



Consiglio Nazionale delle Ricerche

Nota Interna

An Integrated Work Environment for I.N.D.U.C.E.

*Daniele Azzarelli, Edoardo Bozzi, Renzo Bozzi, Massimo Chimenti
Massimo Martinelli, Salvatore Minutoli, Ovidio Salvetti*

B4-15
lug-2000

I.E.I.
ISTITUTO DI
ELABORAZIONE DELLA
INFORMAZIONE

IST. EL. INF.
BIBLIOTECA
Posiz. ARCHIVIO



Project N° BE 97 - 4057
Contract N° BRPR - CT98 - 805



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 1/48
			Appendixes -

An Integrated Work Environment for I.N.D.U.C.E.

*Daniele Azzarelli, Edoardo Bozzi, Renzo Bozzi, Massimo Chimenti,
Massimo Martinelli, Salvatore Minutoli, Ovidio Salvetti*

IEI-CNR
Pisa Research Area, S. Cataldo
Via Alfieri 1, 56010 Ghezzano SGT (Pisa)



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 6/48
			Appendixes: -

An Integrated Work Environment for I.N.D.U.C.E.

"One of the primary challenges of the aeronautical industry is the tireless adaptation of increasing quality levels to costs and cycles reduction imperatives. This challenge can be met by the adoption of unified life-cycle engineering concepts that merge aspects of concurrent engineering and total quality management around the design. This is just the goal of INDUCE"

Table of contents

1	Introduction.....	7
2	Global Project.....	7
2.1	Project requirements.....	8
2.1.1	Easy of use.....	9
2.1.2	Maintenance and updating.....	10
2.1.3	Modularity and flexibility.....	10
2.1.4	Efficiency.....	10
2.1.5	Security.....	11
2.2	Security project.....	11
2.3	Remote acquisition project.....	14
2.4	Database project.....	15
3	The implementation.....	16
3.1	Security implementation.....	16
3.2	Web-Server Site Implementation.....	19
3.2.1	Public environment implementation.....	19
3.2.2	Private environment implementation.....	22
3.3	Remote Acquisition Implementation.....	32
3.4	Database implementation.....	39
3.4.1	Pre-Analysis.....	39
3.4.1.1	Use of a database server.....	39
3.4.1.2	Use of ACCESS.....	40
3.4.2	System structure.....	40
3.4.2.1	Point to point connection.....	40
3.4.2.2	Connection by server process.....	41
3.4.3	Implemented prototypes.....	41
3.4.3.1	Connection between a client in Linux and server SQL 7.....	41
3.4.3.2	Connection by server process.....	43
3.5	Procedure to update the system.....	47
3.6	Backup.....	47
4	Conclusions.....	47



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 7/48
			Appendixes: -

1 Introduction

The aim of this task has been to design and realise an 'Integrated Work Environment' (named I.W.E.) constituting a Web-Server Site for the I.N.D.U.C.E. project. I.W.E. is useful in the frame of Non Destructive Testing (N.D.T.), is accessible remotely via network and it is also able to implement different levels of interactions (Figure 1). It has been designed to offer an actual and efficient instrument for giving world-wide the knowledge of the I.N.D.U.C.E. project and, at the same time, to make easy sharing software and hardware resources or other peculiar information among the members of the consortium. The information and the resources of I.W.E. have been adequately protected with appropriate security instruments.

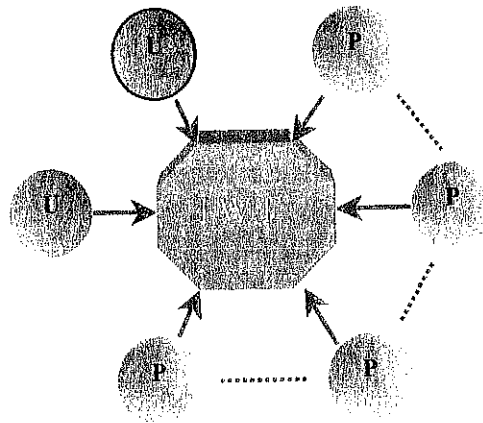


Figure 1: Visiting Users (U) and Partners (P) accessing I.W.E.

2 Global Project

I.W.E. general functionality (Figure 2) mainly satisfies requirements for:

- *Informing*: in terms of specific and industrial aims of the project, areas of interest, adopted techniques, general and particular references or contacts, links to industrial and academic international laboratories and centres operating in the NDT field.
- *Monitoring*: in terms of diffusion of approaches and results by implementing a reserved *channel* of communication inside the consortium. To this end, the Site has been organised into pre-defined sections where each partner can find technical documents, results and software produced by any other partners. This facility could be used also for distributing prototypical software asking B-test for assessment. Monitoring allows also any partner to access special *environments* where to find relevant information, suitably organised and extracted from international databases by performing a continuous search on the network.
- *Remote Processing*: in terms of activation via network, through the Server-Site, of application programs or procedures. In this way, a member can perform remotely a number of operations, such as acquisition of images, data visualization or analysis. Besides, I.W.E. makes available different tools that can be easily increased by adding specific software. At present the Server



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 8/48
			Appendixes: -

has installed IEI proprietary programs for image pre-processing, coding and analysis; extension are under study to make available also commercial software, such as MATLAB™ or AVS™.

- *Database consulting*: in terms of remote access to a relational database, local to the Server, where images, other data and information are stored. At present, a kernel of the INSIDE NDT database, developed by Alenia, has been linked to the Server via a Web-based interface.

I.W.E. can be accessed by two kinds of users: a generic user, mainly interested to have a general overview and information about I.N.D.U.C.E. and the activities performed inside it, and a member himself of the consortium, mainly interested to exchange with other members detailed technical data. For these reasons, I.W.E. implements two separate ways of communication:

- anyone can access a *guest environment* that contains public interest information, general goals of the project, a broad description of the main techniques and functions developed, a reference to the state of the art, some explanation or presentation demos, and so on;
- only the members of the consortium can access a *restricted private environment* that contains specific technical and scientific results obtained in the different fields of activity, detailed descriptions of the test performed, selected databases of images, algorithms and programs designed to solve complex problems and executable by remote control on the Server.

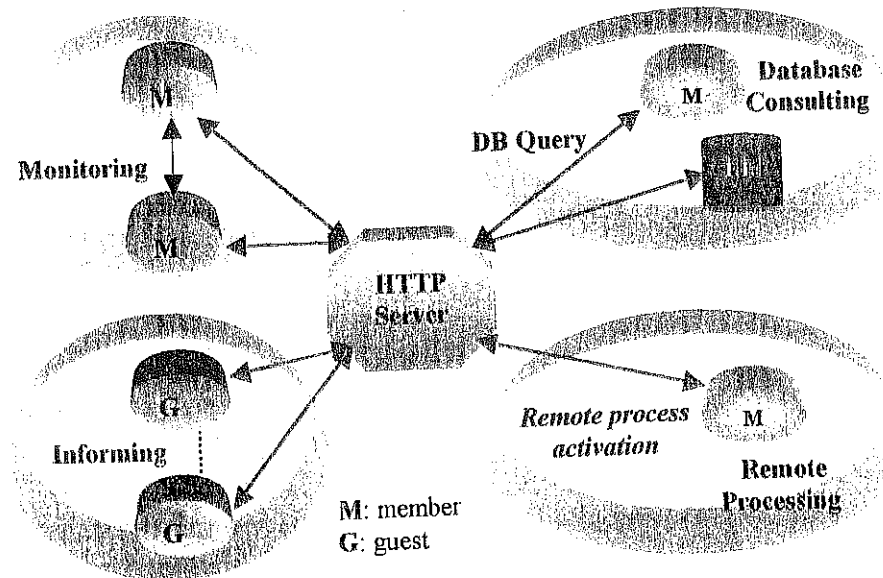


Figure 2: Different functionality of I.W.E

2.1 Project requirements

In order to implement the different environments mentioned in the previous section, I.W.E. separates the functionality of *server* from that of *application*. In the first case, security services and the Web-based techniques have been implemented (Figure 3), while for the application functionality we have satisfied needs like information query, data



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 9/48
			Appendixes: -

consulting, graphics, simulation and in general the control of remote process (e.g. thermographic acquisition, US inspection, and other).
In the following, the main requirements and constraints considered for designing a complete 'Integrated Work Environment' able to fully implement the Web-Server Site for the I.N.D.U.C.E. project are described.

2.1.1 Easy of use

The user is able to easily understand how to communicate and interact with the Server, and the types and the operational characteristics of the services provided. To this end, all the resources available in I.W.E. have been organised *selectively*, according to different user *categories*: this capability has been achieved by means of ad hoc user-friendly interfaces in order to make easy the use of the *system* also by non expert users.
Main interfacing tools have been concentrated on:

- A Web-server interface. The navigation through the Site is very intuitive; images, graphics, animation and colours help the user to orient himself and to understand how the site has been organized, what sections he has just visited and which are still the unexplored ones, which are his rights to access information. To this end, a unique meaning has been assigned to an icon or symbol or colour: for instance, a red button always identify *a gate* to access a restricted area, while a green one indicates just the way to freely access some information. Each *page* has an essential graphic suitable to synthesize and make understandable its information content and long text has been usually avoided as well as the display of the information often stimulates continuing the exploration and the search on the Site. In some cases, colours and *scene depth* have been introduced to increase the readability of the presented information.
- A database access interface. A browser-based interface has been specifically designed to access and query a database via network. In this case, pop-up menus, keywords, thesauri and windowing tools are provided to the user in order to make his query as simple as possible, without any need to know or understand the database structure itself. (Note that, at present, I.W.E. includes only the INSIDE NDT database, developed by Alenia using MS-ACCESS®; another specific database, developed by IEI using SQL 7 Server, is under linkage).
- A remote acquisition program interface. A *batch-oriented* interface has been designed to satisfy this functionality, so that all the information or parameters needed to run a program are given without feedbacks or several input steps. In any case, an approach similar to the one above described has been followed.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 10/48
			Appendixes: -

2.1.2 Maintenance and updating

A main feature of I.W.E. is that it implements a *dynamic system*, able to develop itself for the whole period of the project and then to increase its power in terms of both information stored and functions supported.

Consequently, the design of this environment has been oriented to realize an expandable and exportable system, as much as possible based on standard tools and platforms. For these reasons, the structure of I.W.E. has been implemented to make easier its updating and maintenance, mainly thinking about documents and results production. In this context, we have introduced the figure of a *Web Master*, whose role seems to be crucial since he can control and allow members of the consortium to download or store documents in I.W.E. according to pre-defined access protocols.

2.1.3 Modularity and flexibility

I.W.E. structure has been organized into many levels in such a way to satisfy different user needs. In particular, the information has been organized in a hierarchical structure to allow simply reaching different investigation levels of knowledge details (Figure 3). Users have been then *categorized* and consequently also the access methods and the information structure and relationship have been realized as modular and flexible as possible.

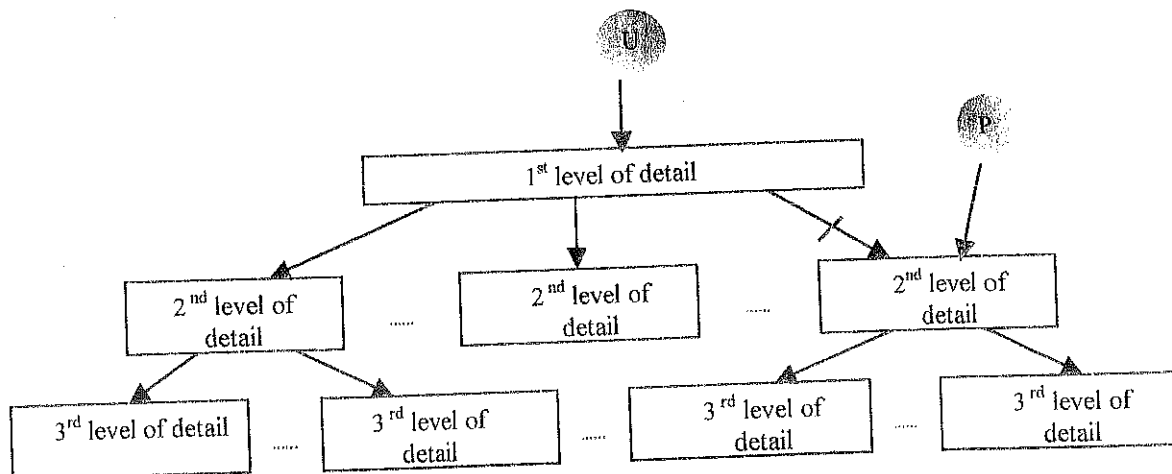


Figure 3: Hierarchical structure of the information.

2.1.4 Efficiency

The navigation through I.W.E. is highly independent from the user hardware resources and from the connection to the network. For this reason, graphic and image use has been implemented to be at the same time winning and essential.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 11/48
			Appendixes: -

2.1.5 Security

In order to guarantee a correct spread or exchange of the information stored in I.W.E., a rigorous access protocol has been defined based on the implementation of an advanced security structure.

2.2 Security project

The architecture of I.W.E. has been studied and developed considering to use adequate hardware and software systems. The security has been realized in order to provide at the same time a high level of protection and a guarantee that the global functionality of the system used would have run. In particular, the problem and the technical requirements and constraints have driven this project pointing out the need to distinguish case by case a *logical and physical division* among the possible functions. For instance, in order to separate two kinds of users, i.e. 'guest' and 'member', we could think to implement two physically divided environments (Figure 4). By the way, this choice could require to dispose of double resources and information, as well as to impose the partners to put the same information on two different environments and consequently the *system administrator* (i.e. the Web Master) to check their consistency.

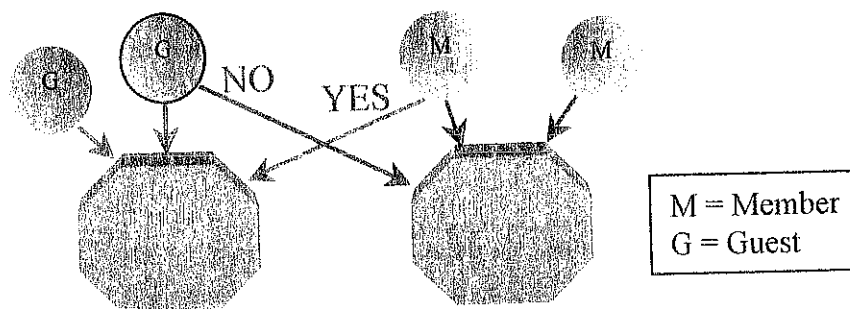


Figure 4: Two physically distinct environments

On the other hand, I.W.E. could be a unique environment keeping logically separated a guest from a member, and sometimes one or more members from the others, maintaining at the same time the capacity of modifying its physical configuration by adding new hardware or software tools.

