

Complements to the architecture of the SIMES information system

*Deliverable number : D1.3
Nature:P*

Task WP3.1 : Specification of the common software bases

*Nom du rédacteur : Jean-Claude Derniame
Institut : Loria
[E-mail](mailto:Jean-Claude.derniame@inria.fr) : Jean-Claude.derniame@inria.fr*

Abstract:

This document is a complement of the deliverable D 1.1 , entitled “Spécification de l'Architecture du Système d'Information SIMES” dated on : *14th March 1999* .

Its objective is to take into account the evolution occurred since this date, in our way of thinking the architecture, as presented during the review in Brussels, on *29-30th September 1999*.

The main evolution is the abandon of the central idea to rely completely on CORBA as a central firmware, between the Web user interface and the databases, local or distant. In the new proposal, Corba will continue to be used to access to tools only available on distant sites, but to access to local data, we propose to use another firmware allowing to interface a Web server with any database compliant either to ODBC or SQL standards. We propose to choose for that the software called “Coldfusion” .

Keyword List:

Information System, Components integration, Meta-modelling, Corba bus, Internet/Web

1.	INTRODUCTION.....	6
2.	PRINCIPLES OF THE PREVIOUSLY PROPOSED ARCHITECTURE.....	6
2.1.	THE CLIENT –SERVER APPROACH	6
3.	THE THREE TIERS APPROACH.....	8
3.1.	REQUIREMENTS FOR THE SIMES ARCHITECTURE	8
3.2.	THE INITIAL PROPOSAL	9
3.3.	THE QUESTIONING	14
3.4.	THE NEWLY PROPOSED SOLUTION	14
3.5.	THE NEW BROWSING INTERFACE	19

Illustrations table

Figure 1: The two tiers client server model	6
Figure 2 : Classical extension mechanisms for the Web	7
Figure 3 : Functional architecture	10
Figure 4 :The three tiers architecture for PCIS2.....	11
Figure 5 : The Corba Architecture	12
Figure 6 : Corba usage in SIMES	13
Figure 7: Schema of the proposed solution.....	15
Figure 9 : Access form one site to another one.....	18
Figure 10 : Functionality's of the proposed interface.....	19
Figure 11 : One possible screen organization	20
Figure 12 : A simulation of the proposed user interface.....	21

1. Introduction

The WP3 objectives are to identify and build the system architecture and the software platform needed to support any SIMES environment (Task 3.1) as well as to establish the software components integration processes (task 3.2). The 18 first months have been devoted to tasks 3.1 and 3.2.

A big effort has been required to find and evaluate tools available at a reasonable price to be supported by the partners. Another part of the effort has been to realize the taking over for these tools by African partners.

The defined architecture aims at allowing the user to easily access to all services available on a SIMES network from any Internet connected machine, as well as preserving the autonomy of the sites providing data and/or processing algorithms.

More precisely, the architecture is thought in terms of components, some of them being designed to be generic and easily integrated on a platform.

Firstly, this document will recall the main principles of the architecture proposed in D 1.1, then we will come back on the notion of firmware and the three tiers architecture. Expected qualities of this firmware will be discussed and proposal evolution presented.

2. Principles of the previously proposed architecture

2.1. The client –server approach

A first principle as described in D 1.1, section1, is to rely on the Web technology to define our user interface, and to build specific browsing metaphors and specific browsing tools and specific information extractors.

Traditionally the Web architecture relies on a client-server model as sketched in figure 1.

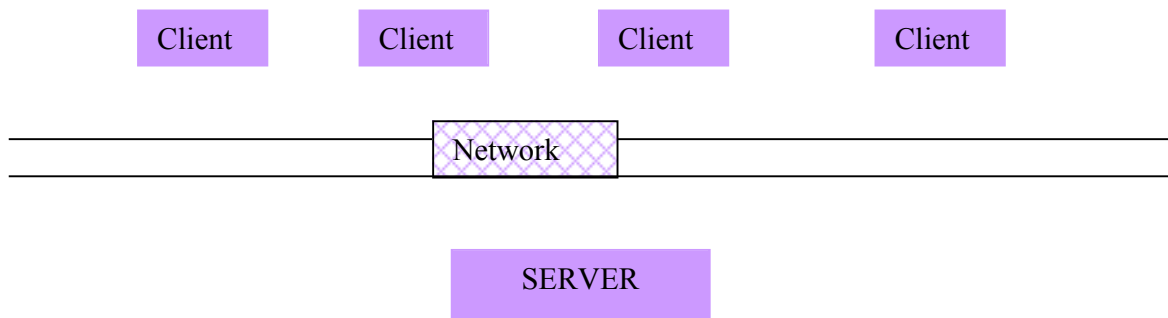


Figure 1: The two tiers client server model

Usually applications can be introduced on the Web through different programming techniques as shown in figure2.

