MASTER IN COMPUTER SCIENCE

A case base as a learning tool for the design of human computer interfaces

Michiels, Rudy

Award date:
1999

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX, NAMUR

INSTITUT D'INFORMATIQUE

RUE GRANDGAGNAGE, 21, B-5000 NAMUR (BELGIUM)

A case base as a learning tool for the design of Human Computer Interfaces

Rudy Michiels
Gaëtan Prévot

Mémoire présenté en vue de l'obtention du grade de

Maître en Informatique

Année Académique 1998 - 1999

Acknowledgements

Nos remerciements les plus sincères s'adressent :

Au Professeur François Bodart, notre promoteur, pour le temps qu'il nous a consacré, pour l'attention et l'intérêt qu'il a témoigné tout au long de ce mémoire, ainsi que pour les nombreux conseils qu'il nous a donnés.

To Professor Mark Guzdial, Georgia Institute of Technology, Atlanta, for his friendly reception, for his availability, for his contribution to our work and for accepting to be a member of our jury.

To Colleen Kehoe and all people of Georgia Tech.

A Jean-Marie Leheureux et l'équipe du projet Vesale pour leur disponibilité et leur apport lors des réunions du projet.

A Micheline Vandenbol pour le temps consacré à corriger notre anglais dans certaines parties de ce mémoire.

A tous nos amis de 3ème maîtrise pour ces années passées ensemble et les périodes de détente.

Enfin, nous tenons à remercier nos familles pour leur support et leur patience pendant les moments difficiles.

Sans toutes ces personnes, ce travail n'aurait pas été possible. Merci encore infiniment.

Résumé

Ce mémoire décrit le développement d'une base de cas comprenant des exemples (bons et mauvais) d'interfaces ainsi que de leurs critiques. Cette base de cas doit servir d'outil d'apprentissage pour la création d'interfaces et la compréhension des règles et critères qui s'y rapportent. Ce travail s'inscrit dans le cadre du projet Vesale de l'institut d'informatique et fait suite au stage de 5 mois passé au College of Computing du Georgia Institute of Technology (Atlanta, Etats-Unis).

La tâche principale attachée à cette base est la création d'une application permettant d'insérer une interface et ses critiques. Le défi est alors de créer une application simple à utiliser, performante du point de vue des possibilités offertes et des libertés laissées à l'utilisateur, tout en gardant à l'esprit le concept de « bootstrapping » cher au projet Vesale. Des réflexions concernant la consultation de cette base seront aussi abordées.

Abstract

This thesis describes the development of a case base containing examples (bad and good) of interfaces and their critiques. This case base must serve as a learning tool for the creation of interfaces and the comprehension of some rules and criteria that exist in this domain. This work is part of the Vesale Project of the Institut d'informatique and follows a five month internship period at the College of Computing of Georgia Institute of Technology (Atlanta, USA).

The main task linked to this base is the creation of an application that lets the users insert interfaces and their critiques. The challenge is then to create an application easy to use, powerful if we speak about the possibilities and the freedom let to the users but by keeping in mind the « bootstrapping » concept used in the Vesale Project. Some information concerning the consultation of the base will also be examined.

Table of contents

Introduction

The title of this thesis is "A case base as a learning tool for the design of Human Computer Interfaces" but a possible other title would be "Summary of the last year of our Master degree".

During the first semester of this academic year, we were welcomed by Professor Mark Guzdial in Georgia Institute of Technology in Atlanta (GA, USA) to make an internship. During this internship we were confronted with a new environment and new technologies. American mind is not the same that European one. We discovered a new way of working and living which were very profitable from a personal point of view. There we also learned a new way of teaching which is very interesting for the realization of our thesis. This approach was also a great inspiration for us.

We met and appreciated this teaching approach in several situations:

Firstly, we were present during some courses. We observed that the studies in the USA are not organized in the same way. So, there isn't a "second mastery degree year" like in Belgium. There is an obligatory list of courses and the very attractive point is that each student can organize himself his program of course. To help the student is this task, the professors give a same lesson several times in a week. This approach gives to the student a larger intellectual freedom.

Secondly, when we discussed with Professor Mark Guzdial and Colleen Kehoe. We confronted our point of view and they gave us advice and some bibliographic references to help us in our research. So, we could take advantage of their experience.

Last but not least, when we met other Georgia Tech Professors. Each week, some education professionals discuss about their research project, recent publication or about experiences, concerning teaching and apprenticeship.

Beyond this theoretical contribution, we acquired also some practice by realizing a web site aimed to teach some HCI concepts. This web site, realized with tools developed at Georgia Tech, has been commented and criticized by students testing it and also by Professor Mark Guzdial.

During the second semester, we had to take a part in the Vesale project in Namur and to develop a case base for HCI. We had a lot of meetings with Vesale team to see how this case base can be implemented and how to use the experience acquired during our internship. This thesis and the developed application is the summary of all these things.

In the first chapter, we will explain the context of our thesis, i.e. the subject, what we realized at Georgia Tech, and then we will also briefly present the Vesale project.

The second chapter will present the pedagogical approach we have followed. This chapter contains of course two main parts since it explains the pedagogical approach followed in the USA and also how we have integrated it in our Thesis. A small theoretical point is also presented.

The third chapter will detail one of the two main functions of our application: the case insertion in the case base. We will explain what we have developed and also why we chose to do that in that way.

The fourth chapter will be similar to the precedent since it will explain the second main function: the consultation of the case base. The only difference is that this point will be less concrete because the consultation has not yet been implemented.

The fifth chapter will be more concrete since it will present the implementation of the consultation. We will not provide code application but we will explain the different choices made.

The last chapter will criticize our developed interface and will demonstrate in this way the importance of "bootstrapping".

After a conclusion and bibliography, some appendixes with different documents will be proposed.

Chapter 1:
Thesis context

1.1. Subject of our thesis

The title of this thesis is "A case base as a learning tool for the design of Human Computer Interfaces". Our work is thus to create a case base that will help users to learn and to understand design principles in HCI. Cases are interfaces that users desire to present or to criticize. Two main tasks must be realized: the case insertion and the case consultation. These two tasks are essential for the concept of apprenticeship of the base. The first one will help the user to learn how to present an interface and how to criticize it. The second one will let him see and criticize interfaces made by others.

1.2. Work experience at Georgia Tech

Our internship was made at the Georgia Institute of Technology in Atlanta. At this place, we worked with Prof. Mark Guzdial and Colleen Kehoe on the creation of a case base that had the same goals that the one we would have to create for the Vesale project. The indications and recommendations of Prof. Mark Guzdial were very helpful for our discovery of the task and frequents reunions with other Education professionals gave us an idea of the different approaches we could follow to create a real learning tool. For example, the CoWeb environment (described later in this paper) was a perfect demonstration of what a learning tool had to offer to users. The possibility to discuss with students and to confront them with our work was also a good motivation for us.

We had also the great opportunity to participate to the "International Conference of the Learning Sciences 1998" [BRUCKMAN and AL 1998].

A detailed presentation of our developed CoWeb site is provided in appendix 2

1.3. Vesale Project

1.3.1. Introduction

"Teaching and research related to Human Computer Interface (HCI) have as aim the theories, the models, the methods and the tools necessary to all the steps of life's cycle of a HCI. Consequently, would it be useful to employ the specific knowledge to this field in the development of software tools of support to a computer-aided teaching? If moreover, this teaching is in the field itself, the opportunity is given to use its content in self-illustration, knowledge and techniques specific to the field of HCI and thus benefiting the learning. This principle of "bootstrapping" is fundamental to the VESALE project (Visual user interface design Education Supported by a computer-Aided Learning Environment), a multimedia environment of support to the teaching of HCI.

This teaching's support concerns four situations of operation:

- The apprenticeship as part of the live teaching : teacher and students are face to face
- The complementary apprenticeship to the live teaching which essentially stands the evaluation of knowledge and review of syllabus contents
- The indirect teaching or distant learning, i.e. the self-apprenticeship enriched by the interactions with the teacher
- The co-operation to the teaching's enrichments. In particularly, the enrichments for reasoned cases and for illustrations of multimedia technologies.

1.3.1.1 Global presentatio n of the Vesale project modules

a) Logical Architecture

Each module/base of the Vesale architecture are related. The figure bellow (Fig XXX) describes those relationships.

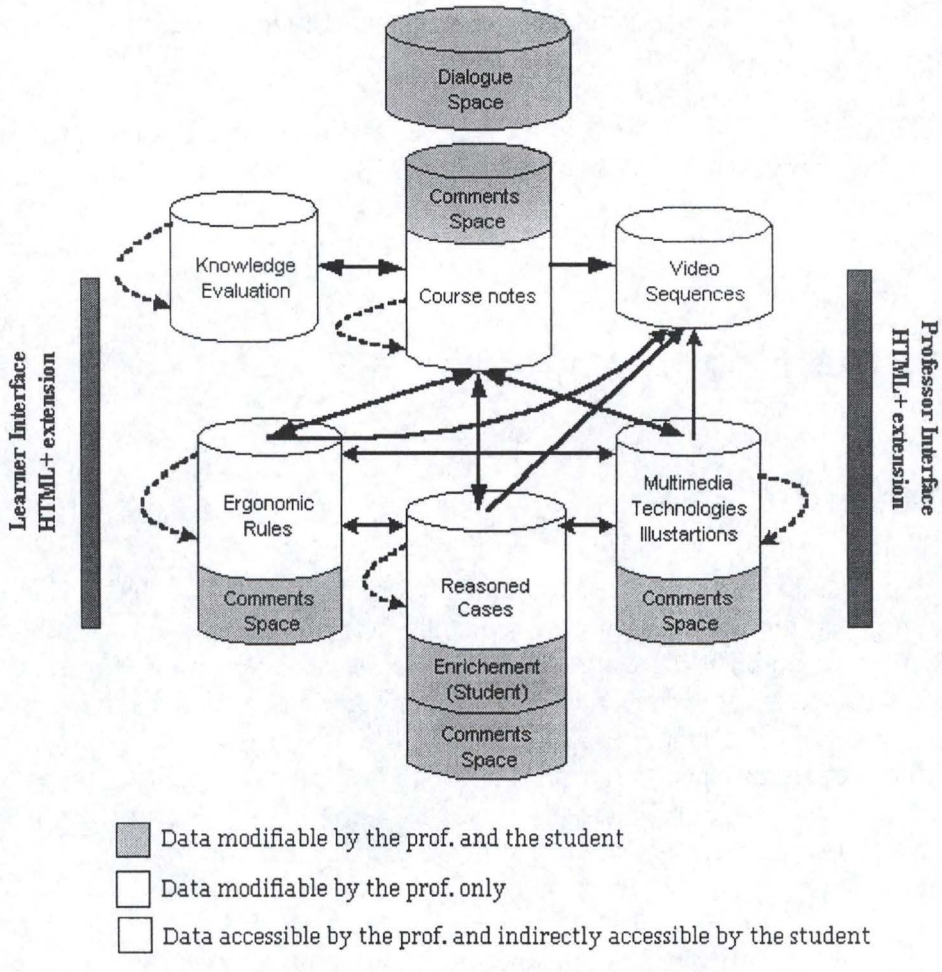


Figure XXX. The relations between the different modules of the Vesale architecture

b) The modules

- *The course notes module*
  The course notes module contains the notes of the Human Computer Interface (HCI) course presented in the form of hypermedia software.

- *The reasoned cases module*
  The reasoned case module contains critical examples of interface including some examples of design. This base can be enriched by cases suggested by the students or other interested persons.

- *The multimedia technologies illustration module*
  The multimedia technologies illustration module is related to the interaction objects and technologies.

- *The ergonomic rules module*
  The ergonomic rules module is intended to illustrate the use of the ergonomic criteria and rules in order to build useful and usable interfaces. It will consist of a subset of ergonomic rules of design of HCI.

- *The knowledge evaluation module*
  The knowledge evaluation module contains exercises, questions and plans of evaluation of knowledge acquisition by the learner. It must allow a summative and formative evaluation.

- *The video sequences module*
  The video sequences module contains video sequences of introduction of certain parts of the HCI course or sequences to illustrate the handling of certain interactive objects.

- *The dialogue space*
  The dialogue space support the dialogue between the professor and the student on general questions.

- *The comments space*
  The comment space is associated with each particular base. This space makes it possible for the students to express comments and eventually for the professor to react to these comments." [BODART and AL 1999]

The complete specification document can be found in appendix 1.

1.3.2. Our work in the Vesale Project

We must construct the case base that will appear in the reasoned case module. This base will contain good examples of quality interface conception and bad examples of interface conception that does not respect the design criteria of the HCI course.

Chapter 2:
Pedagogical approach

2.1. Introduction

The purpose of our thesis is to create an application that will help students to realize good interfaces. The main difficulty is to really understand the task: what are the students waiting for? What pedagogical approach must we follow to reach these objectives? The internship made in the USA at Georgia Institute of Technology was very helpful for us. We had the opportunity to discuss with students and professors about the philosophy used in the USA. The discussions with Professor Mark Guzdial were particularly useful and interesting. His recommendations helped us to improve our design of the site. The discovery of the CoWeb environment has also been essential for the comprehension of our task. With the CoWeb, we were able to test different organizations and evolutions of our site without the problems linked to the implementation.

The application that we have realized for the Vesale Project is thus a combination of the pedagogical approach we learned from the USA and the learning practice of design that we acquired during our studies in Namur. The resulted application is a prototype that we propose to the students. It has not been realized with the help or recommendations of pedagogues. We have just used our feelings and observations of what a student could hope from this kind of work. The future use of this prototype will show us the modifications that the work needs. It is now the role of pedagogues to analyze the results and to see if the proposed application corresponds to these criteria.

This chapter introduces the pedagogical approach that we have used to realize the application. The first section will discuss about the pedagogical approach followed during the internship. It will explain the web site we have realized at Georgia Tech, the principles of the CoWeb environment and a set of principles helping to evaluate a site called « ABLE principles ». The second section will discuss about the pedagogical approach followed during the design of our application used in Vesale project. It will explain the main guidelines used in our application and will show which principles of ABLE reached by our work. The last section will describe the 6 learning paradigms of Leclercq - Denis and analyze if these paradigms can help us to validate our application.

2.2. Pedagogical approach followed during the internship at Georgia Tech

2.2.1. Work performed during the internship

Firstly, we will explain the work we realized at Georgia Tech. The purpose of the work was to construct a web site that taught some rules of Human Computer Interfaces domain and that contained a case base. A user had to be able to consult a case, to criticize a case, and to add his interface. You will find a more detailed presentation of our web site in annex.

Our web site[1] is divided into three parts:

- The first part contains different tutorials :

  1. A tutorial concerning the design criteria studied in our HCI course (compatibility, consistency, management of errors,...). Each criterion had a definition, some examples of good or bad utilization and an explanation for the different sub-criteria that exist.

  2. A tutorial concerning the AIO. Each generic AIO was described and a representation of each specific AIO was shown.

  3. A tutorial concerning the ergonomic criteria: field description, user description (type, task experiment,...) and environment description. Each part was defined and explained.

  4. A tutorial concerning design criteria: these criteria were used in the different classrooms in the USA to classify the cases. They concerned the way to identify the case how it was created, which programs were used,... Each part was defined and explained.

- The second part contains the case base: a list of the cases was insert in this page. A simple click on the name of a case gave access to it. A search engine helped also the user to find a case in the base. Design criteria and ergonomics criteria were used to classify cases in the base.

- The third part contains another case base: this base was realized to correspond on the need of students in Georgia Institute of Technology. The cases of this base are made by students and analyzed by Colleen Kehoe[2]. They are classified in the case base by using the design criteria commonly used in the USA. As in the other case base, a list of the cases was presented and a simple click on the name of one of the cases gave access to it. A search engine was also created to help the user.

As everyone can see, the consultation of the case is not a problem. What a user has to do is just to click on different hypertext links. To add a critique or an interface in the base, it is also really easy to do. In order to explain these operations, we now have to introduce the environment that we used: the CoWeb.

2.2.2. Presentation of the CoWeb

We will now present the creation tool of web pages that we used at Georgia Tech to create our site: it is called the CoWeb. This tool has been generated by Professor Mark

[1] Our site is at the following URL : http://pbl.cc.gatech.edu/interfaces/1.html

[2] Colleen Kehoe was the teaching assistant for CS4753 – Human factors in Software Design at Georgia Tech

Guzdial and allows a real interaction between people who participate to the creation and the evolution of the web site. We present in the following section advantages and disadvantages of the CoWeb. These information come from the article of Professor Mark Guzdial titled "Collaborative Web site to Support Open Authoring Community on the Web" [GUZDIAL 98].

2.2.2.1 Supporting Authoring on the Web

"The Web has potential educational benefits. Creating Web site can be a motivator for students because of the worldwide audience that a web page can reach. Actually, web pages can offer interactions: combining the wide audience and interaction, it can enable collaboration that can support complex and motivating student work and the development of improved, shared conceptualization. But the problems linked with that are considerable. In fact, it's impossible to describe to someone how to create a web page without using words as « servers », « HTML editor » or « FTP ». While the potential of communication thanks to the web is real, the HTML language (and other languages used in web pages as JAVA,...) prevents the least technical users from accessing the web's potential. The idea behind a tool like the CoWeb is to facilitate the creation of web pages, to provide to the user the possibility to create real web pages without the problems of technical languages."

2.2.2.2 The CoWeb

a) What's the CoWeb?

"The CoWeb is an open authoring environment. In an open authoring environment, everything that is created with the authoring tool can be inspected (to learn how it works) and any user can create new things (including copying and modifying old things). That's exactly the purpose of the CoWeb. The basic idea is that any page in a CoWeb web site is directly editable by any reader of that page and that any editor can create pages in the web site."

"A "CoWeb site" looks like a fairly traditional web site. A CoWeb page can have essentially any kind of media or formatting as any other Web page can. A key feature of a CoWeb page, however, is the link in the upper left corner « Edit this Page ». When the reader of the page chooses this link, he gets a new page. The text that appears is the text of the page but the reader can change it, to add or to remove something from the original text. Then, he just has to click on the « Save » button and the page is automatically updated to reflect the changed text. Other readers will see the new page."

b) Advantages of the CoWeb

Now that we know how the CoWeb works, we can examine its advantages:

- Editing a page is really easy with the CoWeb. The user never has to deal with creating files or making these files accessible by a Web server. Beside that, editing

rules of the CoWeb are obvious. In fact, CoWeb pages can be written using the same editing conventions used in E-mail: text can be entered as paragraphs (with or without return key at the end of line), and a blank line separates paragraphs. To add a new page, user just has to put the name of the page between asterisks (the same way is used to add a hypertext link. In this case, the name of the page between asterisks must exist or a new CoWeb page is created).

Images can also be incorporated into a CoWeb page. The image must be located on a web server. User enters then the image URL between asterisks (for example: image.jpg).

If the user knows any HTML or JAVA, it can be intermixed with CoWeb-style text without any problem.

- The CoWeb provides supports that work across the CoWeb pages.

A « Recent Changes » page is available for every CoWeb. It lists each page by title in the CoWeb by the day on which it was changed in reversal chronological order. « Recent Changes » serves as an automatic table of contents for the CoWeb.

The entire CoWeb is searchable from any page in the CoWeb site. This enables users to find what others have done, even if long ago and far down the « Recent Changes » list.

A more esoteric feature « Display All Pages » which displays a page and the entire CoWeb pages that it references, all as a single Web page. This feature can be very useful in collaborative writing activities.

- The CoWeb meets the requirements of an open authoring environment :

Users can take advantage of the key aspects of the Web: links to anyone's work (collaboration), pages combining multiple media that are readable by anyone (powerful communications), and implicit interactivity since anyone can edit any page.

Initial use of the CoWeb to create pages requires no more knowledge than what can be expected by any user of a web browser. As users gain knowledge in HTML, they can incorporate those features integrated with their previous work. There are no discontinuities in the learning curve, though there are limitations discussed at the end of this paper.

All these qualities of the CoWeb are used in our site. Later in this chapter, we will explain where they are used and why they are important for the apprenticeship of our site.

c) Disadvantages of the CoWeb

Of course, the CoWeb has some disadvantages. These are essentially linked to the security.

The CoWeb offers little in the way of traditional security. Rather it relies merely on the power of social conventions. People do not normally destroy one another's contributions. People generally identify themselves with their contributions. Further, users make sure that ideas are not lost, if someone inadvertently (or otherwise) deletes important text, « housekeepers » make sure that the text is repaired. In this way, even protection becomes a collaborative task.

In reality, the CoWeb does have some security mechanisms. For example, the tool does make backups of every page when new text is saved, so that it is possible to recover seemingly lost text. The backup facilities have rarely been used, however. In general, people behave well and want to make real contributions.

2.2.2.3 Utility of the CoWeb for our work

As we said before, the two purposes of our two bases were:

1. To consult the different cases
2. To let the users add their own cases and critiques.

The first purpose was really easy to reach. As it was a web site (embedded in CoWeb pages), the user just has to use the hypertext links to navigate into the site and find the different cases.

But what about the second purpose? In a traditional web page, it is not really simple to add a critique concerning an existent case that already exists or to add a page containing a new case. To do that, user has to know certain things like HTML language, use of a server, how to transfer files from a place to another,... But in a CoWeb page, all these problems disappear! You don't need to learn the HTML language or to transfer files. All you have to do is read the page of recommendations called « formatting rules » (you can find this page by following the link « formatting rules » in the welcome page of our site) and you are ready to create new CoWeb pages that can contain structured text, picture, and so on...

That's the main reason of our choice in favor of the CoWeb: the work of a person who wants to insert a case is really simplified. Moreover, by using the CoWeb environment, we also recuperated some useful tools:

- The « recent changes » page that offers a view of the changes in the base day after day. It's a really useful function in a case base. For example, a professor who wanted to see the new cases that his students have put in the base doesn't have to search in the entire base. He just has to go to this « recent changes page » to see the new added cases.

- The « display all pages » page offers a view of the different pages that reference the one you are reading. It also a very useful function in a case base because when a user want to see all the pages that have a connection

with the one he is reading, he don't have to follow all the links, he can have a summary by using this function.

2.2.3. Pedagogical recommendations

To realize the web site, we discussed with Professor Mark Guzdial to find the best way to construct the site. From these discussions we were able to consider some recommendations that our pages had to follow. In this section, we will describe these recommendations.

2.2.3.1 A student site

Our first idea was to create a site, which could be used by novices and professionals. So we divided it into two parts: one with many explanations and examples for the novice and the other containing no rules or descriptions of the AIO (or just a summary) for the professional.

But Professor Mark Guzdial was above all interested by a web site he could used in his classroom. So the professional part was abandoned.

2.2.3.2 A « non traditional » site

Another recommendation was that the site had to contain a lot of links between the different pages. The idea was that the student didn't have to follow a traditional lesson with one chapter concerning the theory, a second one concerning examples and a third one concerning exercises. All the parts of a common lesson had to be mixed together, each part giving access to elements of the other parts. The better way to obtain this sequence was to introduce a lot of hypertext links in the pages. The principle of the CoWeb lets also the pages renewed themselves: if a student added a case in the base, he could also change the theory pages by adding a link from a theoretical point to his example (if this one illustrated a theoretical point).

2.2.3.3 The students have the choice

It was also decided to let student choose the kind of information he prefers. The idea was to give the students different ways to obtain needed information. For example, to study theory, student had to be able to select level of details that he wants. We had also to propose different ways to present or to search a case in the base.

2.2.4. Apprenticeship-Based Learning Environments (ABLE)

During the design of our web site and during our meeting with Professor Mark Guzdial, we have obtained a lot of information about apprenticeship. The different properties that software designed for apprenticeship learning requires will be explained according to the work of Professor Mark Guzdial and Colleen Kehoe about the ABLE structure. ABLE is the abbreviation of Apprenticeship-Based Learning Environments.

The two following points are extracted from "Apprenticeship-Based Learning Environments: A principled approach to providing software-realized scaffolding as Hypermedia" [GUZDIAL and AL 98]

2.2.4.1 Helping students become skilled practitioners

"Becoming a skilled practitioner involves gaining deep conceptual knowledge as well as process knowledge in order to solve problems. Conceptual knowledge is the information that one most often thinks about as propositions about a field: definition, patterns,... Process knowledge is the information about how one goes about activities: heuristics for how to decide the next step, how to use the tools and techniques of a domain, approaches that work well given kinds of problems. Skilled practitioners have an ability to undertake a complex process in the domain of their expertise and to change their process to suit a given problem in the domain. But it's not the same for the students. So, in general, there is a need for students to learn about process as well as concepts. "

"The most common way across the ages in which students have learned process on the way to becoming skilled practitioners is through apprenticeship. Apprenticeship has students learn process through active participation in the task. Student participation may be very limited at first while students gain an understanding of the process through observation and making small contributions but the involvement develops into full participation and eventually task ownership."

"Apprenticeship learning has several characteristics, which makes it effective:

- *Scaffolding*

  Students are supported to both (a) be successful in their process and (b) learn the process. Scaffolding is faded over time as the student gains in competency and can take on more of the process without support. Scaffolding has three component to it:
  - Communicating Process: the process is shown or demonstrated to the student, often in a structured and simplified form
  - Coaching: Support is provided in response to student activity – often in response to student failure
  - Eliciting Articulation: Encouraging the student to talk about and reflect on the process in an explicit way

- *Authenticity*

  Because all the process learning is in terms of active participation in the task, the authenticity of the learning and the relationship between daily activity and learning goals is more obvious than it is in a traditional classroom. Authenticity can help to improve transfer of the learned knowledge and student motivation.

- *Sequenced tasks*

  While not all traditional apprenticeships offer well sequenced tasks, effective apprenticeship environments provides students with the right task to undertake when the right kind of process (and concept) learning is needed. Too complex tasks can reduce student motivation, while too simple task would not encourage learning.

- *Collaboration across a range of abilities*

  Such an environment provided ample opportunity for just in time scaffolding as professionals (or senior students) were available to more novice students to answer questions and provide demonstrations (coaching). In addition to supporting novice students, the opportunity to work as teacher helps more senior students articulate and refine their learning, as in a reciprocal teaching setting.

However, creating an apprenticeship learning experience is hard in traditional classrooms, Thus, it is not surprising that education researchers have looked to technology as a potential solution of the problem of gaining the advantages of an apprenticeship model in traditional classroom settings."

2.2.4.2 A principled approach to providing software-realized scaffolding as Hypermedia

"ABLE structures have been designed around a set of eight principles that are based on findings about how students learn and how to support student learning. These principles define the kind of information to be included in a case, how it should be structured, and what the student's interface to the case should include. We will now describe these principles.

a) Principle 1: Provide adaptable scaffolding through levels of detail

Scaffolding, in a sense, is about « how much of the answer you give away ». There is no single answer in a design context, but the issue of the design learning context is how much support to provide to a student and how much to ask to the student to do himself. Scaffolding is inherently about the tradeoff in task responsibility between the master and the apprentice. A skilled teacher provides enough to support student's success without impinging on the student's learning, that is, enough to remain in the « Zone of Proximal Development », which is that range of activity where a student is challenged (but not overwhelmed) and can succeed with help.

There have been several examples of software-realized scaffolding in educational technology, but few have actually supported adaptive or adaptable scaffolding. Emile did provide an adaptable form of software-realized scaffolding for constructing physics simulations. Students could turn on or off various scaffolds such as guides for the students process, enforced prompts for articulation (such as predictions about the outcomes of physics simulation experiments), and supports for entering in program code. One of the observations from the Emile experience was that students who attempted their task with the least amount of scaffolding seemed to learn the most. This may be just a correlation, not a causal relationship. Our conjecture is that students should be provided with as little support as possible, but with the option of adapting their scaffolding for more support.