In our case, embracing the second model, the security systems used impose some fixed protections together with other protection mechanisms changing when I.W.E. changes. The today's technology gives us instruments - as *routers* and *firewalls* - which can be used to realize different changeable *security filters*. And I.W.E. just takes relevant advantage from the protection of a router and a firewall (Figure 5).



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 12/48
			Appendixes: -

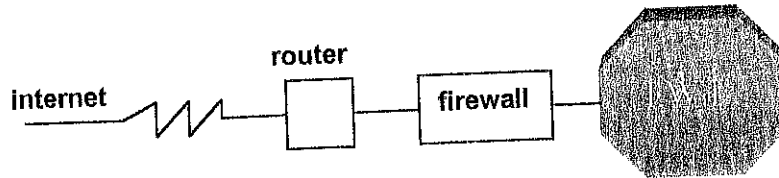


Figure 5: The network architecture

Besides, I.W.E. provides a security policy that is realized through two distinct computers: a *server platform*, *SP*, where all the user's requests come, and an *application platform*, *AP*, where instead the request are satisfied.

The user interacts with the system through a browser talking with a Web-server installed on *SP*. In this way, the filters installed on both the router and the firewall do not allow connecting directly to *AP*, while only *SP* can access *AP* through appropriate interfaces, eventually made available to authorized users (Figure 6).

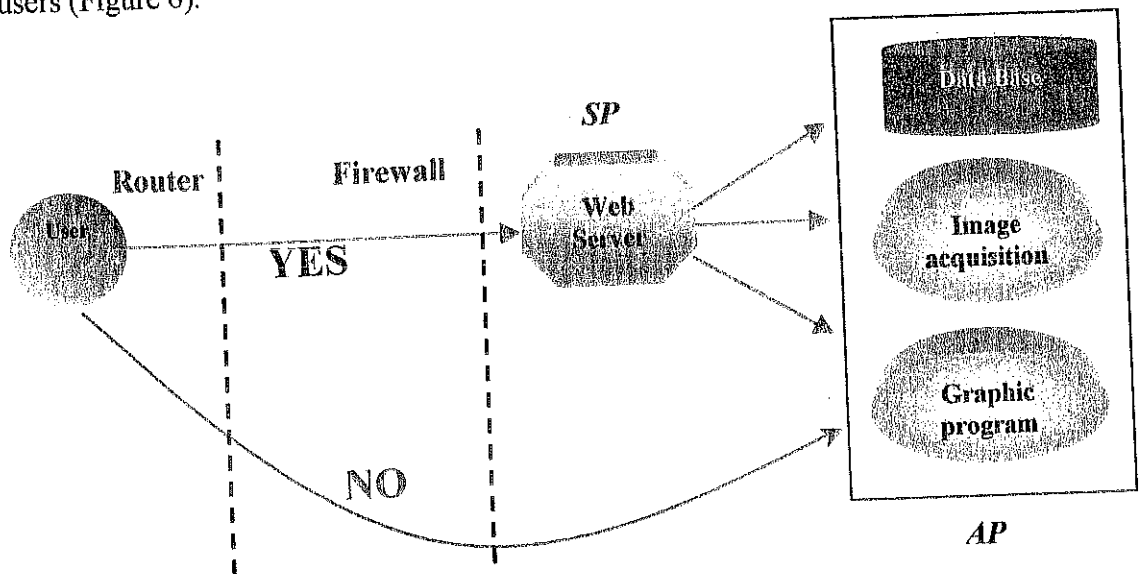


Figure 6: Functionality and security of the private environment

SP performs the functions of both interfacing the user to the system via web and checking security as well as administrating the access policy.

When a member needs to exchange private information with another member by means of I.W.E. he must be sure that no unauthorized person ("man in the middle") can see the data:

- it is necessary an encrypted transmission.

He must be sure that no other computer impersonate the server he thinks to be connected to:

- he needs the certificate of the server.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 13/48
		Appendixes: -	

On the other side, *SP* must be sure that the user requiring information is really who says he is:

- it is necessary another certificate.

Again, *SP* must allow the right, i.e. authorized, user to access just his private *area* of I.W.E. and not another area reserved to other users(Figure 7):

- this is obtained using an *IP access policy* and a *user password* (sent in encrypted session, from a certified client that starts the connection from a determinate IP).

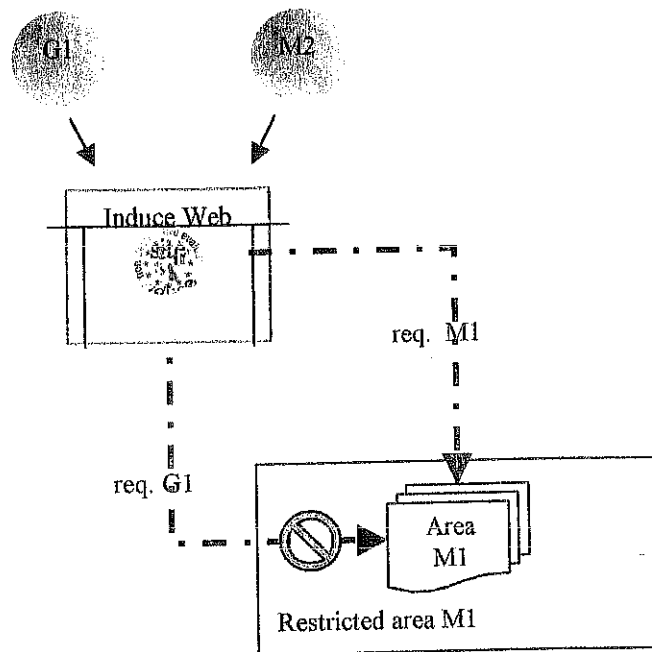


Figure 7: Only the member user M1 can access his private area

I.W.E. is updated continuously in two ways:

1. the Web Master updates the information relative to the monitoring and to the common areas;
2. the I.N.D.U.C.E. partners update all the rest of the information connecting to their private areas and informing the Web Master about the data eventually moved into the common areas. (This is done using file transmission protocol and a shell, both of them encrypted).



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 14/48
			Appendixes: -

2.3 Remote acquisition project

One of the functionality that I.W.E. implements is the capability to run remote application programs. At present, I.W.E. gives only the possibility to acquire remote images, or sequences of images, by means of a thermocamera. The aim is to run an acquisition program, with an appropriate interface inserted in the browser, that returns as answer to the user images, sequences of images or particular elaborations (integral, derivative, etc). According to the architecture of I.W.E., the acquisition program runs on the application platform AP (Figure 8).

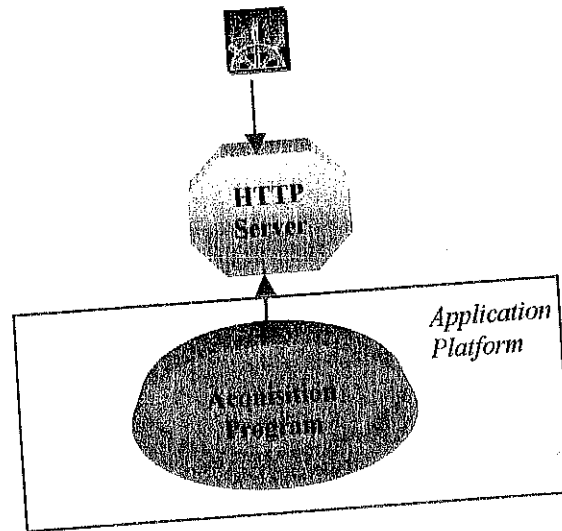


Figure 8: Remote acquisition functionality

The acquisition program is composed of two modules (Figure 9):

acquisition server: this module opens a network communication port which the clients can connect to from a remote system. Once established the connection between the client and the server, this last one waits for an acquisition request from the client side. At this purpose, a communication protocol usable from the client and the server must be established to exchange the information. When the server receives an acquisition request it executes the requested procedure using the functionality that the 'acquisition processor' module makes available. At the end, the client sends a termination message and disconnects itself. At this point, the server is ready again for the next connection.

acquisition processor: this module supports the functionality to acquire image sequences using an acquisition board. In particular, it allows the acquisition of image sequences at a specific time rate; the number of images that constitute the sequence must be an input parameter and an initial delay associated to the external event can also be specified. For efficiency reasons, it is also possible to insert into this module some algorithms in order to send the client only the results of an elaboration instead of, for example, a whole images sequence.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 15/48
			Appendixes: -

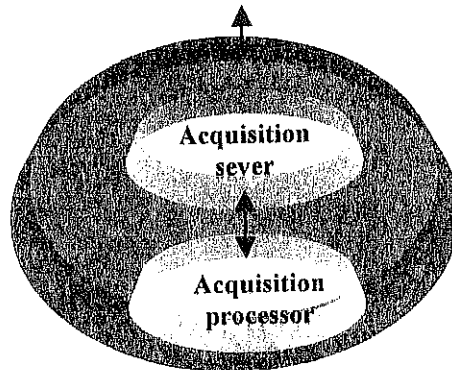


Figure 9: Modules decomposition

In order to allow a remote user acquiring images, a client program has been realised that runs on the http server. This client program allows requesting all the functionality to the server and it has been integrated in a web browser: it shows a graphic interface containing peculiar controls that allow the user to use the functionality of the server.

2.4 Database project

I.W.E. permits also to access remotely information stored into a relational database by means of an user-friendly interface. The information stored in a database may be quite different, ranging from raw data to results of complex simulation or elaborations. For instance, a database will contain:

- Various information about the structure of aeronautic components and materials
- Thermal, US or other images obtained from acquisition equipments
- Processed images
- Information related to inspection or analysis processes.

In any case, an I.N.D.U.C.E. member is able to perform queries, while the update and the consistency control of a database is in charge to the Web Master.

At present, only the INSIDE NDT database, developed by Alenia under MS-ACCESS®, has been ported on I.W.E. and is accessible by any member via an Web-interface reproducing the original one.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 16/48
			Appendixes: -

3 The implementation

A first prototype of I.W.E. has been implemented at the 'Istituto di Elaborazione della Informazione' (I.E.I.) of the 'Consiglio Nazionale delle Ricerche' (C.N.R.) in Pisa. In the following, the technical aspects of the realisation of the Web-Server Site for the I.N.D.U.C.E. project are described in more detail.

3.1 Security implementation

I.W.E. has been located in a pre-existing network structure, as shown in the Figure 10:

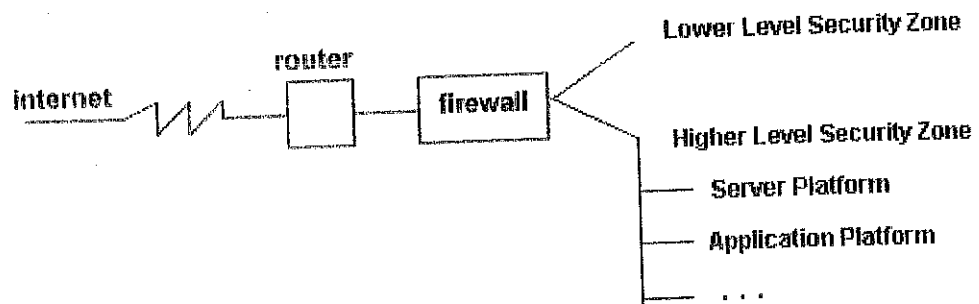


Figure 10: The network configuration at I.E.I.

On the router, some filters have been applied from outside to inside:

- ftp and telnet are not allowed from outside
- http is allowed only to the web servers through the port number 80
- https is allowed only to the web server of our project through a particular port
- ssh is allowed
- no other protocol is allowed to our computers.

No other firewall has been introduced in order to separate our computers from the others of the Institute and not to risk a double bottle-neck like the one shown in the following (Figure 11):



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 17/48
			Appendixes: -

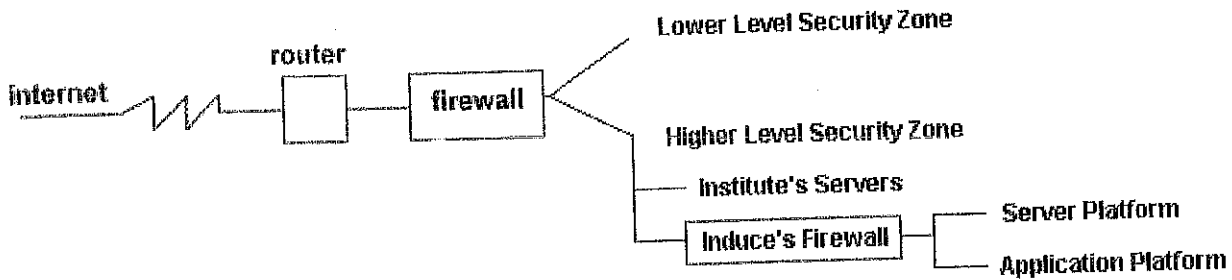


Figure 11: A possible double bottle-neck

Among the many possible hardware/software solutions for realizing I.W.E. - a single computer, many computers, a mainframe or a personal computer - particular emphasis has been turned to aspects like costs of development, updating and maintenance, diffusion degree and up-to-dateness of the chosen technology, correspondence of the adopted solutions towards international standards.

In more detail, the computer acting as a server is a personal computer with Linux O.S: (Red Hat 6.1), on which a server web (Apache 1.3.12) with SSL (openssl-0.9.5, mod_ssl-2.6.2) has been installed

The application platform is a PC with Windows NT (4.0) on which a database server and image acquisition and graphic programs have been installed.