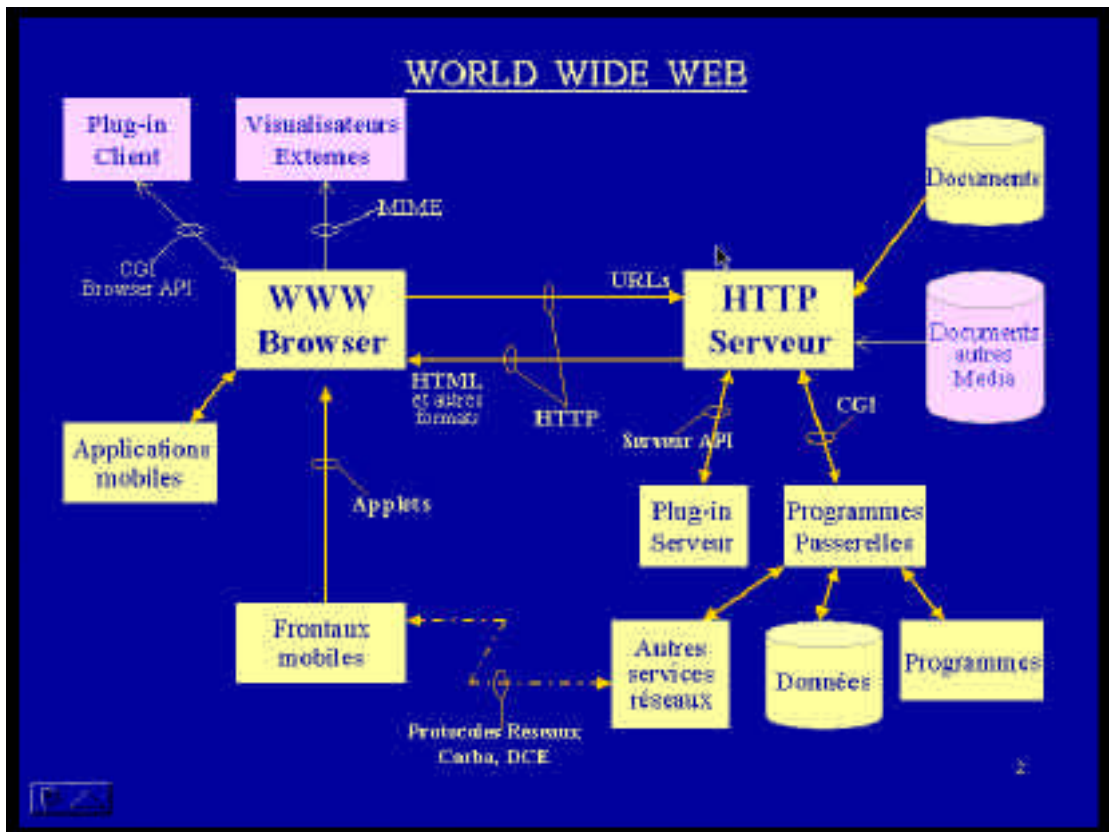


Figure 2 : Classical extension mechanisms for the Web .

On the server side, different solutions exist. The first one is to develop interfaces between the application and the server, using directly the services provided by the server through its API (Application Programming Interface), as with any tool having a public API its the more frequently used technique for public plug-ins. A more specific solution is to use the CGI approach to program the gateways between the server and applications, data, or databases. Other programming styles can be used such as Perl programs or Java serviettes.

On the client side, Java applets, JavaScript's, CGI can be used to develop programs able to be downloaded and executed on any client site, for instance for capturing forms, for interactive applications and for interfaces with other network facilities (for instance the client side of some firmware).

The main characteristics of all these technics, after their efficiency is the fact that they are reserved to programming specialists, and that the programming effort can be too large for frequently used services.

3. The three tiers approach

To reduce programming effort, other organizations are possible :

- *Extend an application to support Web usage and extend the application client to be integrated to the Web browser. It is the case for some database management systems, such as ORACLE for instance in its version 8. Obviously, the efficiency is remarkable for any access to Oracle databases, but its is a proprietary solution usable only with Oracle. Adopting it means that SIMES implementations will be limited to environment systems relying on Oracle. We don't think it would be wishable for the SIMES platform dissemination.
- *Use a firmware. It consists in grouping some services used by the most frequently used applications in an intermediate layer of software between the data and the clients. Several firmware can be found in the market, from RPC (Remote Procedure Call), to OLE by Microsoft, or CORBA ,which is a largely adopted standard to access to remote objects. Corba has been defined by the OMG (Object management Group) and has been described in the deliverables D1.1 and D 3.

The kernel of the architecture is the concept of **software bus** built in three tiers : an application layer, a client layer unifying the user interface and a third layer grouping all services supporting the applications.

Using a firmware leads to more open solutions, largely independent of the database system used in the user environment.

3.1. Requirements for the SIMES architecture

A Simes implementation is thought to be distributed in several servers on an internet like network, and accessible from anywhere.

On server sites data are either documents in different text editor formats (Word .doc, Framemaker ., latex .latex, simple texts .text, ASCII 8 bit .tx8,or interchange formats such as .rtf, .dif or .mif, or charts .xlc or spreadsheets .xls, or questionnaires in ad'hoc formats, or images in several formats such as .jpeg, .gif .ilm,.pict...)

Tools are distributed and but not necessary on each server sites and must be invocable from anywhere .

The architecture must rely as far as possible on accepted and emerging standards such as : HTML for describing all exchanges with users, ODBC and SQL for database interfaces, CORBA for distribution, etc.

As much as possible, the user interface must be simple and uniform.

For obvious economical reasons, in order to easier the dissemination of the future Simes platform, t is recommended to use free software as far as possible, and to buy Commercial of the Shelf Tools (Cots) in other cases, and only when unavoidable, to develop new functionality's with Java, Java scripts, C++ and/or CGI.

3.2. The initial proposal

As well for the requirements as for the architecture design, we gained profit from other ambitious projects such as PCIS2¹ and we have decided :

For the proposed SIMES system, the perspectives mainly concern information extraction, data integration, tool integration and heterogeneity management : data type conversion, meta-data definition, traders, etc..

Another direction of effort has been identified and worked on : integration of the user interface services. Here the objectives are to support the end-user navigation and information search either through traditional hypertexts and/or through geographical data, and/or also through both simultaneously. It is detailed in the deliverable D 1.1 section 1, and D 2, and it has not been changed since delivered except the development of a prototype, presented below.

The architecture design works, and this complement have been developed jointly by Inria Lorraine (JeanClaude Derniame), Dschang University (Georges Kouamou), IRD Orleans (Patricia Dzeakou) and ESI Bobo Dioulasso (Ali Kaba).

The first image processing tools have been identified by INRIA as well as statistical tools.

The results can be summed up in the following schemas.

The figure 3 shows a functional view of the architecture limited to the user interface and data access, independently of the distribution.

The figure 4 sketches the three tier architecture decided for PCIS2 and envisaged for SIMES.

The end user part is made of specific to the application software and particularly the browsing tools proposed in D1.1, actually prototyped and detailed in this complement. Vertical tools are for application tools administration (image processing and others), sites administration and servers administration, and some specific project tools such as extractors seen in the functional view for instance.

To take into account the distribution, the main idea of the initial proposal is to rely on CORBA (Common Request Broker Architecture), a proposal of the OMG (Object Management group) which becomes a de facto standard. It allows to access to any published object from any site equipped with Corba software. D1.2 and D3 have detailed this approach, here we mention only the figure 5 to recall the general architecture.

¹ Portable Common Tools environment : common project between US Navy and french MOD in charge of defining a platform to support secure software engineering environments on Internet. Loria has been responsible of the architecture specification . The system is currently under development by Sema group. The second version is operational since January 2 000.