Of course, it is difficult for the software to sense how much a student does or does not know. Computer scientists refer to software that senses the users needs and automatically changes to meet those needs as being *adaptive*. But the option that one could prefer is having the software being *adaptable* (having the student change how the system supports his activity).

b) Principle 2: Strategy information is available but not immediate

The second principle of ABLE is that strategy information should not be immediately accessible. If requested (e.g. through a click), strategy information should be available to help students understand why a step was undertaken. But by initially hiding the strategy information, students are given the opportunity to think of their own strategy for the given step.

c) Principle 3: Outcome information is available but not immediate

Some projects have shown the important benefits of prediction in science learning. By making a prediction, students articulate their conceptualizations, which may not be accurate with respect to the simulation and the view of scientists.

The outcome step, that is, what the student should see if the step were to be executed, is similar to the result of a simulation. An expert practitioner should be able to predict the outcome of a step. Students need to be provided the opportunity to make that prediction. The third principle of ABLE is that the outcome of a step should not be immediately available. By pushing it one click away, the user is provided the opportunity to predict the outcome and compare the prediction to the recorded one.

d) Principle 4: Where possible, suggest potential problems and solutions

The fourth principle of ABLE is to recognize the difficulty of identifying problems and solutions in all domains, but wherever possible, common problems and solutions should be identified.

e) Principle 5: Use multiple representation

There is an enormous literature on how to create a good representation of information, how we understand representation of information, and how to get computers to generate good representations of information. In fact, the benefits of multiple representations for information are obvious.

The fifth principle of ABLE is to use multiple, linked representations to help describe the project. Through use of multiple representations, the student can be shown different aspects of the case. By linking the representations, the student can be led to see connections in the different aspects of the same case.

f) Principle 6: Design use in practice

The sixth principle of ABLE is to make feasible the use of the case material while the student is undertaking the project. By integrating the ABLE with the task context, we create a better opportunity for transferable learning.

g) Principle 7: Support sense of learning

The seventh principle of ABLE is to support the sense of community in which cases are placed. The goal is to integrate the students into a community of practitioners, and provide both more senior and more junior students the benefits of an apprenticeship community."

After the test of an instance of ABLE called STABLE (for Smalltalk ABLE), Professor Mark Guzdial and Colleen Kehoe have decided to add another principle.

h) Principle 8

Principe 8 is the following: "Provide support to enable students to navigate the hypermedia in a comfortable manner".

2.2.4.3 ABLE in our web site

Now, let's see if our web site respect the different recommendations of ABLE.

a) Principle 1: Provide adaptable Scaffolding through levels of details

As we said before, the web site was created to let the student choose the level of details he wanted for his information. The entire site was made with this idea. We can give an example of this characteristic:

When the student want to learn how to use the criterion of consistency, he doesn't receive at first time the complete criterion with full of explanations and a lot of examples. At the beginning of his apprenticeship, each criterion is described by a simple sentence. After that, he can choose to see some examples of bad or good use of this to try to understand the complete criterion by himself. If it's not clear or if he wants to learn more about the criterion of consistency, he has the possibility to click on a hypertext link to receive the complete explanation of the criterion. At this time, he has again the choice to look at the different examples or cases that concern this criterion in the base. If the student thinks he really understands the criterion, he can now click on another link to have access to the different sub-criteria concerning by the consistency criterion.

We think that this principle is very important for our web site. The way of learning of students is often totally different from one student to another. One prefers reading as little theory as possible and discovering rules by using examples. Another prefers maybe reading all the theory before going to examples and cases of the base. With this principle implemented in the site, it is the student who chooses his program. He can then also choose his own rhythm and improve himself, as he wants.

What we said here for the theory is also true for the other parts of the site. For example, when a student accesses to a case, he first receives a summary of the case with its good and bad properties. If he wants more information, he can click on a button to have access to the entire case with more explanations: picture, comments,... If it is not enough, he sometimes has the possibility to click on a part of the picture (or on a hypertext link) to consult the different sub-cases.

b) Principle 2: Strategy information is available but not immediate

How can we see if a student has a good feeling with the theory he is studying? In a traditional lesson, teacher will ask questions to verify if the student understand what he says. So the task of the student is:

1. Watching the teacher performing some task
2. Predicting what the teacher will do or say next. If the student predicts the good thing, the theory is understood. When the predictions don't match with what the teacher does or says, the student has to understand what was wrong with his reasoning.

But where is the strategy information in our site? We can't ask to a student to find the theory by himself. No, the real strategy information is of course the critiques of the different cases. What the student wants to learn by using our web site is how to use the rules concerning HCI and how to criticize an interface.

How can a student do that? In fact, it's relatively easy. When a student access to a case, he receives first a summary of the case with its critiques. For example he can read: « in this case, there is an example of bad use of the criterion of compatibility ». The following step is the presentation of the case and its interface, and next the real critique. So before reading the critique, the student has the possibility to practice himself and to make his own critique.

After, he compares his critique with the one made by the teacher (or another student) to see if it matches.

The same way of work is possible in the other case base (« Story of a case »). First, the student accesses to the work and the grade that the work obtained. At this time, he can try to understand why the work received such results. After he just has to compare what he said with the decisions of the teacher.

c) Principle 3: Outcome information is available but not immediate

This principle is in our situation not really different than the second one. The system is always the same: the information (outcome information or strategy information) is available with a simple click on a hypertext link or a button but is not immediate. The student has the possibility to use it or not.

d) Principle 4: where possible, suggest potential problems and solutions

The web site we have created doesn't suggest potential problems and solutions. We don't have a system that analyzes the case that someone submitted and find the errors. But it was not the purpose of the site. In fact, the goal is exactly the opposite: what we want is a site that let students think by themselves without the help of a system. In fact, they are other students who make the suggestions of potential problems and solutions: one of them proposes an interface he has found (or created) and put it in the case base. And the others try to exercise themselves by critiquing this case.

e) Principle 5: use multiple representations

Using different representation was one of the directives we received to create the web site. It is indeed easier for the student to understand something if he has different representations of the same thing. We have tried to let the choice to the student as often as possible. The utility of a representation depends of course of what student searches of wants to do. We can give some examples to explain the different situations:

- *The list of the cases.*

  There are two ways to find a case in the base. First, you can use the entire list that is available in one of the pages of the site. This is a very good solution if you know the name of the case and if the list is not too long. In the case of the « story of a case » base, you have the list of the different cases with, for each case, a summary of the good and bad points and the grade obtained. If you're looking for cases with a grade of 17/20, you just have to read the list and select the cases with this kind of grade. Secondly, you can use the search engine. To do that, you have to fill in a form (a JAVA applet in fact) with the description of the case you're looking for: name, particularity and grade... The search engine scans the list of cases and selects for you the relevant ones.

- *The specific AIO.*

  When you want to have a description of a specific AIO, for example a scrolling bar, you have some different descriptions available. For example, you have a textual description with the name of the object, items that compose the object, and so on. You also have a graphical representation of the object. By following the links, you can find other graphical representations of the object if there are interfaces containing it in the base.

Having different representations of something is really important for student. He can so test his knowledge in an easier way.

f) Principle 6: Design for use in practice

From a well designed learning perspective, our intelligent behavior is deeply related to the spaces in which we practice and the tools that we use in our practice. By placing the learning of design in the context of doing design, the opportunity for students is increased to note where the theory of learning applies and to see a need for the learning in their practice. So, if the exercises that students realize are close to what happens in real life, they will learn and understand more easily the theory.

That is exactly the situation that we try to establish in our case base. The different cases that we have studied and put in the base are real cases existing on the web or in programs as Microsoft Word. Some of the most interesting cases to study can be found on the web. The students can analyze and criticize the work of professionals. They can thus link the two things together: improve themselves in the domain of HCI and propose cases that are not created just to explain the theory but that really exist. Our base is a good means to bring closer to the real workplace.

g) Principle 7: Support the sense of community

An important aspect of apprenticeship is the integration of the student into the community of practitioners. While students sometimes create a community around a class and a set of assignments, not all students may be involved, or may feel part of the community, or may be sharing with the community.

Our web site gives the possibility to create strong consistency between the work of the student: for example one part of a classroom can create the cases and the second part can criticize them. Moreover, it can bring closer students from different levels: senior students can for example create the case for one of their works and let junior students test them and try to understand the HCI rules by applying them to senior students cases. All the students have to do is to put their cases on a page of our site and to wait for the critiques of other students.

The connection between our site and the facilities of use of the CoWeb let us imagine real interactions between junior students, senior students and teachers. A professional can also be really implicated in the systems because he don't have to go every day to the university to give advises to the students, he can do that really easily the evening after his work.

h) Principle 8: Provide support to enable students to navigate the hypermedia in a comfortable manner

The challenge in applying this principle is that it is not an initial design principle but an iterative design principle. Sometimes, after using a new system, students imagine things that would have been very useful and that are not represented in the system. The creator has to make the changes after the creation.

In our web site, this problem is relatively easily solved. It is indeed not a problem to change the structure of a page or to add functionality to the base. The CoWeb gives the opportunity to test things. Everybody can change a page and see with others if the result is better. If it's ok for everyone, the page is keeping. In the other case, you just have to reload the last page. An important thing is also that the students are involved in the improvement of the web site. They don't need to wait for the teacher or creators to make the changes, they can do it by themselves.

2.2.5. Conclusion

In summary, the web site that we have created seems to correspond with the majority of the principles described by Professor Mark Guzdial and Colleen Kehoe. Some parts can be improved to match the needs of the Students but they can modify the structure of the pages and their contents to obtain the site that they really want.

This is with this acquired knowledge concerning apprenticeship that we began the conception of our Vesale application. That's what we will explain in the next point.

2.3. Pedagogical approach used in our thesis

In this point, we will present the application developed for the Vesale project. This application has been realized thanks to the knowledge and the experience acquired in the United States.

First, we will explain the principle of the case base. Next, we will present the main pedagogical points of our application. Finally, we will see if the application fit to the ABLE principles [GUZDIAL and AL 98].

2.3.1. The case base

The goal of the application is to familiarize users with the design of interfaces. For that purpose, we have chosen to construct a case base that will content interfaces with or without critiques. We can see two goals to this application according with either the case insertion in the base or either the case consultation.

First, by inserting cases, user will get practice of interface design or interface critique. He will acquire therefore a certain experience in the HCI field. More, he will receive comments from other people about his work, people of all horizons what is impossible in a normal context.

Secondly, the case base constitutes a sort of library in which people desiring to construct an interface will be able to immerse oneself. They will discover there, good and bad examples of interfaces and will be able to receive some profit for the construction of their own interface. It is an attitude that we all have: when we have a work to realize, we attempt to see what has already been made in the field to have some inspiration and not to commit the same errors.

Note that we choose a different approach of the one used in the United States with the CoWeb. In this environment, we had left a lot of freedom to users by structuring only very few critiques format and insertion procedure. The problem was that the provided information were not easily reusable to index the case. So, we tried to find a certain balance between a constraining formalism and a freedom difficulty reusable.

2.3.2. The six main points of the application

Now that the context and the goal of the application are fixed, let's see the six main points of our application concerning pedagogy, namely qualification of the users, types of cases, possible critique types, graphical tool, help and finally indexation of our case base.

2.3.2.1 Qualification of the user

When a user desires to insert a case, he has to identify among the following types of users: student, incumbent professor of the course or external (novice or expert). The goal of this qualification is to put a certain label of quality on cases and critiques. Indeed, during the case consultation, one can grant more confidence to a case inserted by the incumbent professor of the course than to a case inserted by a student. Concerning the third category, we have decided to put it a bit aside because critiques of external people (that's to say people from outside the theoretical world of universities or from outside our European mind) are to take at another level. Indeed, as we have seen during our internship, idea of interfaces can be very different from a place to another. Externals bring therefore very interesting critiques and a more innovative style. So, it is interesting not to mix with others.

2.3.2.2 The three types of cases

As we will see more in details in the following chapter, three types of cases can be introduced in the case base. We decided to divide the cases into three categories because we saw during our internship that the base can be used in different situations and this categorization seems to take into account the observed situations.

The first one is the *"presented case"*. This type of case consists just in the presentation of an interface and its different windows. It can be useful to a student who has created an interface and who desires to have some comments from other people or to the professor who wants to propose exercises to his students.

The second type of case is the *"presented and criticized case"*. Here, in addition to the interface presentation, at least one critique must be realized. This case can be seen in two situations. If a student inserts a case and criticizes it directly to know what others think about or if the critique is realized after the case consultation. When students want to practice to criticize interfaces, it is to this type of case that they will direct.

The last type of case is strong similar to the precedent, it concerns *"reference case"*. In reality, we could name it: "presented and criticized case made by the incumbent professor of the course". It concerns indeed a realized and criticized case made by the professor and therefore judged as reliable, or a case where a realized critique suggested by another person and validated as an interesting case by the professor. For a simple consultation to find examples, it is this kind of case that will be consulted by the users.

2.3.2.3 The three types of critiques

The critique of an interface is something very complex because it can be made to a lot of levels. In an opposite way of what has been realized in the United States, we have chosen to structure the critique to guide the user. We are entirely aware of the constraining aspect of this choice but it facilitates the work of user and improves grandly possibilities of the search engine since we can work on more precise elements. We have therefore chosen to allow three critique types, by leaving from the most global to the most particular.

The first critique is a global critique allowing the user to give his impression on the interface as totality of windows and sequence. To optimize this critique, the presence of a video sequence presenting the interface and the different sequences is very useful. However we have to be aware that it will be present only in very rare cases because it needs an important work on the part of the person that has inserted the case. So, we propose to the user a scrolling of the different window of the interface. A global critique being difficult to make, we have chosen to allow free comments.

The second critique is the inter-windows critique. This critique allows the comparison between two particular windows of the interface and the notice of some inconsistency or other details. This type of critique allows student to become aware, just as for the precedent critique, that an interface is composed by a totality of windows that must have links together.

Finally, the last possible critique type is the intra-window critique. Here, user criticizes the interface of a particular window, without any reference to the others.

Notice also that a vocal critique will have to be allowed. We have not implemented it by lack of time.

2.3.2.4 Graphical tool

To facilitate the task of users, we equipped our system with a graphical tool. This graphical tool is the counterpart of the imagemap technology used in our web site but it seems to us easier to use for the students. His goal is to facilitate the critique and the perusal of the critique. It is divided into two parts: drawing tools and zoom. Note that the graphical tool is available for inter and intra-windows critiques

a) Drawing tools

When somebody has to comment an image and different points, it is far easier to approach of it and to show to other the place whose one speaks. In our case, it is an impossible thing. It would have therefore been necessary to describe precisely the object or the zone whose one spoke. To avoid this problem, the user disposes of squares to mark different zones of the image. When he draws square on the image, it is automatically numbered and this number is postponed in the zone of comment. So, each notice is easily locatable. To facilitate more the perusal, positive critiques are made with the help of green squares (comments are then put automatically in the positive comment zone) and red squares if they are negative.

In the case of an inter-windows critique, squares go by two because it concerns both windows. If the user wants to insert a remark on only one window, he can draw crosses.

b) Zoom function

To be able to criticize an interface and to have its representation on the same screen, we had to reduce strongly the interface picture. Therefore, it is not very easy to draw precisely forms over. To tackle with this problem, we created a zoom function that opens the image in a new window at its real size. The user has always the graphical tool to annotate the picture. Once the picture is annotated, the user returns to the critique window and can comment his critique. Remark that the size and the placement of the different elements of the image are automatically adapted each time.

2.3.2.5 Help

There is of course an indispensable element in the application and that is the help provided to users. This help takes two forms.

Firstly, the help obtained in small info-bubbles. It is, by far, the most developed help in our application. When the user puts its mouse on different buttons or fields of the interface, a small explanatory bubble appears. For example, when the mouse is on a button «following», the info-bubble indicates what is the next window. The huge advantage of this type of help is to be rapid, small and very easy to use.

Secondly, the help file. This type of help has not been developed by lack of time. It is thanks to it that links will be provided to notions of the course of HCI present in the case base.

2.3.2.6 The indexation of the case base

One of the most important question to ask when someone creates a case base is the indexation: "how to index efficiently each case of the base in order that users could find what they seek with the less difficulties? ". Indeed, a good part of the process of apprenticeship that we want to develop depends on the facility to find appropriate cases in the base and therefore on the indexation chosen for these cases.

So it is useful to really think about the choice of the different keywords that we are going to use to cover our base. Notice that two requests can be made. The first one is relative to a case: one will seek a (or several) particular case(s) in the base. The second is relative to a critique: one searches more a special case in the base but rather a relative critique to a certain element (that could be an ergonomic rule or a particular AIO). The different keywords allowing the indexation of cases and critiques will have therefore to allow to find cases according to these two types of research.

In the following section, we are going first of all to expose the different keywords that we have selected. Then, we will try to show that these choices were judicious and will allow us to construct a good process of apprenticeship.

a) Choice of keywords

As we have explained previously, two types of keywords are used to index the case base. Notice that these two types have not to be exclusive: a user has to be able to undertake a combined research between these keywords. The search engine has therefore to be sufficiently developed to allow all possible combinations. As example, we can give some request that a user could have undertaken as well as results that the search engine would have to return:

- A user wishes to obtain a list of all cases of the base that have been inserted by Gaëtan Prévot. He makes therefore a request on the **name of the producer** and the former is well only dependent of cases and not of critiques. The results provided by the engine would have to be for example:

  1. Microsoft Word (Gaëtan Prévot)
  2. ATM Banking (Gaëtan Prévot)
  3. Exercise A (Gaëtan Prévot)

- A user wishes to obtain a list of all cases of the base that have received a critique concerning the design criterion: Consistency. He thus makes a request on the **criterion of consistency**. The former is dependent on the critiques. The results provided by the search engine would have to be for example :

1. Microsoft Word (Gaëtan Prévot)
   Inter-window critique (Rudy Michiels)
   Global critique (Patrick Kreps)

2. Microsoft Excel (Olivier Davreux)
   Inter-window critique (Elise Remy)

Notice that the engine informs only the relevant critiques. It is possible that the Microsoft Word case has received other critiques but that they did not concern by the consistency criterion. Engine keeps only relevant results and therefore ignores the former.

- A user can also want to make a combined research. He wishes, for example, a list of cases that belongs to pharmaceutical field and that have received a critique concerning a scrolling menu. He makes therefore a research on the **application domain** and on **a specific AIO** that is the scrolling menu. The results provided by the search engine would have therefore to be :

  1. Pharmacy A (Rudy Michiels)
     Intra-window critique (Gaëtan Prévot)

  2. Pharmacy B (Olivier Davreux)
     Intra-window critique (Gaëtan Prévot)

Now that we know the type of results that the engine would have to provide, it is time to make the list of keywords that we have chosen to index our base.

i) Keywords relative to a case

Keywords that are going to serve to index cases of the base are:

- Name of the person who inserts the case
- Name of the interface
- Domain of the interface
- Type of the case (presented, presented and criticized or case of reference)

ii) Keywords relative to a critique

Keywords that are going to serve to index criticized cases of the base are classified in two families: one that concerns design criteria and another that concerns the different generic AIO (these AIO are after particularized in specific AIO).

In the family of criteria of design, one will have keywords:

- Compatibility
- Consistency
- Workload
- Adaptability
- Control of the dialog

- Representativity
- Guidance
- Management of errors

In the family of generic AIO, one will have following keywords:

- static AIO (label, separator, group box, prompt, icons)
- Scrolling AIO (scroll arrows, slider, scroll bar, thermometer, frame)
- Action AIO (menu, menu item, action bar, pull-down menu, pop-up menu, cascade menu, submenu, embedded menu)
- Control AIO (control AIO of action, control AIO of information)
- Dialog AIO (window, help window, logo window, textual publishing window, graphic publishing window, dialog box, expandable dialog box, repetitive dialog box, file open dialog box, panel, control panel)
- Feedback AIO (information message, warning message, help message, action message, progress indicator, pointer)

b) Justification of chosen keywords

i) Theoretical notions

To justify the choice of our keywords, we begin first of all by reminding some theoretical points explained in the book of Janet Kolodner: " Case-based Reasoning" [KOLODNER 93]. These reflections aimed to release some properties that a system of indexation must possess in order to judge it satisfactory and efficient.

We will now try to fix properties that we would like to see verified by our system of indexation [KOLODNER 93 pp. 193-198]:

- It has to be able to anticipate the vocabulary that a user can use to make its request. Thus, the system of indexation has to be predictive.
- It has to be able to anticipate circumstances by which the user will want to obtain a result of the search engine. Thus, suitable predictions have to be useful predictions, they have to satisfy goals that the user has fixed.
- It has to be enough abstracted in order to make a case useful in the maximum of possible situation.
- It has to be concrete enough in order to be easily identifiable for future situations.

- Finally, it has to be useful and not superfluous. It is useless to institute an indexation that will never be used.

One can therefore release from these notions several qualities that our system of indexation must verify [KOLODNER 93 pp. 198-202]:

- *Abstractness of indexes*

  The indexation of case has to be chosen such that the case could be used in the maximum of possible situations. That often means that chosen keywords have to be more abstracted than detailed for a particular case. For example, if we speak about a cooking book, it is often useful to have a keyword " Meat " regrouping a series of more specific terms such as " Meat of beef " or " Meat of pork ". The generic term will allow to consult a list of all the dishes containing meat.

- *Concreteness of indexes*

  The danger of an abstracted indexation is that it can be so much abstracted that some cases found by the engine will have nothing to do with what the user asked. The inference asked to the user is then too important to be really appropriated. A typical example of this situation is a search engine on the web. Results found by this type of engine are often far from being efficient because it uses too vague notions to really allow to target the research.

  This is why the system of indexation has also to be relatively concrete to be able to be recognized with a light inference.

- *Usefulness of indexes*

  Finally, it is sure that the system of indexation has to be chosen by keeping well in head the criterion of usefulness. It is useless to create a system of indexation that is not relevant and that is not useful to the user. It is therefore necessary to understand goals of the user in order to be able to choose the best keywords.

ii) Application to our system of indexation

It is time now to see if the system of indexation that we have chosen is relevant with all theoretical notions presented in the last section. In order to do that, we will resume each notion and will explain why and how it is verified by our system.

1. Abstractness of indexes

Our system of indexation has been studied to be able to account of abstractive keywords. To prove this fact, let's take a look at the different possibilities of the indexation concerning abstraction:

- First of all, **the domain of the interface** is an important keyword that will have an abstract character. Indeed, the type of data expected in this field is not something like " Medical Area linked to the study of neurology with the help of computers ". This field will be rather filled with information like " Medical Area ". The importance of the abstraction is here really

demonstrated: the case characterized by " medical area " will be more often found and used in the case base than the one characterized by " medical area linked to the studied of neurology with the help of computers ".

- Keywords corresponding to **criteria of design** (Consistency, compatibility,...) have also been taken in order to be sufficiently abstracted. Thus, one would have been able to specialize far more these keywords by using the sub-criteria relative to each of chosen keywords. However, we think that the formers would not have been so usefulness for our base. Indeed, presumed user of the base will be novice students in the area of human computer interface, it is therefore more useful to present to him the great principles governing this field rather than to particularize so much the research that it furnish no convincing result.

2. Concreteness of indexes

Our system of indexation has also been studied in order to be the most concrete possible. It is important (to improve the character of apprenticeship linked to the case base) to present some concrete points in order to be able to understand each notion. This will be also true for AIO. So, it has been useful to choose relatively concrete keywords:

- Keywords linked to AIO required unhesitatingly to be presented with a concrete form. To limit choice of user with generic AIO would not have been very useful. Nobody speaks in term of generic AIO: from the person who makes the critique to the one who uses the search engine, everybody uses the specific AIO:

  - Nobody will emit the following critique: " The static AIO has not been well aligned as compared to the other objects of the interface ". The emitted critique will be rather: " the label X has not been aligned as compared to «Cancel» and «Next» buttons figuring on this window ".

  - It is more useful to make a research like "I want to obtain all critiques concerning the utilization of scroll bars".

Notice that even if someone wishes to remain to a more general level of abstraction and not to enter in the detail of specific AIO, he will always have the possibility to do it by selecting all keywords corresponding to the generic AIO. Similarly, if a critique relates not to a specific AIO such as the scroll bar but to the totality of scrolling AIO, the person just has to select all keywords corresponding to the generic scrolling AIO to characterize the case.

3. *Usefulness of indexes*

Some keywords have been added in order to make the search engine the most efficient and the most useful possible. They cover a majority of researches that a user can think to undertake. Moreover, combined with the other keywords, they allow the search engine to be very efficient and very selective if needed. These added keywords are the following:

- Name of the person who inserts the case: it can be very useful in some situations to research cases of the base according to the name of a person.

- Name of the interface : evident because this name is the identifier of the case

- Domain of the interface: this allows to see all cases that have already been studied in a particular field. This can be useful notably if somebody has to create an interface in a certain area and that this person wants to avoid often-committed errors in this domain.

- Type of the case: it is useful to allow researches like «I want to see all the case of reference of the base».

- Date of the critique: this criterion is interesting to verify changes in the case base since the last time that one has consulted it. One can for example research the totality of critiques that have been emitted since a certain date or between two dates, and so on.

The totality of these justifications proves that keywords of our case base have been chosen according to the different properties exposed by Janet Kolodner.

Another justification concerns the number of keywords chosen. It is evident that we would have again been able to add others keywords to refer our base. However, too numerous keywords are inefficient because the work of the person who inserts a case becomes impossible: he cannot verify all the keywords and so let the application choose these words. This would destroy of course all apprentice notions linked to these keywords. As this notion is essential for the user since it allows him to verify what he said in his critique or to find relevant case (concerning the consultant), it has therefore been necessary to limit the choice of keywords to increase the character of apprenticeship linked to the case base.

2.3.3. ABLE or not ABLE ?

2.3.3.1 Principle 1: Provide adaptable scaffolding through levels of detail

This principle is not met in the current developed application but it will be very useful and not too difficult to provide this adaptable scaffolding during the insertion or during the consultation.

Concerning the insertion, we can say that the scaffolding is principaly possible for cases proposed by the teacher. The insertion of the case must then be decomposed into three steps. First, the teacher inserts his interface as everybody, following the procedure currently developed. Secondly, he has to predefine some shapes or zones on the window, shapes that will show some places where critiques can be made. Third, he has to put some tips or

comments about the predefined zones that he has selected to give information to the student about the critique that can be made. So, when a student will criticize the proposed interface, he can either criticize the interface without help, either ask some help to do it. In this case, we can say that adaptable scaffolding through levels of details has been provided

Concerning the consultation, we can follow the same guideline. When the user consult the interface, he just see the illustration and by asking more details, he can see to appear the shapes, brief comments and total comments.

2.3.3.2 Principle 2: Strategy information is available but not immediate

Once again, this principle is not present in the developed application. Indeed, the user doesn't have any information concerning the strategy used. The reason of this choice is not to flood the user with useless information. We have estimated that the strategy was transparent enough and letting the possibility to the user to have information about this subject would risk misleading it uselessly.

2.3.3.3 Principle 3: Outcome information is available but not immediate

This principle can be encountered during the case consultation. Indeed, it is necessary that the consultation unfold as follows. The user sees first of all the illustration of the interface. He can at this time decide what is correct or not. He clicks then on a button that has for effect the display of the different rectangles and cross on the illustration. That confirms or infirms his first ideas on the interface. The last is the display of the text of the critique realized. In this way, the user has to his disposition all the information of the critique in a gradually manner that allows him to compare its expectations to the reality.

