interacts directly with the database level (on the Server). Three-Tier architecture adds a new level, the *middleware level*, between the Client program and the database level. The middleware level makes the Client independent by the access problems to the information on databases.
In the Three-Tier architecture the following three levels could be identified:

- Presentation Level
- Intermediate Level (Middleware)
- Database Level

The Presentation Level represents the Application User Interface by which he gets and sends all the necessary information. The Middleware level forwards all the User information/requests to the Database Level and vice versa. The Database Level contains all information needed by the Application Server (Fig.2).
Web Based Applications have a lot of advantages. On the Client side they allow:

- **Independence of Client Platform**: there is no need to develop specific Client interfaces because the Internet Browser can make efficient connections to every application and the Internet Browser is an available program on all hardware/software platforms.
- **Single program to access to the applications**: this is a consequence of the previous point; the program to access applications is always the same and it has the same interface. So users have less difficulties to learn the interface functionalities and behaviour.

On the Server side all software upgrading and maintenance concerns exclusively the Server without Client modification. Software running on the Client (Applet) is upgraded on the Server and downloaded automatically to the Client.

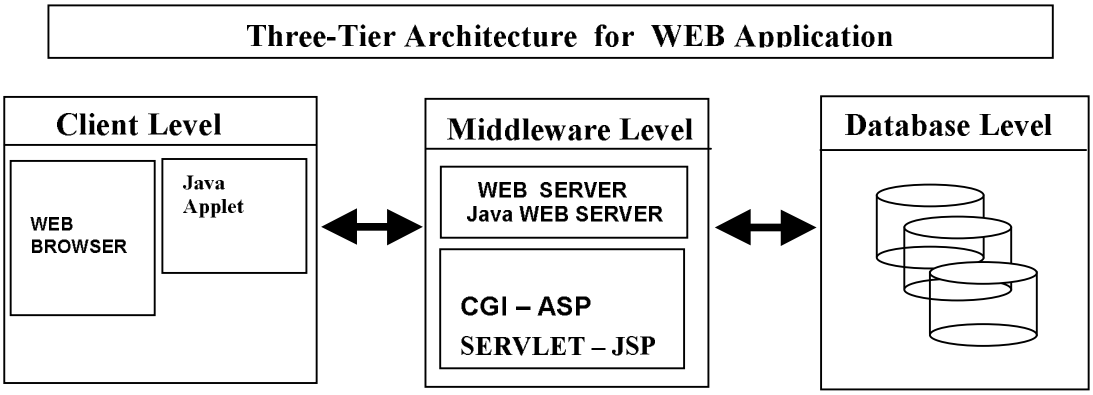

Fig. 2 – Application Server Architecture

The Client Level, on fig. 2, consists of the Internet Browser and of some Java Applets. They are downloaded automatically to the Client from the Web Server and run inside the browser.

Middleware Level, shown in more detail on fig. 3, is composed by a couple of Web Servers: the first one, is an Internet Information Server by Microsoft. It obviously supports Active Server Pages (ASP) and also CGI programs. The second one is a Java Web Server supporting Java Servlet and Java Server Pages (JSP).

The third level, the databases level, consists of some relational databases (Access) and Excel WorkBooks.