The platform configuration is shown in the following (Figure 12):

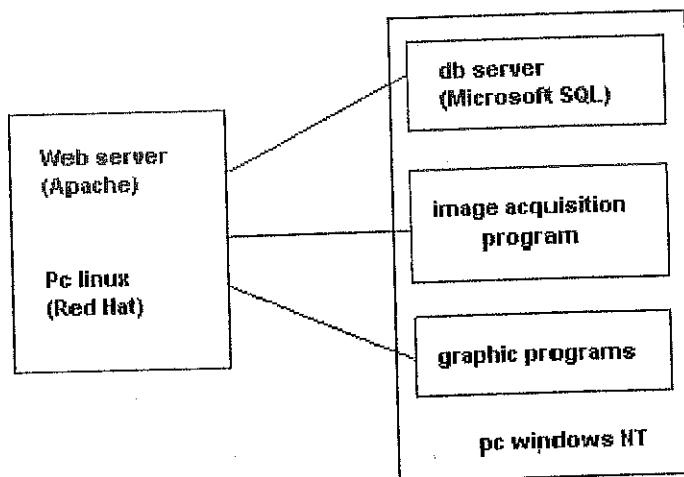


Figure 12: Actual platforms configuration

The ssh access to the web server is checked also by the tcp_wrapper program, so that connections are allowed only from particular IP addresses.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 18/48
			Appendixes: -

Besides, a self signed Certification Authority has been created in such a way that only authorized and certified clients can try to access our web server. Furthermore, some web accounts have been created too, so that each I.N.D.U.C.E. partner can access both an assigned own area and a project common area. Only a super-user can access all the areas and only from the console.

On the server platform also a sniffer (tcpdump) is installed able to catch all the connection tries, what's more logging them to permit us adding other filters and checks in future.

In the following figure we can see an example from this log.

```
Log.txt:10:04:26.568288 eth0 < pc-salvetti.iei.pi.cnr.it.1466 > inducesrvs.iei.pi.cnr.it.www: P
334:334(0) ack 19129 win 8760 (DF) (ttl 128, id 35855)
Log.txt:10:04:26.625146 eth0 < pc-salvetti.iei.pi.cnr.it.1468 > inducesrvs.iei.pi.cnr.it.www: P
0:353(353) ack 1 win 8760 (DF) (ttl 128, id 36367)
log.txt:10:04:26.627258 eth0 < pc-salvetti.iei.pi.cnr.it.1469 > inducesrvs.iei.pi.cnr.it.www: P
0:354(354) ack 1 win 8760 (DF) (ttl 128, id 36623)
10:24:11.106322 eth0 B arp who-has inducesrvs.iei.pi.cnr.it tell pc-martinelli
10:24:11.119196 eth0 < pc-martinelli.1257 > inducesrvs.iei.pi.cnr.it.ssh: S 27556
07:2755607(0) win 8192 <mss 1460> (DF) (ttl 32, id 35110)
10:24:14.382092 eth0 < pc-martinelli.1257 > inducesrvs.iei.pi.cnr.it.ssh: S 27556
07:2755607(0) win 8192 <mss 1460> (DF) (ttl 32, id 35366)
```

Moreover, a particular software that keeps trace of any change happened on our computers has been installed so that we can check the system integrity (tripwire). It creates a log and mails it to the super-user.

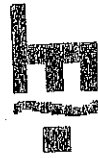
```
Directory tree is valid
Changed files/directories include:
changed: -rwxr-xr-x root 1144 Mar 13 17:15:56 2000 /etc/rc.d/rc.local
changed: -rw-r--r-- root 1192 Mar 14 10:15:33 2000 /etc/sysconfig/hwconf
. . . . .
changed: -rw-r--r-- root 26007 Mar 14 10:15:23 2000 /lib/modules/2.2.12-20/modules.dep
changed: -rw-r--r-- root 12054 Mar 12 04:22:48 2000 /usr/lib/perl5/man/whatis
```

In addition, many processes have been disabled, like the dangerous sendmail and lpd, so that the relative ports cannot be used to try hacking.

All these functions have been realized on platforms under different operating systems mainly for the following reasons:

- 1) the costs of the software
- 2) the possibility to use a standardized database familiar with most of the partners
- 3) the security level of a web server installed on a Windows NT
- 4) the recovery of advanced proprietary software packages for graphic and image acquisition and processing running only on windows NT.

To check I.W.E. security a test phase has been also activated involving our partners asked to try hacking the system: in this way, all the information acquired could be used to increase the security level of the system.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 19/48
			Appendixes: -

3.2 Web-Server Site Implementation

The graphic interface of the Web-Server Site has been subdivided into two separate parts. To access the public environment it is necessary to use the http://hostname.domain URL, instead for the private one the URL is https://hostname.domain:portnumber. The philosophy at the base of this subdivision is that the less it is possible to imagine, the less it is possible to understand; substantially, by dividing physically the public from the private environment a first security level is realized. In fact, who is connected to the public site cannot imagine the presence of the private site and so it is not stimulated to violate the security. Remanding the deepen of these aspects to the following paragraph we limit ourselves to list here the various levels of security that has been realized:

1. Logical division of the public from the private environment (also through built-in options of the apache web server) (Figure 13) ;
2. Connection and exchange of encrypted information through the SSL protocol;
3. Server certification;
4. Client Certification;
5. Access to the private area through a password;
6. Access allowed only from specified IP addresses;
7. Update of the system using encrypted shells.



Figure 13: Division of the public from the private environment

The web site, that represents the graphic interface of the Server, has been implemented using HTML language and Fronte Page 2000 as base program; in some cases, to make more standard the visualization of the pages and to minimize the incompatibility problems among the various browsers, it has been necessary to edit manually. The images and the relative graphic has been realized using instruments like Word Art, Paint Shop 4, Corel Draw, and other applications oriented to image manipulation.

3.2.1 Public environment implementation

The page of the public environment is characterized by the presence of a table of contents, on the left side, that allows to navigate from a section to another, and by a sub-index, on the right side, that is relative to the actual section (Figure 14). It has been chosen this structure because of its simple intuition also by the least expert navigator and since it allows to pass from a section to another limiting the possibilities of loosing inside the various HTML pages. In Figure 15 the *home page* to enter this environment is shown.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: I	Pages: 20/48
			Appendixes: -

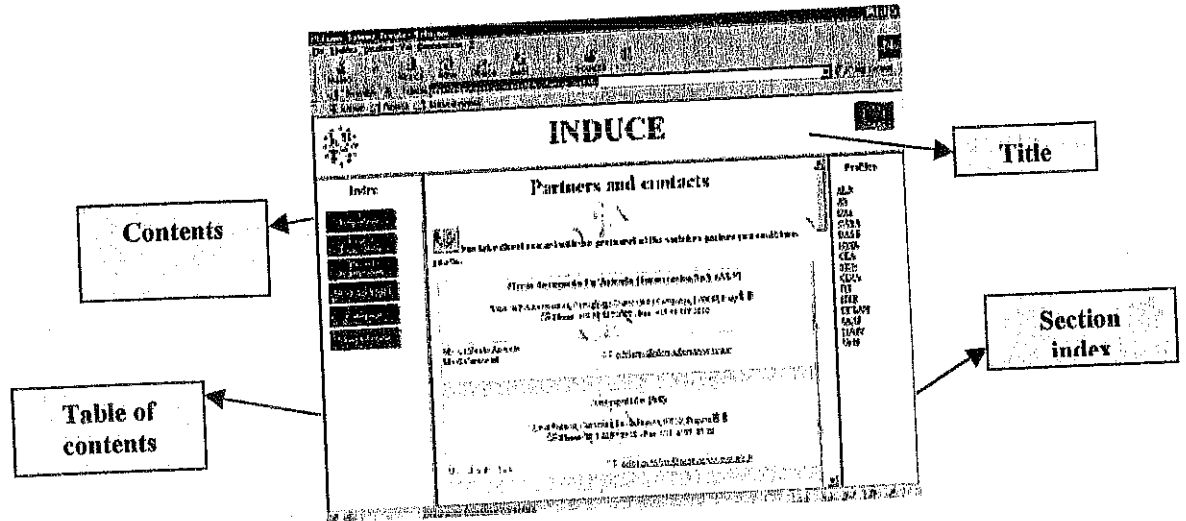


Figure 14: Public page structure

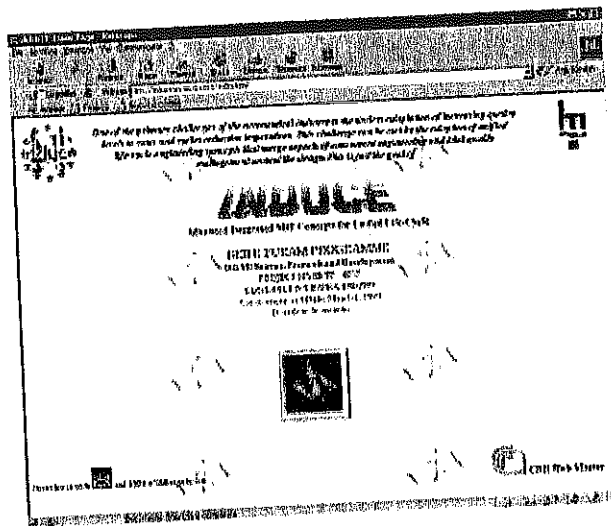


Figure 15: Home page



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 21/48
			Appendixes: -

The principal sections are:

1. *Consortium*: all the data relative to the various partners (Name, Responsible, Telephone, Address, Email) are provided and a page is associated to each partner containing a brief description of his history and of his developed activities and functions within the project; in this page, additional information about the working staff (photos, roles,...etc.) can also be inserted.
2. *Project Description*: a general description of the I.N.D.U.C.E project is presented with attention to the technical goals and the approaches carried on to solve the problems faced; a description of the NDT techniques used, of the Industrial goals, of the problems relative to the project and of the state of the art are presented too.
3. *Project Organization*: a graphic schema of the consortium structure is provided.
4. *Work-Packages*: the various work-packages are described, providing the goals to be reached and sometimes the involved partners (Figure 16).

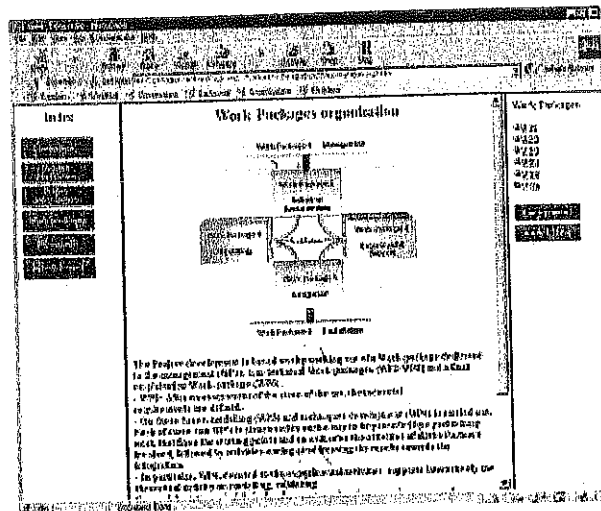


Figure 16: Work-packages organization



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 22/48
			Appendixes: -

3.2.2 Private environment implementation

This environment can be accessed only by the members of the consortium connected to the <https://hostname.domain:port> URL.

In the first phase of the connection the server is certified; the server certificate is shown to the remote user and, in a transparent way, the SSL protocol is activated (from now on all the information transferred will be encrypted).

Once the certificate has been verified, the private home page is displayed (only for the users having an authorized IP address) and, clicking on the entry button, the username and the relative password are requested.

Note that a partner has only the rights to visit common areas and his private *folders*, but not the private folders of the other partners.

To better understand the concepts up to here mentioned, an example of a practical *work session* is shown in the following.

While we leave you finding information about the project, we start a certified and encrypted connection on the private site.

As we can see in Figure 17, our browser signals us that the certification authority is not one of the default set since we are using a self signed one.

By default, the https port is the number 443, even if we have chosen a different one in order to mislead attacks and systematic scanning of hackers.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 23/48
			Appendixes: -

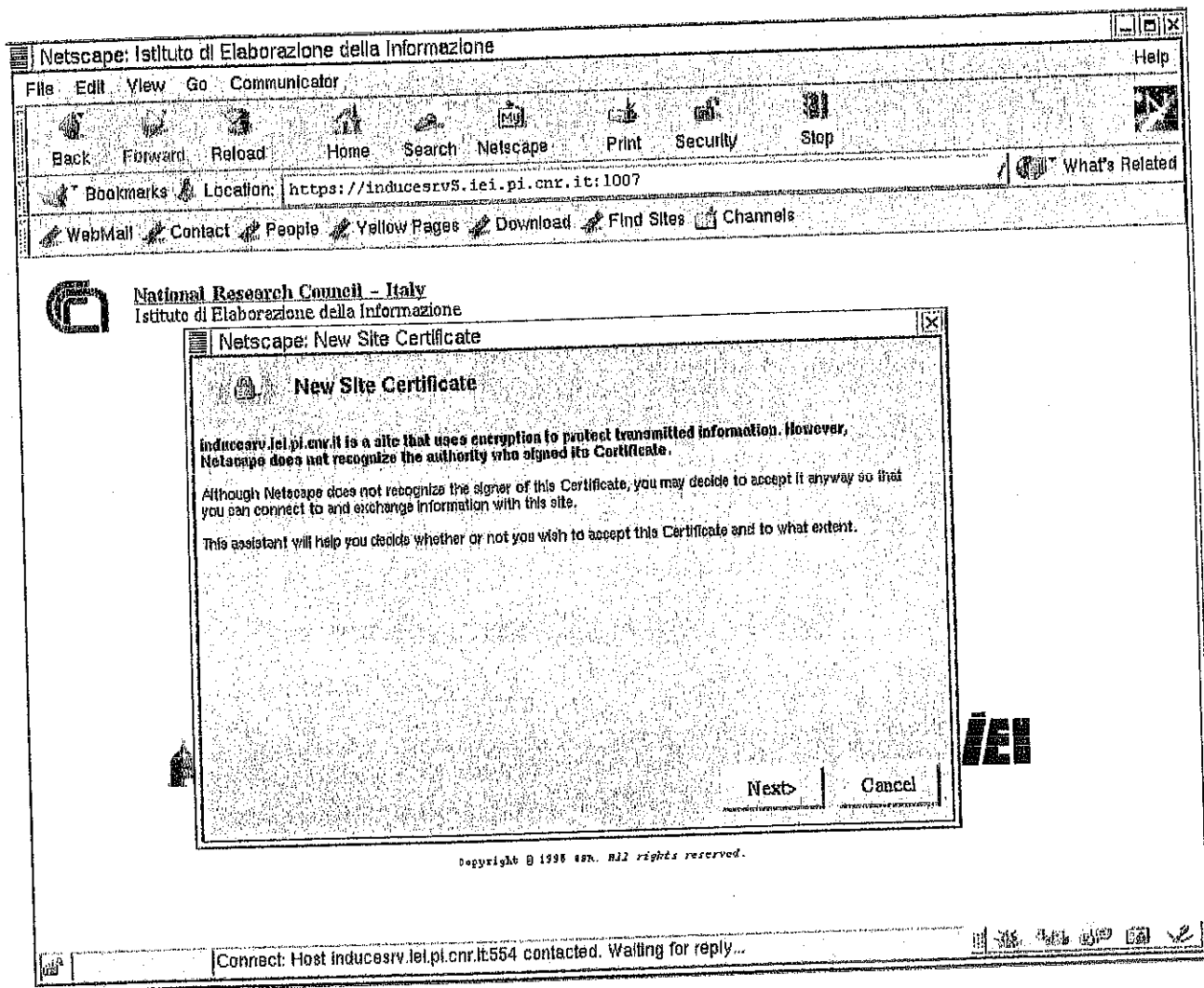


Figure 17: New Site Certificate

Then we can see the certificate the server presents us (Figure 18).