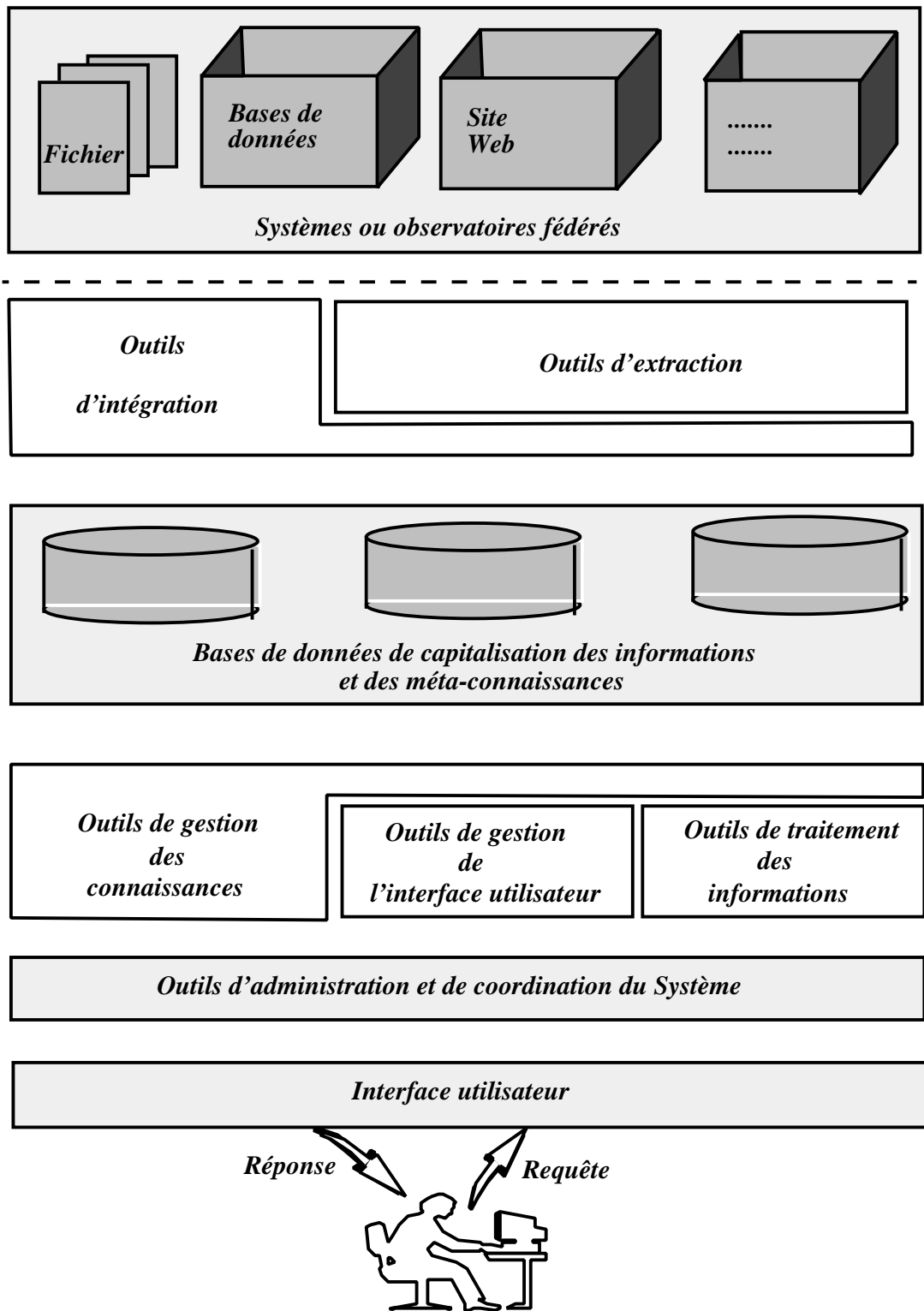


Figure 3 : Functional architecture

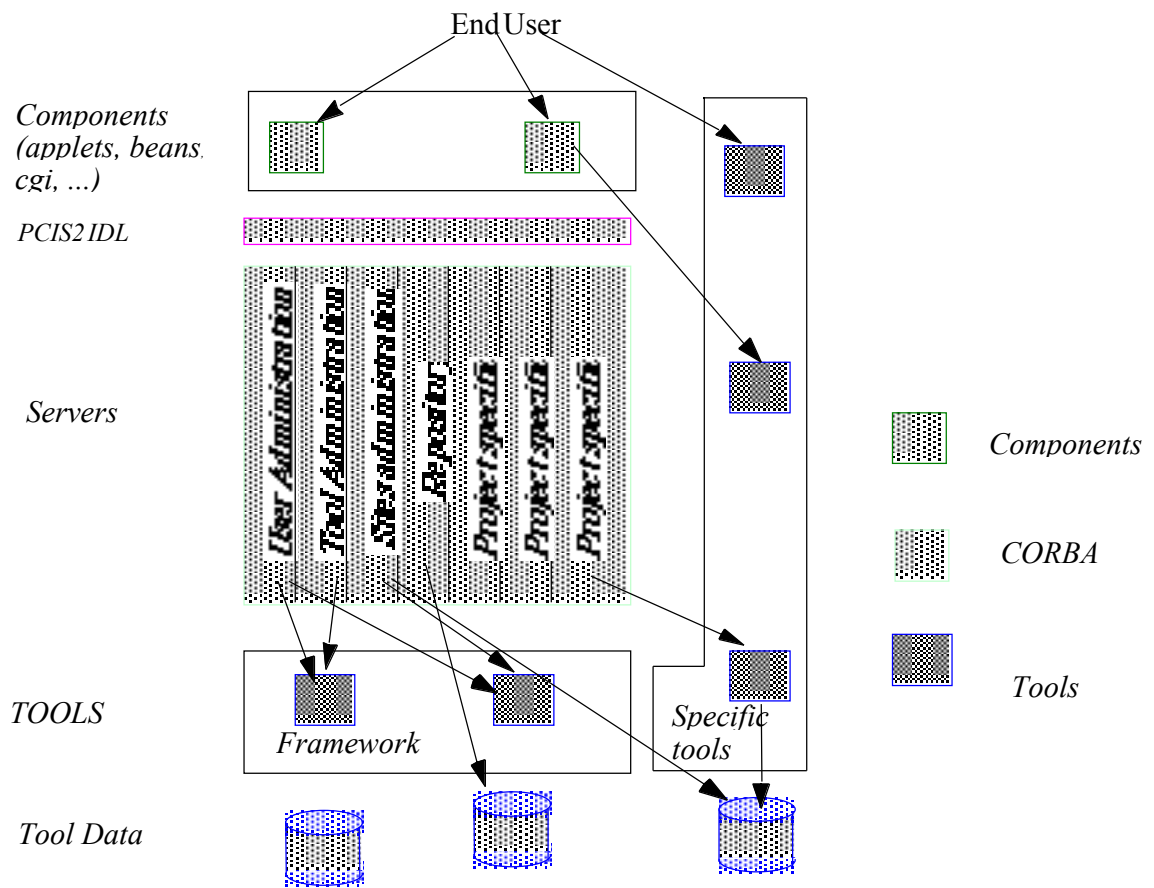


Figure 4 :The three tiers architecture for PCIS2

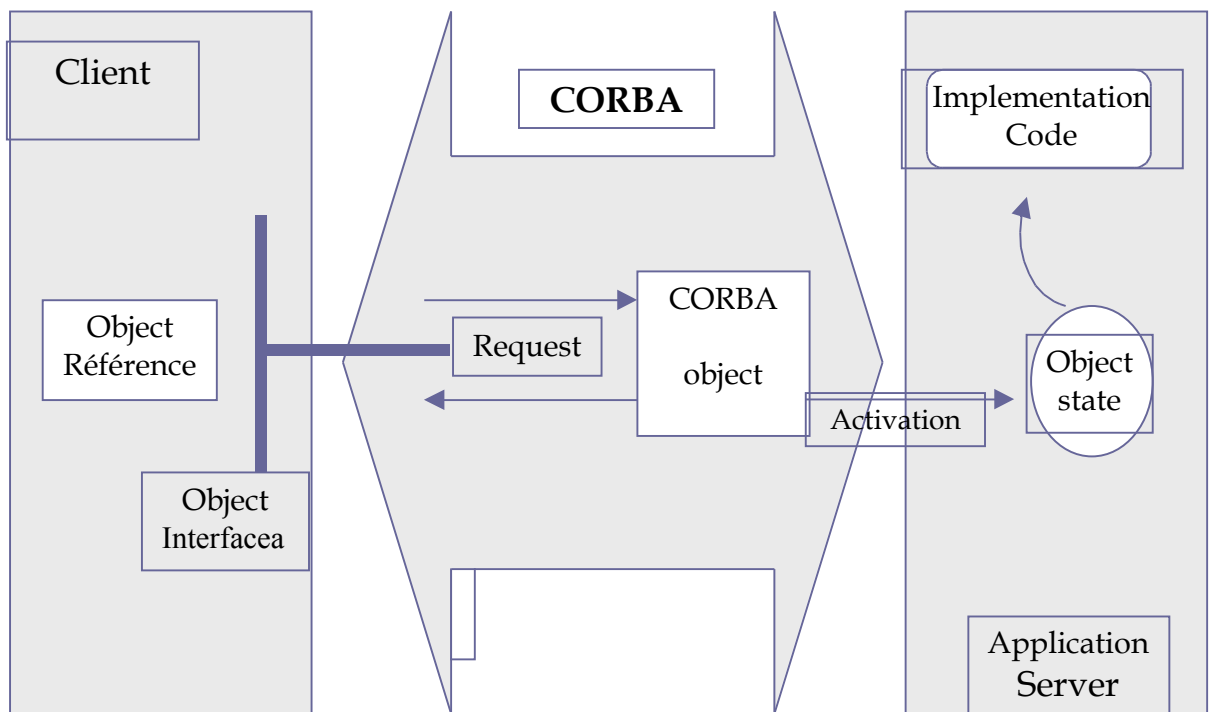


