

CAE application software moves to the WEB

R.Allaria

FIAT G.S.A.

Via Caraglio 56, 10141 Torino, renzo.allaria@fiatgsa.it

W.Aldinucci

X-Technologies

Via Corsica 2-15, 16128 Genova, w.aldinucci@pn.itnet.it

G.Alemanni

Studio Alemanni

Via della Liberta' 19-2, 16129 Genova, g.alemanni@pn.itnet.it

Abstract

The main target of CAEWEB project is moving the CAE application software of FIAT AUTO into an enterprise-wide distributed environment. In the paper we describe the overall software architecture and the porting strategy. We point out also the main goals of the porting: maintenance cost reduction, diffusion of the CAE tools and overall exercise cost reduction.

Keywords

CAE, Java, CGI, Fortran, xy graphs

1 INTRODUCTION

The CAEWEB project has to push the diffusion of CAE methodologies in the world wide context of Fiat Auto and, at the same time, reduce the managing and maintenance costs of the existing hardware and software systems. The initial analysis pointed out the necessity of keeping the

engineering know-how of the CAE tools, while using new up-to-date tools, like Java.

2 CAE TOOLS

The mission of the Engineering Departments of Fiat Auto is the possibility of verifying the behaviour of the vehicle through the use of CAE tools; in the years we analysed and developed procedures in order to allow this type of analysis. The CAE tools are a set of 100 computing procedures that can perform analysis in different engineering fields: internal and external aerodynamics, vehicle performances and fuel consumption, elasto-kinematics analysis of suspensions, chassis fem analysis, stamping, crash analysis, engine behaviour analysis ...

All these types of CAE tools are the basis of Fiat Auto design departments, consequently they are constantly used by the designers and are constantly supported by the IT department the CAE tools reaching a high degree usefulness and security. CAE tools today represent the Fiat Auto engineering know-how: they stand for the IT translation of more than 20 years of experience.

It is obvious that the IT solution had to follow the constantly increasing needs of design as well as the innovative technology: in the years we had to modify the procedures causing a high difficulty in maintenance. We had to face some technology problems, such as:

- Architecture based on departmental VAX's, using VMS o.s.
- Use of Fortran to develop algorithms and interfaces
- Use of proprietary tools such as FMS or SMG to handle user interface
- Use of proprietary tools to handle data base (RdB)
- Dependence on old libraries, like peripherals interfaces i.e. VDU or plotters
- At the same time the new challenges as globalisation push the use of CAE tools in the new plants: i.e. Brazil, Turkey, India ... It is clear that a solution based on VAX could no more be adopted

3 THE ACTUAL TECHNOLOGY

In these years we made different tests in order to understand the reference architecture:

- PC: we used PC when we had to use a simple and friendly tool to access data, with no interaction with geometric data

- Workstation: we used the CAD workstation when the interaction with the geometry was heavy

The coming of new technologies allowed the possibility of analysing new solution such as Java in order to take advantage of new market opportunities.

4 THE ARCHITECTURE

The architecture of the system has been implemented as a client/server model: the GUI component is on the client, while the computing component is on the server and the data management component is shared between client and server, via Java classes and CGI scripts.

Each user may access the CAEWEB system via a Web browser (as Netscape), connect to the server, select the program he needs through HTML pages, activate Java interface to select a test case, execute the program and get the results as text files or graphics.

4.1 CAEWEB Server organisation

The CAEWEB server is made of the following system components:

- Web Server Software (manages http protocol)
- A collection of Web pages (transferred to the clients via http)
- Java classes (inserted into the Web pages and transferred to the clients via http)
- CGI scripts
- NQE (a special implementation of FTP protocol)
- Fortran solver programs
- Data configuration files of CAEWEB system
- Input data for solver programs (test cases, configuration tables, etc)
- Output data (transferred to the client via http or via NQE)

4.2 Web server configuration

The web server of the CAEWEB project is Netscape Enterprise Server. All the parameters connected with the functionality of the server may be set via configuration files. The implementation of the CAEWEB system is very flexible and efficient: for instance it is possible to distribute the application programs on different servers for a better availability of the configuration files and data areas.