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 24/46
			Appendixes: -

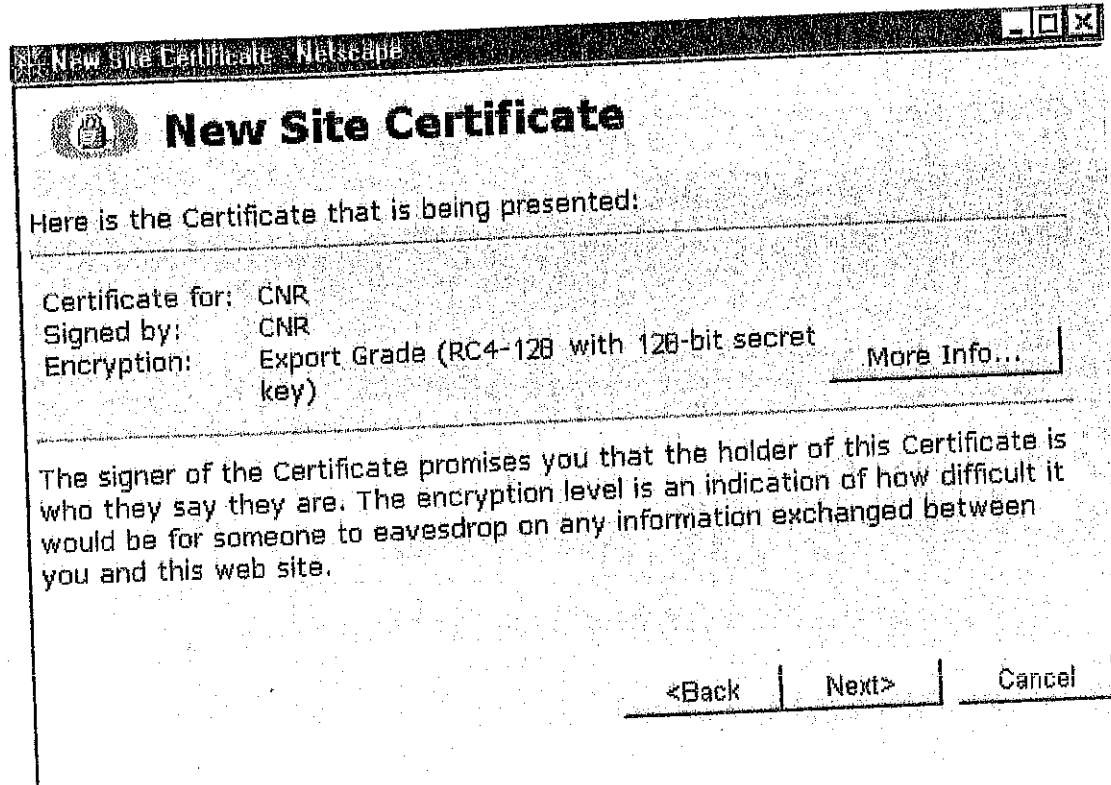


Figure 18: Certificate presented by the server

Clicking on the "More information" button, other information is displayed about this certificate (Figure 19).

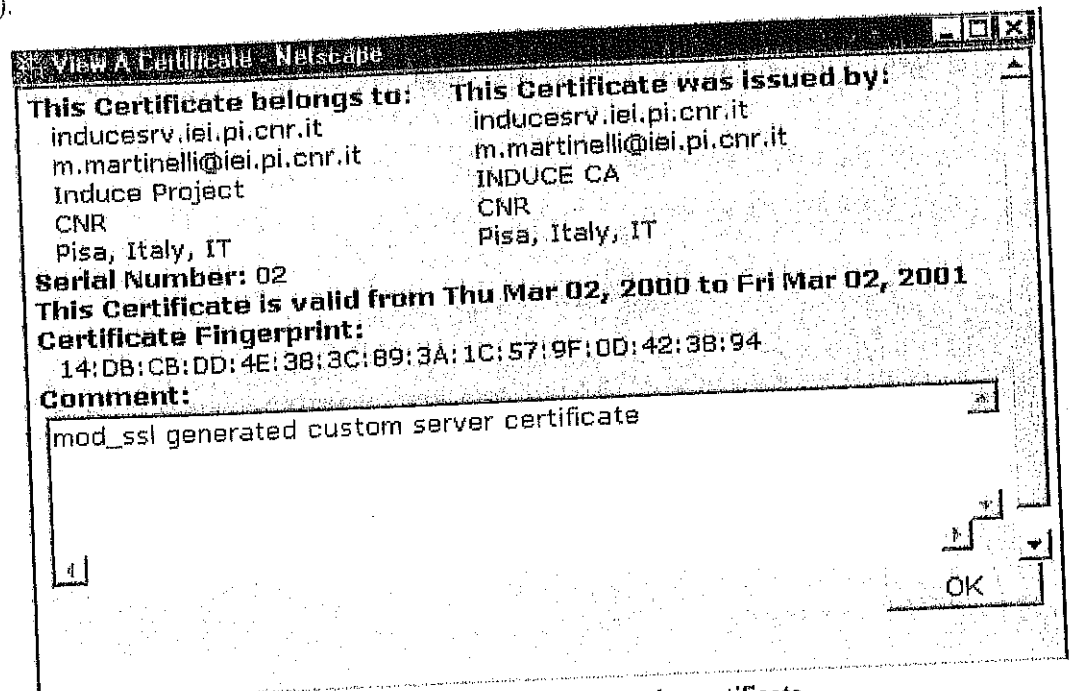


Figure 19: More information about the certificate



Date: 15.03.2000	Reference N°: <u>TR-2140-IE1-01</u>	Revision: 1	Pages: 25/46
			Appendixes: -

At this point, we can choose whether or not to accept the certificate and, if yes, we can accept it for this session or forever (Figure 20).

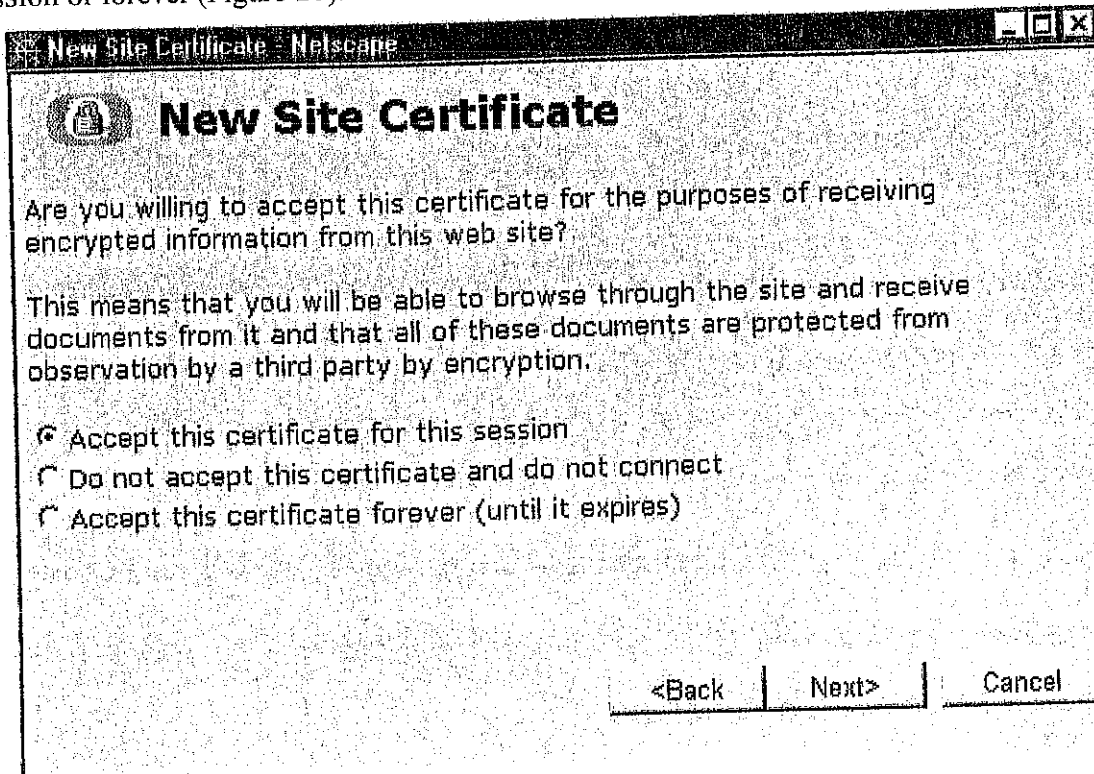


Figure 20

On demand, we can also be warned every time before sending information to the site (Figure 21)

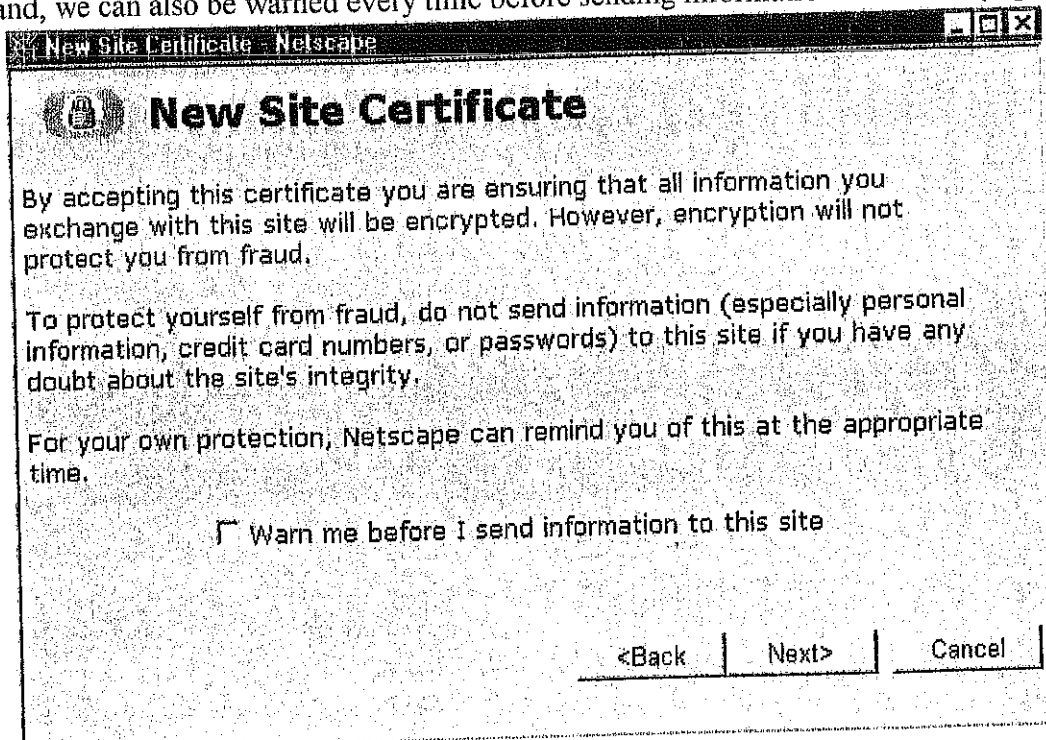


Figure 21



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 26/46
			Appendixes: -

The browser shows a summary of the steps we did (Figure 22).

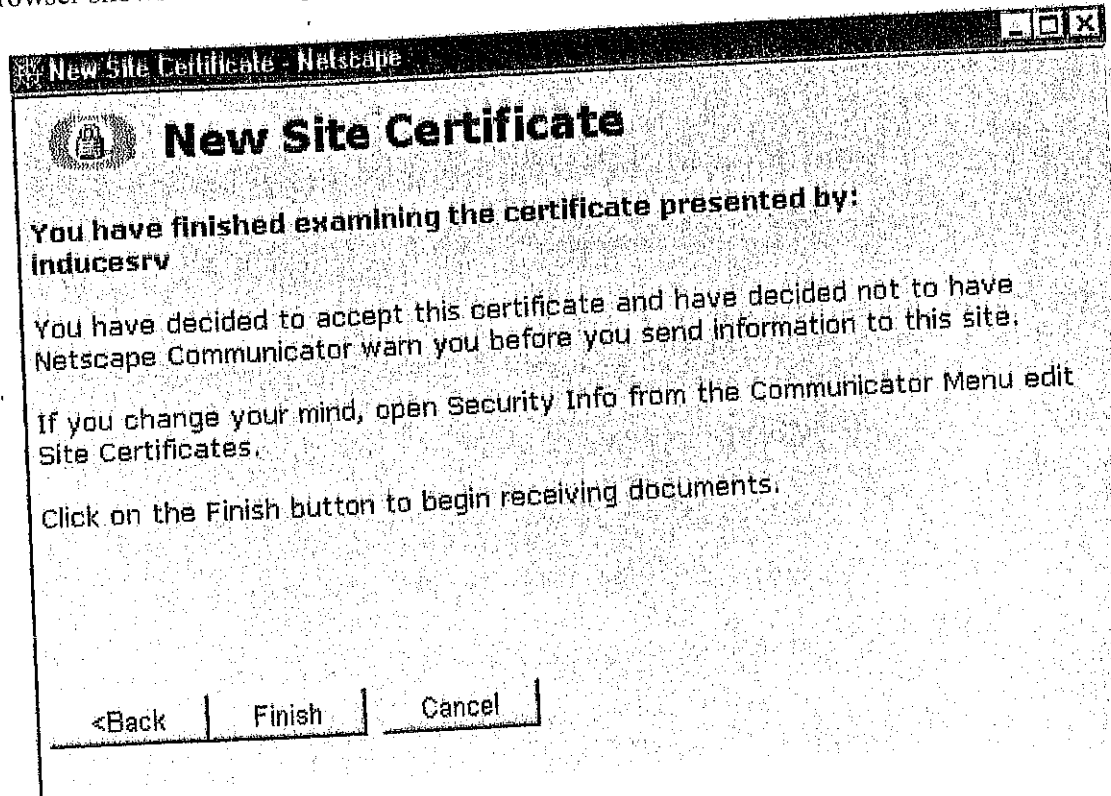


Figure 22

Now the server requires a valid certificate from our client (Figure 23).