Figure 5 : The Corba Architecture

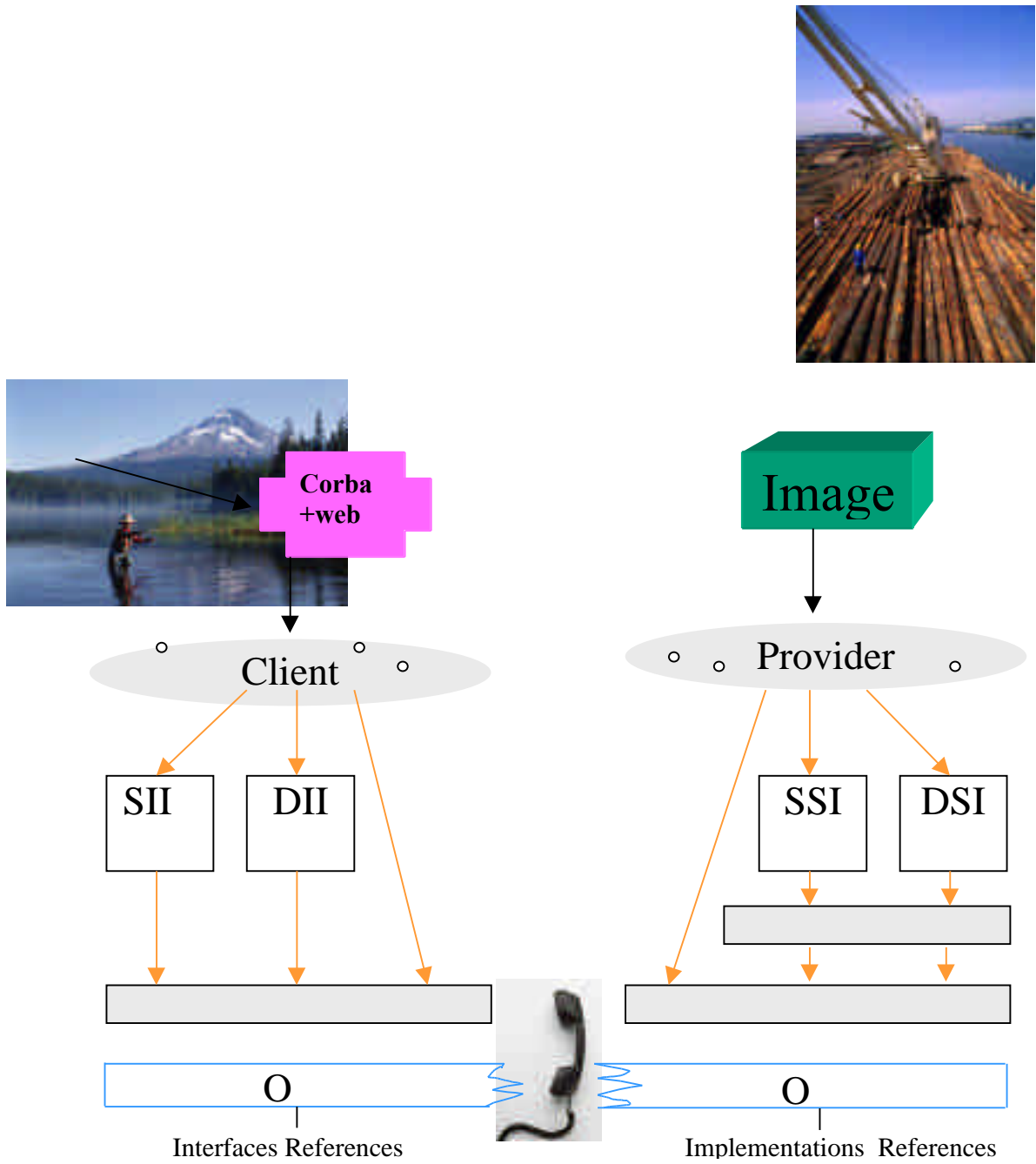


Figure 6 : Corba usage in SIMES

3.3. *The questioning*

Having the Web, do we really need to be all object oriented?