4.3 CGI Scripts configuration

The CGI scripts are activated by Java procedures and manage the access of the remote clients to the file system of the server.

The CGI scripts map http references to absolute addresses into the server file system. The general configuration of the mapping is again stored into a configuration file: so it is very easy to change the reference to the absolute location of data areas of the system, modifying only one file.

The CGI scripts invoke the execution of Fortran programs and set the correct path to the configuration files for data input to application programs.

4.4 Users definition

The system needs a set of defined users to access the server utilities. The users will be organised in groups, as in Unix world, in order to manage the access rights to system objects. CGI scripts will use those users for each operation on the server: this will activate a correct account procedure. The same users, structured in groups, will be defined in Web world too. This double definition is useful for access security at every level.

4.5 Security management

The security management in the Web Server is performed by internal protection methods, implemented by Netscape Enterprise Server. Those mechanisms work both in terms of access protection of subset of public file system and in terms of cryptography of HTML pages. By default Netscape Enterprise Server makes no restrictions to the users of the Web server.

Some areas of the file system must have a restricted access only to authorised users: for instance Web pages with links to application programs.

Other pages are public domain and have no restrictions: for example documentation.

The policy of management implemented for the CAEWEB system defines no access limits for public-domain areas and gives access rights for particular areas only to authorised users or groups of users. The access restriction is managed via a password request when the browser is connected to a protected URL. The Web server acquires the password only once per session and if it is correct allows the user to navigate in the reserved area.

The Web server Administrator manages of the configuration of the access rights to the system. The protection of the information transferred via http is performed by a Security System based on SSL system, offered by Netscape Enterprise Server. A Site Certificate located on the server is able to authorise and encrypt the communication from/to clients.

4.6 CAEWEB client

The CAEWEB client part is very easy to implement: the only component necessary to access the system is a Web browser, configured to execute correctly the CGI scripts and the Java applets. Currently the Web browser used is Netscape Communicator 4.0.

4.7 Data access

A typical session follows this schema:

- application program selection
- access validation
- project selection
- object selection
- input file selection
- new data set-up
- calculation execution
- output set-up

The application program selection is made by navigating through the HTML pages of the Main Programs Archive (Programmoteca).

A set of connected HTML pages has been prepared to supply an easy access to application programs. For each program is currently available:

- a link to documentation, which is public-domain.
- a link to the execution point, which is protected.

The definition of the user is the first key selection of the operating sequence. The verification of the user is made by the Web server access protection system. It is activated as soon as the user tries to access the link to the execution point, then the application program starts (Figure 1).

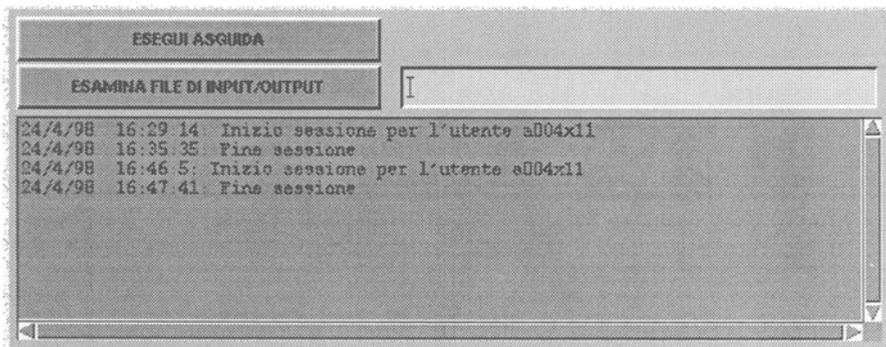


Figure 1: Application program execution.

The next step is the selection of the project, from a list of possible projects, made available by the system manager (Figure 2).