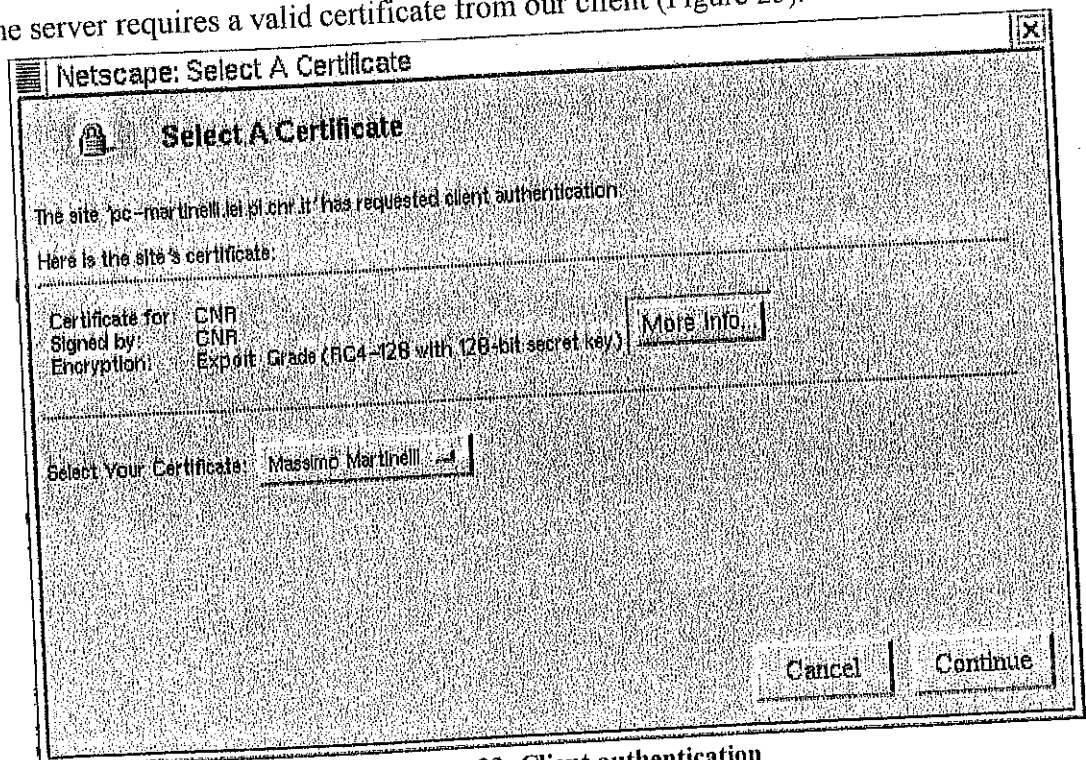


Figure 23: Client authentication



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 27/46
			Appendixes: -

The server recognizes our certificate and allows the connection (Figure 24).

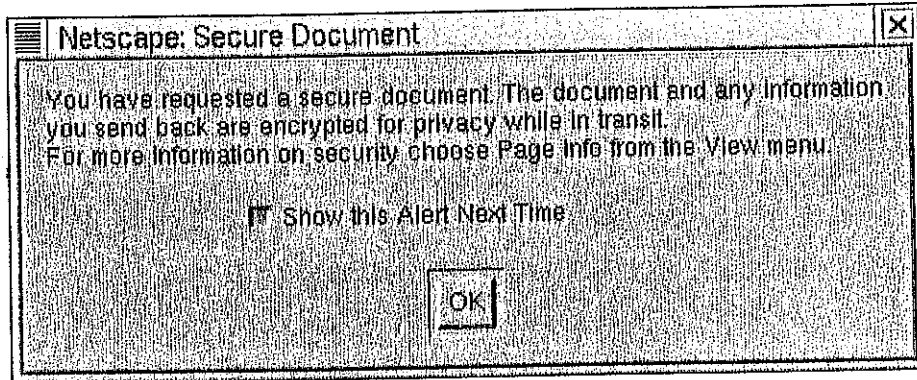


Figure 24

Finally, we are connected.

As we can see, the locks in the evidenced icons are closed, to signal us that we are using an encrypted connection. (Figure 25).

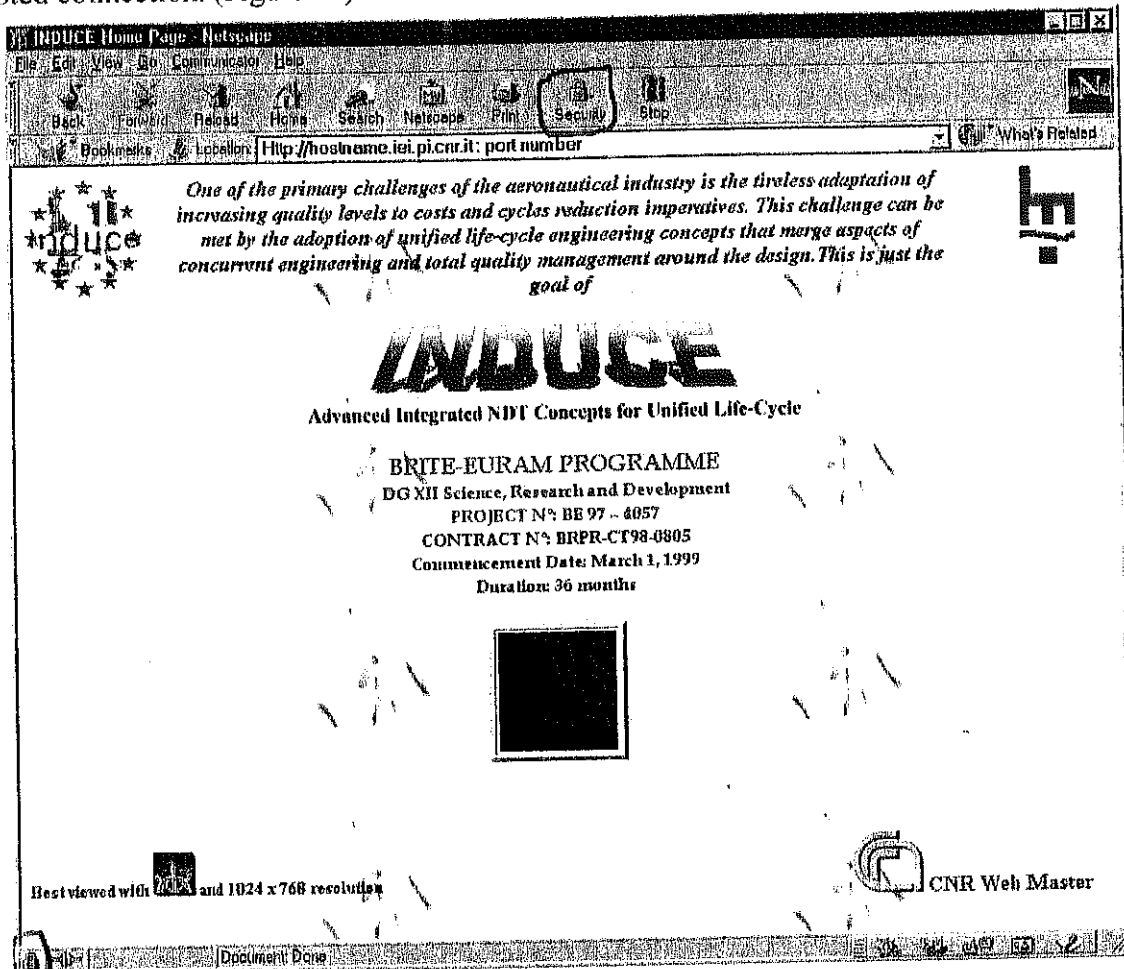


Figure 25



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 28/46
			Appendixes: -

Clicking on the central image (red surface), the server requires a *user id* and a *password* (Figure 26).

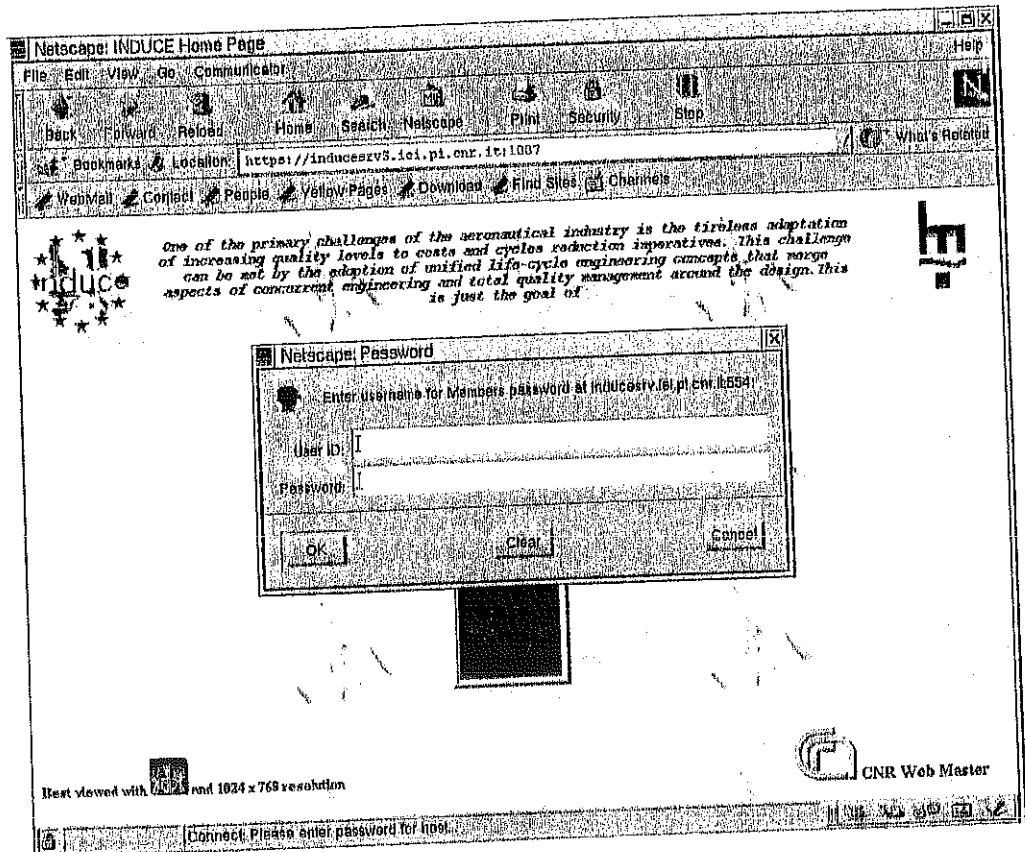


Figure 26: Username and Password

Giving the *user id* and the *password* of any partner of the project we reach the introduction page of the I.N.D.U.C.E. Web-Server Site (Figure 27).



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 29/46
			Appendixes: -

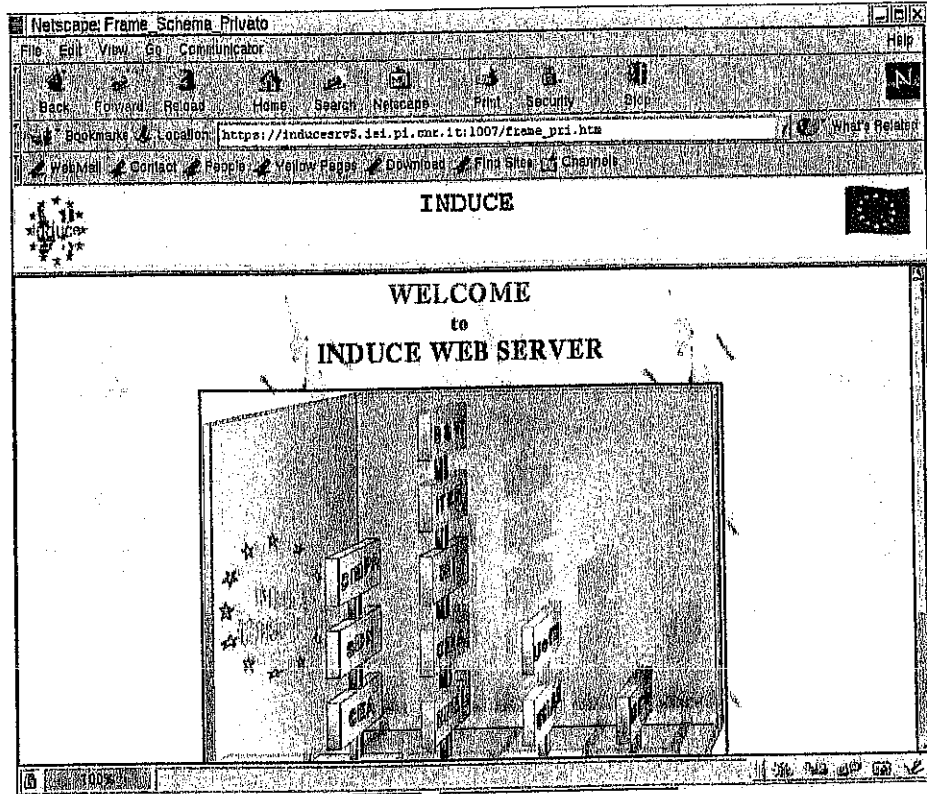


Figure 27

This area is reserved to any valid user, being a common part for all the partners.

Clicking anywhere inside the presentation page, six different environments are displayed (Figure 28):

1. *Communication Environment*: this environment allows to exchange information among the partners (Figure 30).
2. *Algorithms Collection*: this environment contains a collection of algorithms and their documentation (codes, test reports, comments, etc.). The algorithms are subdivided into categories and each one is inserted in a row of a table. Each row, in its turn, is subdivided into a fixed number of fields where the main features of the algorithm reported are described. Each field of the row is clickable allowing to download the code and the documentation (Figure 33).
3. *Image Archive*: this environment allows to access a digital image archive, simply managed as a file system. Some queries are possible to recover the images in their raw version or after processing (Figure 32).
4. *Local/Remote Processing*: this environment permits to run proprietary or commercial applications in a local or remote way: possible operations are image processing, film acquisition and analysis or simulation using Inspector[®], AVS[®], Matlab[®], etc (Figure 31).