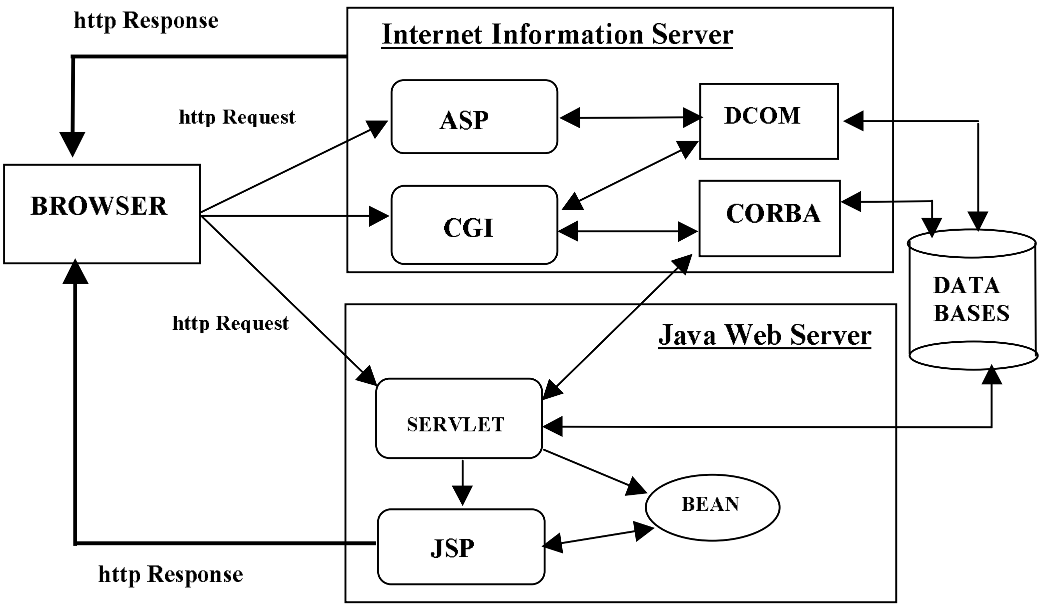

Fig.3 Middleware Level in detail.

As shown in the figure, the Client level is only composed by an Internet Browser. The user see in the Browser the Application Server Home Page. He makes some choices according his requirements and each choice is an *Http request* toward the Application Server. The Http request can activate another HTML page, an ASP page, a CGI program or a Java Servlet. The ASP page and the CGI program may communicate with some DCOM server components. CGI program could also activate CORBA

server components. Both ASP pages and CGI programs are activated by the Internet Information Server. They get the Http request and, after the elaboration, produce a response to the Browser. DCOM/CORBA components interact with Databases and also with Excel WorkBook.
At the same time also Java applications, running on Java Web Server, can Read/Write on Excel WorkBook through the CORBA Server. In fact, as it is shown, the Browser can also do a request toward the Java Servlet. This one can access directly to a database or to a Java Bean that is Java component. The Servlet can also access to a CORBA Server to easily read e/o write to an Excel WorkBook. After all elaborations the Servlet redirects the request to a JSP (Java Server Page) that produce the Response toward the Client.

4. Toolkits implementations and test

The above described application server architecture is taken as reference for the implementation of various emergency management set of tools. The first one, described in [3], is relative to the mitigation of serious industrial accidents whose consequences have impacts outside the plant toward the social community. The second one, described in [1] is relative to the prevention of accidents/faults inside the plant that may generate risks for plant personnel/operators and economical losses.
The third one, described in the next paragraphs is relative to the realisation of safeguards able to increase the reduction of waste materials and pollutions coming from the manufactory and process industries.

4.1 Safeguard tools to prevent dangerous Environmental effect

To help the manufacturing and process industries to include environmental safeguards into the design and redesign of production processes (Design For the Environment), a set of tools are under development that can be used to assess and/or substitute technologies and other pollution reduction options. DFE toolkit can provide companies with a systematic way to improve their operations for better environmental performance. DfE toolkit includes tools to support the process designers in the respective process optimisation strategies by the introduction of the Best Available Techniques (BAT) inside the production lines. The objective is to realise appreciable reductions of the environmental pollution without too great increasing of the production costs.

4.2 The model of the production line

The model of a single production line is composed, as in the fig 4, is by a process logically decomposed into more production phases. Every phase is consequent to a previous one and utilizes, as input, materials and energy. Every phase produces dangerous pollutions and generate production costs. To one or more phases a set of BAT can be applicable to improve the process with reduction of pollutions and minimization of additional costs.