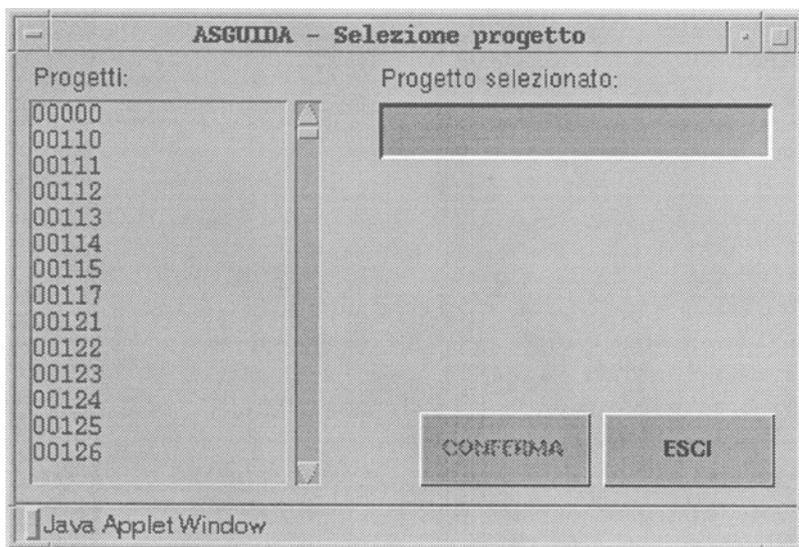


Figure 2: Project selection.

The selection of the object to analyse is based on the selection from a list of available objects, coming from the official company database. Each user may select a proprietary subset of the objects, to work with (Figure 3).

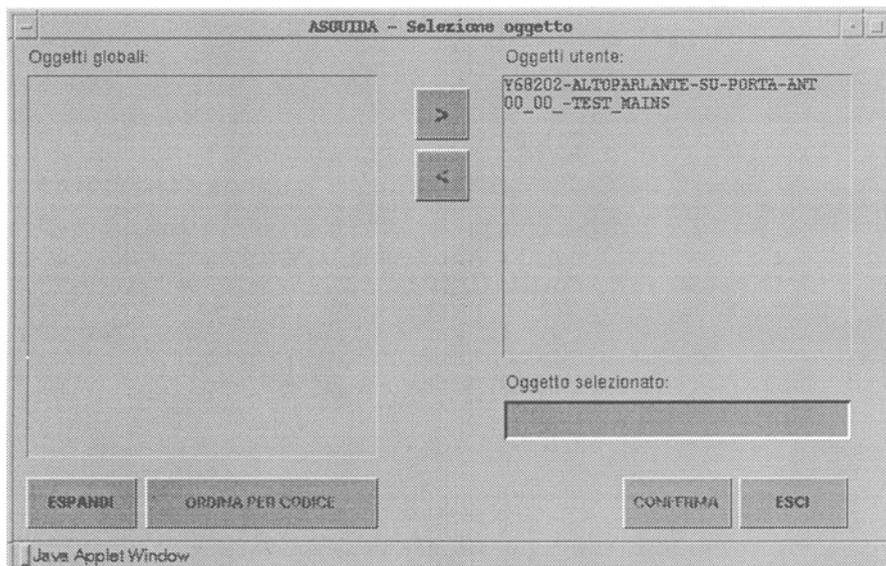


Figure 3: Object selection.

Based on the selection of the previous parameters (program, user, project, and object) the system is able to set up automatically the user directory on the server. In this directory are stored the data file related to the application program required. Other data files may be retrieved into a special directory, which contains the basic test cases for each program.

The selections described below are made via GUI, written in Java language and presented on the screen as a set of panels.

5 SYSTEM FUNCTIONALITIES

5.1 Main area

The following main functionality have been prepared:

- application execution
- output visualisation
- access to documentation files
- errors alarms

The interface is structured into two buttons and a text area. The first button starts the execution of the program. The second button allows the user to examine the complete set of output files of the previous job sessions. The

text area is the area where the GUI visualises the system messages (console area).

The application execution opens a panel that allows the selection of the project from a list of available projects. Two buttons allow the user to apply the selection or to cancel the request.