WWW is a killer application from years 1990's, offering convivial interface to access resources. WWW is quite an (object oriented) operating system with :

- Uniform naming : URLs
- Distributed persistent entities : the HTML pages in the Web servers file systems
- One meta protocol : HTTP between clients and servers
- * System extensibility : Applets, CGI, Servlets

But extensibility is also needed in terms of resources and there is an important need of a deep integration with the whole information system. In another hand the access to preexisting tools is not really supported except through important wrapping.

CORBA supports heterogeneous applications interoperability (mainly due to the OO approach and IDL) and allows effective existing systems Federation. So the marriage between WWW and CORBA has a lot of advantages: it is a **complete** infrastructure for distributed services grouping probably the most used standards in new software developments. Many solutions to integrate them exist and are available on the market and or in free versions. The only points is the heaviness when accesses are only local and the necessity to develop interfaces with databases.

The figure 6 is an illustration of one possible usage of CORBA, viewed mainly as a way to easier the access from a local site to some specific tools (specific to a platform for instance or too much resource consuming, or too expensive to be installed in each SIMES exploitation center, such as some image processing tools).

But we definitely believe that using only and inevitably CORBA will be too heavy for current usage, mainly for text and or structured data stored locally.

3.4. *The newly proposed solution*

It can be sketched by the following schema (Figure 7).

To access to local texts and data stored in various data bases, is proposed to use a tool interfacing the Web and databases. Thus interface must be uniform and able to exchange data with different styles of databases. The only way to reach that is to rely on tools offering interoperability with s databases compliant to some standards;

We have studied the market of these interfaces and the offer done by the Allaire Company with the tool ColdFusion appeared to be a good compromise between price and functionality's.

ColdFusion is a platform to develop interactive applications on WEB.

This software provides an interface between a set of WEB pages on a WEB server and a data base provided that it is SQL or ODBC compliant.

Cold Fusion follows a Client-Server approach : the main part is on the server side, on the client side are only forms to fulfill for querying.

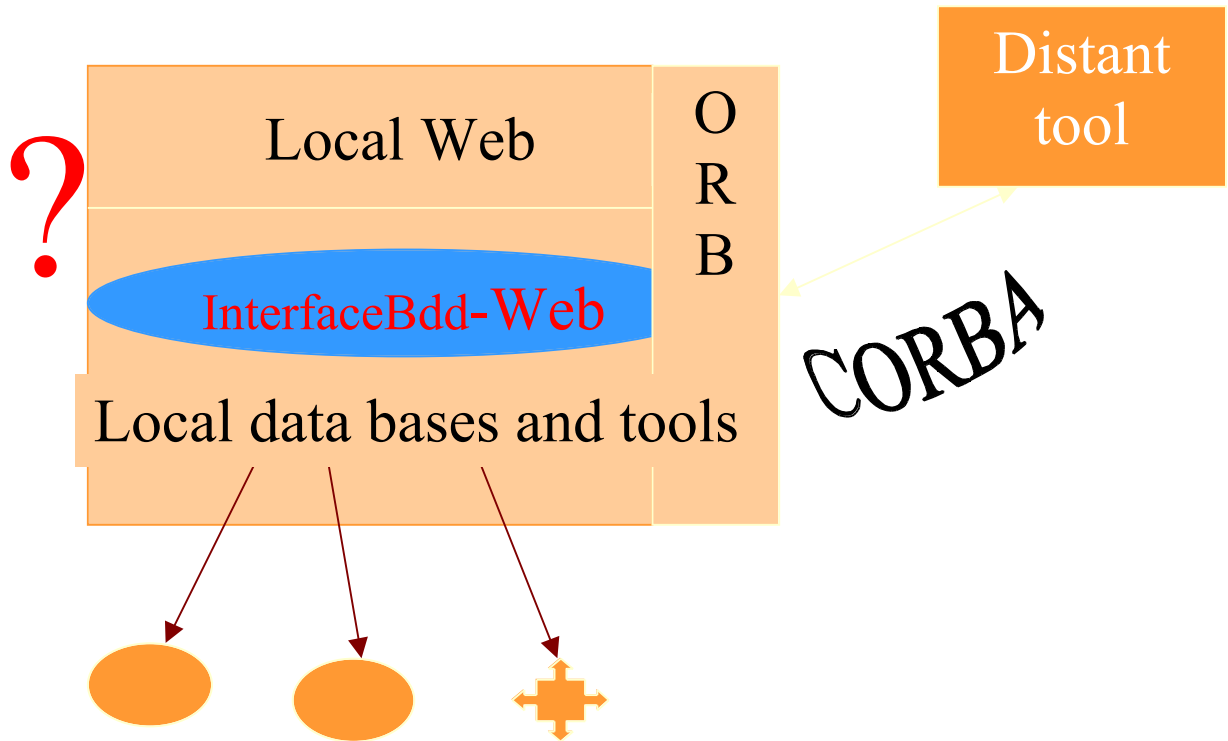


Figure 7: Schema of the proposed solution.

We have studied the market of these interfaces and the offer done by the Allaire Company with the tool ColdFusion appeared to be a good compromise between price and functionality's.

ColdFusion is a platform to develop interactive applications on WEB.

This software provides an interface between a set of WEB pages on a WEB server and a data base provided that it is SQL or ODBC compliant.

Cold Fusion follows a Client-Server approach : the main part is on the server side, on the client side are only forms to fulfill for querying.

It supports an easy construction of a set of HTML pages to fully manage a data base. It supports dynamic sites development through interactive forms and *JavaScript* usage to create animations.

A ColdFusion application is only made of Web pages, HTML enhanced to CFML (specific to **ColdFusion Markup Language**), interpreted by a '**ColdFusion server**'.

Using tags to do all operations makes it easy to use for someone already HTML aware. The main part of the tool are a *Query Builder* for **assisted** SQL_request creation, an interface for *mails management*, another for *files management* (upload, download, etc) an integrated *debugger*, integration with *FTP* and *HTTP* protocols.

The main advantages are

- * Easy and fast installation.(Wizard)
- * Data base Independence.
- * Server management through the browser.
- * Convivial interface : all commands are accessible through tool bars.
- * Real programming language.

The most important one is the fact its supports also the connection to systems using Corba C/C++, VBScript, JavaScript ? Doing so it is possible to implement the schema of the figure 7. The figure 8 is an illustration of this situation.