From the previous figure it is possible to evidence that for every phase more waste

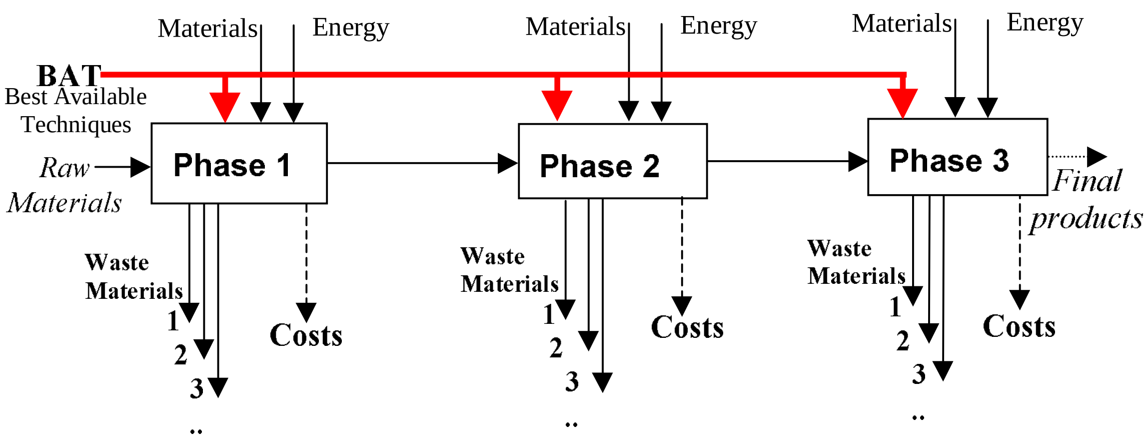

Fig. 4 – Dfe process model

materials could be produced; generally the application of a certain BAT reduces the production of a certain type of waste material but could increase the generation of another one. The optimization algorithm increases its complexity while the number of different waste materials increase.
For this reason the DfE toolkit will be tested using two different benchmark test cases: The first one is relative to the manufacturing process of wood painting where only two types of waste material are produced, the second one is relative to the glassware production process involving several types of waste materials.

4.3 DfE Toolkit

The DfE toolkit has a Client/Server architecture, component oriented. The Client includes the following described services, offered by the following three types of tools:

Configuaration tool:
The configuration tool allows the user to configure the model of his own production line. He must define the number and the type of phases, the involved materials, the costs, the necessary energetic consumptions etc. The configuration tool, using information coming from a configuration data base and following the indications of the user, produces an Excel sheet model for every production phase and an additional sheet for the set of applicable BAT.

***Analysis tool*:**
After the configuration, the analysis tool allows the user to modify one or more configured parameters and consequently to analyze the obtained results in order to execute a sort of sensitivity analysis. This functionality is important to make the users confident about the results provided by the numerical models implemented inside the Excel sheets and produced by the configuration tool.

Optimisation tool:

The optimization tool allows a sequential optimal application of all defined BAT inside the configured sheets; as a result, an optimal graph, as the one visualized in fig. 5 is produced.