The GUI shows a new form containing two lists: a global object list and a user object list. The user can move objects from a list to the other using two selection buttons (> and <). The selection of the object to work with is made by an OK button. A Cancel button provides the possibility to reject the selection.

The system shows now a form to:

- select an input data file, from a list of possible test cases
- open a new input data file

The form allows the user to open a preview window of the input data file, via a preview button. The OK button and the Cancel button maintain the usual functionality.

The aspect and the sequence of the forms described below are the same for all the programs of CAEWEB system.

The user interface of the following forms is on the contrary partially application dependent, because each program has its own specific set of input data.

The window has a left common vertical tool bar, which contains:

- an exit button
- a reload button
- a save button
- a save as button
- an execute button
- two move previous/next page buttons (< >)

And a right panel, which contains the input data typical of the application.

- The exit button stops the execution and returns the control of the application to the main applet on the HTML page
- The reload button reloads the content of the input data file
- The save button enables the user to save the data to the file on disk
- The save as button enables the user to save the data to a new file on disk
- The execute button starts the execution of the application program. This button is activated only if the file has been saved or saved as.
- The two move buttons allow the user to move forward or backward if there are more panels containing the input data.

The data inserted in the form fields are validated against a set of boundary limits (minimum, maximum value), format (text, number, etc) and correlations each other. All those controls are performed by the interface and implemented in the Java procedures.

Error messages are provided to the user to allow the correct filling of the fields.

During the execution of the program the system messages are shown on the console area. At the end the output files related to the current session are listed in the bottom part of the current window of the browser.

The user may visualise on the screen the content of the list files by clicking on the name of the file, in the output file list.

The visualisation functionality for text files is provided directly by the browser.

The graphic representation on the screen is provided via a graphic set of routines (jGraView) into a new window.

5.2 Development environment.

The GUI has been written in Java language, via the development tool Cosmocode from Silicon Graphics.

The CGI scripts have been developed via Bourne shell (sh) under Unix system.

5.3 Solver

Old Fortran programs were all-in-1: solver + user interface + graphic output.

Fortran has been retained for its best usage as pure solver: changing to other languages implies full rewriting of existing algorithms, which in turn implies very high costs and long times.

New programs instead act as black boxes: they are Fortran standard, and carry out only the solver task, no user interface; they can be invoked from the web via CGI scripts, but they still can be invoked as stand-alone applications from the specific operating system (Unix shell on ws, or dos windows in PC).

Numerical test of each application (which implies multiple run of different test cases) can be performed quickly and easily, without invoking the WEB interface, which is best suitable for the end user single run.

5.4 Graphics

Graphics have been designed with the following goals:

- minimize the programmer's effort to obtain a standard graph
 - number & complexity of API reduced to a minimum;
 - intermediate file in ASCII format and easily readable, in order to help in the debugging phase,
- minimize the data amount to be moved through the network
 - intermediate file in compact form, with every redundant digit eliminated, so that size is reduced to a minimum
- minimize the user effort to view the graphs
 - intermediate file is complete, no need for other information to be supplied by the user in order to obtain the graph: just click on the graph name, done.
- allow graphs to be viewed in different environments
 - graphs should be visible anywhere from the web,
 - on windows PCs, graphs can be inserted into other documents
 - on windows PCs, graphs can be printed on all windows compatible devices.

5.5 FORTRAN

In order to renew the Fortran modules, we developed a set of new procedures, such as:

- new utility “xmake”, to manage serial compilation and test of multiple applications on different platforms
- storage of official test cases (average of 5~8 per application) to allow easy check of results between different platforms
- new generic libraries host small set of machine dependent subroutines base string manipulation, parsing, other general tasks input/output library (file opening, program start/stop...) and creation of CSV files (ASCII worksheet)

Obviously we had to clean up the software, adding:

- standard error handling, through library routines
- old functionalities embedded in scripts, hardware dependent Unix Vs VMS, moved into Fortran, now hardware independent
- syntax changes so that all statements are accepted by most Fortran compilers (on PC, on SGI, on SUN machines) no tabs, no H characters in formats, alignment of variables in common

- argument arrays with size 1 converted to *, to allow runtime range checking
- all variables declared explicitly

A set of I/O subroutines has been implemented, and then inserted into each application, to obtain a similar behaviour of all applications. For example, standard units include:

- input data file can refer to other files
- listing results in human readable format
- log all major events
- exit a 1-line file that stores the completion status (ok/bad)

This standardisation process is particularly useful to the WEB modules that can treat every application in the same standard way.