**Local or distant Access to a SIMES site
from a work station to query a data base or to
launch a tool on the same site**

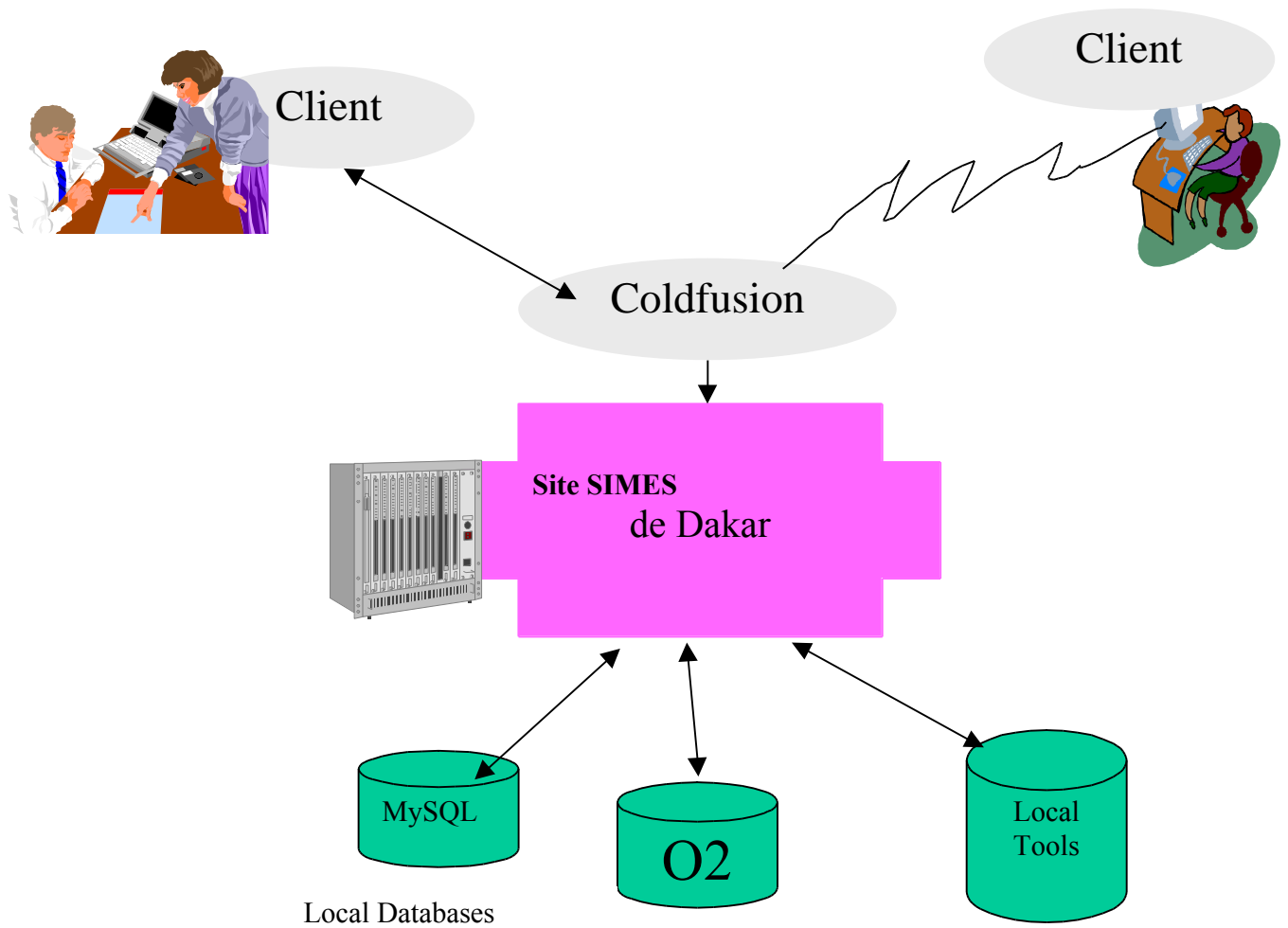
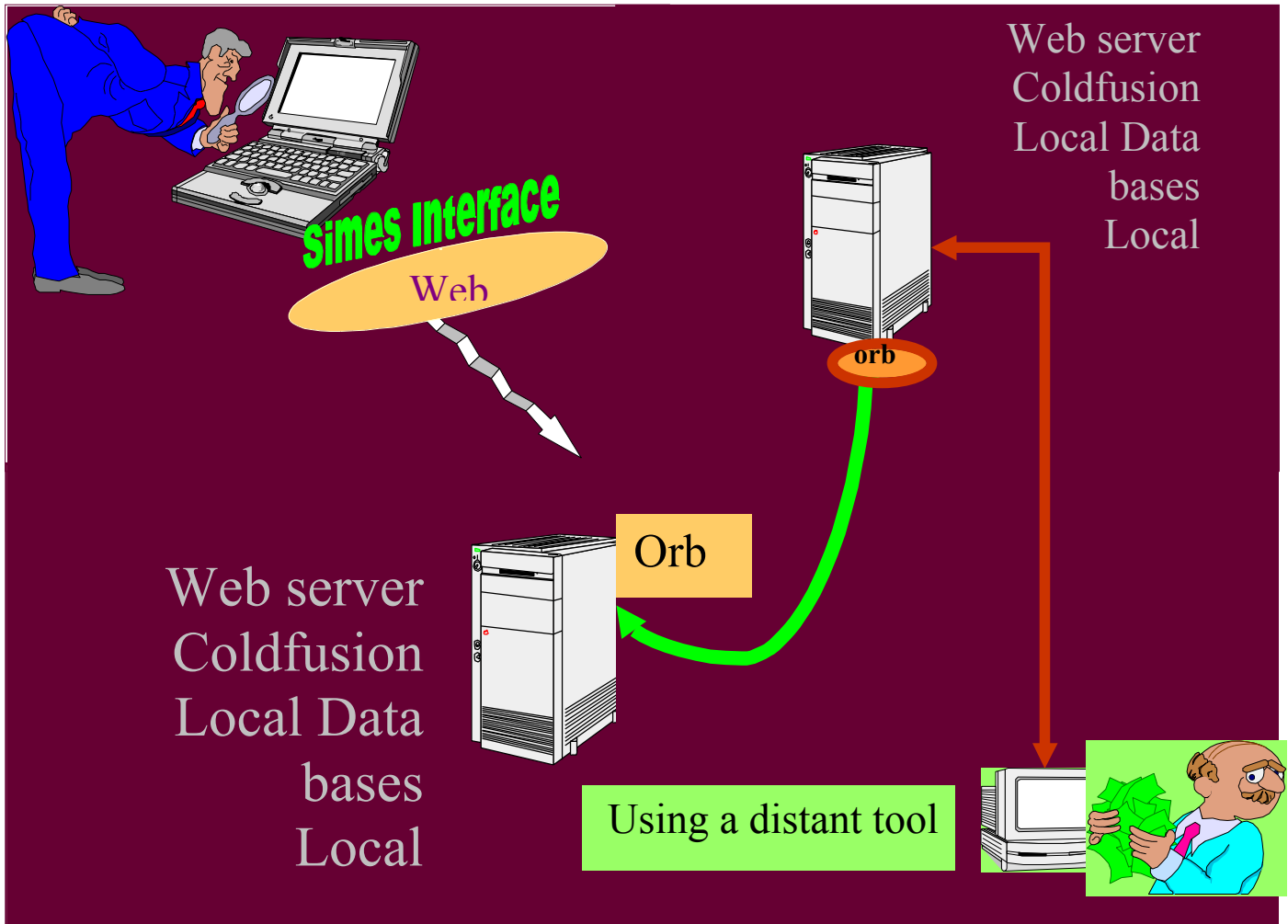


Figure 8 : Access to the Dakar SIMES site (from a station directly connected and used to query data bases or to launch tools based on the same site.