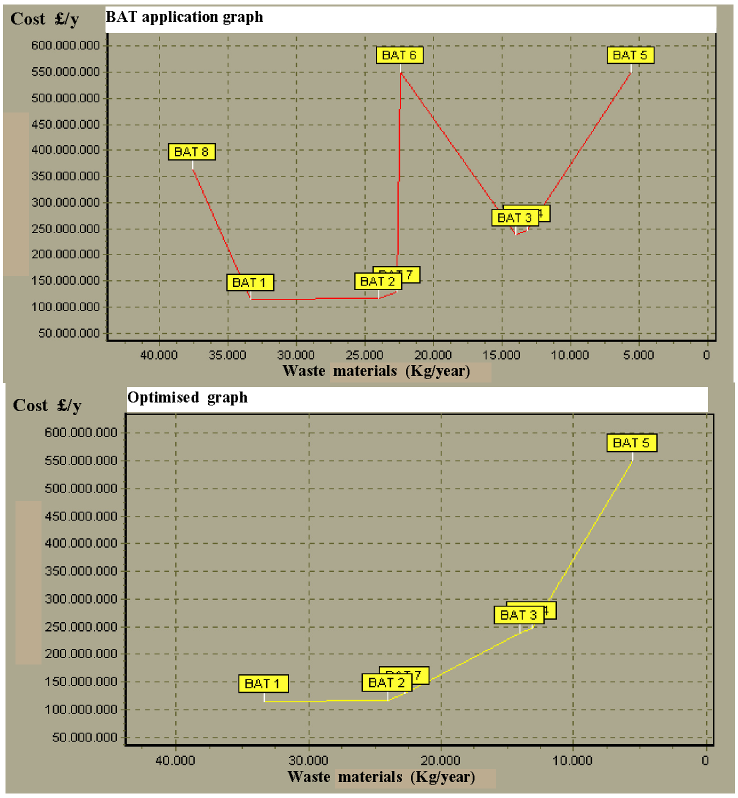

Fig.5 - Optimised sequence of BATs

Every BAT on the curve represents a "set of technologies" applicable inside the manufactory process as candidate to realize environmental improvements. BAT 1 is the set of technologies actually in use. The set of BAT visualized in the bottom graph indicate the set of improvements able to minimize the dangerous pollutions with the minimal possible additional cost. The user has the possibility to insert rules/constraints to avoid the indiscriminate application of every combination of BAT. In such case is possible to reduce the computing time necessary to produce the solution.

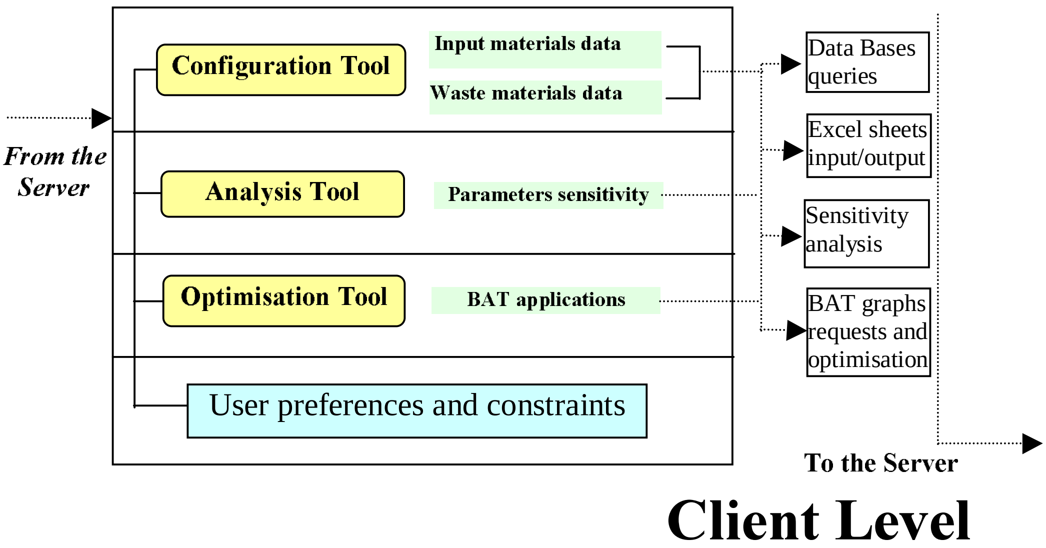

Fig.6 - Client side implementations

4.4 Dfe architecture

On Client level only a Web oriented interface of these three tools is implemented as visualized in fig. 6. Their real implementations reside on Server level, as Middleware components, inside the three different soft agents visualized in fig. 7.
More Clients have the capability to access, at the same time, the Server components to realize a concurrent design process in which more users have the possibility to optimize the proprietary production line with different environmental technologies and cost strategies.