2.3.3.4 Principle 4: Where possible, suggest potential problems and solutions

This principle is applicable but not automatically. Indeed, during the insertion of a critique, no mechanism allows to give a feedback to the people who insert the case. However, two external possibilities are left. Firstly, during the consultation of the inserted case, the other persons can send comments or again realized a new critique of the same case. Inserant receives then a sort of feedback of his critique. Secondly, it is possible to insert a new case that is the correction of an existent one. Thus, a student having inserted an interface that he has created will be able to see alternatives proposed by others persons.

2.3.3.5 Principle 5: Uses multiple representations

The multiple representation is present at two levels, as we will see. Firstly, the same interface can be criticized several times by the same or by different persons. It will sometimes concern critiques about the same points but expressed in a different way.

Secondly, a same interface can have several representations. What we mean there, is that it is possible to insert interfaces that are the correction of an existent interface. So, the interface, in general sense, will have several representations.

2.3.3.6 Principle 6: Design for uses in practice

This sixth principle is entirely achievable but does not depend on the application in itself. Indeed, it can be associated with a project about the design of an interface and in this case, it can be very useful. This will be the case here because the case base finds its place in Vesale project. The student will use therefore the application in the framework of the apprenticeship of the various concepts of HCI.

2.3.3.7 Principle 7: Support the sense of community

It concerns of course a very important point. Indeed, the case base can be interesting only if many cases are found there and these cases are inserted by users. These users or other persons realize the critique of interfaces. It is therefore essential that a group uses the system and communicates together. That's why the Vesale project has been created. It will regroup students and professionals of HCI around a case base. So, students will be integrated in a group.

2.3.3.8 Principle 8: Provide supports to enable students to navigate the hypermedia in a comfortable manner

This principle can't be encountered at this time. It must be study later after use and evaluation of the current application

2.3.4. Conclusion

We have seen in the preceding point that our application meets only some ABLE principle. With some improvements (for example, the consultation in three steps), majority of them will be met.

In the last section of this chapter, we will introduce a pedagogical methodology concerning apprenticeship.

2.4. Attempt of theoretical justification

As said before, our first approach is not explicitly based on a pedagogical methodology. Nevertheless, pedagogy must be an important point. We think so that it can be useful to see if our application meet some pedagogical principles. In order to do that, we will explain first these principles and we will try to see how our web site and our Vesale application meet the paradigms.

2.4.1. The six paradigms of Leclercq - Denis

The goal of the project that we developed is to allow users to learn notions of HCI. But the apprenticeship is something complicate and many models and paradigms exist about it. For example, Piaget has developed a model named: "the piagetian model". In this model, the subject acts to transform the reality, to question it; the former resists and obliges him to adapt himself by the elaboration of new cognitive structure. To re-establish his balance between elements of the situation and cognitive means, he has to start a process of optimal solution research or «equilibration». Thus, learning is a construction of the knowledge that goes through levels of balance succeeding to moments of imbalance. This model is based therefore on the sequence: equilibration - desequilibration - reequilibration.

Another model that we will see more in details is the "Paradigms apprenticeship/teaching " from Leclercq and Denis [LECLERCQ 98].

First of all, let's begin by giving a definition of the two terms. For this purpose, let's refer to the glossary proposed in pedagogical course in Namur [DONNAY 97]. We can read this:

Apprenticeship: Active, constructive and cumulative process that happens when learner handles actively the new information, modifying thus his cognitive structure. The information can come from the own long-term memory or from outside the person learning.

Teaching: System of tasks that can develop, enrich, elaborate gradually. The different tasks are those linked to the three teaching phases: the preparation (preactive phase), the professor-student interaction (interactive phase) and the evaluation (postactive phase).

This system aims the establishment of pedagogical relationship, which derives from the presence of the 3 following relationships: the didactic relationship (professor-discipline), the mediation relationship (professor-student) and the apprenticeship or study relationship (student-discipline).

This system of tasks takes place inside a definite time and several environment levels (from the values of the society to the temperature of the classroom).

We can see, through these definitions, that the two terms are strongly linked. Therefore it is not surprising to link them within paradigms. Leclercq and Denis propose us 6 that we can divide into two categories. Indeed, the three firsts are rather on the initiative of the teacher while the three others depend more on the learner. We will see them more in details now.

2.4.1.1 Paradigm 1: Impregnation/Modelisation - Referenciation

The humans learn a lot by looking around them and by living in a particular environment. That's what Leclercq and Denis named «impregnation». For example, number of rules of the road regulations are known by children, not because they studied it but well because they have observed the behavior of their parents when they drive. This impregnation can be realized thanks to all our perception organs. The formative can favor this impregnation but most of time, it is either automatic either non-existent. For example, parents serve as model for their children without they will it explicitly.

2.4.1.2 Paradigm 2: Reception/Transmission

This paradigm is about the most common apprenticeship in various institutions. It regroups conferences, syllabi... that's to say everything that concerns transmission and reception of information.

2.4.1.3 Paradigm 3: Practice/Guidance

"It's on the ground that we learn the most". This sentence summarizes alone a good part of this paradigm. We learn a lot by practice. However, certain guidance is necessary. It allows us to earn time, energy and especially keep our motivation. The formative has therefore to try to propose the most possible practice to the student and especially to help him in his apprenticeship

2.4.1.4 Paradigm 4: Exploration/Supply

The apprenticeship requires of course a certain will of learning. This will has to translate into an exploration of the different documents or tools he disposes. This exploration can have two aspects: it can be free or oriented. In the first case, learner is entirely free to do what he wants and to consult information, as he wants. In the second, certain formalization (more or less strong according to circumstances) can be applied, restricting a bit, for consistency reasons, his learning freedom of apprenticeship. The role of learner is therefore essential in this paradigm but it is necessary to specify that the formative has to put information at the disposal of the student.

2.4.1.5 Paradigm 5: Experimentation/Reactivity

There are cases where learner has to be able to experiment, that's to say to manipulate the environment and, if needed, to modify it, by exhausting and combining possibilities that he judges significant. The formative, in this context, has to offer reactions, that's to say to make accessible to learner not only an handable environment (whose parameters are modifiable) but also a reactive environment. The typical method to settle the apprenticeship by experimentation consists in confronting learner to case problems.

2.4.1.6 Paradigm 6: Creation/Confortation - Confrontation

Finally, we learn also by creating new things, by realizing things by ourselves. The role of the formative will be, in this case, to provide an environment favoring and allowing this creation. More, he will have to reinforce and confront learner. To confort him, he has to encourage him, to sustain him and to motivate him. To confront, it is to submit his own work to the notice of others.

The various paradigms that we have seen are combinable together but not in the same proportion. In reality, it is rare to find an apprenticeship situation that combines the six paradigms in the same proportion.

Let's now see if our web site and our application meet these paradigms.

2.4.2. The 6 paradigms in our web site

2.4.2.1 Paradigm 1: Impregnation / Modelling - Referenciation

As Leclercq said in his book, we learn a lot of things by looking at what the other people do and by doing the same.

It's why it is very important to have a lot of cases in our base. When someone wants to create an interface for an application, he can begin by looking in our base if there are examples of interfaces that correspond to what he wants to realize. He can then see the common errors that people do or be inspired by what he sees. The importance of reference cases is clearly shown with this paradigm. It is of course better for a novice to have examples of really good cases in order to be impregnated before beginning his work.

2.4.2.2 Paradigm 2: Reception / Transmission

This kind of apprenticeship can of course be made with our web site. The first part of this site (concerning tutorials) is a kind of syllabus that we offer to the students. They can read information, have different examples to illustrate it,... It works exactly like a reference book about HCI, and we try to transmit information by using these tutorials.

Another example of this paradigm in our site is the reference cases. The incumbent professor who shows an interface with its critiques makes these cases. The students can receive some information about the use of the different rules and criteria concerning HCI.

2.4.2.3 Paradigm 3: Practice/Guidance

Our entire web site is devoted to the idea of practice. What is really important for the students, is not to consult the different cases of our base. They will learn better if they practice by themselves: if they put some new cases in the base, if they add a critique on a case, and so on. The guidance is important but in our site, we have decided to let the user do the maximum without help. In one page, we explain how a user can add a case or make a critique, we explain some rules and criteria that an interface should respect in order to facilitate the work of the student. But the facility of the CoWeb allows the user to find his own way to criticize a case or to add something. Little knowledge are necessary and little rules have been demanded to let the site be as open as possible.

The guidance is also assured by the fact that the different students could work together. As we have explained before, senior students can work in collaboration with junior students. Older students can thus help the others to understand criteria and guide them to a better comprehension of the domain.

2.4.2.4 Paradigm 4: Exploration/Supply

This paradigm concerns a certain wish of learning something. If a student is motivated and makes researches about what he studies, everything will be easier for him. The notion of exploration is important and always present in our site. As we explained before, the site is not realized like a sequential book. The different sections are linked together in order to let the user choose what he really wants or needs at this time. He can thus explore what he believes to be the more useful things for his study. All important information concerning HCI have been put to the disposal of student. Now all he has to do is to explore the site and the different cases to find what he is looking for (things like hypertext links and search engines have been created in order to facilitate his researches)

2.4.2.5 Paradigm 5: Experimentation / Reactivity

The CoWeb environment lets the student to experiment the different possibilities of creation and critiques of interfaces thanks to the "sandbox" section of this environment.

2.4.2.6 Paradigm 6: Creation / Confortation - Confrontation

The main purpose of this web site is of course to let the student adding new cases in the base and to let him learning how to make a relevant critique. The environment and the design of each page have been thought to allow a student to realize these tasks. A student is also motivated to realize a good job by the facts that other students or professors can criticize

his work. In this case, the students are confronted to the reality and the opinion of others can help them to improve themselves.

2.4.3. The 6 paradigms in our Vesale application

2.4.3.1 Paradigm 1: Impregnation/Modelisation

This paradigm is well present but can appear harmful in some cases. The user will be most of time a user familiarized with the data processing and its universe. It has therefore already seen numerous programs and their interface. If these last were good interfaces, it's perfect. In the opposite case, this paradigm becomes a problem to correct because the user has bad example in mind.

It goes similarly for our application: the interface that we have developed has to respect and to illustrate the different principles that we attempt to teach. We see therefore all the importance of the bootstrapping (see chapter 7 for more information about it).

2.4.3.2 Paradigm 2: Reception/Transmission

This paradigm is found in the Vesale project but is less present in our application. Indeed, we have no lessons properly speaking, we just have links that will have to be created to the on-line course. However, one can consider the different critiques realized by students and especially by professors (reference case) as a sort of transmission. More, main notions will be briefly explained in the application help.

2.4.3.3 Paradigm 3: Practice/Guidance

The teacher can insert cases to criticize. In this way, students can practice different critiques over cases. The teacher can then consult the different critiques and give a feedback to the student by telling him what was good and what was wrong.

More, learner can construct interfaces and submit it to the critique of the professor or other consultant. So, he can practice the design of interfaces and receive guidance from the professor.

Concerning guidance again, we can say that it is more present than in our web site thanks to formalism required by the use of form to perform the critique.

2.4.3.4 Paradigm 4: Exploration/Supply

Users can experiment the interface design by creating their own interface. The reactivity is provided, in our case, by the reaction of other users who can send comments or post a new critique.

2.4.3.5 Paradigm 5: Experimentation/Reactivity

This paradigm is not very present in what we have developed. Certain experimentation is possible but the reactivity associated will not be automatic because there are no good or bad critique. It is therefore very difficult to react according to the experimentation of learner.

2.4.3.6 Paradigm 6: Creation/Confortation - Confrontation

The creation aspect is entirely possible for the learner because he can insert new cases or new critiques. It will remain however to give him the aspiration and the motivation to do it. So, for the teacher, the most important role will be without doubt the confrontation. Indeed, the goal of the application is to comment critiques or to realize some news. The confortation will be present only to motivate the user to insert his own cases.

2.4.4. Conclusion

Both applications seem to be in concordance with the different paradigms despite the fact that these paradigms have been studied afterwards. Finally, it seems that our experience and our feeling have been a good basis to elaborate our study.

2.5. Conclusion of the chapter

The design of the web site and the ideas of Professor Mark Guzdial were very important for us and for the design of the Delphi application that we have realized for the Vesale Project. The difference between the way of apprenticeship in the Georgia Institute of Technology in Atlanta and the "Facultés Universitaires Notre-Dame de la Paix" in Namur are obvious. In the USA, the tendency in apprenticeship is to be widely opened and let the students choose their own structure. Americans seem to be more active in discovering new ways to teach. In Europe, the tendency is to structure a little more the applications in order to facilitate the work of the students. Both methods have their advantages and their disadvantages. So, it was very helpful for us to have another idea of what apprenticeship could be before starting the job on the Vesale Project.

Chapter 3:
Case insertion

3.1. Introduction

In chapter three, we explained the pedagogical approach we have followed to realize our work: the goals we had to reach, the recommendations we had to follow, and so on. We made these observations for the web pages in general or for the entire application. Now, it's time to analyze in details the first task of our application: the case insertion. We have to understand what the user really wants to do during an insertion, we must study the different procedures we will need, and other things that will help us to construct our application.

We will begin this chapter by fixing the objectives of a user when he inserts a case in the base: we will describe the different cases he wants to enter, the kind of critiques he wants to make. This section will also briefly explain how a user can discuss with another one to obtain more information about a case.

After this section, we will know exactly the goals and needs of the users. It will be a good time then to realize the task analysis. This analysis will let us find the different functions and objects (presentation units, windows...) we have to implement to realize the task.

These two sections will show us what the user wants and what we need to satisfy him. After their analysis, we will be able to create our application. The third section will explain the graphical tool we have created. We will also give a scenario of insertion that will present step by step what the user has to do to enter his interfaces.

The last section of this chapter will show a detailed description of a window with the different CIO[1] we have used.

You will find in appendix 4 the conversation conception of the case insertion. This appendix will detail the chaining between windows of our interface.

3.2. User objectives for an insertion

3.2.1. Introduction

In order to realize the task of insertion, the first thing we have to do is to understand the objectives of the users. Once we will be sure of the goals, it will be easier to implement solutions.

[1] Concrete Interactive Object

A user who wants to insert an interface in the base needs three tools:

- *A tool to present the interface.*
  A lot of things can be employed to present an interface: windows, textual explanation, video sequence, ... The application that we have to realize must offer the possibility to present all these parts of an interface.

- *A tool to criticize the interface.*
  A user also desires to enter critiques concerning the interface he inserts. So, we have to think about the different kind of critiques a user may want to make. Each critique can have different purposes and then can require different tools.

- *A tool to discuss with other users.*
  A user must also have the possibility to contact other people; sometimes to have more information about a case, sometimes to say to someone that his case has been corrected.

We will discuss these three points in the next sections of this chapter.

3.2.2. Presentation of an interface

In this section, we will first categorize the information that a user may want to enter when he inserts a case. After that, it will be interesting to describe the different possible cases of the base.

3.2.2.1 Needed information

When someone presents an interface, he has to give more information than just the name of the case. People who will consult the case need to know the environment of the case, the windows that correspond to the interface, and so on. These information can be classified into two categories:

- *Data concerning the user.*
  The user gives his name, his E-mail address, ... He also has to specify his qualification in the domain of Human Computer Interface.

- *Data concerning the interface.*
  As we said before, the user can give a lot of information concerning his interface. He can make a video sequence concerning its use, he can describe how the case has been created, with what kind of tools, he can present the different windows of his interface, and so on.

The application that we will create must give the opportunity to the user to enter all these information. We will see later in this chapter how it is realized in our base. We also need to see the different kind of cases that can be put in our base. The next section describes the three possible cases.

3.2.2.2 Description of the different possible cases of the base

The different interfaces presented in the base can be gathered in three types of cases:

- Presented cases
- Presented and criticized cases
- Reference cases

Presented cases are cases that have not been submitted to a critique. Presented and criticized cases have at least one or more critiques made by a user of the base. Finally, reference cases are cases that have been created or corrected by the incumbent professor of the course.

Now, we will describe more in details each of these cases.

a) Presented case

This type of case is the simplest of the case base. During the insertion, the user will provide all necessary information about the case. He will also provide to other users screenshots of windows that he would want to see criticized.

The different information to be provided will be the following one's:

- Concerning himself: first name, last name, Email, qualification and comments.

- Concerning the interface: name, domain of application, type of user, type of case (here it is «Presented Case»), existence of a video sequence, correction of an existent case, comments about environment of the task, about the pedagogical approach and finally, free comments.

Once these information are provided, the case is inserted in the base and becomes available for the consultation. Each user is then able to submit his critique for this case.

b) Presented and criticized case

When a presented case is completed with a critique, it belongs to the second category of case. The addition of a critique for an interface can be undertaken during two moments: during the insertion of the case (the critique is made by the person who inserts the case) or during the consultation of the case (the critique is made by the consultant). Of course several users can criticize a same case. Notice also that, no guarantee is given concerning the precision and justification of these critiques. Indeed, novices can realize the critiques (nevertheless the user is warned about the level of knowledge of the person that proposed the critique).

Several types of critiques are possible:

- Intra-windows critique: critique about a specific window of the global interface.

- Inter-windows critique: critique about the comparison between two windows of the global interface.

- Global critique: critique about the global interface.

For more information about these different critiques, one can refer to the section 3.2.3 "Critique of an interface".

c) Reference case

The last type of available case in the base is the reference case. This one constitutes a subset of the type "criticized and presented case". A reference case is a case that has been created by the incumbent professor. So it has been verified and validated by him. This case respects rules and criteria that the professor has chosen to serve as reference to his course.

These cases can notably serve as an illustration for the course.

3.2.3. Critique of an interface

As we have already said before, three types of critique are possible for an interface:

- Intra-windows critique

- Inter-windows critique

- Global critique

These critiques are not exclusive but are truly complementary. Nevertheless, it is not necessary to provide inevitably each type of critique.

We will now present these three types of critique in details. Notice that the application has been designed to accept all types of critiques. It is organized to allow each user to criticize the interface according to his own desires and knowledge. This is why the critiques do not follow a formal form but are made essentially with unstructured comments.

3.2.3.1 Intra-windows critique

When the user chooses to make an intra-windows critique, the list of all available windows is given. This list is divided into two sub-lists (already criticized and non-criticized windows) to facilitate the work of the user. He must then choose the window he wants to criticize.

Once this choice is made, the user can see the illustration of the chosen window, a graphical tool and a text zone allowing him to put his critiques. The graphical tool serves to

annotate the illustration so that the comprehension of the future consultant will be facilitated. Its use is explained in the corresponding section 3.4.2 ("Graphical tool").

Two zones of text allow to criticize the window. The critique must concern the illustrate window. The different critiques will be automatically annotated with numbers corresponding to the different squares made (thanks to the graphical tool) on the illustration. Moreover, these critiques will be organized into negative and positive critiques thanks to different colors (green for positive one's and red for negative one's) and two different text zones.

3.2.3.2 Inter-windows critique

When the user chooses to make an inter-windows critique, the list of windows that can be criticized is proposed to him. He must then choose the two windows he wants to compare.

Once this choice is made, the user can see illustrations of the two selected windows, a graphical tool and a zone of text allowing him to give his critiques.

The critiques must be critiques that compare the two windows. The main points in this type of critique are the consistency between the two windows (for example, absence of a button in one of the windows compared with the other, uncoordinated style of handwriting,...). Once more, positive and negative critiques will be separated.

3.2.3.3 Global critique

When the user chooses to make a global interface critique, he obtains: a tool allowing him to follow the different windows that compose the interface (it allows the user to see each window at any given time of his critique), a video sequence of the utilization of the interface (if the case has one) and finally, a text zone for the positive critiques and another for the negative ones.

The waited critique in the text zone is a critique that relates to the set of windows of the interface and not to one particular window. One expects therefore here critiques like "the choice of the size of characters is not optimal and does not allow a good perusal",...

It is also a good place to make critiques concerning the environment of the interface or concerning the approach that has been employed to create the interface. All those things are not dependent of a particular window but rather from the choice made before the creation of the interface. Of course, it is only possible if the creator of the interface has given this type of information.

3.2.4. Comments

A user of the case base may want to have contact with others to discuss about the interfaces. For example he can discuss with the creator of the case to have more information about the conception or the interface environment. He can also discuss with other persons in order to have their opinion about his work.

The conception of these comment spaces does not depend on our application but rather on the Vesale Project. At this time, the comment spaces and dialog spaces have not yet been implemented but the used tools are the following ones:

- BSCW : BSCW (Basic Support for Cooperative Work) enables collaboration over the Web. BSCW is a 'share workspace' system which supports document upload, event , notification, group management, and so on.

  You can access this workspace at www.bscw.gmd.de .

- Email

You can have more information about the comment space of Vesale in this paper: « Visual user interface design Education Supported by a computer-Aided Learning Environment: specification document »[BODART and AL 99]

3.3. Task analysis

Now that we know the needs of the users, we can analyze the task in order to find the better way to realize our application. To do that, we will use the Trident method [BODART and AL 95] to find the procedures and the objects (presentation units, logical windows, ...) that we will have to implement. We will begin by the description of the task and the different procedures that compose it. After we will analyze the different sub-tasks thanks to the TKS method [VANDERDONCKT 97b]. We will end this chapter by the conception of the chaining graph of functions and the definition of the presentation.

3.3.1. Description of the case

3.3.1.1 Textual statement of the interactive task

The interactive task "Case insertion in the case base" can be decomposed in two main sub-tasks: the interface presentation and the interface critique. Points that follow describe completely these different sub-tasks.

a) Interface presentation

When a person wants to enter a case in the base, he has first of all to give information about the case to facilitate the comprehension of the other users. For the presentation of an interface, we can consider that there exists three different procedures: the two firsts are filled in by the person who inserts the case while the third is automatically accomplished to facilitate his work. The three procedures are the following one's:

- The user has to provide data concerning himself. We will call this procedure "Data concerning the user (données sur l'insérant)".

- The user has to provide general information concerning the interface he wants to present. We will call this procedure "Interface presentation (présentation de l'interface)".

- To facilitate the work of the user, the application will provide a summary of the different information concerning the case. We will call this procedure "Summary of the presentation (résumé de la présentation)".

We will now explain each procedure.

i) First procedure: "Data concerning the user (données sur l'insérant)"

When a user wants to propose an interface for the case base, he has to provide information concerning himself. These information will allow the other users to better know this user and to judge his level of expertise concerning the HCI domain.

Requested information concerning the user are: First name, Last name, E-mail address, Qualification and other free comments.

ii) Second procedure: "Interface presentation (présentation de l'interface)"

Once the user has provided some data concerning himself, he has to present his interface. He has to be precise enough to allow the other users to understand the case and he has to give all necessary elements for the critiques (screenshots, for example).

At this level, requested information are: Name of the interface, Domain, Type of the user, Type of the case, Presence of a video sequence, Correction of another case, Comments, Windows composing the interface.

iii) Third procedure: «Summary of the presentation (résumé de la présentation)»

This procedure is an assistance provided to the user. Once he has entered data concerning his interface, the application provides a summary of the information he has inserted up till now.

b) Interface critique

A user may want to criticize the interface that he observes. It is necessary to give him tools to criticize the case on different ways: in general, window by window or by interrelating two windows together. In the case of an interface, we can consider that there are five different procedures: three of them depend on the user while the two others help him in his task. The five procedures are the following one's:

- It is necessary to give the possibility to the user to choose the type of critique that he wants to make. This first procedure has to provide possible choices. We will call it: "Orientation of the critique (orientation de la critique)".

- The user may want to make a global critique. The second procedure helps him to insert his critique. We will call it: " Global critique (critique globale) ".

- The user may want to make an inter-window critique. The third procedure helps him to insert his critique. We will call it " Inter-window critique (critique inter-fenêtre)".

- The user may want to make an intra-window critique. The fourth procedure helps him to insert his critique. We will call it " Intra-window critique (critique intra-fenêtre)".

- To facilitate the work of the user, the application will provide a summary of all the information concerning the critique. We will call this fifth procedure " Summary of the critique (résumé de la critique)".

We will now explain each of these procedures.

i) First procedure: "Orientation of the critique (orientation de la critique)"

When a user wants to make a critique, it is necessary to help him to create this critique. The procedure of orientation of the critique lets therefore the user choose between a global critique, an inter-window critique or an intra-window critique. Outside of this choice, no other information is requested.

ii) Second procedure: " Global Critique (critique globale) "

Once the user has decided to make a global interface critique, the application has to help him to realize his critique. The user has to be able to emit positive and negative critiques concerning the interface. Moreover, it is necessary that he can choose keywords that will classify his critique among the other available ones. The second procedure collects these different information concerning the global critique.

iii) Third procedure: " Inter-window critique (critique inter-fenêtre) "

Once the user has decided to make an inter-window interface critique, the application has to help him to realize his critique. It is necessary first of all to allow the user to choose the two windows that he wants to compare. Then, he must have text zones and tools allowing him to make his critique. Finally, it is necessary that he can choose keywords that will classify his critique among the other available ones. The third procedure collects these different information concerning the inter-window critique.

iv) Fourth procedure: " Intra-window critique (critique intra-fenêtre) "

Once the user has decided to make an intra-window interface critique, the application has to help him to realize his critique. It is necessary first of all to allow the user to choose the window that he wants to criticize. Then, he must have text zones and tools allowing him to make his critique. Finally, it is necessary that he can choose keywords that will classify his critique among the other available ones. The fourth procedure collects these different information concerning the intra-window critique.

v) Fifth procedure: " Summary of the critique (résumé de la critique)"

This procedure is an assistance provided to the user. Once he has entered data concerning his critique, the application provides a summary of information that he has inserted up till now.

c) Relative parameters of the task

After examination of this textual statement, we are now able to determine relative parameters of the task.

- **Knowledge that a user must have:** average.
  The user must have some knowledge in the domain of interfaces to use the application and to criticize.

- **Productivity:** important
  The task consists of the insertion of case in a case base. It is therefore a task that will produce a lot of new examples of cases and critiques.

- **Objective Environment of the task:** Partially non-existent.
  The system itself is enough to realize the entire task except that the user needs a tool to capture the pictures of his interface and a disk to stock these screenshots.

- **Reproducibility of the environment:** practicable.
  The reproducibility of the environment is said practicable because it can be transposed in the framework of the system.

- **Structure of the task:** moderated.
  The task has been studied in order to leave the greatest possible freedom degree to the user: numerous optional fields, free comments, and so on. Despite that, a certain structure of the task has nevertheless been necessary.

- **Importance of the task:** important.
  This task allows to enter new cases in the base and to criticize them what is essential for the apprenticeship concerning the HCI.

- **Complexity of the task:** moderated.

3.3.1.2 Description of the population

The population of the users must contain the two following stereotypes: student and incumbent professor.

a) Student user

- **Experience of the task:** moderated.
  The student is a beginner in the domain of interface critique.

- **Experience of the system:** average.
  The student studies computer science and has therefore a certain experience of this type of system.

- **Motivation:** important.
  The student is motivated to present and criticize cases to improve his knowledge.

- **Experience of a complex interaction means:** rich.
  The student studies computer science and is therefore always confronted with this type of interaction means.

b) Incumbent professor user

- **Experience of the task:** rich.

- **Experience of the system:** rich.

- **Motivation:** important.

- **Experience of a complex interaction means:** rich.

3.3.1.3 Description of the environment

- **Type of processing:** multi processing
  A workstation can be used to realize different tasks or sub-tasks.

- **Capacity of processing (repetition of the task):** moderated.

3.3.2. Analysis of the sub-tasks

3.3.2.1 Choice of the analysis method

The structure of the interactive task being moderated, the analysis method of the task is the TKS method (Task Knowledge Structure) that consists of an extension of the hierarchical task planning[VANDERDONCKT 97b].