5.6 Graphics

A very simple set of graphics API has been implemented. The object names and the API names are as follows:

- “xFolder” opens the folder
- “xPage” begins a page
- “xGraph” defines a graph inside a page
- “xValues” creates a set of values with a name
- “xCurve” links 2 sets of values to create a curve

With these API the Fortran programmer can ignore the graphics details and obtain the standard types of graphs (xy and bar charts) with the minimum effort, just calling the high level API.

Graphs produced by Fortran applications are stored in intermediate files, .GRA; these files are like paper folders that hold one or more pages. The page is identified by name, not numbers, to allow easy browsing by the user. Format is compact and ASCII, so those files can be read and checked by the editor, just in case.

From the WEB, graphics files are visible via jGraView applet. When the applet is started, .GRA file is retrieved from the server, its content is parsed, and a data structure is created, that contains pages, values (arrays of real numbers), axes specifications (min max step...) curves properties (colour red, style dashed), etc; the first page is displayed, the user can "browse" the folder with navigation buttons (1st page, prev, next, last). And with a choice control (comboBox) which lists the page names and allows the user to skip to a specified page.

PC solution: the same .GRA file can be viewed/edited on the PC with a Windows application, xGraView that extends the functionality of the JAVA version and takes full advantage of the features offered by Windows.

- Edit: graphs can be edited, e.g. properties can be changed like axis range, curve style and colours, and the like
- Save: the modified folder can be saved to disk
- Copy: a page can be copied to the clipboard: it is stored in windows metafile format, which is far better than a bitmap (much more compact, can be resized, full quality,..). The clipboard can then be pasted in other applications, such as Word
- Export: all numbers, grouped in values (columns), can be exported in CSV format. This allows further post processing on the results, with no need to modify the application software
- Print: each page (or the whole folder together) can be printed. Print is done with the Windows print manager; the printed page is in windows metafile format, so the print time is fast and quality is high.

6 THE ACTUAL SITUATION

The project reached the goals:

- All the CAE tools are used on WEB
- We reduced the overall costs of more than 50%: 50% on system maintenance and more then 60% on software maintenance and drastic decrease in client expenses
- The use of a browser, NETSCAPE or INTERNET EXPLORER, allowed the PC's and workstations to use directly CAE tools
- All our foreign sites could now access the WEB

7 REFERENCES

Bud Tribble (1996), Java Computing per l'azienda (White Paper), *Sun Microsystems Italia*

8 BIOGRAPHY

R. Allaria received his degree in Physics at the University of Genoa, Italy, in 1973. His experience is in the analysis, design and implementation of Computer Aided Engineering and Computer Graphics Systems in industrial

environment. He is among the authors of the APPLE SAP system, a CAE program for structural analysis. At present he is in charge of CAD, CAM, CAE and PDM systems development in the Fiat G.S.A. Automotive Division.

W. Aldinucci, electronics engineer degree at University of Genoa, Italy, 1976. His experience is in the analysis, design and implementation of Computer Aided Engineering and Computer Graphics Systems and in project management for software development. He worked in industrial environment as well as in hardware and software international companies. Actually is chairman of X-Technologies.

G. Alemanni, physics degree at University of Genoa, Italy, 1978. His experience covers the fields of tennis, until 1980, and technical software, from 1981. After two work experiences for engineering companies, he started his personal studio in 1988. Now a senior developer, he works at a keyboard, rather than at meetings or shows. His attention is focused on the quality of the software and on the abstraction level of the software from specific platforms. His plans for the future: keep on developing software hopefully until Windows 2030 will hit the shelves.