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 30/46
			Appendixes: -

5. *Dissemination*: this environment contains documents released by the partners or technical and scientific articles suitable to be spread outside the consortium.
6. *Product Validation Environment*: this environment defines a laboratory for testing software products developed by a partner.

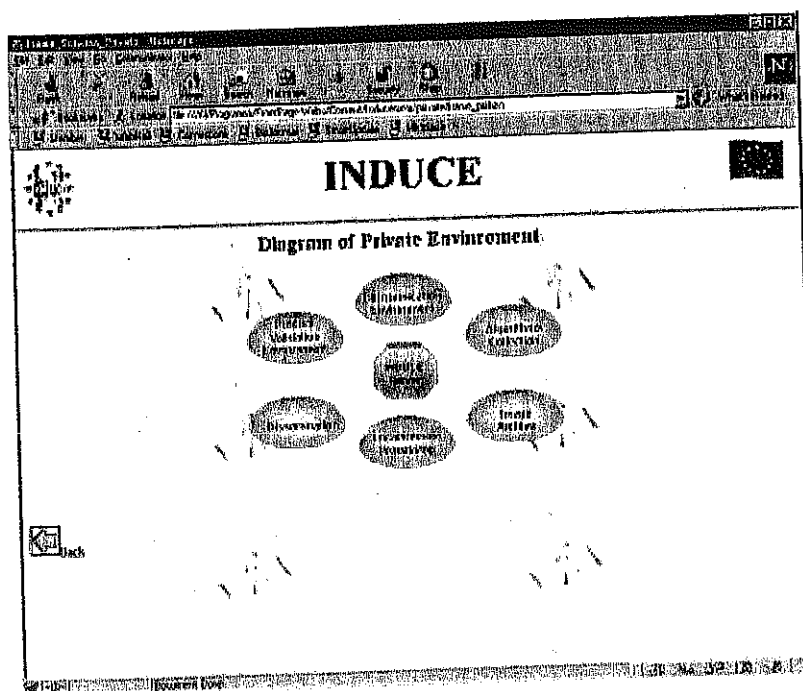


Figure 28

The *Communication Environment* is in its turn subdivided into four sub-environments:

- *Share box*: this area is used for exchanging messages and draft documents to be completed.
- *EU-Box*: this area is used for exchanging messages between the consortium and some representative of the European Community; the access to this box is allowed only to the representatives and Alenia.
- *Partner Box*: each partner can access a private area through an IP address and a password. This box can be used by some partners to exchange personal information not relevant for all the consortium members.
- *PSC/PCC-BOX*: the members of the PSC and PCC Committees can use these boxes to exchange information not relevant for all the consortium members.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEL-01</u>	Revision: 1	Pages: 31/46
			Appendix: -

In Figure 29, an example is shown regarding an access deny from the server following an attempt to enter a reserved partner area.

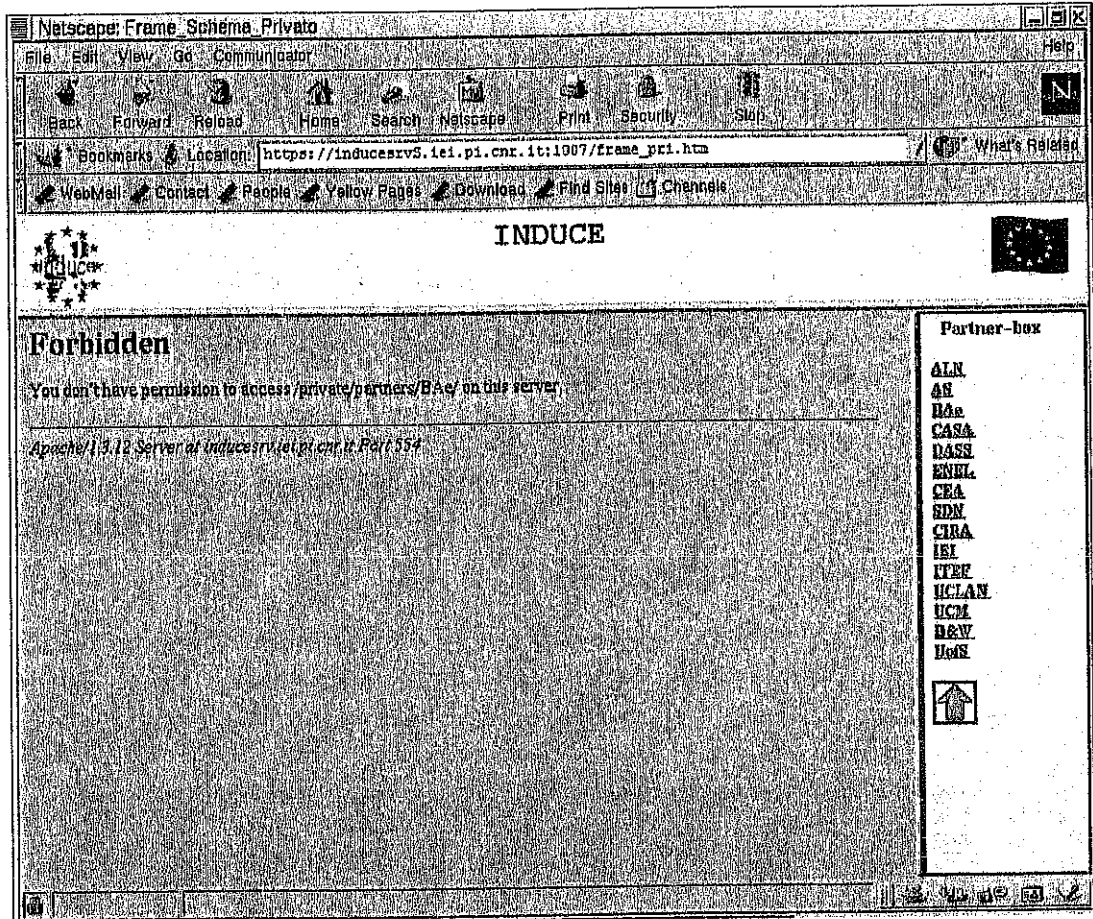


Figure 29: Access denied

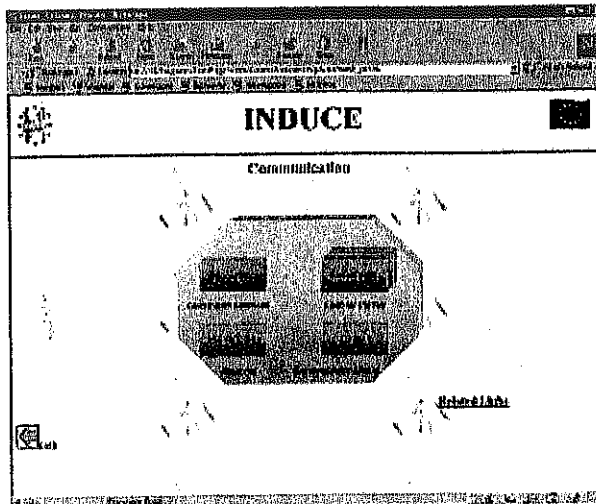


Figure 30: Communication Environment

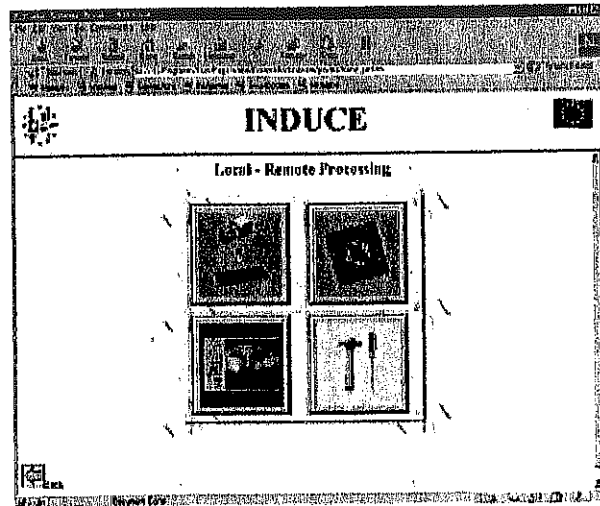


Figure 31: Local - Remote Processing Environment



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 32/46
			Appendixes: -

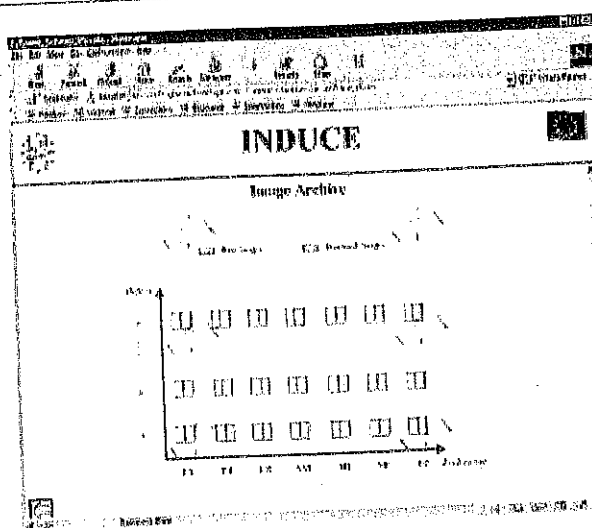


Figure 32: Image Archive

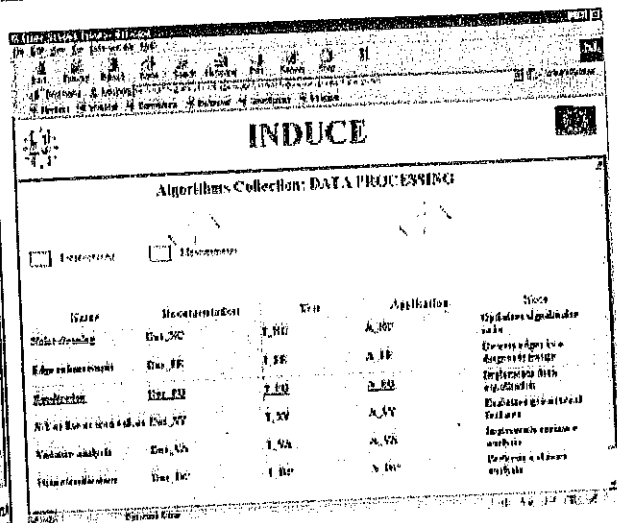


Figure 33: Algorithms Collection

3.3 Remote Acquisition Implementation

The remote acquisition program is based on a procedure specifically implemented at I.E.I. for image acquisition by a thermocamera. In this case, the user interface has been substituted by a server module in order to be used by a web browser. The user interface of the (local) acquisition program is shown in Figure 34.

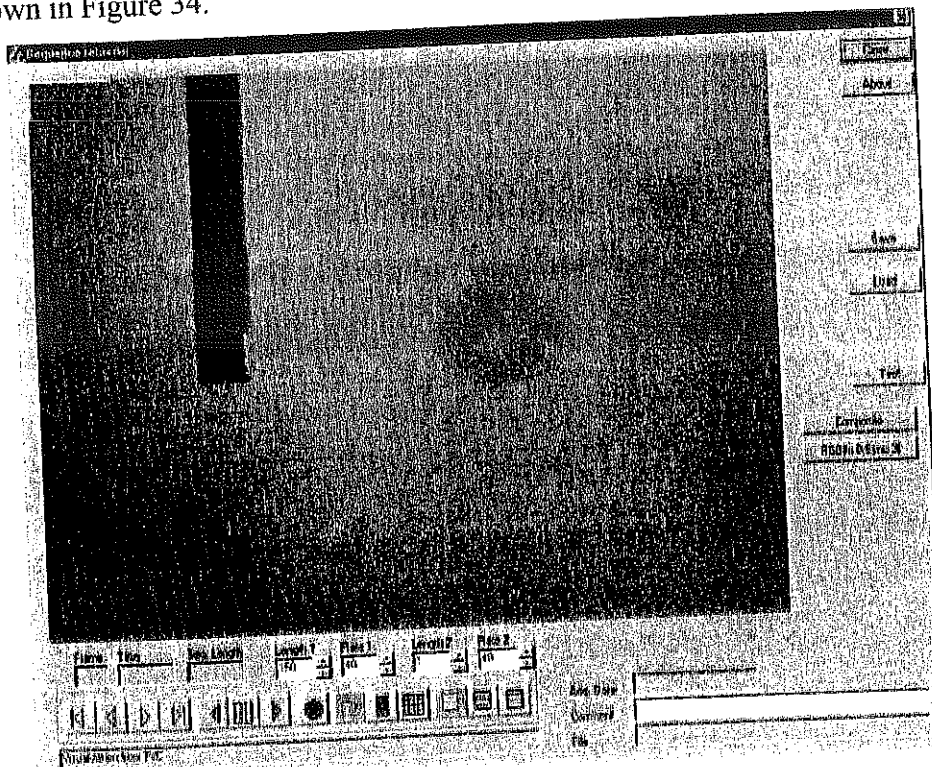


Figure 34: Interface



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 33/46
			Appendixes: -

This program allows to acquire a sequence of images by a thermocamera. The sequence is characterized by the number of images to be digitised and by the frequency of image acquisition: both these parameters can be set through specific buttons. Once the sequence has been acquired, its frames can be viewed using the VCR type buttons. Also the set of frames can be viewed as a series of icons.

On the lower left side of the window shown in Figure 34, some buttons allow the user to load/save an image sequence from/to the hard disk of the computer.

The text boxes have the following meaning:

- Frame: number of the currently visualized frame
- Time: time at which the current frame was acquired (offset from the first frame)
- Seq. Length: number of frames of the current sequence
- Length 1: number of frames in the first section of the sequence
- Rate 1: acquisition rate for the first section
- Length 2: : number of frames in the second section of the sequence
- Rate 2: acquisition rate for the second section

The buttons are defined as follows:

	select the first frame of the sequence		select the previous frame
	select the next frame		select the last frame of the sequence
	play backward		stop playing
	play forward		acquire a sequence
	live visualization		recorded sequence visualization
	thumbnails visualization		full frame visualization
	odd field visualization		even field visualization



Date: 15.03.2000	Reference N°: <u>TR-2140-IEL-01</u>	Revision: 1	Pages: 34/46
			Appendixes: -

The scheme of the program is the following (Figure 35):

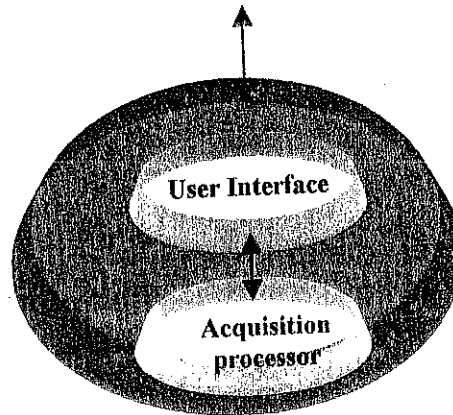


Figure 35: Local program structure

This scheme, as we can see, is similar to the one shown previously in the project description. The difference is mainly due to the presence of a graphic user interface instead of the server. However, it is possible to use the 'acquisition processor' module with minimum changes for realizing the program (server side) of remote acquisition.