This method includes:

- the decomposition in goals and sub - goals. Each goal will be labeled by :
  - p if it concerns a (sub -)goal of preparation ;
  - t if it concerns a (sub -)goal of transformation ;
  - s if it concerns a (sub -)goal of selection ;

3.3.2.2 Case insertion in the base

a) Decomposition in goals and in sub - goals

0. case insertion (insertion d'un cas)
    0.1 Interface presentation (présentation de l'interface) (p)
    0.2 Interface critique (critique de l'interface) (p)

b) Diagram of goals and sub–goals

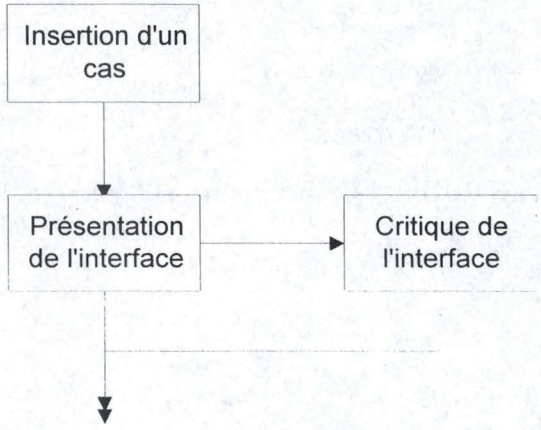


Figure 3-1: Diagram of the case insertion

3.3.2.3 Interface presentation

a) Decomposition in goals and in sub - goals

1. Interface presentation (présentation de l'interface)
    1.1 Data concerning the user (données sur l'insérant) (p)
    1.2 Information concerning the interface (renseignements sur l'interface) (p)
        1.2.1 Data concerning the interface (données) (p)
        1.2.2 Correction of another interface (correction d'une autre interface) (p)
        1.2.3 Existence of a video sequence (séquence vidéo) (p)
        1.2.4 Windows composing the interface (fenêtre composant l'interface) (p)
    1.3 Summary of the presentation (résumé de la présentation)(p)

b) Diagram of goals and sub-goals

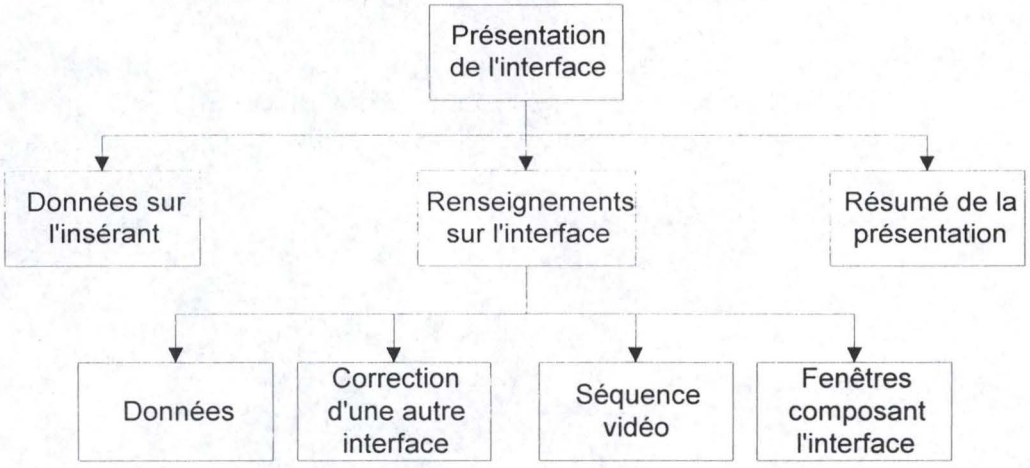


Figure 3-2: Diagram of the interface presentation

3.3.2.4 Interface critique

a) Decomposition in goals and in sub-goals

2. Interface critique (critique de l'interface)
    2.1 Orientation of the critique (orientation de la critique) (s)
    2.2 Choice of global critique (choix critique globale) (p)
        2.2.1 Global critique (critique globale) (p)
        2.2.2 Keywords (mots clés) (p)
        2.3 Choice of inter-window critique (choix critique inter)(p)
        2.3.1 List of windows to criticize (liste fenêtres) (p)
        2.3.2 Inter-window critique (critique inter) (p)
        2.3.3 Keywords (mots clés)^(p)
    2.4 Choice of intra-window critique (choix critique intra) (p)
        2.4.1 List of windows to criticize (liste fenêtres) (p)
        2.4.2 Intra-window critique (critique intra) (p)
        2.4.3 Keywords (mots clés) (p)
    2.5 Summary of the case (résumé du cas) (p)

b) Diagram of goals and sub-goals

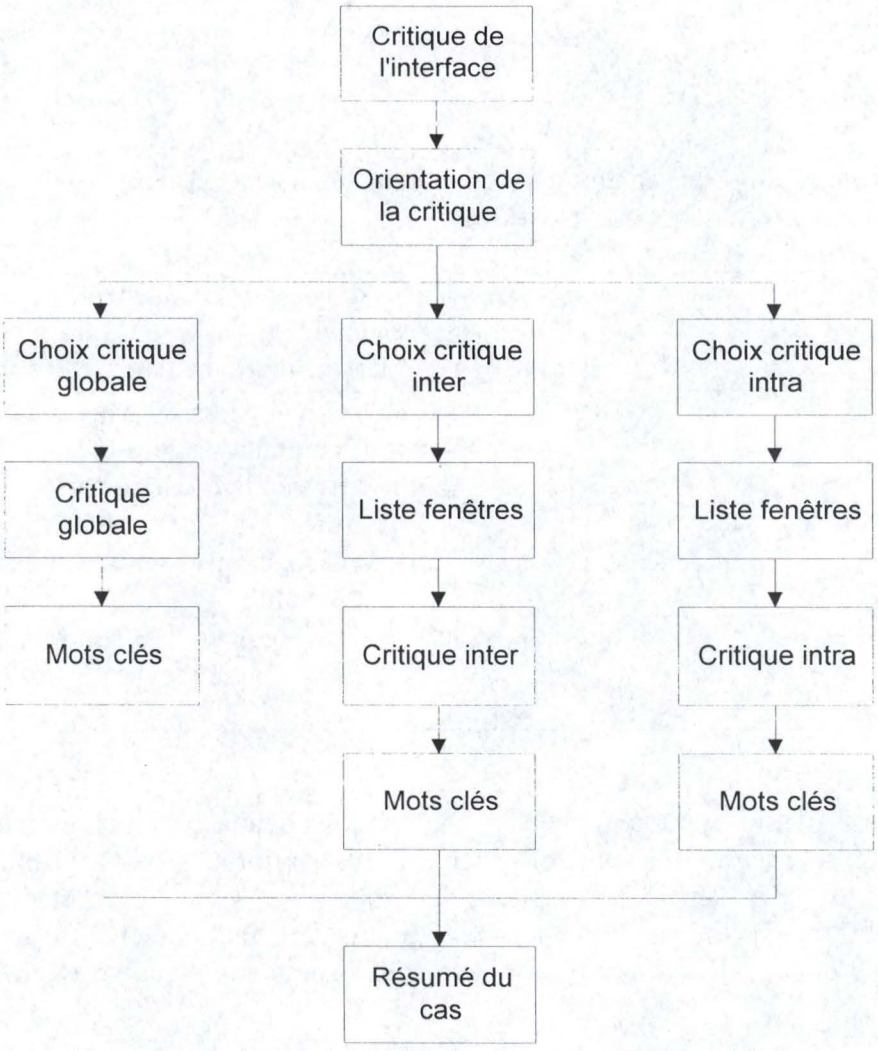


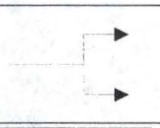
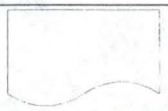
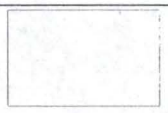
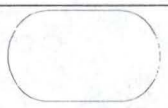
Figure 3-3: Diagram of the critique of an interface

Of course, when you have finished one critique, you don't have to follow all the process again. The application lets you make the number of critiques that you want. The summary of the case appears only when you have finished all your critiques.

3.3.3. Chaining graph of functions

3.3.3.1 Conventions

Before exposing the chaining graph of functions, we propose a table recapitulating graphic conventions employed for its writing.

	This drawing is used when there is a choice to make. The graph follows one of the two arrows
	This drawing is used when an information message is created between two functions
	This drawing is used to show the different functions or procedures of the graph
	This drawing is used to delimit a sub-graph in the graph

Because of the important size that the chaining graph of functions can reach, it is strong difficult to draw it on a sheet of paper with the A4 format. To avoid this problem, we have decided to divide it in sub - graphs. These sub - graphs are represented with rectangle with rounded form. The other conventions used in this table the same that the ones used in the HCI course. When it was necessary, we have added some comments to explain our various diagrams.

3.3.3.2 Chaining graph of functions

The chaining graph of functions can be decomposed in two sub-graphs, which correspond to the two sub-tasks analyzed previously. For reminder, it concerned the interface presentation and the interface critique.

The basis diagram of our graph is:

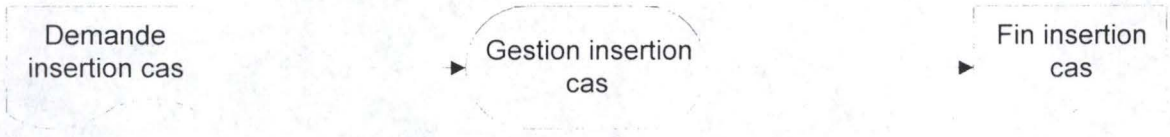


Figure 3-4: Main chaining graph of function

It the same diagram than the following one's:

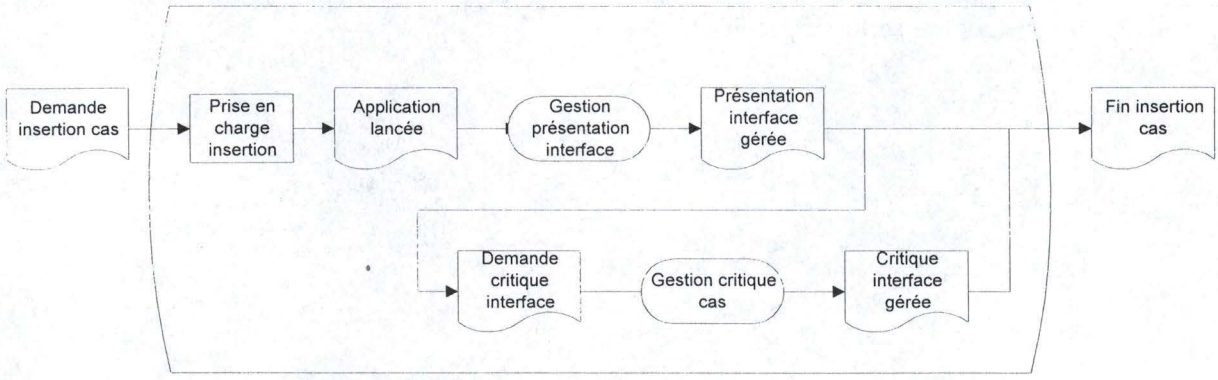


Figure 3-5: Detailed chaining grapf of function

Beside the management functions of the interface presentation and of the interface critique (that are described after), another function interferes in management of the case insertion. It concerns:

- *The plug in cost of the insertion* (prise en charge de l'insertion)
  When a person wants to insert a new case in the base, it is necessary first of all to download the application and to launch it.

a) Management of the interface presentation

The interface presentation has to be able to provide the following information:

- *Data concerning the user*

- *Information about the interface itself.*
  This function being complex enough, it will also be decomposed in several sub-functions.

In addition to that, it is useful to provide to the user a summary of information that he has inserted during this presentation. We have therefore added a function that creates this summary:

- *Summary of the presentation*

The sub-chaining graph of functions corresponding to the management of the interface presentation is thus the following one's:

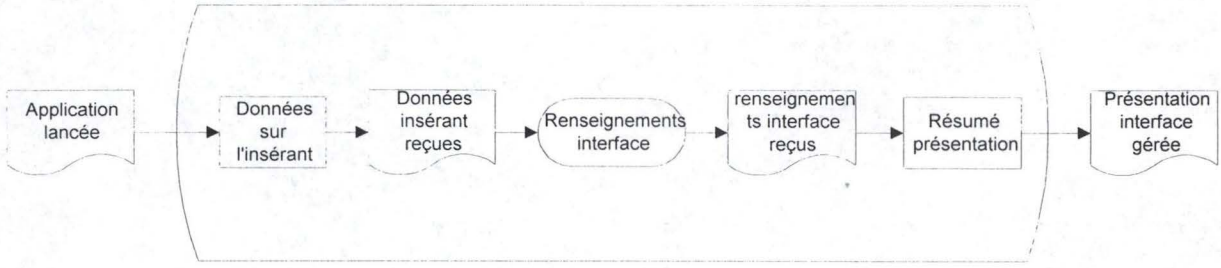


Figure 3-6: Functional chaining graph of the management of the interface presentation

i) Information concerning the interface

The function «information concerning the interface» serves to collect information on the interface. It is indeed necessary to store some information in order to be able to classify the interface in the base. One can decompose this function in four sub-functions:

- One sub - function that collects data on the interface. This one is called *«management about data concerning the interface»*.
- One sub - function that asks if the inserted interface is the correction of another interface that is in the base. This one is called *«correction management»*.
- One sub - function that manages the possible presence of a video sequence. This one is called *«video sequence management»*.
- One sub - function that manages the different windows of the interface. This one is called *«window management»*.

The sub-chaining graph of functions corresponding to the management of information of the interface is therefore the following one's:

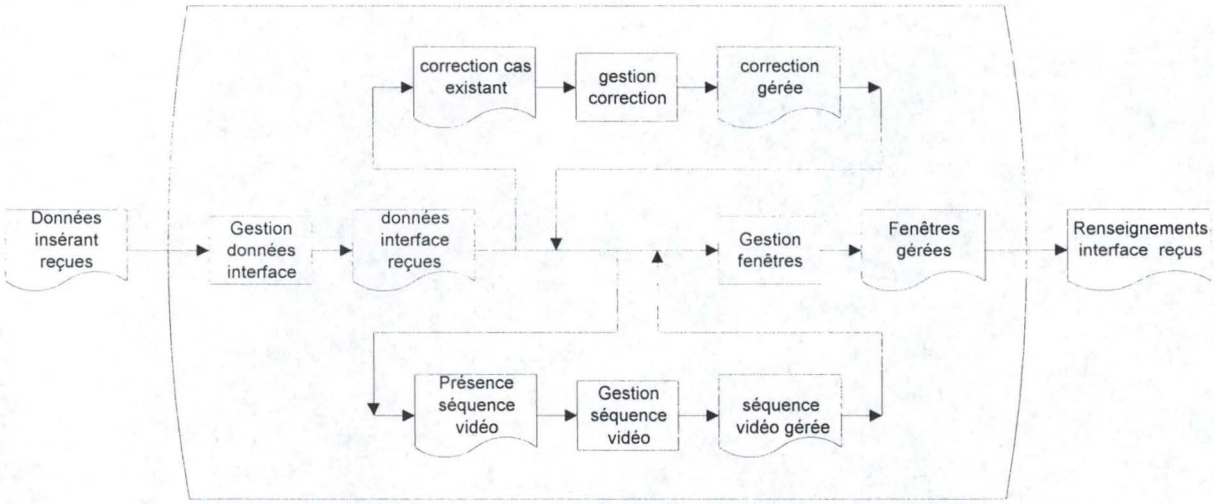


Figure 3-7: Functional chaining graph of the management of information of the interface

b) Management of the interface critique

The management of the interface critique can be divided in five sub-functions:

- One sub-function that orients the user to the type of critique of his choice. This one is called «*orientation management*».
- One sub-function that manages the global critique. It is called «*management of the global critique*».
- One sub-function that manages the inter-window critique. It is called «*management of the inter-window critique*».
- One sub-function that manages the intra-window critique. It is called «*management of the intra-window critique*».
- One sub-function that provides to the user a summary of information that he has entered during the insertion of this case. It is called «*management of the summary of the case*».

The sub-chaining graph of functions corresponding to management of the interface critique is therefore the following one's:

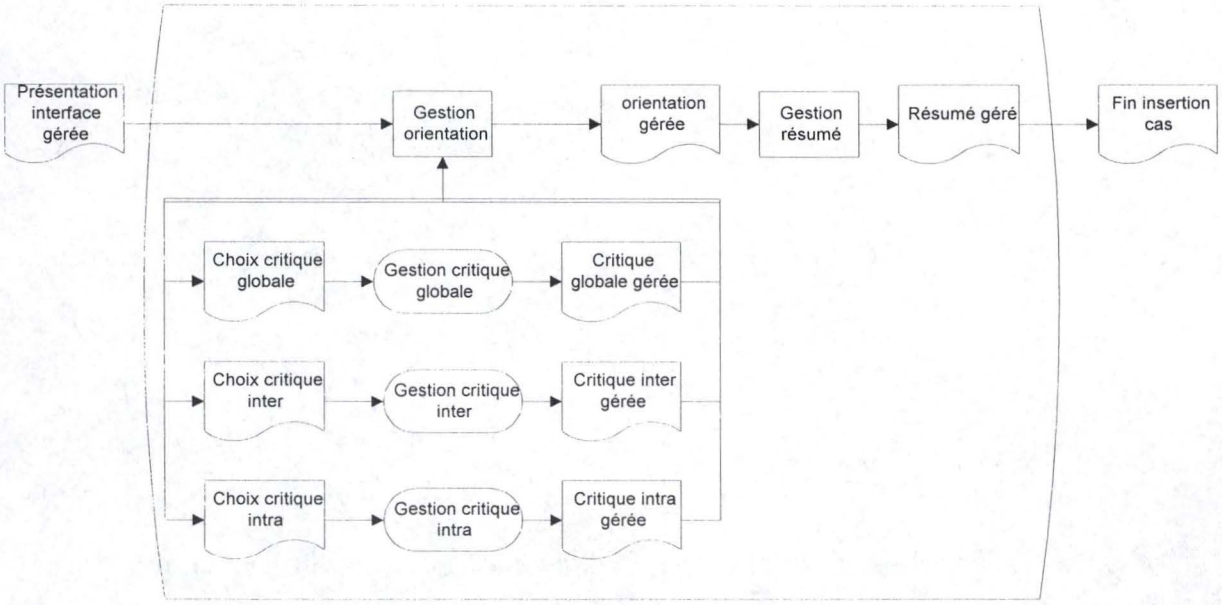


Figure 3-8: Functional chaining graph of the management of the interface critique

i) Management of the global critique

When the user chooses to make a global critique, it is necessary to help him in the realization of this critique. This management of the global critique is divided in two sub - functions:

- One sub-function that manages the critique. It is called «*global critique*».
- One sub-function that helps the user to choose representative keywords for his critique. It is called «*selection of the keywords*».

The sub-chaining graph of functions corresponding to management of the global critique is therefore the following one's:

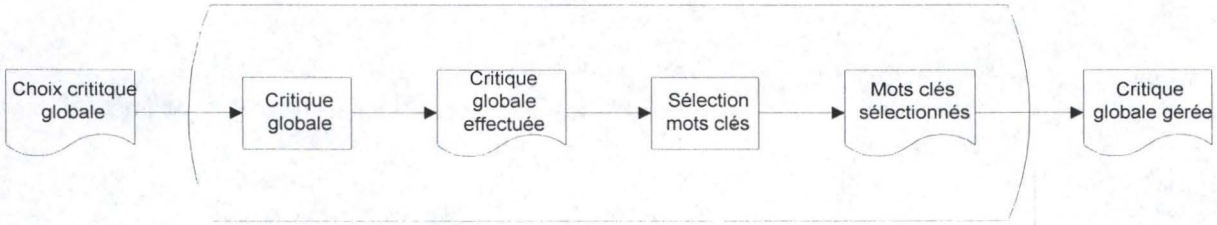


Figure 3-9: Functional chaining graph of the management of the global critique

ii) Management of the inter-window critique

When the user chooses to make an inter-window critique, it is necessary to help him in the realization of this critique. This management of the inter-window critique is divided in three sub-functions:

- One sub - function that allows the user to select the two windows in order to make the critique. It is called «*window selection*».
- One sub - function that manages the critique. It is called «*inter-window critique*».
- One sub - function that helps the user to choose representative keywords for his critique. It is called «*selection of keywords*».

The sub-chaining graph of functions corresponding to management of the inter-window critique is therefore the following one's:

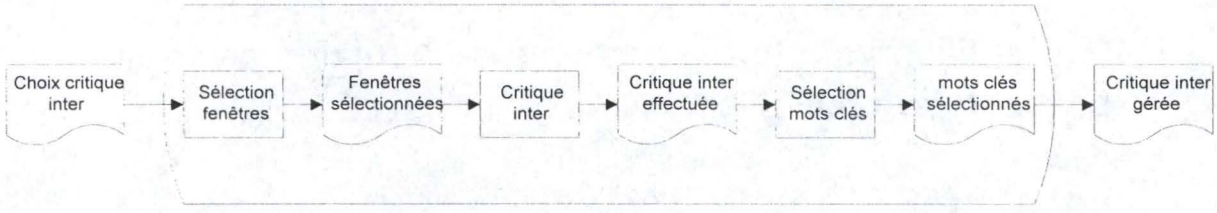


Figure 3-10: Functional chaining graph of the management of the inter-windows critique

iii) Management of the intra-window critique

When the user chooses to make an intra-window critique, it is necessary to help him in the realization of this critique. This management of the intra-window critique is divided in three sub-functions:

- One sub - function that allows to select the window in order to make the critique. It is called «*window selection*».
- One sub - function that manages the critique. It is called «*intra-window critique*».
- One sub - function that helps the user to choose representative keywords for his critique. It is called «*selection of keywords*».

The sub-chaining graph of functions corresponding to management of the intra-window critique is therefore the following one's:

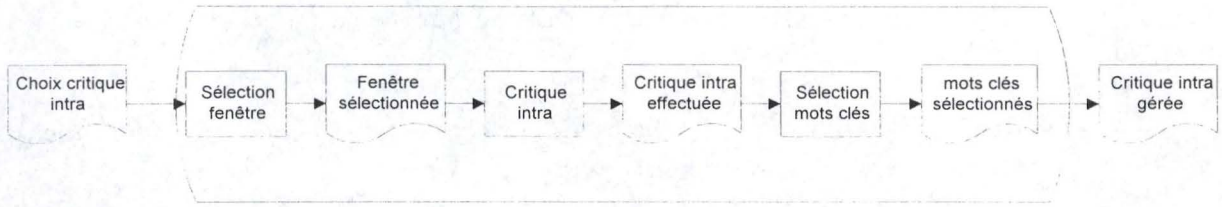


Figure 3-11: Functional chaining graph of the management of the intra-windows critique

3.3.4. Definition of the presentation

3.3.4.1 Identification of presentation units (unités de présentation (UP))

From the chaining graph of functions, presented in the another section, we can observe two presentation units. Each of these units corresponds to one sub - task. This is inspired by the methodological rule: "one UP by sub - task".

1. UP1: Interface presentation
2. UP2: Interface critique

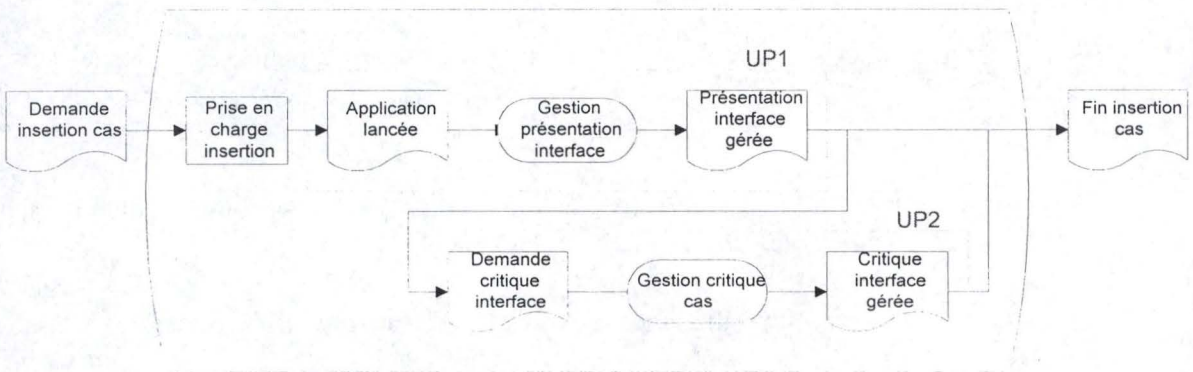


Figure 3-12: Functional chaining graph of both UP

3.3.4.2 Windows identification

a) Justification of the windows identification

From the chaining graph of functions, we can partition the graph of each UP in sub-graphs that materializes windows. It exists different identification criteria of windows, each determining decomposition more or less fine of the UP. Finer is the decomposition, more generally adequate for a novice user it is, because that structures increasingly the interface.