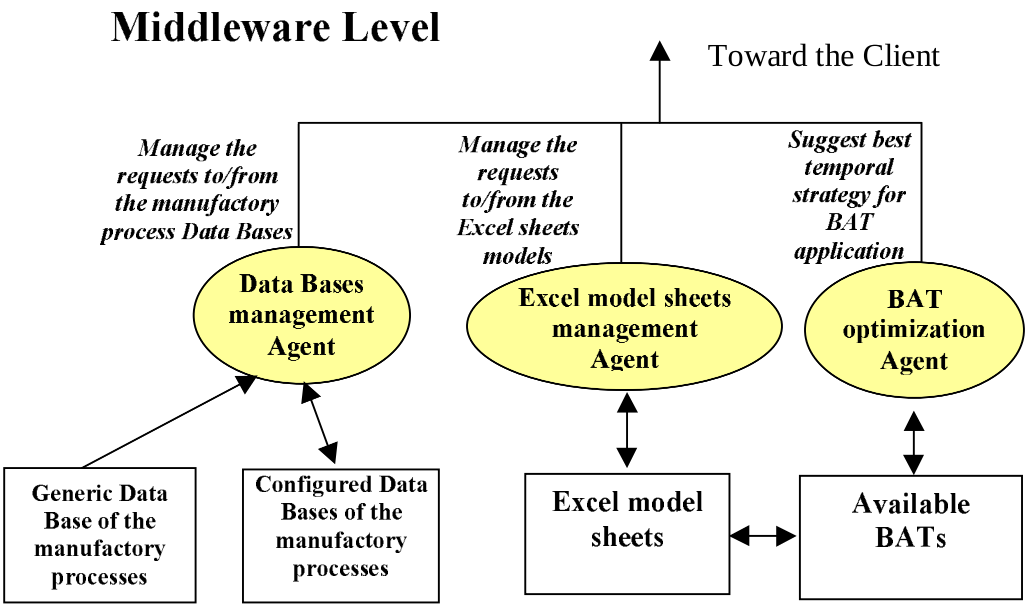

Fig.7 - Middleware level of Dfe system

5. Conclusions

Dfe toolkit is already tested for a quite simple process domain as the industrial painting process of wood articles. The global system architecture as well as the user interactions functionalities are in agreement respect to the design specifications and users expectations.
Further experiments are actually in progress, aimed to implement and optimize the models of glassware production processes. Such industrial process is more complex and produces more types of waste materials respect to the previous one: generally, improvements realized through the application of a certain BAT, may reduce the production of a certain waste material, but may increase the pollution due to another one.
A different type of optimization process, driven by evolutionary algorithms, will be adopted in this case, and a more free type of user interface will be implemented, able to explore more efficiently all the available possibility.

References

[1] Balducelli C., Brusoni F. "A CBR tool to simulate diagnostic Case Based operator's models", Proceedings European Simulation Symposium and Exhibition Oct. 1996, Genoa, Italy

[2] Balducelli C. "DSS systems for emergency management and training based on distributed components object models", Proceedings Of Enlarged HPG meeting on MMI systems research, May 1999, Loen Norway.

[3] Balducelli C., D'Esposito C., 2000. "Genetic agents in an EDSS system to optimise resources management and risk object evacuation", Safety Science 35 n. 1-3, Special Issue on Emergency Management, June-august 2000.

[4] Klein U. "Simulation-based distributed systems: serving multiple purposes through composition of components", Safety Science 35 n. 1-3, Special Issue on Emergency Management, June-august 2000.

[5] Hayzelden A.L.G., Bigham J., "Agent technology in communications systems: an overview", The Knowledge Engineering Review, Vol. 14:4, 1999, 341-375, Cambridge University Press.

Authors biographies

Dr. Claudio Balducelli is a senior scientist working at ENEA as project manager since 1983 in the field of AI technologies applied to operator decision support systems for emergency industrial accidents. His interests include operator models, knowledge formalisation, planning, computerised procedures, plant diagnosis, case based reasoning, learning and fuzzy algorithms. In the last years he was also involved in the coordination of research projects in the field of distributed AI applications.

Dr. Giordano Vicoli from 1988 to 1992 took part in research projects in the field of design and development of expert systems for diagnosis and control of industrial plants. From 1993 he has been working in the field of design and development of decision support systems and training applied in emergency management of high risk industrial plants. He worked in national and European projects. Actually he is interested in the development of distributed application with Java and CORBA technologies.