To access and use tools installed on another site, the interface ColdFusion –Corba will be taken to access the local ORB and through the CORBA Architecture, to arrive to the ORB of the distant site and to invoke the distant tool, as it can be seen in figure 9.

Figure 9 : Access form one site to another one



3.5. The new browsing interface

Here our objective has been to refine the proposals done in the deliverable D 1.1, in order to define:

- * an **interface model** which could be specific to environment observatories and could offer a better legibility and adaptability for non professional users..
- * **An interface generator** able to automate some tasks in the design of the navigation interface of one knowledge base in one environment observatory.

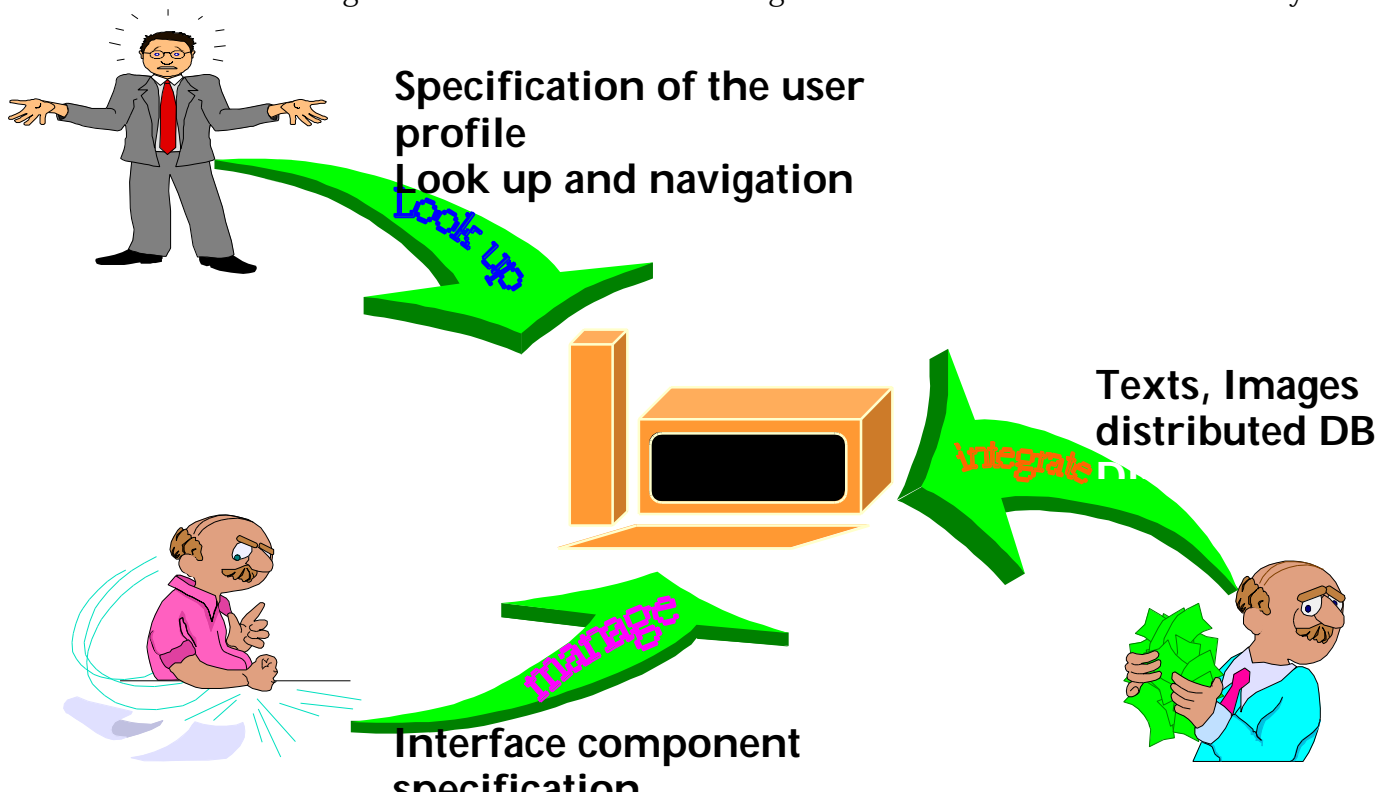


Figure 10 : Functionality's of the proposed interface

To search information in the diverse databases present in the SIMES sites network, different strategies and different ways of working can be used. We have defined a view as the way to consider, select, present data during a session of work: the view cannot be the same if the user wants to consider statistical data, historical data, photos collection, texts, geographical maps, etc...

This research being sometimes complex and long, we have introduced the notion of context of work: the context is a subset of all the documents present accessible on the network.

Contexts are persistent objects (they subsist after the session and can be retrieved for the following session), and they can be build form existing contexts by selecting (using the select mode of one or several successive views) components or by fusion of several contexts, etc. The figure 11 is one possible way to organize the screen as it is currently experimented by Patricia Dzeakou.

The left part of the screen is used to display the history of the last operations ordered in the current context. The upper buttons allow to select the desired view. The navigation image is devoted to the display of data (a map for instance) useful to build the next state of the context. The “navigation instruments“ bar allows to choose one tool specific to the selected view. The navigation space is a portion of the screen used by the selection tools. The following figure 12 shows a simulation of this interface usage. The user is supposed to have firstly selected the geographical view. In his context he has only the Niger map (if he would have several, the navigation space waould allow him to slect one). Choosing the first tool in the navigation instrument bar, he can select a geographical zone on the map which appears in the navigation image area. He chooses the “Batamani” region which appears in a red rectangle: doing so, the context evolves: it is now limited to documentst concerned only with this region.

Then he chooses the third button in the navigation instruments tool bar: *select photos*. The number of documents in the context is again decreasing, only photographs subsist in the context. Choosing the fifth button he can navigate through the list of photos by alphabetic order of authors, which is a specific to photos way of navigation in a context. Then two arrows appear on the screen allowing to enumerate the photos, which appear in the navigation space. Current work attempts to identify useful views and to precisely define the associated navigation tools.