For the two UP, we have chosen an identification in / out. This identification consists in the gathering of all external information in entry in a window, and the gathering of all external information in exit in another window. The decomposition is compatible with the level of expertise of the user that, for reminder, is average. With this identification, we have attached the functional decomposition, which regroups in a simple window external information of each function. This roundup is also compatible with the level of expertise of the users.

b) Logical windows of UP1

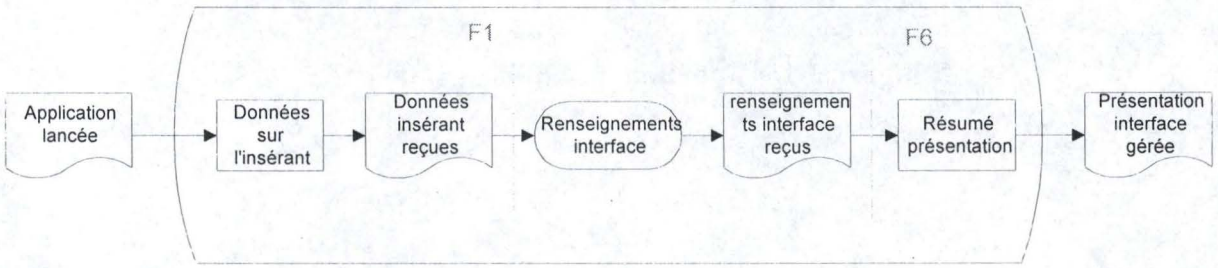


Figure 3-13: Functional chaining graph of UP1

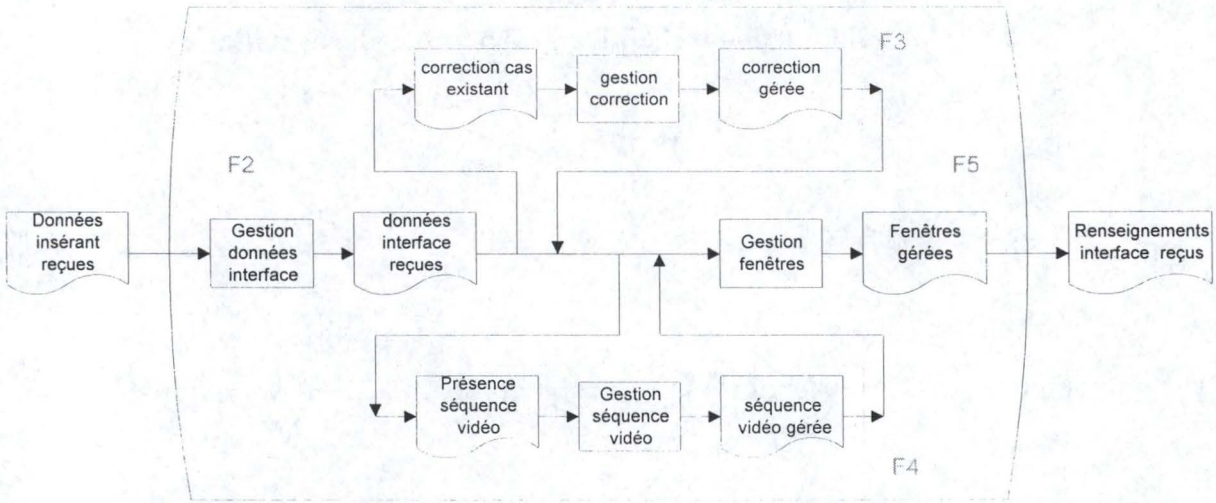


Figure 3-14: Functional chaining graph of the interface information

c) Logical windows of UP2

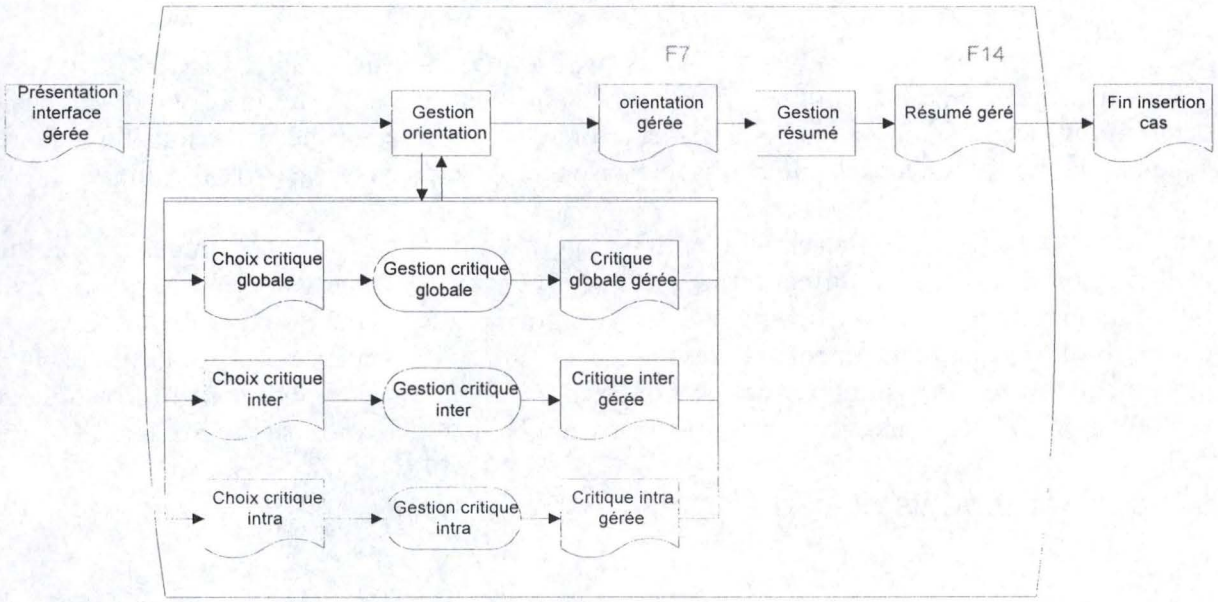


Figure 3-15: Functional chaining graph of UP2

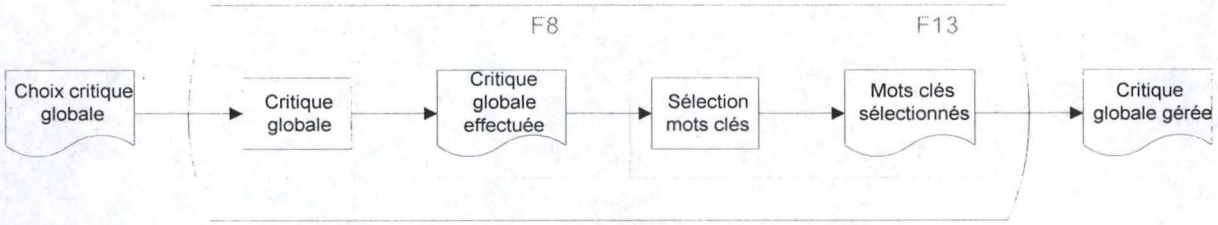


Figure 3-16: Functional chaining graph of the global critique

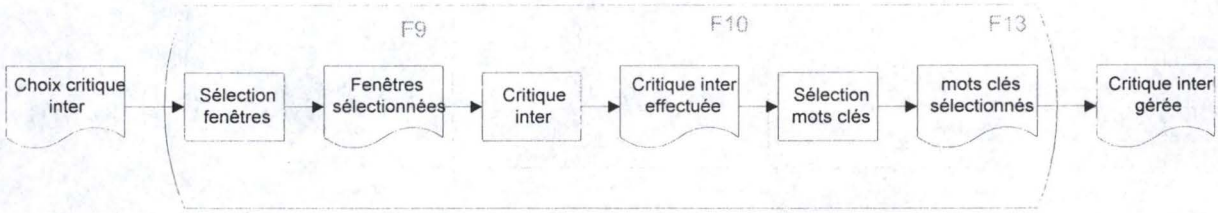


Figure 3-17: Functional chaining graph of the inter-windows critique

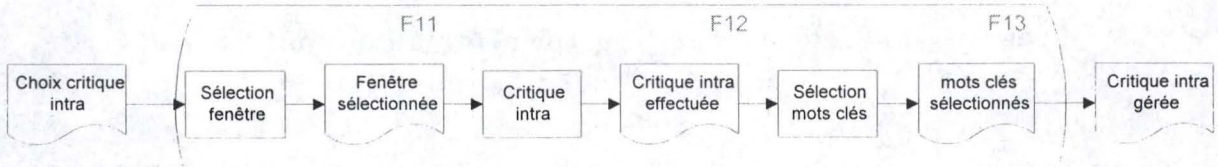


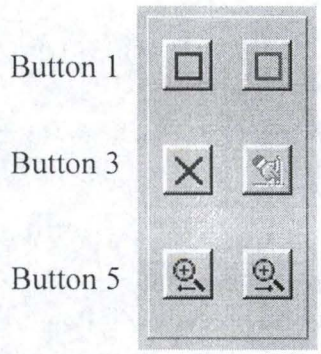
Figure 3-18: Functional chaining graph of the intra-windows critique

3.4. Insertion process

3.4.1. Introduction

Now that we know exactly what the user whishes and how to realize it, we can describe our application. We will begin by the graphical tool we have created to help the user to mark zones on interfaces. After that we will give the insertion scenario. To end the section, we will describe in details a window of our application. The other windows will be examined in a chapter of the annexes.

3.4.2. Graphical tool



The graphical tool serves to annotate the illustration of a window (during the intra- or inter-windows critique) with the aim to facilitate the comprehension of the future consultant.

The use of this tool is relatively easy thanks to different buttons.

- Button 1: This button allows to draw a red rectangle on the illustration. This rectangle serves to delimit a positive critique zone.
- Button 2: This button allows to draw a green rectangle on the illustration. This rectangle serves to delimit a negative critique zone.
- Button 3: This button allows to mark a zone of the illustration with a cross. This allows to signal that an element is missing in the interface.
- Button 4:This button allows to suppress a selected rectangle or cross.
- Button 5: This button allows a zoom on the illustration that displays then in full mode screen.
- Button 6: This button allows a zoom on the second illustration during of an inter-windows critique. Note that this button is not present for the graphical tool linked to the intra-windows critique.

We will see now how to use this tool in the case of an intra- or inter-windows critique.

3.4.2.1 Use of the tool to realize an intra-windows critique

Remind that after selection of the window to criticize, the user can see the illustration of the window, the graphical tool and a zone of text.

The critique of the window is made following this way: with the graphical tool, the user draws a rectangle on the illustration. This rectangle includes a number. Automatically, with the creation of the rectangle, the same number appears in the corresponding critique zone (left - or positive critique zone - if the drawn rectangle is green, right - or negative critique zone - if the drawn rectangle is red). The user has then just to add his critique aside the corresponding number. His comment will be red if it is in the negative zone and green otherwise.

3.4.2.2 Use of the tool to realize a inter-windows critique

Remind that after selection of the two windows to criticize, the user can see the illustration of the two windows, the graphical tool and a zone of text.

The inter-window critique is made following this way: with the graphical tool, the user draws two rectangles. The first on the left illustration, the second on the right illustration. These two rectangles serve to select a zone on each window to be able to compare them. It is important to draw a rectangle on each illustration because it is well an inter-windows critique. Notice also that the two rectangles must to be of the same color.

Here again, when the two rectangles are drawn, they include each the same number. This number appears also in the corresponding text zone: left for green rectangles and a positive critique, right for red rectangles and a negative critique.

He can also draw crosses on only one window to mark, for example, a missing element. These crosses are "numbered" thanks to letters.

3.4.3. Scenario for the case insertion

In this section, we will explain the step that the user has to follow to insert a case in the case base. We suppose that the user is already on the web site of the Vesale project.

3.4.3.1 To access to the case base

Once the user is on the web site, he's able to access to the case base. In order to do that, he has to follow links that allow him to join the web pages relative to the case base. There, the user has to choose between the different possibilities of the base. It will be able therefore:

- To consult the case base
- To insert a new case in the case base
- To leave the case base

Concerning this section, he has to follow the link " To insert a new case in the case base".

3.4.3.2 Delphi Application

To insert a case in the base, the user has to download a Delphi application. This application allows him to store the different data concerning his case as well as the various information that he wants to propose to the other users: video sequence, windows composing the interface, and so on. It also allows him to make critiques concerning the case that he wants to insert (if nevertheless he desires it).

The user has the possibility to download this application by clicking on the hypertext link in the web page concerning the insertion of cases.

Once the case is inserted (thanks to the application), the user returns on the web page concerning the insertion of cases. He can then continue to insert new cases in the base, or return on the welcome page of the case base.

3.4.4. Interface presentation structure

We will now present in details a particular window. Less detailed description of other windows of this application is provided in appendix 6.

We decided to present the "Critique intra-fenêtre" window. This critique will be divided into three points: AIO transformation into CIO, screenshot and finally a description of the different components of the window.

3.4.4.1 AIO transformatio n into CIO

a) Set of "static AIO" used

AIO	CIO
Label	TLabel
Prompt	":"
Group Box	TGroupBox
Icons	Timage

b) Set of "scrolling AIO" used

AIO	CIO
Scrolling Bar	TScrollBar (predefined in Extended Edit Box)

c) Set of "action control AIO" used

AIO	CIO
Draw Button Tool Bar	TBitBtn TspeddButton + TGroupBox

d) Set of "information control" AIO

AIO	CIO
Edit Box Extended Edit Box	TEdit TMemo

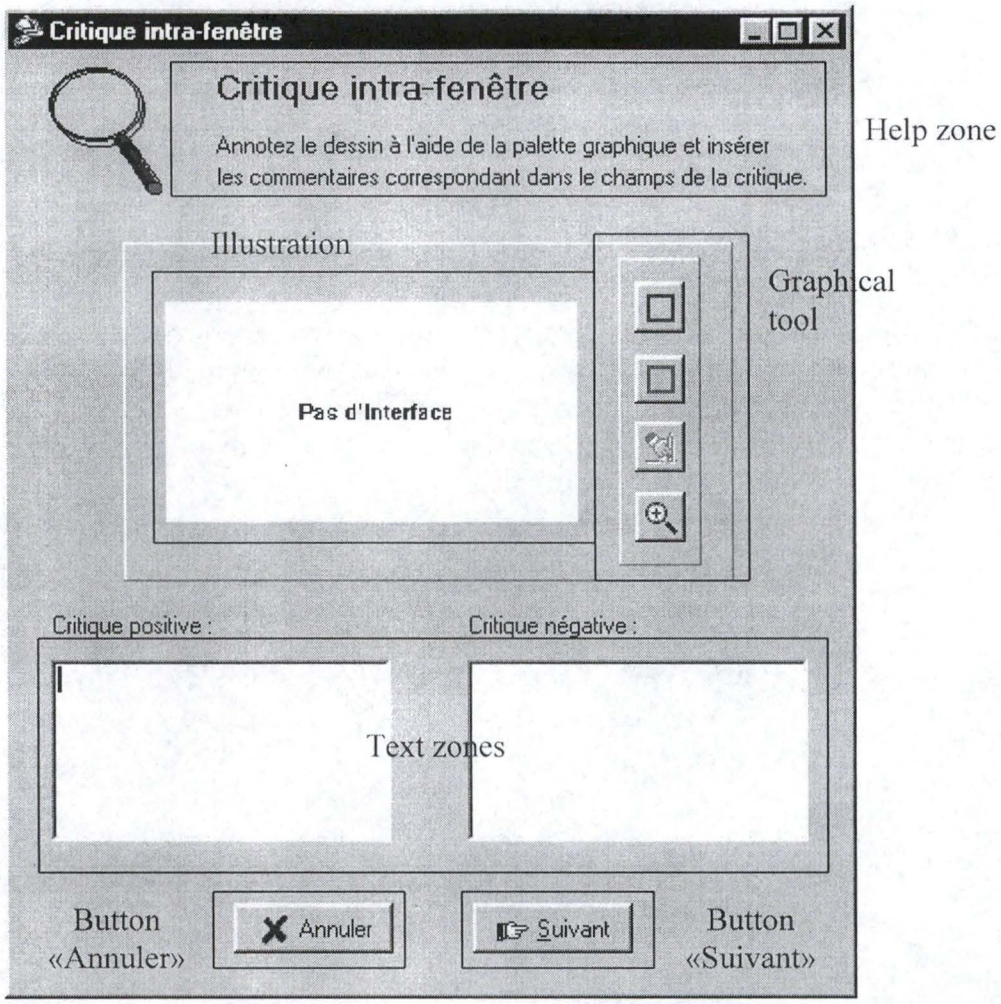
e) Set of "dialog AIO" used

AIO	CIO
Window	Tform

f) Set of "feedback AIO" used

AIO	CIO
Contextual cursro	Cursor property of TForm

3.4.4.2 Screenshot of the window "Critique intra-fenêtre"



3.4.4.3 Description of the different components

This window allows the user to make an intra-window critique of the selected window.

The window includes following elements:

- Graphical Tool: this tool allows the user to draw rectangles on the illustration of the window, to enlarge this illustration, and so on. The utilization of this graphical tool is explained in the section 3.4.2.
- Illustration: the user can see the illustration of the window that he is criticizing.
- Text zones (positive critique and negative critique): thanks to two Memos, the user can comment his intra-windows critique. This field is mandatory.
- Button «Suivant»: this button allows to go to the window "Mots clés".

- Button «Annuler»: this button cancels the intra-windows critique and allows the user to return to the preceding window, that is to say the window " Choix de la critique ". Information concerning this critique are not then saved.

- Help zone: this is a small comment about the window and about what user has to do.

3.5. Conclusion of the chapter

In this chapter, we have described how we have undertaken the task of insertion. We began by searching the needs of the users. Then we analyzed the task in terms of these needs. Afterwards we were able to propose solutions and to construct our application. The same sequence will be followed in the next chapter concerning the second main task of the case base : the case consultation.

Chapter 4:
Case consultation

4.1. Introduction

Now that we have analyzed the first task of our base (the case insertion), we will discuss about the second main task: the case consultation. The user must have the possibility to find cases in the base and to consult them. In order to do that, he needs powerful research tool and well designed pages.

We will begin this chapter with the study of the objectives that a user could have when he wants to consult: what he wants to do, what he needs, and so on.

The second section will be use to make the analysis of the task. We will describe the task environnement and the different procedures that must be implemented.

To end this chapter, the consultation process will be explained. We will give a scenario for the consultation case, we will describe the search engine and give some information about what must be implemented in the future.

4.2. User objectives for a consultation

4.2.1. Introduction

What wants a user when he consults the case base? We suppose that he could want three things :

- He wants to search cases that have some properties. To do that, he needs **a search engine**.
- He wants **to consult a particular case**. In other words, he wants to read and see all information that the case contains.
- He wants **to consult a particular case and to add a critique to this case**.

These three points will be explained in the next sections of this chapter.

The users may also want to discuss with each other about the different cases but as for the case insertion, the dialog space and comments space that must be created don't depend on our work. To have more information about it, you can consult the Vesale paper provided in appendix 1.

4.2.2. Need of a search engine

To help the user to find cases in the base, it is necessary to implement a search engine. With that tool, the user will be able to give some information to the engine and will receive a list of relevant cases present in the base.

The search engine isn't a simple improvement but rather a real necessity for the base. If the base is too big, the user will be lost and not able to find the cases he really wants. The introduction of a search engine helps him to find cases but also to improve his theoretical knowledge because if he doesn't understand the theory, he will not be able to make good requests to the engine.

The search engine must have two main qualities. First, it must be powerful enough to offer a great choice of requests to the user. Secondly, it must be simple enough to facilitate the task of the novice user.

4.2.3. The case consultation

The main part of this task consists in the simple case consultation. When a case has been introduced in the base, it must be possible for users to consult it. The users must have access to all the information concerning this case: the video sequence, the different windows, the critiques, and so on. The presentation of the cases must be clear enough to facilitate the work of the user. It must also be adaptative to improve its usability.

4.2.4. The critique of a consulted case

A user must be able to criticize someone else's case. Thus, during the consultation, a system must be created to let the users criticize the case. In fact, this system will be a Delphi application. This application will be approximatively the same as the one created for the case insertion. We will see later the relations between the two applications.

4.3. Task analysis

Now that we know needs of the users, we can analyze the task in order to find the better way to realize the consultation. To do that, we will use the Trident method [BODART and AL 95] to define and analyze the task. We will describe the task and the different procedures that compose it. After we will analyze the different sub-tasks thanks to the TKS method [VANDERDONCKT 97b].

4.3.1. Description of the case

4.3.1.1 Textual statement of the interactive task

The interactive task " Case consultation in the case base" can be decomposed in three main sub-tasks: the selection of a case in the base, the consultation of the selected case and the critique of the selected case. Points that follow describe completely these different sub-tasks.

a) Selection of a case in the base

When a user wants to use the case base to consult (or to criticize) an existing case, the first thing he has to do is to find the wanted case. The selection of a case helps thus the user to realize this task. It contains some procedures that will facilitate the work of the consultant. In order to complete this selection task, three procedures are necessary: the consultant realizes the first and the third ones. The second one is automatically generated by the system. The three supposed procedures are:

- The user has to explain the kind of case that he is searching. He wants for example the complete list of the cases in the base (to obtain it, he just has to click on a button) or he wants a particular kind of cases (to obtain these cases, he has to use the search engine). The procedure that controls these functions is called «Research of the relevant cases (recherche des cas pertinents)».

- The system has to give to the user the list of the relevant cases found in the base (or the complete list of the cases if it was the choice of the user). The procedure that controls this part of the selection is called «list of the cases (liste des cas)».

- When the user obtains the list of the relevant cases, he has to choose between all cases the one that he wants to consult. The procedure that controls this selection is called «choice of the case (choix du cas)».

We will now explain in details each of the procedures.

i) First procedure: «Research of the relevant cases (recherche des cas pertinents)»

When a user wants to use the case base to consult different cases, he first has to give information about the cases that he is looking for. Two possibilities are offered. The first one gives to the user the entire list of the cases of the base. The user just has to click on a button called «complete list» to launch the function. Another possibility is to use the search engine. With this tool, a user can for example search a case called «Microsoft Word» or search all the cases in the base that contain a groupbox. The search engine tool will be developed in details in another section of this chapter.

The different information asked to the user in this procedure are the information of the search engine.

ii) Second procedure: «list of the cases (liste des cas)».

This procedure is only necessary to give to the user the list of the relevant cases that the search engine has found or the complete list of the cases in the base. This list of cases is presented with the name of the case and his creator. The name of the case is a hypertext link that lets the user access to the case.

iii) Third procedure: «choice of the case (choix du cas»).

Now that the user has the list of the different relevant cases that he needs, he has to choose the case that he wants to consult. In order to do that, he just has to click on the hypertext link corresponding to the case he wants to consult. Once the choice is made, the user has access to the case and the different information concerning it (video sequence, critiques, windows of the interface,...). The first sub – task «selection of the case» is then finished.

b) Consultation of the selected case

At this moment, the user has access to a determined case. This second sub–task has to give him the possibility to consult all the information of the case. So, different procedures will be in charge of these information. Five procedures are necessary to do that:

- A procedure that lets the user seeing a summary of the case. This procedure is called «summary of the case (résumé du cas)».
- A procedure that lets the user seeing the presentation of the case. This procedure is called «presentation of the case (présentation du cas)».
- A procedure that lets the user seeing the video sequence. This procedure is called «video sequence (séquence vidéo)».
- A procedure that lets the user seeing a list of the cases that correct this one and the cases that are corrected by this one. This procedure is called «correction (correction)».
- A procedure that lets the user seeing the different critiques of the case. This procedure is called «critiques of the case (critiques du cas)».

We will now explain in details each of the procedures.

i) First procedure: «summary of the case (résumé du cas)».

One of the most interesting things that the user wants to see is a summary of the case he consults. This procedure will furnish to the user a summary of the information presented in the case: the name of the case, the domain, the number of critiques for each type, the presence of a video sequence... That is the first page that will be presented to the user in order to let him decide if the case really interests him or not.

ii) Second procedure: «presentation of the case (présentation du cas)».

If the user thinks that the case is really interesting, he can now have access to all the information concerning it. The first thing he receives is a presentation of the case: why was the case created, which programs have been used, what users are aimed for... The different windows of the interface are shown, the explanation of its use, every information that the person who inserts the case had furnished is presented here.

iii) Third procedure: «video sequence (séquence vidéo)».

If the person who has inserted the case had furnished a video sequence, this procedure is used to let the consultant see this sequence. The consultant accesses to this function by clicking on a button.

iv) Fourth procedure: «correction (correction)».

It is possible that the presented case is in fact the correction of another case. Or, on the opposite side, that the presented case has been the object of a correction. This procedure gives to the user the list of the cases related to the consulted one by a relation of correction. The user has access to this function by clicking on a button.

v) Fifth procedure: «critiques of the case (critiques du cas)».

The consulted case can already have some critiques (intra-window, inter-window or global). This procedure gives to the user the possibility to see these critiques.

c) Critique of the consulted case

When a user has selected a case (and maybe consulted it), he may want to criticize it. As for the case insertion, he will have the choice between an intra-windows critique, an inter-windows critique or/and a global critique. When his critique is realized, the user has also to choose representative keywords. In fact, this sub-task is the same that the one developed for the task «case insertion in the base». So, the fifth procedure for this sub-task will be the same. We remember here the name of these procedures:

- "Orientation of the critique (orientation de la critique)".
- "Global critique (critique globale)".
- "Inter-window critique (critique inter-fenêtre)".
- "Intra-window critique (critique intra-fenêtre)".
- "Summary of the critique (résumé de la critique)".

The description of these procedures has been made in the chapter three.

d) Relative parameters of the task

After examination of this textual statement, we are now able to determine relative parameters of the task.

- **Knowledge that a user must have:** average.
  The user must have some knowledge in the domain of interfaces to use the application and to criticize.

- **Productivity:** average / important.
  The task consists of the consultation of case in a case base. The users who will only look at the different cases will not be very productive but the ones who will add some critiques to the existing cases will improve the case base.

- **Objective Environment of the task:** non-existent.
  The system itself is enough to realize the entire task.

- **Reproducibility of the environment:** practicable.
  The reproducibility of the environment is said practicable because it can be transposed in the framework of the system.

- **Structure of the task:** moderated.
  The task has been studied in order to leave the greatest possible freedom degree to the user: numerous optional fields, free comments, and so on. Despite that, a certain structure of the task has nevertheless been necessary.

- **Importance of the task:** important.
  This task allows to enter new critiques in the base and to consult cases what is essential for the apprenticeship concerning the HCI.

- **Complexity of the task:** moderated.

4.3.1.2 Description of the population

The population of the users will be the same as the one studied for the other main task « case insertion in the base ».

4.3.1.3 Description of the environment

- **Type of processing:** multi processing
  A workstation can be used to realize different tasks or sub-tasks.

- **Capacity of processing (repetition of the task) :** moderated.

4.3.2. Analysis of the sub-tasks

4.3.2.1 Choice of the analysis method

The choice of the analysis method is again the TKS method [VANDERDONCKT 97b].