The server for the acquisition should be executed at the system start. At this purpose, the 'Acquisition Service' module has been added since it allows the automatic activation of the server at the bootstrap of the operating system, in this case Windows NT. The role of the 'Acquisition Service' module is to communicate with the Service Control Manager (SCM) of Windows NT. It has the function to accept from SCM the Start, Stop, Pause and Continue commands and to let SCM know the state of the Service (Running, Stopping, etc.). Once the service has received the start command from SCM it will start the acquisition server. The program implemented includes also the functionality to install and uninstall the service in the system. The modality by means of which it is activated (manual, automatically at the start) can be set through the Windows NT control panel. Thus the program scheme can be completed as follows (Figure 36):



Date: 15.03.2000	Reference N°: <u>TR-2140-IEL-01</u>	Revision: 1	Pages: 35/46
			Appendixes: -

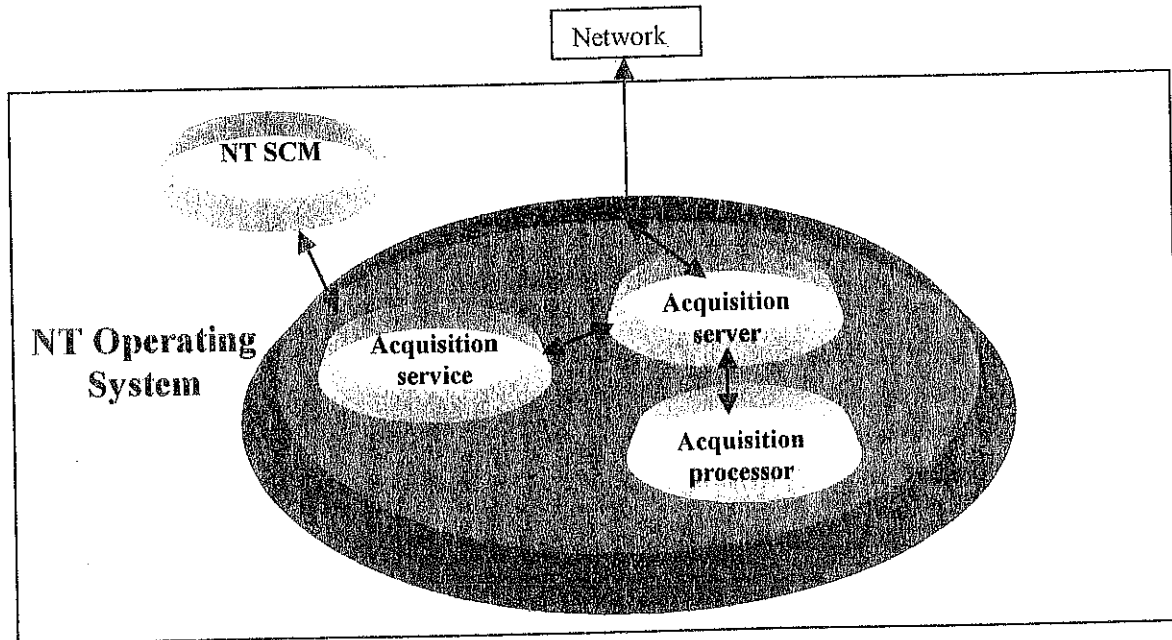


Figure 36: Remote program structure

The acquisition program has been developed in C++ using Microsoft Visual Studio. The acquisition Matrox Meteor card has been used to implement the acquisition processor module and the Mil-Lite 4.0 libraries has been used for his management. The module, with few changes, can be employed to support the Matrox Pulsar card.

The Microsoft Win Sock2 library has been used to realize the acquisition server module that manages the connections via Network. MPEG format could be used for image transmission.

A *client program* able to support the communication protocol used by the acquisition server has been implemented for a test phase. The client sends a request asking for the acquisition of a certain number of images, waits until it receives an image and then it stores each one on a single file. The program was designed to be executed under a DOS shell, specifying the name of the computer on which the server was located, the port number used by the same server and the number of images to be acquired. The server acquires a sequence of the requested number of images and sends them to the client one at a time. The client stores the received images in single files, compressed as Tiff format.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEL-01</u>	Revision: 1	Pages: 36/46
			Appendixes: -

```
D:\Induceserver\NetAcq>client pc-acquisition 9999 25
Host to connect: pc-acquisition
Size of data: 25
Connected with server
Image 00 correctly received
Image 01 correctly received
Image 02 correctly received
.
Image 21 correctly received
Image 22 correctly received
Image 23 correctly received
Image 24 correctly received
```

In a second time, the client has been modified to run under a browser. In this way, a sequence of images can be acquired by clicking on a button created in the remote processing environment. An HTML page is loaded and shown to the user, then specific acquisition parameters can be sent to the client by filling in the fields of this page (at the moment just the number of images). The client is a CGI that provides to connect the acquisition server, sends the parameters and once received the images saves them and produces an HTML page displaying a series of icons for giving a preview of the acquired images (Figures 37, 38, 39). In a first phase, only the icons are transferred to give the user the possibility to choose interesting images, and then, in a second moment, the selected images can be downloaded. A button gives also the possibility to download the entire sequence, compressed to create a unique file.

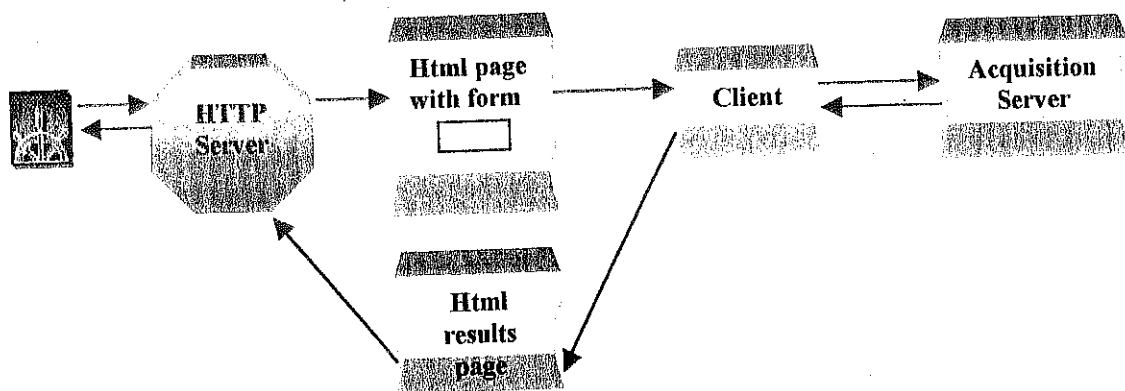


Figure 37: Interaction with program acquisition



Date: 15.03.2000	Reference N°: TR-2140-IEI-01	Revision: 1	Pages: 37/46
			Appendixes: -

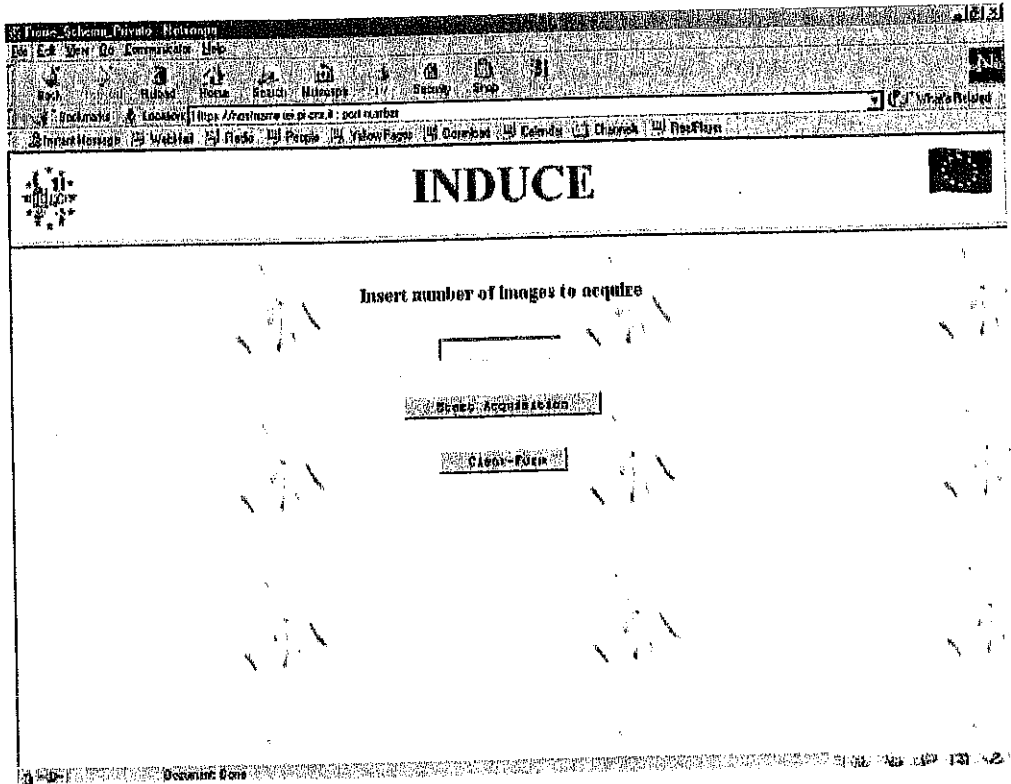


Figure 38: Example of the test-oriented interface for remote TC acquisition

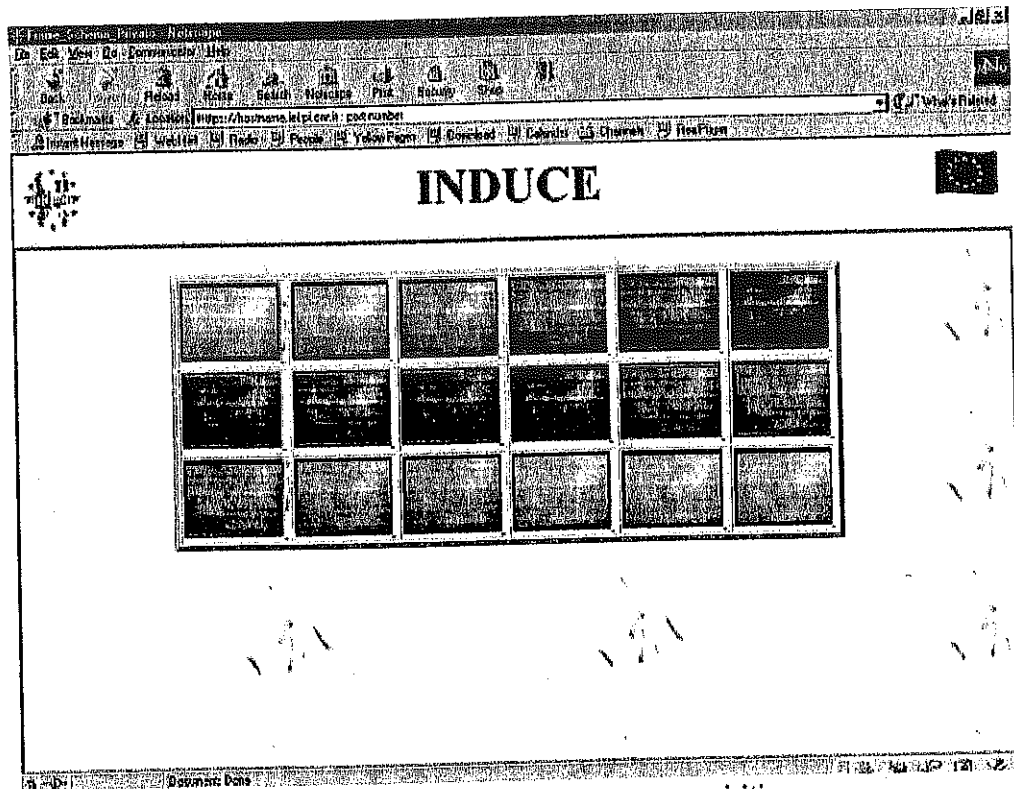


Figure 39: Result of image sequence acquisition



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 38/46
			Appendixes: -

3.4 Database implementation

In the following, the approach to make accessible via Web browser the INSIDE NDT database is described. The database was implemented using Microsoft ACCESS 97.

3.4.1 Pre-Analysis

In order to allow remote hosts to access information contained in the database, a server program must be running on the system where the database resides. The server program will wait for requests from processes running on other hosts, and will send them the information retrieved from the database. If the Database Management System (DBMS) used provides a database server, this can be used to allow remote requests. As an alternative, an ad hoc server can be used to directly access the local database or to interact with an available database server. Microsoft ACCESS does not provide a database server. Application software using ACCESS databases is implemented by means of specific libraries and drivers that allow them to directly access the information stored in a local database.

We have thus two choices for the implementation of the system:

- Use the ACCESS database
- Use a DBMS that provides a database server.

3.4.1.1 Use of a database server

If we choose to implement the system with a database server, we need to convert the database in order to be accessible by the database server. The migration of a database between different DBMSs can, in general, cause some incompatibility problems due to the specific implementations of the various database features. The migration will require the use, and possibly the implementation, of a conversion procedure. The database obtained by means of this procedure should be equivalent to the original one and the conversion should take into account the database structure (e.g., tables, keys, relations, indexes, constraints, etc.), data already stored and software already implemented.

Among the several DBMS's available in the market, we have considered Microsoft SQL Server 7 as a possible DBMS to which the original database can be converted. This choice is based on two reasons: the availability of a conversion tool from Microsoft Access, and the availability of the Microsoft SQL Server 7 package on the used platforms. While the adoption of SQL Server 7 will cause the migration problems listed above, it can offer some advantages, such as a better performance with a database that has a complex structure and contains a great amount of data. Moreover, it allows expert users to access some control functions of the database and its components.