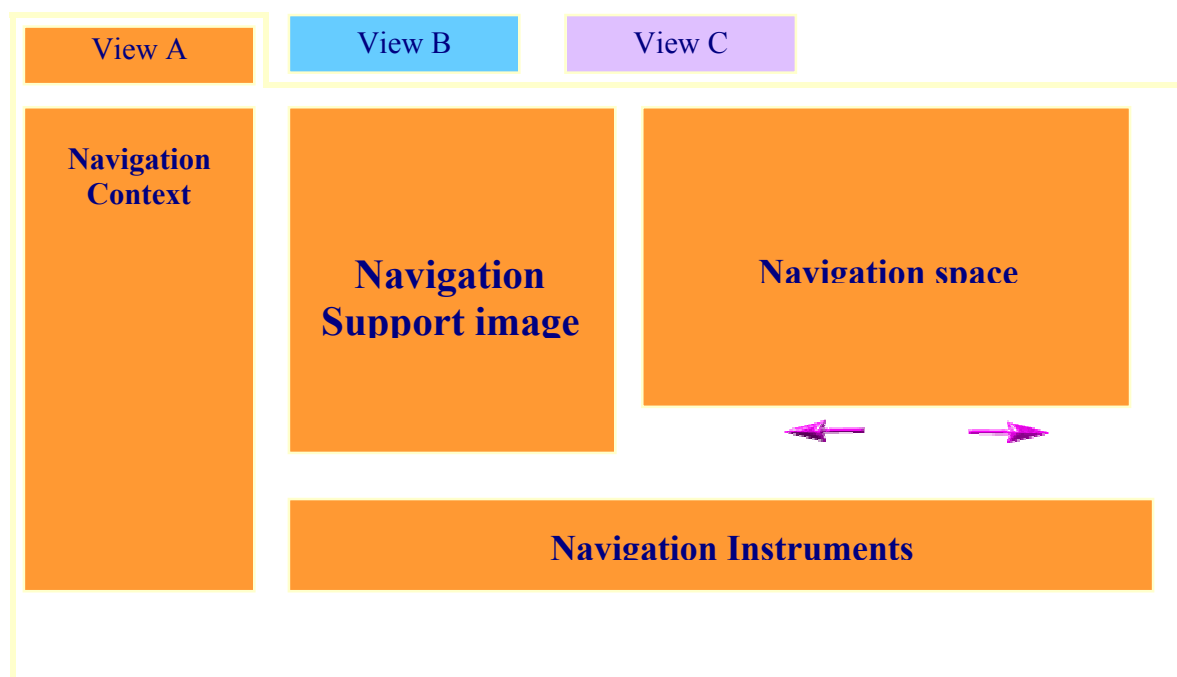


Figure 11 : One possible screen organization

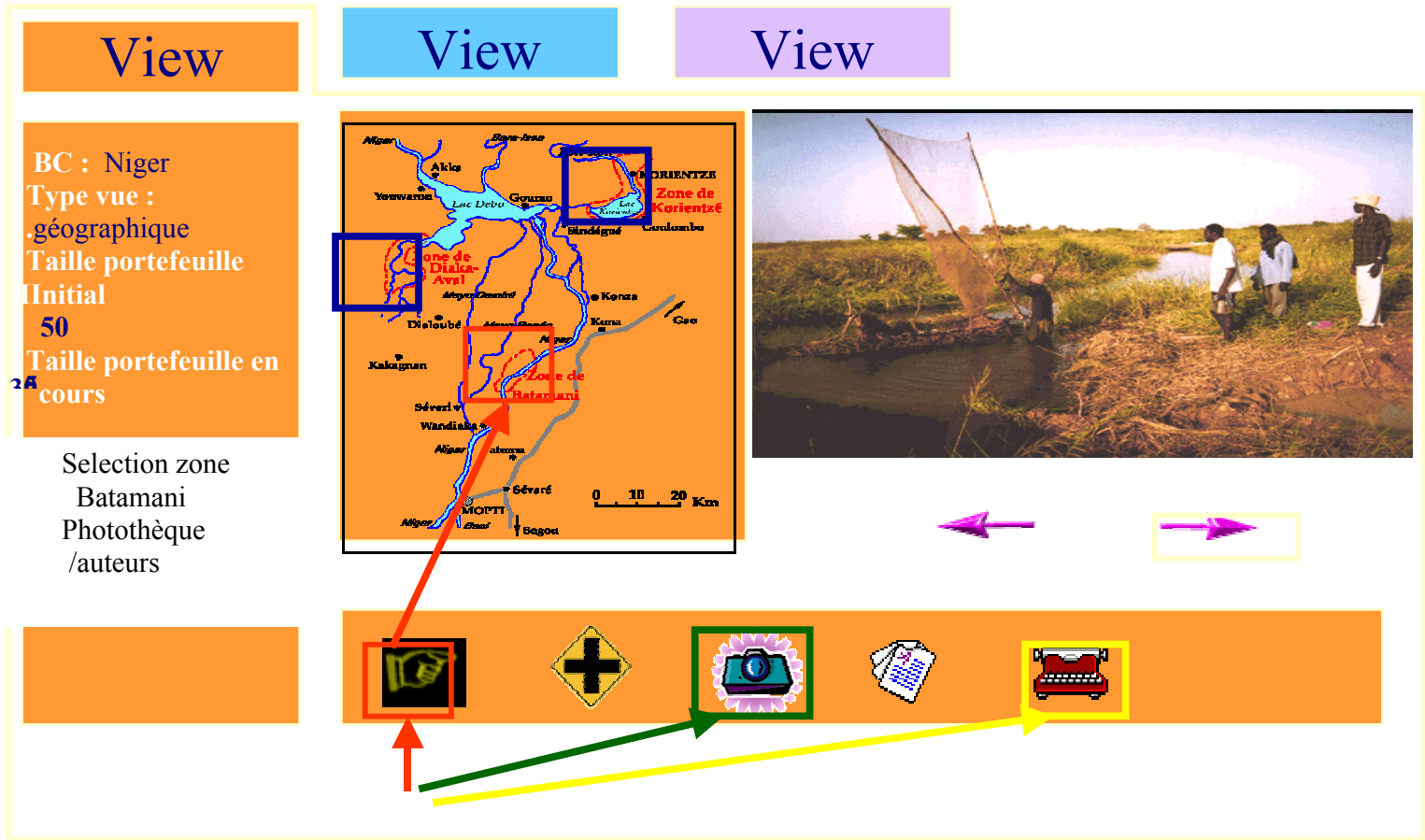


Figure 12 : A simulation of the proposed user interface