4.3.2.2 Case consultation in the case base

a) Decomposition in goals and in sub - goals

0. case consultation (consultation d'un cas)
    0.1 selection of a case (sélection d'un cas) (s)
    0.2 consultation of a selected case (consultation d'un cas sélectionné) (p)
    0.3 critique of a selected case (critique d'un cas sélectionné) (p)

b) Diagram of goals and sub – goals

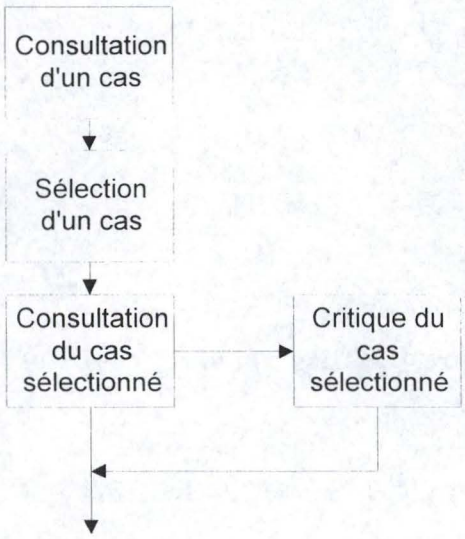


Figure 4-1: Diagram of the case consultation

4.3.2.3 Selection of a case

a) Decomposition in goals and in sub – goals

0.1 selection of a case
- 0.1.1 research of the relevant cases (recherche de cas pertinents) (p)
- 0.1.2 list of the cases (listes des cas) (p)
- 0.1.3 choice of a case (choix du cas) (s)

b) Diagram of the goals and sub – goals

Figure 4-2: Diagram of the choice of a case

4.3.2.4 Consultation of a selected case

a) Decomposition in goals and in sub – goals

0.2 consultation of a selected case (consultation d'un cas sélectionné)
- 0.2.4 summary of the case (résumé du cas) (p)
- 0.2.5 presentation of the case (présentation du cas) (p)
- 0.2.6 video sequence (séquence vidéo) (p)
- 0.2.7 correction (correction) (p)
- 0.2.8 critiques of the case (critiques du cas) (p)

b) Diagram of the goals and sub – goals

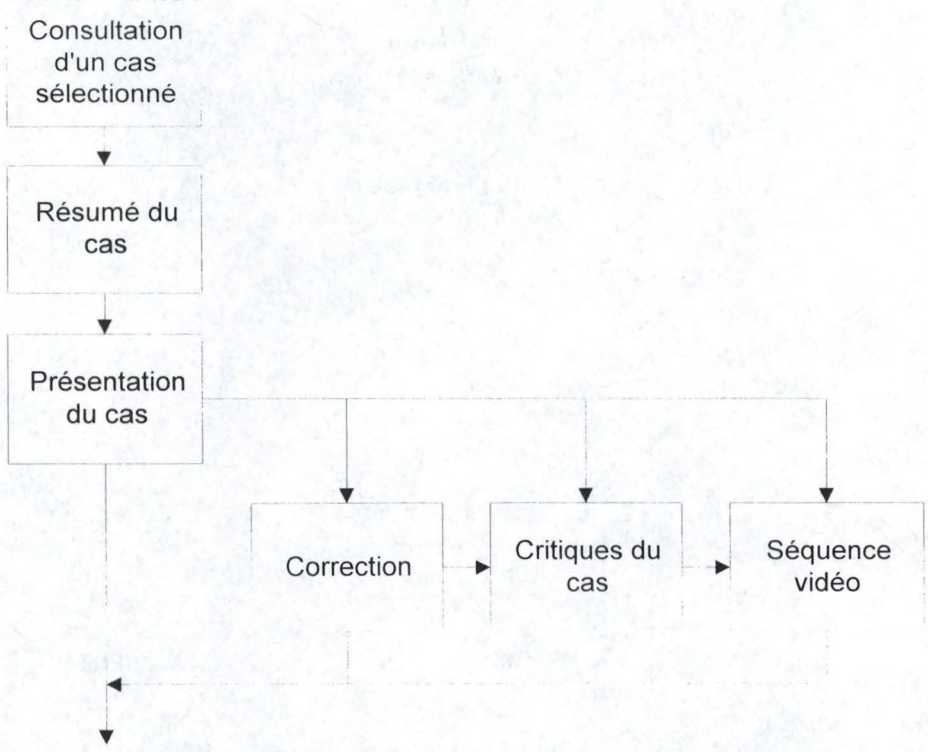


Figure 4-3: Diagram of the consultation of a selected case

4.3.2.5 Critique of a selected case

a) Decomposition in goals and in sub – goals

0.3 critique of a selected case (critique du cas selectionné)
- 0.3.1 Orientation of the critique (orientation de la critique) (s)
- 0.3.2 Choice of global critique (choix critique globale) (p)
  - 0.3.2.1 Global critique (critique globale) (p)
  - 0.3.2.2 Keywords (mots clés) (p)
- 0.3.3 Choice of inter-window critique (choix critique inter)(p)
  - 0.3.3.1 List of windows to criticize (liste fenêtres) (p)
  - 0.3.3.2 Inter-window critique (critique inter) (p)
  - 0.3.3.3 Keywords (mots clés) (p)
- 0.3.4 Choice of intra-window critique (choix critique intra) (p)
  - 0.3.4.1 List of windows to criticize (liste fenêtres) (p)
  - 0.3.4.2 Intra-window critique (critique intra) (p)
  - 0.3.4.3 Keywords (mots clés) (p)
- 0.3.5 Summary of the case (résumé du cas) (p)

b) Diagram of the goals and sub – goals

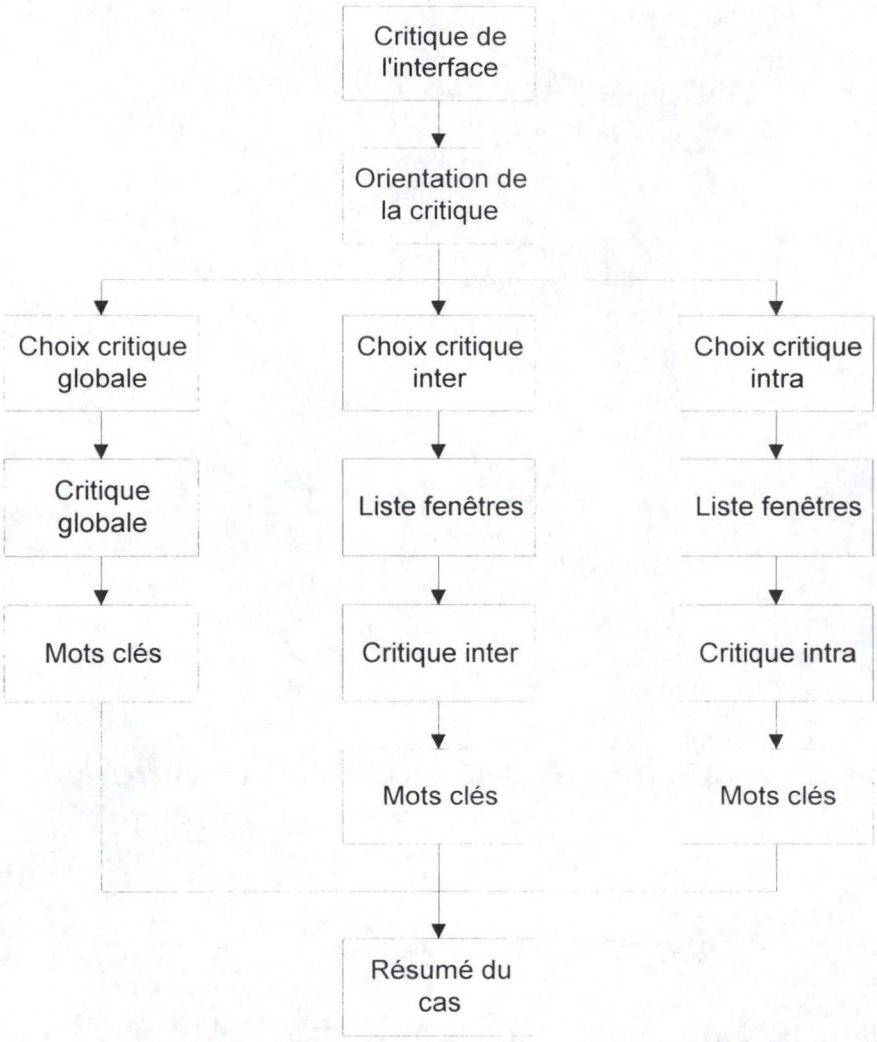


Figure 4-4: diagram of the critique of an interface

4.4. Consultation process

4.4.1. Introduction

Now that we know exactly the objectives of the users and how to realize them, we can speak about the consultation process. The pages concerning the consultation and the search engine have not been implemented. However, we will give some information about this engine to help the persons who will implement it. After that, the consultation scenario is proposed. The end of the chapter explains the work that has to be done later.

4.4.2. Search engine

Now that we have analyzed the task of consultation for our case base, it is time to give some information about the search engine. The results that the search engine must offer to the user have already been discussed in chapter two (indexation of the case base, 2.3.2.6). In this section, we will describe how the search engine should work and we will wake a summary of the results that it should give.

4.4.2.1 Keywords

It is obvious that the search engine must give to the consultant the possibility to use the keywords to find relevant cases in the base. Let's remember the different keywords that could index a case (or critique) in the base:

- the family of criteria will have keywords : Compatibility, consistency, work load, adaptability, dialog control, representativity, guidance, errors management.

- the family of generic AIO will have keywords : for the static AIO (label, separator, group box, prompt, icons), for the scrolling AIO (scroll arrows, slider, scroll bar, thermometer, frame), for the action AIO (menu, menu item, action bar, pull-down menu, pop-up menu, cascade menu, submenu, embedded menu), for the control AIO (control AIO of action, control AIO of information), for the dialog AIO (window, help window, logo window, textual publishing window, graphic publishing window, dialog box, expandable dialog box, repetitive dialog box, file open dialog box, panel, control panel), for the feedback AIO (information message, warning message, help message, action message, progress indicator, pointer)

4.4.2.2 Added words

We have added some other words that could be very helpful for a good research. These words are:

- The name of the person who inserts the case
- The name of the interface
- The domain of the interface
- The type of the case (presented, presented and criticized or case of reference)
- The date of the critique.

4.4.2.3 How will it work ?

The search engine will be a form that the user fills in with the information he wants to find in the different cases. The form will be separated into three parts: the first part will concern the added words (name of the interface, type of cases...), the second part will concern the family of design criteria and the third part will concern the family of generic AIO.

So, when a user wants to find a case in the base, he just has to fill in the form, the engine searches in the base and proposes to the user the cases that matches with the requirements. Of course, he can choose what he wants as keywords, there is no obligation to fill in all the parts of the search engine. On the opposite side, the user can make very precise researches. The different parts of the engine and the proposed keywords can be combined to be as precise as possible.

The results of the search engine will be given on a new web page. In fact, the page will contain the list of relevant cases. A simple click on a hypertext link will let the user see the selected case.

4.4.2.4 Example of a result

A student wants to find all the presented and criticized cases that contain critiques about the compatibility criterion. So, he must select the reference cases in the listbox concerning the **type of the case** and he must also click on the checkbox of the **compatibility criterion**.

After the research in the base, the engine will give the following results:

1. Microsoft Excel (Gaëtan Prévot)
   Intra-window critique (Rudy Michiels)
   Inter-window critique (Olivier Davreux)
   Intra-window critique (Patrick Kreps)

2. Example 1 (Steve Mihy)
   Global critique (Olivier Davreux)

So, we can see that the search engine gives the following results:

- the name of the case
- the name of the person who has inserted the case
- the relevant critiques
- the name of the persons who made the critique

4.4.3. Scenario of the case consultation

In this section, we will explain steps that the user has to follow to consult the case base. We suppose that the user is already on the web pages of the Vesale project.

4.4.3.1 To access to the case base

Once the user is on the web site, he's able to access to the case base. In order to do that, he has to follow links that allow him to acces to the web pages relative to the case base. The user has to choose between the different possibilities of the base. It will be able therefore:

- To consult the case base
- To insert a new case in the case base
- To leave the case base

Concerning this section, he has to follow the link " To consult the case base".

4.4.3.2 Search engine

The user has now access to a search engine. This engine gives him the possibility to research at best cases of the base. It has been created in JAVA and is therefore directly available in the web page. It is presented as a form to fill as we explain it in another section of this paper.

Once the form is filled, the search engine questions the case base and provides to the user relevant cases according to his request. Notice that the user can also consult the complete list of available cases in the base. In order to do that, he has to click on the button " complete list " (that will provide the list of all cases of the base) or not to fill fields of the search engine form.

At this moment, the consultant has the list of cases corresponding to the different criteria of his request. The list of cases appears therefore in a web page and each case contains a hypertext link on which the consultant can click to access to the case.

4.4.3.3 Choice of the case undertaken

Once the case is selected, the user can then consult all information that the producer has given: windows of the interface, video sequence, the different critiques already undertaken, and so on. The user travels between these various information by using hypertext links.

When his consultation is ended, the user can wish three things:

- To enclose his consultation of the case base. So, he just has to click on a button allowing it to return to the welcome page of the case base.

- To return to the list of cases that the search engine had provided him. To do that, he has to leave the case in question to have access again to this list.

- To criticize the case that he consults. It is with this hypothesis that we will continue the consultation scenario.

4.4.3.4 Critique of the consulted case

To have the possibility to criticize the case, the user has to download a Delphi application. In order to do that, he just has to click on the hypertext link of the application.

The Delphi application is in fact a "purified" version of the application created for the case insertion (the former is described in details in chapter three). The different windows corresponding to this application will be:

- «Renseignements sur l'insérant»: the consultation has not (until now) obliged the user to identify himself. It is necessary therefore to make it at this time to be able to classify his critique. This window is identical to the one developed for the insertion of the case. The only difference is that the user has to enter the identifying name of the interface he wants to criticize.

- From the window «Choice» until the window «Résumé de la critique»: these windows will be the same that those developed for the insertion of the case.

Once his critique(s) is (are) undertaken, the user ends by closing the Delphi application. He can then have access again to the list of case selected by the search engine to be able to continue his consultation.

4.4.4. What must be done

All the precedent sections of this chapter explain the work that we have done for the consultation of the case. The scenario of the consultation, the analysis of the task, the proposition concerning the search engine, all these things have been realized in order to facilitate the work of the person who will make the implementation. The insertion of the case has also been realized with this view: the representative keywords, the information concerning the person and the interface, everything is ready for a good classification and thus good performances of the search engine.

Now, what must be done is the implementation of the different tools and pages we have spoke about:

- *The search engine.*
  As all the necessary keywords have been anticipated and analyzed, the only thing to do is to implement the search engine with a JAVA applet.

- *The web pages.*
  At the time we finished this thesis, the structure of the web pages for the Vesale project wasn't well known. So, it was not possible to implement the

presentation of the case in thee pages once a person has inserted it. This work has to must done.

- *The application.*
  An application must be realized to let a consultant enter his critiques. Fortunately, the majority of the windows we have realized for the insertion can be use again here. The only things that must be added are some windows to let the consultant presents himself.

- *Provide adaptable scaffolding through levels of detail*
  As discussed in chapter two, it will be possible to decomposed the consultation process into several steps to make the consultation more pedagogical (see 2.3.3.1). Nevertheless, this is not an indispensable improvement.

4.5. Conclusion of this chapter

In this chapter, we have described how we have undertaken the task of consultation. It is obvious that a lot of things have not been realized but all these things depend on the implementation and choices that need to be done by the Vesale Project team : structure of the Web pages, programming language, and so on. On the opposite side, the analysis of the task and the needs of the users have been studied. That will facilitate the work and choices of the team.

Now that we have analyzed the two main tasks of our case base, it is time to give some information about the implementation. That's what we will do in the next chapter.

Chapter 5: **Implementation**

5.1. Introduction

In the preceding chapters, we analyzed the two main functions of our application, the insertion and the consultation. It's time to look our work on the implementation side. Note that the realization of a very performant application has not been our goal. So, if concept and realization of this application seem to be conclusive, lots of improvements will be necessary.

In this chapter, we will explain the implementation of the case insertion. Case consultation has not been implemented. Note also that the communication between our application and the Vesale project has not been implemented. The main function of the application, from the implementation side, is to collect information about interfaces, windows, critiques..., to structure them and to make a pack with all these information. The pack sending is the aim of our application.

In a first step, we will explain some choices we had to make at the beginning of the conception of our application and in a second step, we will explain the different data structure used.

We didn't insert the code in this section. Nevertheless, the application and its code are available upon request.

5.2. Important choices

At the beginning of the conception, we had to make important choices:

- To use or not to use the CoWeb
- What programming language to use?

5.2.1. CoWeb or not?

As we have seen in preceding chapters, CoWeb is very easy to use and offers a lot of advantages. But it has also a very big disadvantage for us: it is linked with a programming language, which is Smalltalk. Smalltalk is very common language in USA but practically unknown in Europe. So, if we wanted to use all possibilities of Coweb, we would have to implement in Smalltalk and it was a too big workload.

So, despite its big possibilities, we had to forsake Coweb. Nevertheless, we are convinced that when the Vesale project will be more developed, Coweb will be a possible tool to allow powerful communication.

5.2.2. Programming language

The choice of the programming language that we would used was an important point during the beginning of the conception. The programming language had to fill several criteria and mainly two:

- It had to be powerful enough and easy to use to allow the realization of the wanted application
- It had to be possible to link our application with the Vesale project.

It's very difficult to find a language that allows both. We had to choose between both: Java and Delphi. The first one was very easy to link with the Vesale Project. Nevertheless, it would be more difficult to use because it is younger than Delphi and the possibilities are more limited and/or less known by us. The second one was powerful, easy to use and well known by us but more difficult to link with the Vesale project.

After discussions, we decided to privileged the facility of used, conditionally that second criteria would be possible. So, we chose Delphi and we have tried to see if it was possible to send and to receive information from the Vesale Database. It seems to be possible thanks to "Internet components" of Delphi and more precisely thanks to "HTTP components" that allows "Get", "Head", "Post" and "Put" methods.

So, when a user of the Vesale project will want to insert a case, he will first have to load Delphi application. Then, Vesale will send useful information to this application (for example, information about the user such that the "Données sur l'insérant" window can be pre-filled). When the insertion will be finished, the application will send a pack with useful information (between other new inserted information) to Vesale that will insert the new case or the new critique in the case base.

5.3. Data structure of our application

Implementation of an application isn't neuter. We had to make choices and we decided to present hereunder the data structure of our application. You will find the main data structure used with a small explanation of the different components. Remarks that structure choices aren't always optimal but we privileged facility of use to optimal performance.

5.3.1. Interface

This is the main data structure of our application. That is this record that will be send to the database after the validation of the insertion by the user.

```
interfa = record
     generaux : tgeneraux;
     listfen : lpath;
     cglob : tglob;
     cinter : lcritinter;
     inserant : tinser;
     correction : string[50];
     sv : string[50];
   end;
```

An interface is a record composed with these different fields:

- `généraux`: contains all the information provided in "Présentation de l'interface" window.
- `Listfen`: pointer list containing windows composing the interface.
- `cglob`: contains components of global critique.
- `cinter`: pointer list containing the different inter-window critiques
- `insérant`: contains information about the inserant.
- `Correction`: contains, if present, path of the interface corrected by the current.
- `Sv`: contains, if present, path of video sequence.

Let's note that the intra-window critique isn't present. The raison is simple: this type of critique is linked to a window and not to an interface.

Remark also that we limited the strings containing name, path... to 50 characters.

5.3.2. General information

Once again, this data structure is a record because that's a set of different components.

```
tgeneraux = record
     nom : string[50];
     dom : string[50];
     typ : string[50];
     com1 : string;
     com2 : string;
     com3 : string;
   end;
```

This structure aimed to store the different information of the "Présentation de l'interface" window.

So, it's composed of:

- `nom` : interface name. This name is unique.
- `dom` : Field of the interface
- `typ` : Interface type
- `com1` : Environment comments of the interface
- `com2` : Pedagogical approach comments
- `com3` : Free comments

5.3.3. Windows list

For this structure, we have used a pointer list because we don't know how many windows will compose the different interface.

```
lpath = ^path;
path = record
     nomfen : string[50];
     chemfen : string[50];
     hdef : integer;
     wdef : integer;
     critique : boolean;
     cintra : lcritintra;
     svt : lpath;
   end;
```

A window contains this different information:

- `nomfen`: window name
- `chemfen`: window path
- `hdef`: height of window illustration. This information will be useful to adapt the size of illustration in the different application windows.
- `wdef`: width of window illustration.
- `critique`: boolean used to indicate if the current window has received an intra-window critique. It will help for the categorization of interfaces between already criticized window and uncriticized window
- `cintra`: list of critiques concerning current window
- `svt`: pointer to next window.

5.3.4. Global critique

Given that there exists only to the maximum one global critique, a record is the most appropriate structure.

```
tglob = record
     comp : string;
     comn : string;
     mc : lkey;
     da : string[10];
   end;
```

tglob stores the following information contained in the "Critique globale" window:

- `comp`: positive comments about the interface
- `comn`: negative comments about the interface
- `mc`: keywords associated with the global critique
- `da`: date of the realization of the global critique. This date is compute tanks to the "Mots clés" window.

5.3.5. Inter-window critique

An inter-window critique is a pointer to a record.

```
lcritinter = ^critinter;
critinter = record
     fen1 : string[50];
     coord1 : array [1..5] of Tcoord;
     fen2 : string[50];
     coord2 : array [1..5] of Tcoord;
     coordc : array [1..5] of Tcoord;
     comp : string;
     comn : string;
     mc : lkey;
     da : string[10];
     svt : lcritinter;
   end;
```

An inter-window includes elements hereafter:

- `fen1`: path of the first interface selected
- `coord1`: array with the coordinates of the different (maximum 5) rectangles designed on the first window
- `fen2`: path of the second window.
- `coord2`: array with the coordinates of the different (maximum 5) rectangles designed on the second window
- `coordc`: array with the coordinates of the different (maximum 5) crosses designed on the two windows.
- `comp`: positive comments about the inter-window critique.
- `comn`: negative comments about the inter-window critique.
- `mc`: list of the different keywords of the inter-window critique
- `da`: date of the creation of this inter-window critique
- `svt`: pointer to the next inter-window critique

5.3.6. The people who insert

This structure will store the different information contained in the "Données sur l'insérant" window.

```
tinser = record
     nom : string[50];
     prenom : string[50];
     email : string[50];
     qualif : string[50];
     com : string;
   end;
```

In this record, we store the different fields of the "Données sur l'insséran" window, that's to say:

- `nom`: the name of the person
- `prenom`: his given name
- `email`: his e-mail adress

- `qualif`: his qualification (Etudiant, professeur titulaire, autre)
- `com`: some free comments about him

5.3.7. Intra-window critique

This structure will store the different information contained in the "intra-window" window.

```
lcritintra = ^critintra;
critintra = record
     coord : array [1..5] of Tcoord;
     critp : string;
     critn : string;
     mc : lkey;
     da : string[10];
     svt : lcritintra;
   end;
```

The different information stored in the intra-window critique are the following one's:

- `coord`: array with the coordinates of the different (maximum 5) rectangles designed on the window
- `critp`: positive comments about the critique
- `critn`: negative comments about the critique
- `mc`: list of the different associated keywords
- `da`: date of the intra-window critique
- `svt`: pointer to the next intra-window critique

Note that we don't need the references of the window because an intra-window critique is automatically linked with it.

5.3.8. Keywords

This structure will store the different information contained in the "Mots clés" window.

```
key = record
     motc : string[50];
     svt : lkey;
   end;
```

The keyword structure is very simple, it just stores:

- `motc` : the keyword selected
- `svt` : pointer to the next keyword

5.3.9. Coordinates

Coordinates are the different coordinated of the shapes that user can draw on the illustration of the windows.

```
Tcoord = record
     depx : integer;
     depy : integer;
     finx : integer;
     finy : integer;
     rouge : boolean;
   end;
```

Here are the different fields:

- `depx`: x coordinate of the first point of the shape
- `depy`: y coordinate of the first point of the shape
- `finx`: x coordinate of the last point of the shape
- `finy`: y coordinate of the last point of the shape
- `rouge`: boolean indicating if the shape is red or green

5.4. Conclusion of this chapter

In this chapter, we presented our implementation choices. This implementation will need improvements and changes after evaluation of the application but basis and choices are consistent.

In the last chapter of this thesis, we will present a critique of our design interface.

Chapter 6:
Critique of the created interface

6.1. Introduction

In chapter 2, we spoke about the different paradigms of Leclercq and Denis and one of them was about "Impregnation". We said that the student learns interface design through an interface and so, this last had to be a good example. That's what we will develop in this chapter: we will try to demonstrate that our application interface has been well designed.

At this purpose, we will, in a first point, try to define what is "bootstrapping" by giving a definition and some example illustrating this approach. Secondly, we will try to see if our application can follow a bootstrapping approach and if it does.

We will finally make a critique of our interface design.

6.2. Bootstrapping approach

6.2.1. Definition

Bootstrapping is a teaching method. "The principles that are taught must be apply directly in the way of teaching". That's the main concept of this method. That can be in the way of being, in the presentation of the different things,...

In other words, the way of teaching must be an illustration and an application of taught concepts.

The mandatory condition to apply this method is that teaching must be found in the field itself.

So, it is based on the impregnation principle (see 2.4.1.1)

Notice that this method is very important because this is the first relationship between theory and practice. In other words, this is the first real example of theory application to which student is confronted. This is the "bootstrapping" of the concept.

6.2.2. Example

Let's see two examples to illustrate this concept.

- First, during deportment lessons, teacher has to well-behave, to give the example.
- Secondly, anti-alcoholic league chairman discoursing on problems caused by alcohol with a vodka glass in hand is a bad example.

6.3. Bootstrapping application: critique of a particular window of our application

6.3.1. Can we use bootstrapping?

We have seen that the mandatory condition to apply bootstrapping is that teaching must be found in the field itself. In our work, the user has to criticize interfaces through interfaces. So, the first case in our base can be the application interface itself.

We are now sure that we can use this method. All we have to prove now is that our application interface respects the different principles we try to teach. At this purpose, we will use our application to criticize our interface.

6.3.2. "Insertion" case

To prove the bootstrapping of our application, it would be necessary to criticize all the windows of our interface together and alone. Nevertheless, it would be a useless and repetitive process. That's why we will present to you:

- The global critique of our interface
- One inter-windows critique
- One intra-windows critique of the first window: "Données sur l'insérant".

We will also skip the presentation phase because this is not a main point to prove bootstrapping approach.

So, we will present the different critiques realized in our application thanks to screenshots. Nevertheless, these screenshots have two disadvantages:

- They are in French
- They are static and thus the entire text can't be displayed

That's why we will provide a summary of the critique after each screenshot.

6.3.2.1 Global critique

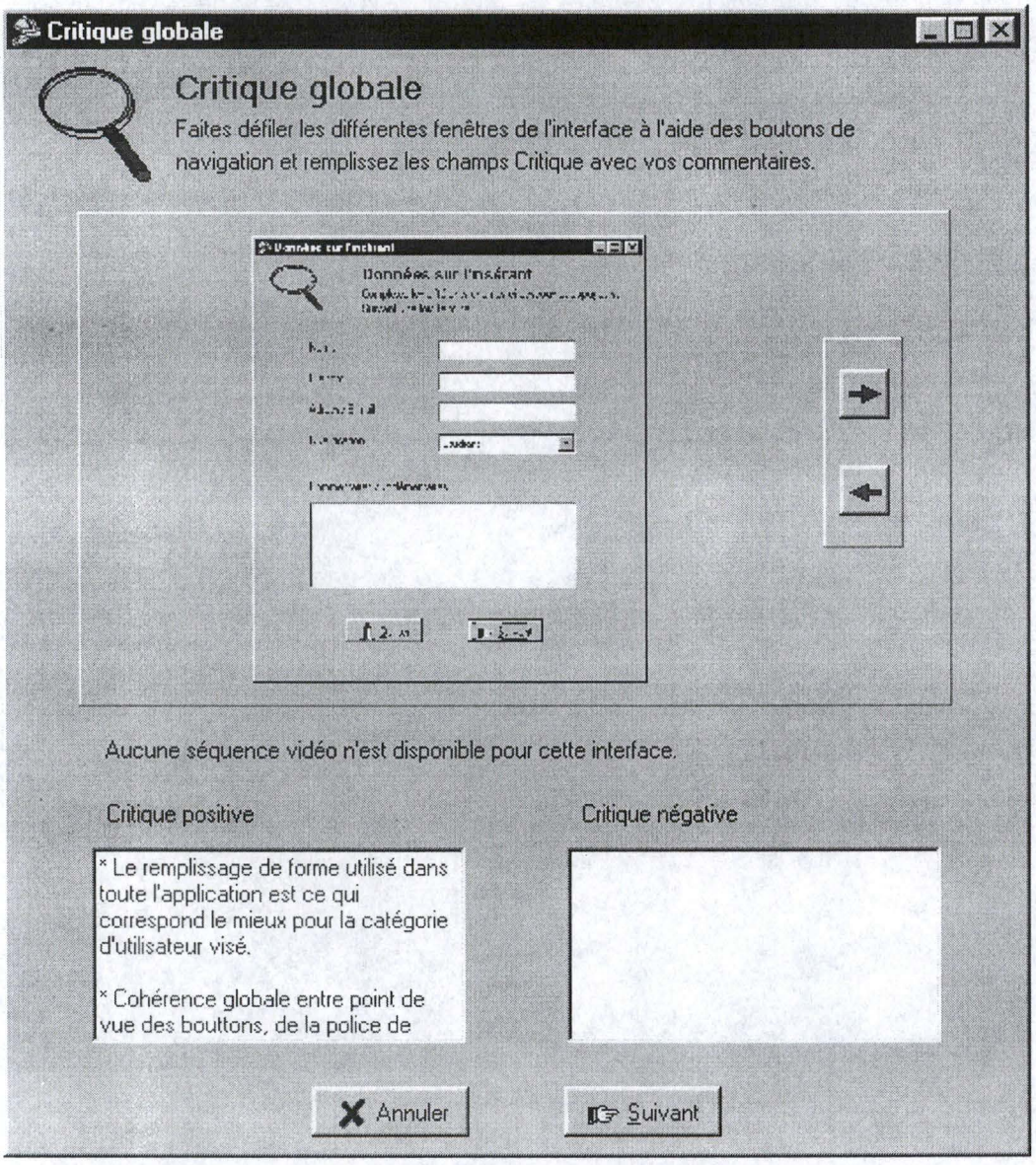


Figure 6-1: Global critique

- The interaction style chosen (form fill in) correspond to the level of expertise of the different users.
- There is a global consistency between the different windows of our interface concerning buttons, fonts…
- Chosen font is readable and fluently used.
- Windows sequence is intuitive and corresponds to the habits of the user.