3.4.1.2 Use of ACCESS

Using ACCESS database requires a server program to process the incoming remote requests. On the receipt of a remote request by a client, the server will retrieve the required information from the database and will send them to the client. An ad hoc server can be implemented for this purpose.

D
A
hc
W
H
d
3
B
1
2
3
/
c
t
t



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 39/46
			Appendixes: -

Alternatively, some other server program can be used as a means of connectivity between remote hosts and the database: for example a Web server can be used to allow for remote requests from a Web browser, or from another Web server (such as the Induce Web Server). In this case some HTML pages, CGI programs or ASP will be implemented to access the information contained in the database.

3.4.2 System structure

Based on the chosen DBMS we can design the system using two different structures:

1. Point-to-point connection.
2. Connection by server process.

3.4.2.1 Point to point connection

A client can be connected with the database server with a point-to-point connection. In this case the client will send its requests directly to the server and will obtain the required information. Using this technique the client program will use a library of functions to communicate with the server using its specific protocol.

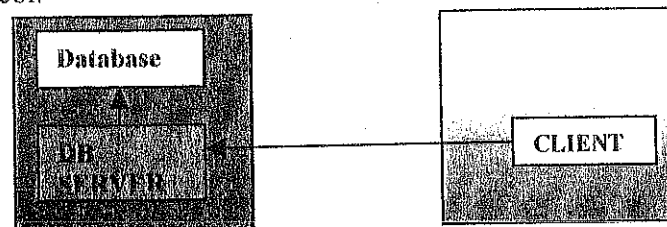


Figure 40: Point to point connection between client and database server

This scheme (Figure 40) can be used when the appropriate development tools, for the connection to the server, are available for the implementation of the client. Moreover, in some cases, the installation of additional software on the database host is not accepted: this can be due to security issues and to guarantee the stability of the system.

3.4.2.2 Connection by server process

A scheme in which a server process is running on the database host can be used. The server process is the means by which the client accesses database information. In this case the database is directly accessed by the local running server process. For the implementation of this server process we can thus use the database development tools provided by Microsoft. This scheme can also be used with SQL Server 7: in such a case the server process will just redirect the client requests to the database server, and the server responses to the client. This can be useful if the appropriate database development tools for the client host (in our case Linux) are not available.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 40/46
			Appendixes: -

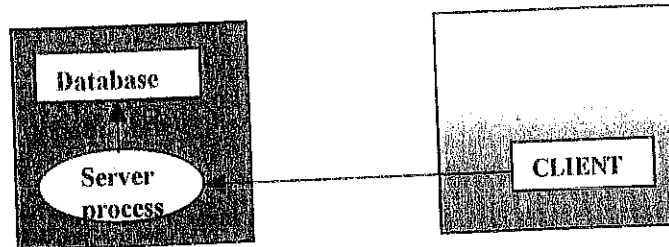


Figure 41: Connection by server process

The advantage of this scheme (Figure 41) is, in our case, the availability of database development tools for applications that must access the database, directly or by means of a database server. Using these tools it is possible to implement a program that waits for requests via network and accesses the database.

Using this additional server program is also possible to add some functionalities. These can be implemented to minimize the amount of data sent via network. Another possible use is for a stronger access security control on the incoming connections and the implementation of some predefined procedures to query the database. A stronger access control can be used, for example to allow connections only from a predefined list of hosts. In our case the access to the database could be limited to the Induce Web Server host, if desired.

3.4.3 Implemented prototypes

Given the different possibilities for the structure of the system, two prototypes have been implemented. The first one is based on a point-to-point connection between the client and the SQL Server 7. The second one is based on the use of a server process, a Web server, that accesses the Access database in response to the remote requests, and sends the required information to the client.

3.4.3.1 Connection between a client in Linux and server SQL 7

We will now describe in details the steps needed to establish a connection between a client running on a Linux host and a SQL 7 database server running on a Windows NT host.

The scheme that we used is the point-to-point connection. By searching for available database development tools in Internet, both in Microsoft site and others, we found there was no such tools specifically available for the connection to a SQL 7 Server by a client running on Linux. In particular Microsoft, in general, does not support development tools for non-Microsoft systems. However a library developed by SyBase, called Ct-Lib, implemented for the connection to the SyBase server has been found. The Microsoft SQL Server 7 uses the same protocol, called TDS; used by the SyBase server and so we tried to use this library for the connection to the SQL 7 server.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 41/46
			Appendixes: -

The original library implemented by SyBase is based on the 'a.out' object format, used on old versions of Linux and thus could not be used on our Linux system. A version of this library for the new 'elf' format was found by means of Internet searching.

This library has been downloaded from URL <http://www.mbay.net/~mpeppler/Sybper/ctlib-linux-elf-dynamic.tar.gz>. This version could be successfully used only for the compilation of the client since, due to some bugs, could not be able to establish a connection with the SQL 7 server. We found a patch, named 'ctlib-rhlinux-fix', for these bugs in Internet.

The next attempt to connect to the SQL 7 server resulted in a login error message from the SQL server. This error was due to a bug in the SQL 7 server. This problem was known at Microsoft that solved it with a patch contained in the SQL 7 Service Pack 2.

At the time of our test this Service Pack was still in Beta version, but we decided to install it anyway. After its installation we could successfully connect the client to the SQL 7 server. The connection from the client to the SQL 7 server can however also fail due to other problems not depending on the libraries used. If, for example, the SQL 7 server is configured to allow only a single concurrent connection, due to license limitations, and a single client program tries to establish multiple connections, it will fail. In our tests we found that the SQLServerAgent, shipped with the SQL 7 server, uses one license when is running; this also applies to the Enterprise Manager, a useful tool for the SQL server administration. So, if there is only a single license available these processes must not be running to allow other clients to connect.

If this scheme will be used, an appropriate client program that uses the Ct-Lib can thus be implemented. This client program could be, for example, a CGI program used by the Web server to obtain the required information from the database.

In the implementation of the client we can decide to let the user specify any query or other command, or we can limit the access to the database allowing only the use of some predefined queries. In the first case the server simply acts as a process that routes information to/from the database server. In the second case the client implements some predefined functions on the database that, though limiting the functionality of the database, can provide a simpler interface even for users having no specific knowledge on database details.

To perform some tests on the possibility to establish a connection with the SQL server we used a sample program provided with the Ct-Lib library. In the code of this program the appropriate values to be used for the connection and the query have been set. Among the most meaningful values we set:

```
#define EX_SERVER      "SQLSERV"  
#define EX_USERNAME   "user"  
#define EX_PASSWORD   "*****"
```

SQLSERV is a name that identifies the database server. In a specific configuration file we set the association between this name and the information to connect to the server, in particular the name of the host and the number of the port on which the server waits for connections. A variable in the program contains the name of the database to use for the query.

```
CS_CHAR *Ex_dbname = "Northwind";
```



Date: 15.03.2000	Reference N°: <u>TR-2140-IE1-01</u>	Revision: 1	Pages: 42/46
			Appendixes: -

In the following line the SQL command to submit to the database server is defined

```
/*  
** Define a sample select statement  
*/  
#define SELECT "select CategoryName from Categories"
```

the following listing contains the output of the client program that shows the messages and information received from the SQL server.

Compute Example
Context allocated

Server message:
Message number: 5701, Severity 0, State 2, Line 0
Server 'INDUCEWS'
Message String: Changed database context to 'master'.

Server message:
Message number: 5701, Severity 0, State 1, Line 1
Server 'INDUCEWS'
Message String: Changed database context to 'Northwind'.

CategoryName

Beverages
Condiments
Confections
Dairy Products
Grains/Cereals
Meat/Poultry
Produce
Seafood
All done processing rows.

3.4.3.2 Connection by server process

Web interface to database reproduces the INSIDE NDT one allowing its immediate use by project's partners. Its structure is composed by a cover (home page) with copyright information, a main page



Date: 15.03.2000	Reference N°: TR-2140-IEI-01	Revision: 1	Pages: 43/46
			Appendixes: -

used as a menu to access all other functionality and a set of secondary pages to view all available information.

The main page content is composed by five groups representing as many sets of object homogeneous from a logic point of view. All object in anyone group are identified by a button/icon that allow to access to a dedicated section to its management:

- “Examination Object” collects all sections e so all information on objects to test, parts composing object, materials which are made and typical defects.
- “Examination Procedure” collects information on inspection procedure and characteristic, used tools, etc.
- “Examination System” allows consulting information about systems and tools needed to do the tests. So are described hardware and software platforms and devices that characterize it other than measurement tools.
- “Examination Data” allows consulting information about raw data acquired during inspection, applied processes to these data and data so produced.
- “Diagnosis” collects information about founded defects during inspections.

What shown below (Figure 42) it's the main page of INSIDE NDT On-Line interface.

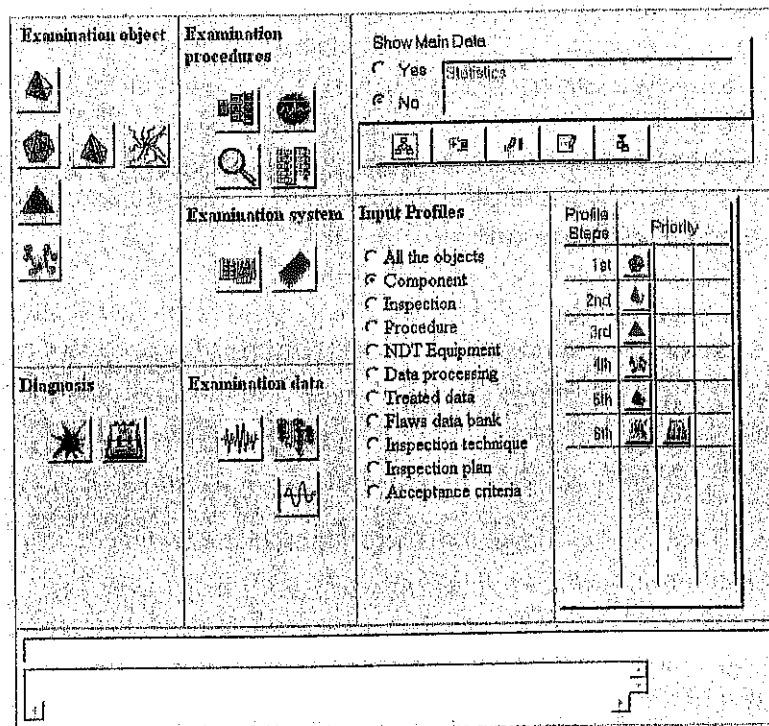


Figure 42: First page



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 44/46
			Appendixes: -

As said above, any section is composed by one or more pages reproducing masks used by original INSIDE NDT interface.

An example section is shown in Figure 43 and reports information about inspection technique and used standards.

INSIDE NDT On-Line: Technique


Method		Technique
IRT		Ispezione Normatizzata 116
Description		Ispezione non distruttiva descritta dal DDL 116/D
Output data type		Digital Images
<< Previous		<u>Next >></u>

Figure 43: Inspection technique

The link called "Next >>" allows to access next records.

Connection between Database and Web server it's realized using Open Database Connectivity (ODBC) driver that allows abstracting from used database. This allowed us using original INSIDE NDT database.

Interface is composed by a set of HTML pages which content it's composed by a fixed part, regarding control elements and base page structure, and a variable one affected by user choices and database content. This last part is built dynamically using the Active Server Page (ASP) technology that allows, With extreme simplicity, to insert within HTML pages, constructs and commands of scripting languages as Visual Basic Script and JavaScript. These are interpreted by the Web Server and allow to use ODBC driver to querying the database.

Actual query structure is very simple. We haven't been used particular technique we have inserted SQL commands directly within the ASP code. These commands are treated as simple string that, if necessary, are processed and modified using Visual Basic Script commands, to insert needed parameters (for example: component identifier, procedure name, etc.) and then passed to relative command to execute it.

This technique also if not seems elegant it has undoubted advantages. Building SQL command allow a very simple system debugging, in fact to view what query is executed it's sufficient adding this string to the page content. Besides the fact that commands to built query are present directly within code, make easy verify what are connections and overlapping, desired or not, between the different part of code allows the final HTML page construction.



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 45/46
			Appendixes: -

3.5 Procedure to update the system

To update the information of I.W.E. a partner should apply the following procedure:

- use the sftp or scp programs to put a file into his/her private areas

For example, a file called file.gif should be sent as follows:

```
scp filename user-id@hostname.domain
```

or

```
sftp hostname.domain user-id  
put file.gif
```

- connect as follows to delete or change a file:
ssh -luser-id hostname.domain
then, look at the directory content using the command
ls -la
then, remove a file using the command
rm file-name.extension
or change the file name using the command
mv filename newfilename
- contact the Web Master to run other command

In any case, the file processed should be only in PDF, HTML, GIF or JPEG formats.
After these operations, please always inform the Web Master.

3.6 Backup

Under the supervision of the Web Master, a backup operation is regularly implemented following a semi-automatic approach, by using an on-line mass storage unit only accessible locally.

4 Conclusions

In this report, the design and implementation criteria of an 'Integrated Work Environment' (I.W.E.) constituting a Web-Server Site for the I.N.D.U.C.E. project were described in detail.
I.W.E. was discussed in terms of its components, functions and internal/external relationships.
Being accessed by a variety of users, a particular care was put on security aspects concerning the Site access rights management.

In particular, in the first prototype of the Server, different security levels were implemented:

- Logical division of the public from the private environment (also through built-in options of the apache web server);
- Connection and exchange of encrypted information through the SSL protocol;
- Server certification;
- Client Certification;



Date: 15.03.2000	Reference N°: <u>TR-2140-IEI-01</u>	Revision: 1	Pages: 46/46
			Appendixes: -

- Access to the private area through a password;
- Access allowed only from specified IP addresses;
- Update of the system using encrypted shells.

In the approach followed, relevant aspects regard the creation of the certification authority and the management of the certificates themselves. In order to improve this process, we are now thinking to use an European certification authority, like the EuroPKI top level certification, that is an organization established to create and develop pan-European public-key infrastructures.

Open problems that we think to solve during the development of the project concern mainly the refinement of the database interfacing, the implementation of new links to application software, the continuous updating of the references to interesting international sites.

Furthermore, we expect data and information from our partners in order to enrich I.W.E. increasing its roles of internal monitoring and world-wide I.N.D.U.C.E. knowledge dissemination.