6.3.2.2 Inter-window critique

We chose to compare "Choix des 2 fenêtres à critiquer" and "Choix de la fenêtre à critiquer"

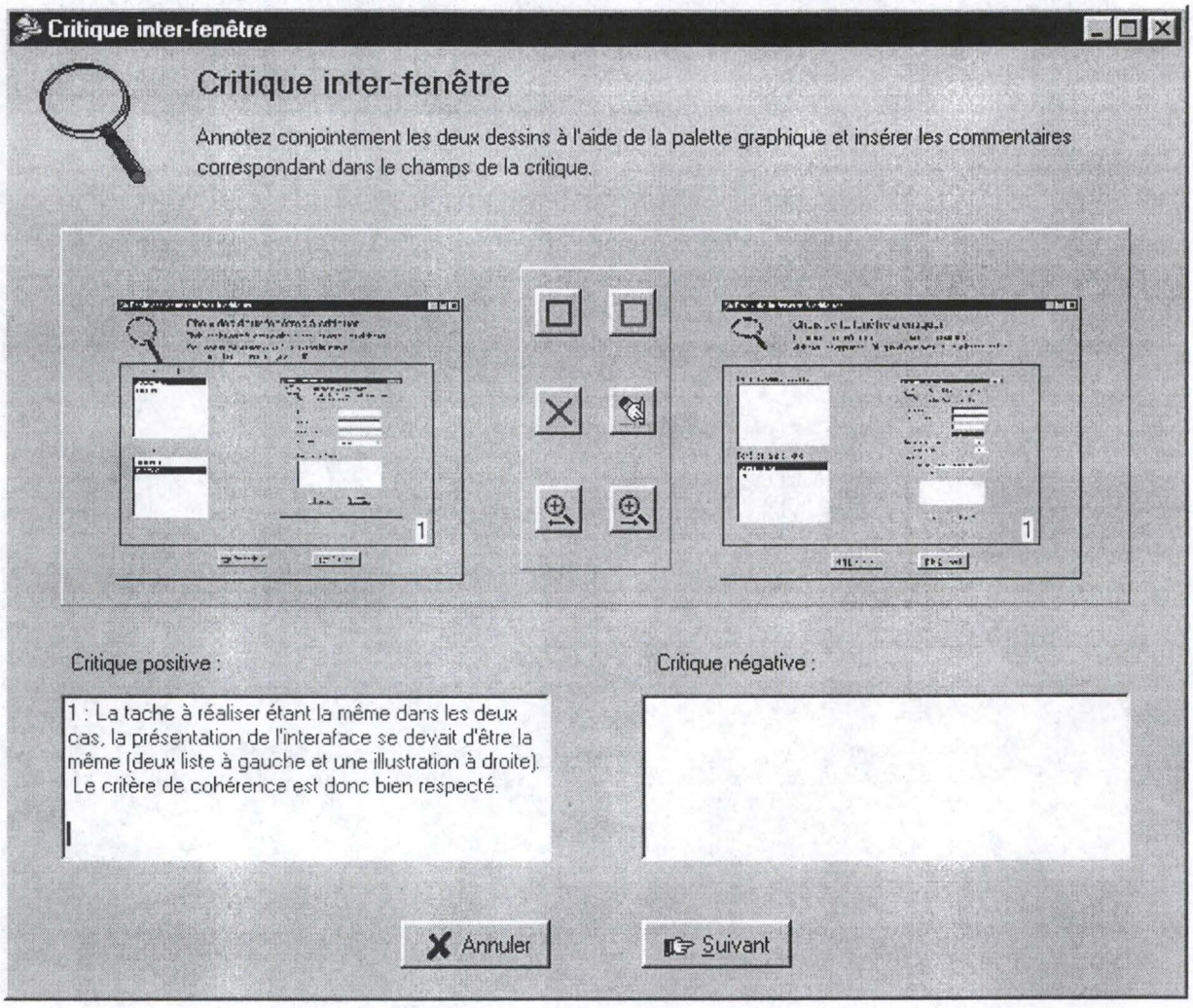


Figure 6-2: Inter-windows critique

- As the task to perform is the same in both cases, the interface presentation must also be the same. That's exactly what we have in our application: two lists on the left and an illustration on the right.

6.3.2.3 Intra-window critique

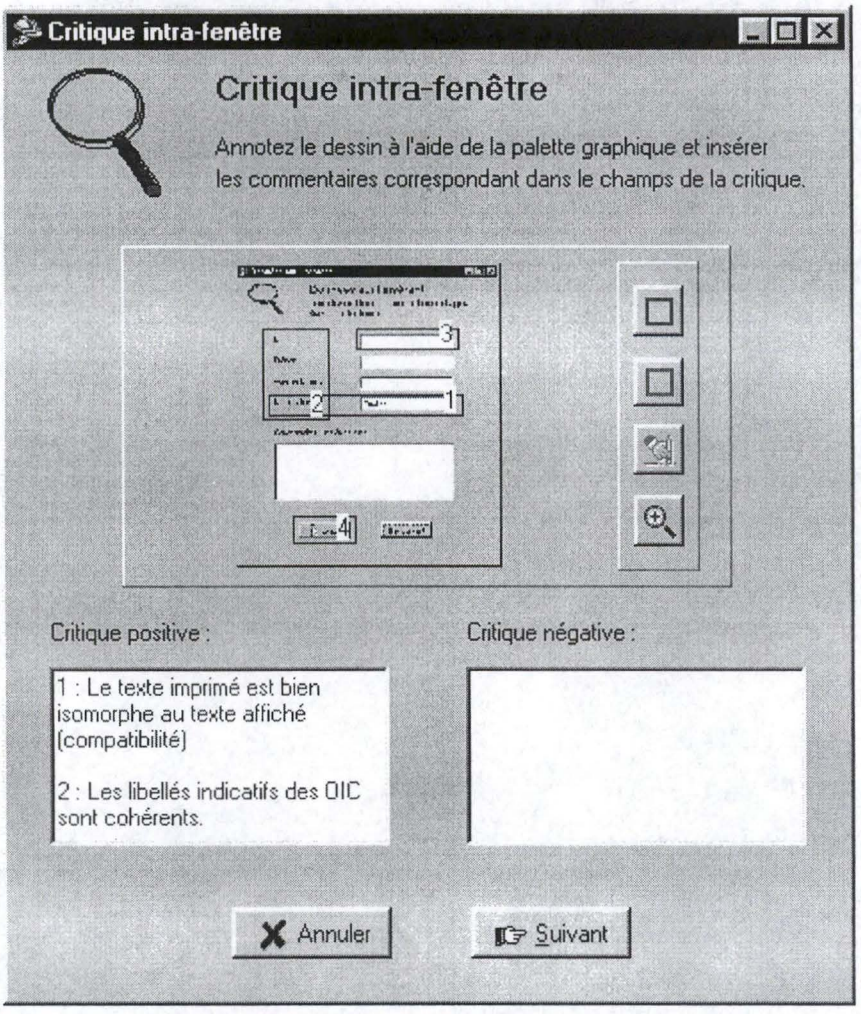


Figure 6-3: Intra-windows critique

- Compatibility
  - Printed text is isomorphic to the displayed text.
  - Position of the different CIO is compatible with the user expectations
  - Data concerning the same task are all on the same window
- Consistency
  - Labels before CIO are consistent
  - The different CIO are consistency placed
- Workload
  - CIO dimension is correctly chosen
  - CIO length is less than 40 characters
  - There is a default value for the ListBox
- Dialog control
  - User must use tabulation key to reach another field of the form

- Representativity
  - Buttons mnemonic is well-chosen

6.4. Conclusion of this chapter

In this chapter, we tried to explain the « bootstrapping » approach. This approach is fundamental for the Vesale Project. So, our work had to correspond to that. We explained thus with an example how our application and its interfaces can be used as a self-illustration for a good interface.

Conclusion

Conclusion

As we have explained in this thesis, we have realized two things. First of all, we have created a web site that explains some different HCI rules and that manages a case base. Secondly, for the Vesale Project, we have worked on the setting of a case base and we have implemented an application that let the users insert cases in the base.

It is obvious that the application and this web site are only prototypes that we propose to users. People must now test these tools and pedagogues have to evaluate the results in matter of learning.

As we had the opportunity to compare the computer-assisted methods from the USA and from Europe, we could say that a lot of progress has to be made in Europe if we speak about on-line course. In the USA, students seem to be familiar with the concept but in Europe, it is only the beginning. Now that computers have taken an important place in our today's life, it is time for the European teaching to adapt itself to the new technologies and to integrate them in its structures. Concerning our application, it's important to say that we have chosen an approach that favors the interactions and collaborations between the different human actors. To do that, we have followed the principles of case-based teaching. Another approach like case-based reasoning can be chosen. In this case, the interaction of the actors falls down and the machine takes a more important place. This approach can also be very interesting and useful in apprenticeship. It can constitute an ideal complement to the work we have done.

From a personal point of view, we can say that the American experience has been essential for us. It has been very interesting to compare two ways of life and two ways of thinking. To the rigorous way of working learned in Europe we can now add the open mind existing in The USA. These benefits is not limited to this thesis but will also help us further in our professional and personal life.

The opportunity to participate at a concrete project of the Institute (the Vesale Project° brings us a lot of satisfaction. The fact that this work will be employed also after the end of this thesis has been a real motivation.

Bibliography

Bibliography

BARBER J., BHATTA S., GOEL A., JACOBSON M., PEARCE M., PENBERTHY L., SHANKAR M., SIMPSON R., STROULIA E., « AskJef : integration of case-based and multimedia technologies for interface design support », *Artificial intelligence in design*, pp. 457-475, J.S. Gero (Ed.), Kluwer academic publisher, 1992.

BARON Georges-Louis, BRUILLARD Eric, *L'informatique et ses usagers*, presses universitaires de France, 1996

BODART F., PYGNEUR Y, *Conception assistée des systèlmes d'information, 2ème édition*, Masson (Ed), 1993.

BODART F., HENNEBERT A.M., LEHEUREUX J.M., PROVOT I., VANDERDONCKT J., ZUCCHINETTI G., « Key activities for a development methodology of interactive applications », chapter 7 in *Critical Issues in User Interface Systems Engineering*, pp. 109-134, D. Benyon and Ph. Palanque (Ed.), Springer Verlag, Berlin, 1995

BODART François, *Cours introductif à la conception des interfaces homme-machine*, Institut d'Informatique, Facultés Universitaires Notre-Dame de la paix, Namur, 1998.

BODART F., BEIREKDAR A., EMBAKI E., LEHEUREUX J.M., *Visual user interface design Education Supported by a computer-Aided Learning Environment: specification document*, Institut d'Informatique, Facultés Universitaires Notre-Dame de la paix, Namur, 1999

BRUCKMAN Amy S., GUZDIAL Mark, KOLODNER L. Janet, RAM Ashwin, *International conference of the learning sciences 1998*, Association for the Advancement of Computing in Education (AACE), 1998a.

BRUCKMAN Amy S., EDWARDS Elizabeth, *Should we leverage natural-language knowledge? An analysis of user errors in a natural-language-style programming language*, College of Computing, Georgia Institute of Technology, Atlanta, GA, 1998b

CATIZZONE Michela, REMY Elise, *Virtual University,* Final thesis, Institut d'Informatique, Facultés Universitaires Notre-Dame de la paix, Namur, 1999.

CHINOWSKY Paul S. and ROBINSON Jeffrey, *Facilitating interdisciplinary design education through case histories*, Georgia Institute of Technology, Atlanta, GA, 1995

DOMESHEK Eric, KOLODNER L. Janet, « Using the points of large case » from *AI EDAM* 7(2), pp. 87-96, College of Computing, Georgia Institute of Technology, Atlanta, GA, 1993

DONNAY Jean, *Cours de pédagogie générale, méthodologie générale, pédagogie expérimentale*, FUNDP département Education et Technologie, Namur, 1997

GAINES B. R., HILL D.R., *International journal of Man-Machine studies*, Department of computer science, The University, Calgary, Albertan, Academic press limited, Canada, 1992

GUZDIAL Mark, KEHOE Colleen, *Apprenticeship-based learning environment: a Principled approach to providing software-realized scaffolding through hypermedia*, http://guzdial.cc.gatech.edu/papers/able , GVU center and EduTech Institute, College of Computing, Georgia Institute of Technology, Atlanta, GA, 1998a

GUZDIAL Mark, *Collaborative Websites to support an open authoring community on the Web*, EduTech Institute and GVU center, College of Computing, Georgia Institute of Technology, Atlanta, GA, 1998b

HUANG T., FLANIGAN J., « Human centered systems in the perspective of organizational and social informatics », Chapter 5 of *Human center systems for the national science foundation*, 1997

JONASSEN David H., MANN Edward, *Scaffolding diagnostic reasoning in case-based learning environments*, Pennsylvania State University and StorageTek, 1994

KOLODNET L. Janet, *Case-Based Reasoning*, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1993

LECLERCQ Dieudonné, *Pour une pédagogie universitaire de qualité*, éditions Mardaga, Belgique, 1998

MAGNIER Marc-Laurent, BODART Thibaut, *A multimedia support for the learning of interaction objects*, Final thesis, Institut d'Informatique, Facultés Universitaires Notre-Dame de la paix, Namur, 1999.

NARAYANAN N. Hari, KOLODNER L. Janet, *Case libraries in support of design education: the DesignMuse experience*, EduTech Institute, College of Computing, Georgia Institute of Technology, Atlanta, GA, 1995

PETRUSHIN Valery A., THOMSON Chris, DESETTO Louis, *Learning ergonomics by doing job analysis*, Georgia Institute of Technology, Atlanta, GA, 1996

RAVDEN S, JOHNSON G., *Evaluating usability of Human-Computer Interfaces*, Ellis Horwood Limited, England, 1989

SCHANK Roger C., *Case based teaching: four experiences in educational software design*, The Institute for the learning sciences, North Western University, Evanston, IL, 1991

SCHANK Roger C., « Goal Based Scenarios: case-based reasoning meets learning by doing » in David Leake (Ed.) *Case Based Reasoning: experiences, lessons and future directions*. AAAI press/MIT press 295-347, 1996

SHABO Amnon, NAGEL Kris, GUZDIAL Mark, 1997, *JavaCAP: a collaborative case of authoring program on the WWW*, EduTech Institute, College of Computing, Georgia Institute of Technology, Atlanta, GA

VANDERDONCKT Jean, *Tâche interactive de l'enregistrement d'un bon de commande client*, Institut d'Informatique, Facultés Universitaires Notre-Dame de la paix, Namur, 1997a

VANDERDONCKT J, PROVOT I., SACRE B, *Description d'objets interactifs abstraits*, Institut d'Informatique, Facultés Universitaires Notre-Dame de la paix, Namur, 1995

VANDERDONCKT Jean, *Introduction à la méthode TKS d'analyse de tâche*, Institut d'Informatique, Facultés Universitaires Notre-Dame de la paix, Namur, 1997b

VANDERDONCKT Jean, *Corpus ergonomique minimal des applications de gestion*, Institut d'Informatique, Facultés Universitaires Notre-Dame de la paix, Namur, 1997c

VITALE Bruno, *L'intégration de l'informatique à la pratique pédagogique, volume 1* , centre de recherche psychopédagogique (CRPP), Genève, 1990

Appendixes

Appendix 1:
Vesale specifications

University of Namur (FUNDP)
Informatics Institute

VESALE

Visual user interface design Education

Supported by a computer-Aided

Learning Environment

Specification document

VESALE Team
Professor François Bodart
Abdo Beirekdar
Efrem Embaki
Jean-Marie Leheureux

Writer : *Abdo Beirekdar* ***Version*** *: 5.1.1999*

TABLE OF CONTENTS

0. INTRODUCTION

Teaching and research related to Human-Machine Interfaces (HMI) have as aim the theories, the models, the methods and the tools necessary to all the stages of the cycle of life of a HMI. Consequently, it would be useful to employ the specific knowledge to this field in the development of software tools of support to a computer-aided teaching? If, moreover, this teaching is in the field itself, the opportunity is given to use it's content in auto-illustration of knowledge and techniques specific to the field of HMI and thus benefiting the learning. This principle of "bootstrapping" is fundamental to the VESALE project, a multimedia environment of support to the teaching of the human-machine interfaces, for which the global design architecture is given below.

This teaching's support concerns four situations of operation :

1. the apprenticeship as part of the live teaching : teacher and students are face to face ;
2. the complementary apprenticeship to the live teaching which essentially stands for the evaluation of knowledge and review of syllabus contents ;
3. the indirect teaching or distant learning i.e. the self-apprenticeship enriched by the interactions with the teacher ;
4. the co-operation to the teaching's enrichments. In particularly, the enrichments for reasoned

cases and for illustrations of multimedia technologies.

From this point of view of general introduction, the project VESALE can be represented according to three views : the global logical architecture of the project, the logical view related to the professor and the logical view related to student. These views are presented in the first part of this document.

In the second part, is given a short description of the modules under consideration in the project.

I- Global presentation of the modules of project VESALE

The project VESALE will comprise the following modules :

- ***the base of course notes***: contains the notes of the Human-Machine Interface (HMI) course presented in the form of a hypermedia software.
- ***the base of the reasoned cases*** : contains critical examples of interface including some examples of design. This base could *be enriched* by cases suggested by the students or other interested persons.
- ***the base of the illustrations of multimedia technologies*** : is related to the interaction objects and technologies.
- ***the base of the ergonomic rules*** : is intended to illustrate the use of the ergonomic criteria and rules in order to build useful and usable interfaces. It will consist of a subset of ergonomic rules of design of HMI.
- ***the base of knowledge evaluation***: contains exercises, questions and plans of evaluation of knowledge acquisition by the learner. It must allow a summative and formative evaluation.
- ***the base of video sequences*** : contains video sequences of introduction of certain parts of the HMI course or sequences to illustrate the handling of certain interactive objects.
- ***the space of dialogue*** : support the dialogue between the professor and the student on general questions.
- ***a space of comments*** : associated with each particular base, this space makes it possible for the students to express comments and eventually for the professor to react to these comments.

The layout of these modules is illustrated in the schema of the global architecture (figure 1). The functionalities intended for the professor are illustrated in the schema of figure 2 and those intended for the student are illustrated in the schema of figure 3.

- **Global logical architecture**

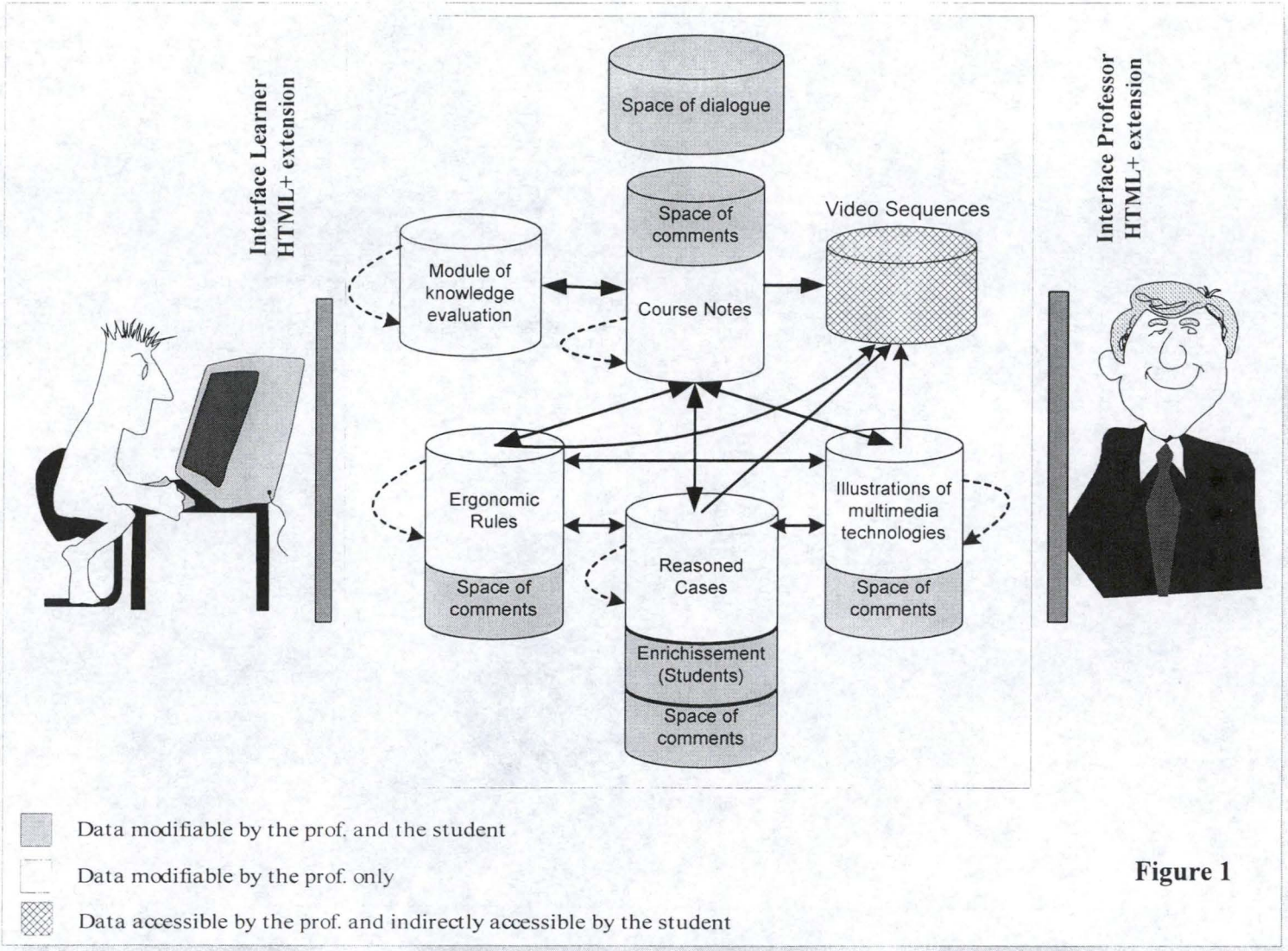


Figure 1

- **Logical view of the professor**

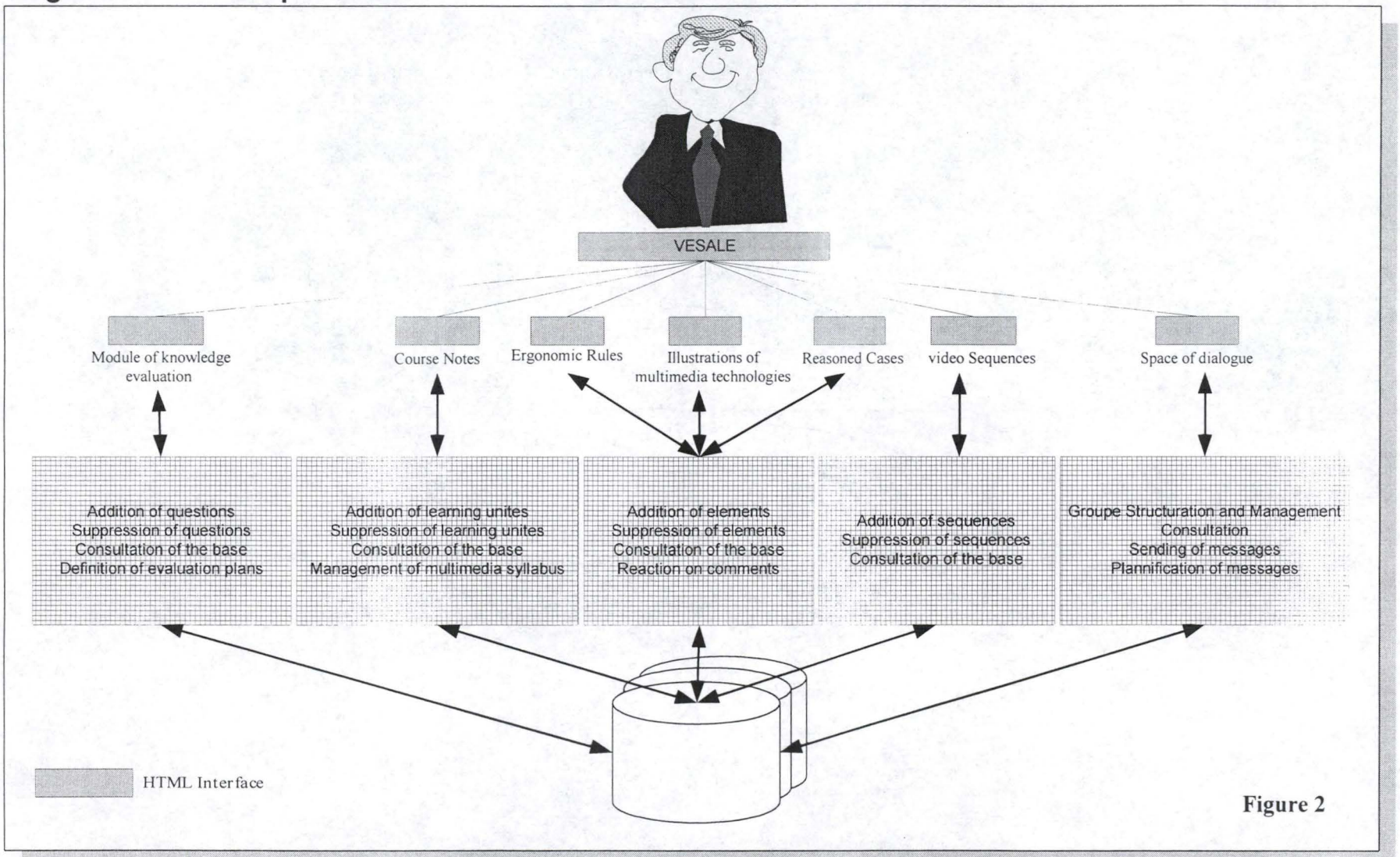
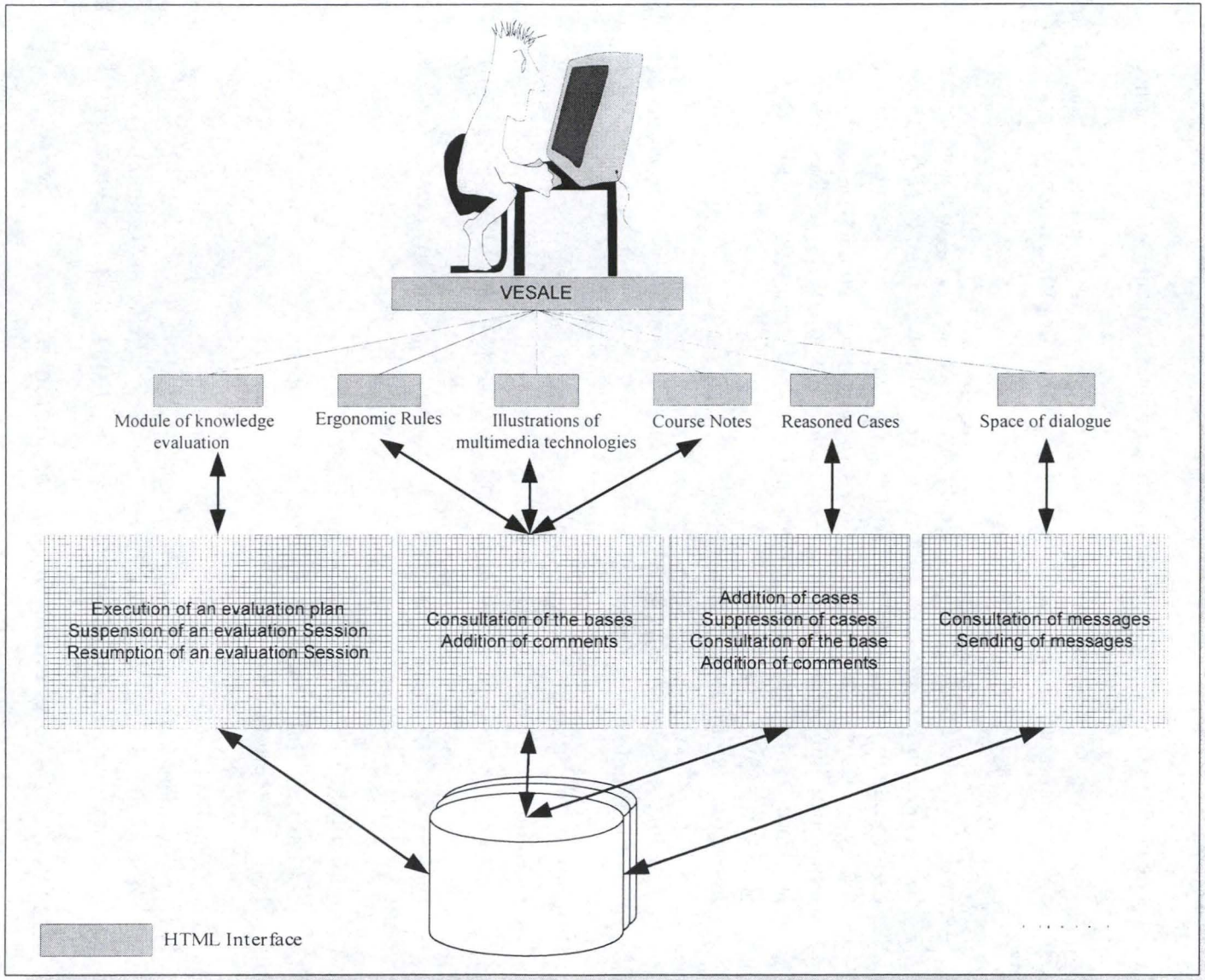


Figure 2

- **Logical view of the learner**



II- Description of the modules

The modules of project ?????? are the followings :

1. Notes of course

The notes of the course will be restructured in units of learning (to be defined).

These notes will be presented to students in the form of a hypermedia syllabus. It consists of a number of HTML pages generated starting from the notes of course and other bases.

A priori, the multimedia syllabus should be structured according to a double structuring :

- by ***conceptual units*** and links between them. Example of conceptual units corresponding to the underlined terms: a HMI is related to an interactive task carried out by an actor (or actors), according to a dialogue mode and a dialogue style which implement interactive objects ... according to a presentation. These elements of the HMI are used by physical devices and software means within the framework of a software architecture.
- by ***regrouping units*** (chapter, section...).

The syllabus is organized and presented in a **dynamic** and **personalized way** according to the profile of the student and his model of learning.

Content of the base of course notes

- *learning units.*
- *a space of comments* (about the course notes) between the professor and the learners or other interested people.
- *references* towards illustrations of multimedia technologies, video sequences, ergonomic rules, questions of evaluation of knowledge and reasoned cases

Functionalities related to the course notes

- *creation, deleting, addition and structuring* of learning units: using an authoring language (editor).
- *consultation.*
- *addition of comments.*
- management of the multimedia syllabus :
  - ⇨ hypermedia layout of the pages (hypertext and hypermedia links),
  - ⇨ creation of the navigation plans according to the selected progression mode.

2. Base of illustrations of multimedia technologies

Multimedia technologies are done via interaction objects and the physical devices associated with them.

Content of the base

- *types of abstract interactive objects (AIO)* described according to the model: french name, english name, definition, abstract attributes, abstract events, abstract primitives.
- *types of concrete interactive objects (CIO)* described according to the model : name, concrete attributes, concrete events, concrete primitives, graphical representation, physical devices of interaction.
- *instances illustrating the CIO*: these instances could be screen copies, representative photographs, etc.
- *examples of CIO execution* explicitly illustrating by direct manipulation the concepts of attribute, event and primitive.
- *applications* using the CIO in a well-defined context.
- *a space of comments* : to allow the students and the professor to exchange specific comments.
- *references* towards the notes of course, other interactive objects, ergonomic rules and reasoned cases.
- possibly *reference* towards video sequences.

Example : AIO ***Command button***

- AIO
- *French name* : bouton de commande, bouton poussoir (BTC)
- *English name* : push button (PBT)
- *Definition* : control used to trigger an action whose result is described by the label. Only one button can be activated at the same time
- *Abstract attributes* : AT_BTC_LIBELLE, AT_BTC_DEFAUT, AT_BTC_FCT
- *Abstract events* : EVT_BTC_SELECTIONNE
- *Abstract primitives* : PR_BTC_CREATION, PR_BTC_DEFAUT, PR_BTC_NORMAL

- CIO (MS-Windows95)
  - *Name* : push button
  - *Concrete attributes* : CAPTION, DEFAULT, LEFT, SIGNAL, WIDTH, HEIGHT...
  - *Concrete events* : OnClick, OnEnter, OnExit, OnKeyDown...
  - *Concrete primitives* : CREATE, DESTROY, CLICK, HIDE, SHOW.
  - *Graphical representation* :

- CIO (Motif)
  - *Name* :
  - ...
  -

- *application*: For example to trigger a process where one must press on the button to record a command.

Functionalities specific to the base of illustrations of multimedia technologies

- *creation, deleting, addition* of new objects: using an authoring language (this language can be limited to some updating forms).
- *consultation*: by name of object, by its nature (input, output, acquisition...), by the author description (student, professor)...
- *manipulation of the illustration instances*: for better understanding of the use and the behavior of certain CIO, one will have some specific applications that permit to manipulate these instances of these CIO (e.g. an application which allows to rotate it, to move a 3D CIO). This manipulation can be carried out using some toolkit-independent libraries (such as the library ***Swing***).
- *passive observation of a CIO behavior*: for certain CIO (complex, nontraditional...) their use will be shown by visualizing a video sequence stored in the base of video sequences.
- *addition of comments (by the professor and/or by learner).*

3. Base of ergonomic rules

This base is intended to illustrate the use of the criteria and the ergonomic rules in order to build useful and usable interfaces. It will contain a subset of ergonomic rules for the design of HMI. Moreover, it will contain references to Web sites relating to existing corpuses of ergonomic rules (general or special to particular fields).

Content of the base

- *ergonomic rules* described according to the model: type, title, statement, linguistic level, justification, positive examples, negative examples.
- *space of comments* : allows professor and learners to add their comments related to this base.
- *references* towards the notes of course, the interactive objects, other ergonomic rules, ergonomic corpuses of rules and reasoned cases.
- Possibly *reference* towards video sequences.

Example of an ergonomic rule :

- *Type* : localization of the OIC, objects of control.
- *Title* : localization of the command buttons.
- *Statement* : the position of the command button by default should be the highest if the buttons are laid out vertically or on the left if the buttons are laid out horizontally.
- *Linguistic level* : lexical level.
- *Justification* : minimization of the articulatory distance.
- *Positive examples* : screens 1, 2..
- *Negative examples* : screens 3, 4..

Specific functionalities to the base of the ergonomic rules

- *creation, deleting, addition* : using an editor of rules.
- *consultation*: by type of rule, by linguistic level, by key words, etc. The possibility would be given to introduce a query of consultation according to a given syntax.
- *addition of comments (by the professor and/or by the students).*

4. Base of reasoned cases of interfaces design

This base will contain positive examples of design and negative examples of design of interfaces that do (not) respect the criteria of design and the taught ergonomic rules.

These examples could be analyzed according to the approach described in the following diagram of the figure 4.

Content of the base

- *examples of interface* described according to a general model to be defined.
- *space of comments* : allows the students to add their comments related to this base.
- *references* towards the notes of course, the interactive objects, the ergonomic rules and other reasoned cases.
- Possibly *reference* towards video sequences.

Specific functionalities

- *creation, deleting, addition* : using a specific editor.
- *consultation*: by the implied CIO, the cases that (don't) comply with a given ergonomic rule, etc. As for the base of the ergonomic rules, one could give the possibility of introducing a query of consultation according to a given syntax.
- *enrichment of the base*: it should be possible to allow the enrichment of this base by any interested person. However, the addition of cases could be restricted to certain conditions.
- *addition of comments*.

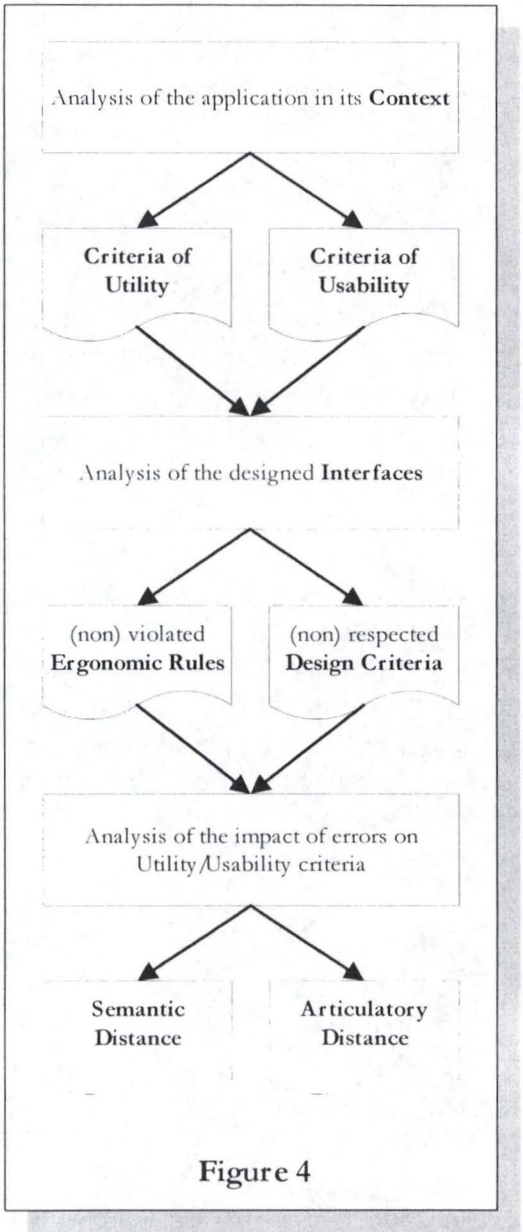


Figure 4

5. Base of video sequences

This base will contain sequences of presentations of certain parts of the course and sequences of illustration of certain CIO or cases.

The base will also contain other information making it possible to manage these video sequences as well as possible (subject, authors, title, keywords, duration, description...).

Content of the base

- *general data.*
- *references* to the files of the video sequences, to the illustrated aspects, etc.

Specific functionalities

- *creation, deleting, addition* of general data.

- *creation, deleting, addition* of video sequences.

6. knowledge Evaluator

The evaluation could be done in two manners : summative and formative.

The objective of this evaluation is to give to student the possibility of evaluating the state of his knowledge. The questions can relate to a section, a chapter, a technique, a step, a concept and its use in all the course, etc. This functionality should also allow the student to optimize his preparation for the examination according to specified requirements.

The evaluation is independent of the fact that the student follows or does not follow the course. The student could evaluate himself at any time. Following a session of evaluation, he can be advised to re-study certain parts of the course (from the evaluation advised).

Content of the base

- *exercises of evaluation.*
- *questions of evaluation* described according to a model to be defined: e.g. by statement, chapter, concept...
- *plan of evaluation and guidance.*

Functionalities specific to the module of evaluation

- *creation, deleting, addition*: using a language author.
- *consultation.*
- *definition of a plan of evaluation.*

7. Space of dialogue

Beside the spaces of comments attached to the various data bases, this space would be used to support the general dialogue (general remarks, request for references, appointment with the professor, etc.) on the one hand between the professor and the student and on the other hand, among the students themselves. This space of dialogue can be seen as a mixture of contextual e-mail, of news, newsgroups (forum).

Specific functionalities

- *Structuring and management of the groups of students:* by the professor and also by the learners.
- *Consultation :* messages...
- *Sending of messages.*
- *management of the diaries of events*

8. Administration of the course

- by the professor in the class room:
  - *on- line support of the course* (screen projection, synchronous interaction between the workstation of the professor and those of the student.
  - *On-line exercises* with possibility of access by the professor to the work stations of the students.

- remote administration by the professor :
  - non interactive diffusion.
  - Asynchronous interaction with the students (off-line answers).
  - synchronous interaction with the students :
    - ⇒ possibility for the student of addressing questions during the lecturing by the professor (e.g. request for control. If this one is granted, expression of the request with possibility of a direct answer to the applicant only or a multicast answer.
    - ⇒ possibility of video conference with sharing of documents on both sides.
- Off-line use by learners:
  - to learn
  - to make exercises
  - to be evaluated
  - to dialogue (to cooperate) with the professor and with the others students.

9. Browsing of the content of the course

There would be several ways of examining the content of the course:

- *by the hypermedia course notes*: approach suggested by the professor. The student chooses his plan of navigation and follows this plan starting from the interface of access to the hypermedia syllabus.
- *by a data base*: another approach would be to directly access from the home page to the interface of one of the data bases and not throughout the course notes. And then the learner should be able to access to all other objects linked to an object of the visited data base. E.g. the re-examination of an ergonomic rule could result in seeing one (some) case (s) of design where the rule is applied, CIO related to this rule and parts of the syllabus where this rule is mentioned.

Appendix 2:
Detailed presentation of our CoWeb site

A2.1. Introduction

A2.1.1. Objective

The main objective of the created site is to allow all program creators to develop a "good" interface. The "good interface" term is however vague enough and to choose criteria of "good" interface, we based on the Trident project [BODART and AL 95], [VANDERDONCKT 97a] and on the course received in our fourth year of our master degree [VANDERDONCKT and AL 95], [VANDERDONCKT 97].

At this purpose, we have developed a base of the site with some cases inside and we hope that users of this web site will enrich it by their comments, questions and of course by their interfaces.

A2.1.2. Targeted users

We rapidly remark that program conceptor can have very different knowledge in HCI and nevertheless both are confront with interface design.
This is why we decided to create two "levels" in this site, namely "novice level" and "expert level".

A2.1.3. Means of realization

We chose to realize this site in a particular environment named "CoWeb" that appears, in reality, to be a very simple author language for the realization of WebPages. Although it is simple, it allows the realization of sophisticated WebPages because it accepts all HTML code, Java...

The most important factor of the CoWeb is, as we have told it above, its simplicity to create WebPages. Thanks to it, everyone can create some pages and/or modify some existent pages. Everybody will be able therefore to add without problem cases, comments or questions.

A2.1.4. Address

To access to our web site, You just have to go to this URL:

http://pbl.cc.gatech.edu/interfaces/1.html

If you want to try the CoWeb and to test it, go the above address and go to the "sandbox".

A2.2. Coweb site contents

We will now describe the different pages composing our CoWeb site.

A2.2.1. FrontPage

A2.2.1.1 Goal

It is the homepage of our site. His goal is to explain briefly to the visitor what is the site and to allow him to enter in the apprenticeship side of our site.

A2.2.1.2 Contents

Beside logos illustrating the collaboration between FUNDP and GATECH, a small introduction welcomes the visitor.

We propose then a choice list that will allow the user to know more about this site:

- About our project: this page will present our project, the Vesale project of The Computer Science Institute of Namur as well as the CoWeb of the College of Computing of Georgia Tech.
- Ergonomic rules: brief presentation of rules on which this site is based. From there, user can enter the tutorial if he wants more information about one particular rule.
- About us: brief presentation (to be continued) of this site creators.
- Formatting rules: basic rules about the use of CoWeb, how to add things,...
- Question page: Sort of discussion forum where everyone can put questions, can reply to questions, can add remarks,...
  This point will be more analyzed later.

The visitor can then decide to enter to our site by clicking on "Enter here". At the bottom of the page, an envelope allows the user to send us an Email and a link to the "Recent changes" allows to see what are the recent changes realized on the different pages of this site. Thanks to this link, it is possible to know if new cases have been added, if replies to questions have been brought,...

A2.2.2. Type of user

A2.2.2.1 Goal

The goal of this page is to allow the user to choose between the novice way or the professional way.

A2.2.2.2 Contents

Address: http://pbl.cc.gatech.edu/interfaces/8.html

After a brief introduction, we explain to the user what is for us a novice and a professional as well as the different things that he will find according to his choice.

During our internship, we have only worked on the novice aspect of the site because we were in a very interesting environment for this part. Consequently, although it is accessible, "professional way " does not contain great things.

A2.2.3. Novice way

A2.2.3.1 Goal

The goal of this page is to propose to novice several tools to learn the HCI concepts following its waits and its preferences.

A2.2.4. Contents

Address: http://pbl.cc.gatech.edu/interfaces/50.html

After a brief introduction, three choices are proposed to the novice:

- The tutorial: we expose principles of the HCI in a theoretical way. This point is more detailed in the following sections.
- The case base: we expose good and bad examples of interfaces to novice so that he could learn from work of others
- The history of a case: we propose here different reports written by students during the conception of interfaces.

In reality, at the beginning, the case base was the only foreseen component. We were rapidly aware that we used in HCI a particular jargon that it would be necessary to explain to

novice. Therefore, we began to construct a small glossary explaining the different used terms. When we saw the number of terms to explain, we decided that it would be a good idea to present these terms in a tutorial and to propose it to the novice who desires to know theoretical things before seeing the illustration. When an HCI term is used in examples, we tried to propose to the user a link to the tutorial part.

The history of a case part has appeared after having consulted HCI courses given at Georgia Tech. The way to present things is not the same than in Europe and we have dedicated a part of this site to the works realized by Georgia Tech students. We find that this dual version of presentation can be very interesting for American or for European students.

A2.2.5. Tutorial

A2.2.5.1 Goal

The goal of the tutorial is to present the different concepts of HCI

A2.2.5.2 Contents

A very important thing is that we have tried to realize the tutorial in a scalable way. So the information is presented in different steps from the most general to the most particular and that's the student who decides where he wants to stop his exploration.
The tutorial is divided into four subject explained in the four following points..

a) Tutorial concerning criteria of design

After a brief explanation of the concept of design criteria, the 8 design criteria taught in the University of Namur are presented. They are:

- Compatibility
- Consistency
- Charge of work
- Adaptativity
- Guidance
- Control of dialogue
- Representativity
- Control of errors

Concerning these 8 criteria, we present in a first time a brief definition. The novice can then continue his exploration and see a more complete definition, the objectives of the design criteria as well as their possible decomposition. If he wants more details again, he can access to sub-rules. We have tried to illustrate rules by textual examples and by links to the case base.

b) Tutorial concerning items

After a short explanation of concepts of CIO and AIO, we present the different groups of AIO, and for each a short explanation. The 6 groups are:

- Static AIO
- Scrolling AIO
- Action AIO
- Control AIO
- Dialog AIO
- Feedback AIO

Once again, if the user desires to explore more, he can click on each AIO group to have a more complete definition as well as a graphic representation.

c) Tutorial concerning ergonomic criteria

We present here different ergonomics criteria useful during the choice of the interaction style. They are:

- Domain of the application
- Description of the user
- Environment of the application

We present then different interaction style with the possibility to see an illustration of the interaction styles from the case base.

d) Tutorial concerning conception criteria

We have summarized in this point the main conception criteria that are studied at Georgia Tech. We find among them:

- prototype methods
- design methods
- evaluation methods
- computer knowledge
- field knowledge
- input
- output
- device
- field

A2.2.6. Case base

A2.2.6.1 Goal

The goal of the case base is to present to user a lot of interfaces in which he will find negative and positive points and some justifications of proposed critiques.

If the user desires, he can target his search thanks to a search engine.

A2.2.6.2 Content

Address: http://pbl.cc.gatech.edu/interfaces/75.html

a) Case List

After a brief introduction, we propose to the user a list of all cases of the site. To allow the user to obtain some information about each case, a certain classification has been realized. Cases are presented in the form of a table containing the following parameters:

- the name of the case (this field remains void in the case of an sub-case)
- the name of the sub-case (this field is filled by " | " in the case of a case)
- the field of application
- the type of targeted user

As mentioned above, we made the distinction between a case and a sub-case. The first one is less precise and proposes a more general interface of the application. It is then decomposed in sub-cases that present particular parts of the interface. For example, one of our cases is "Winword 97" and presents interface with different windows and opened menus. The user can click on each window or menu to have a detailed description.

b) The search engine

Beside this list, the user can access to a search engine that presents a subset of cases in agreement with what the user desires to see. This search engine is implemented in Java and functions on a subset of the base.
More precisely, when the user access to the search engine, he can see a form composed in 3 parts:

- Description of the application field: we propose to the user several possibilities thanks to a scrolling list.
  This is not a mandatory field. If the user want, he can't select any item from the list so that this field will be occult during the search.

- Description of the user: We are in "novice way", so the user isn't very familiar with all nuances of the profile of the future user. This is why we have defined two

typical users, the novice and the expert, so that the experience of the task, the system, the motivation and the mastery of complex interaction means are predefined.
If the novice wants to define the 4 criteria, he must just choose "Other" in the type of user and the 4 fields corresponding to the type of user become accessible.
Once again, this is not a mandatory field. If the user want, he can't select any item from the list so that this field will be occult during the search

- Description of the environment: here, the novice will have to select the type of processing (mono or multi) and the data-processing capacity.
  Once more, this is not a mandatory field. If the user want, he can select no item from the list so that this field will be occult during the search

The user can then submit his research demand or reinitialize all fields. Search engine will be then proposed result of the search to him. The result includes the name of the case and the percentage of correspondence between this case and the user initial demand. A minimal acceptance percentage will have to be fixed. User must then select the case to be able to access to it.

c) Illustration of the search engine

Here is hereafter a screenshot of the Java applet of the search engine.

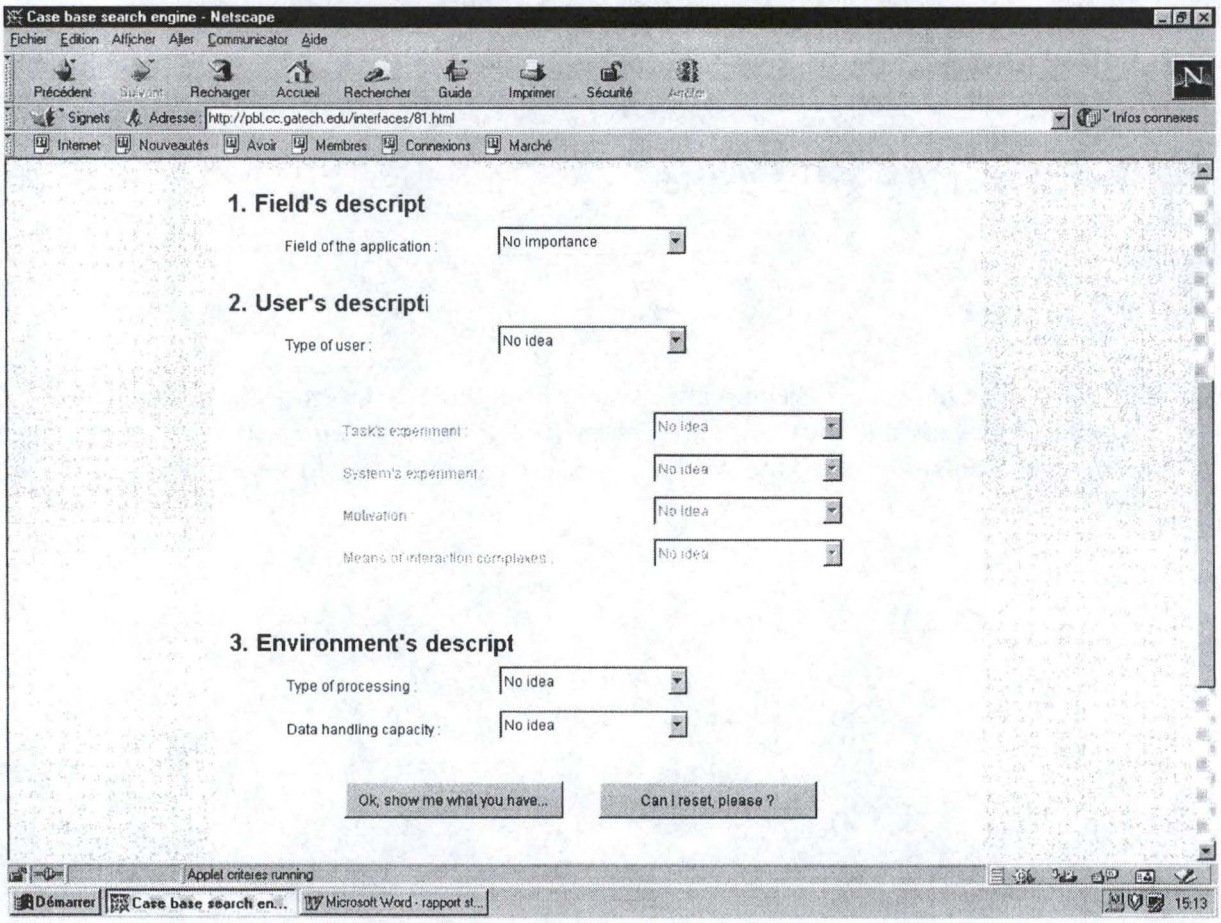


Figure A2-1: Search engine of the case base

A2.2.6.3 Case structure

Cases are structured of different manner following that it concerns a case or one sub-case.

a) Cases

A brief presentation of the case is given in introduction so that the context can be presented. Then, a screenshot of its interface is given. Each element of the interface is accessible by simple click (what brings in the corresponding sub-case) and for persons preferring to see the list of associated sub-cases, this list is presented in the page bottom.

b) Sub-cases

Sub-cases are presented in a more structured form. We can find:

- Screenshot of the interface of the sub-case
- Summary presenting parameters linked to this interface (cf. Criteria of the search engine)
- Presentation of used AIO
- Justification of interaction style choice
- Negative and positive points of the interface.

A2.2.7. History of a case

A2.2.7.1 Goal

This part of the site has been realized after discussions with Professor Guzdial. Its goal is to present to the user a totality of interfaces analyzed. The case base will serve as reference to American students to consult the work of their predecessors and to help them to construct their own interfaces.

A2.2.7.2 Content

Address: http://pbl.cc.gatech.edu/interfaces/77.html

A2.2.7.3 Case list

After a brief introduction, the list of all currently implemented cases is proposed to the user. For each case, these information are provided to the user:

- Name of the case
- Positive points in the idea of this case
- Negative points in the idea of this case
- Grade obtained

Cases present in the base are cases that have been created by students of preceding year. The analysis in negative and positive points of each case has been realized by the incumbent professor.

a) The search engine

As for the case base part of this site, the user can access to the different cases thanks to a search engine. This search engine has been implemented in Java and allows to select some case that concerns directly the user. This search engine uses design criteria used in HCI courses at Georgia Tech) to sort cases. A description of these criteria is available in the Tutorial part of the site.

Search engine does not allow to combine several design criteria. Thus, when the user access to the engine, he sees a list of criteria and he has to choose which one concerns him particularly. Once this choice is made, he can target his search more precisely according to the design criteria chosen.

The search results are also presented in the form of a list that summarize the different cases corresponding to user choice. He just has to click on one of these cases to access to it.

b) The structure of cases

When a case has been selected, a summary of what we can find in this case is displayed. This summary includes next points:

- summary of the case: this part is in fact a summary from positive and negatives points
- comments: this part includes different comments that incumbent professor has emitted during the correction of the work
- classification of the case: this part provides a classification of the case in function of the different design criteria

Then, a link allows to return in the case.

A2.2.7.4 Illustration of the search engine

Here is one screenshot of the Java applet realizing the search engine.

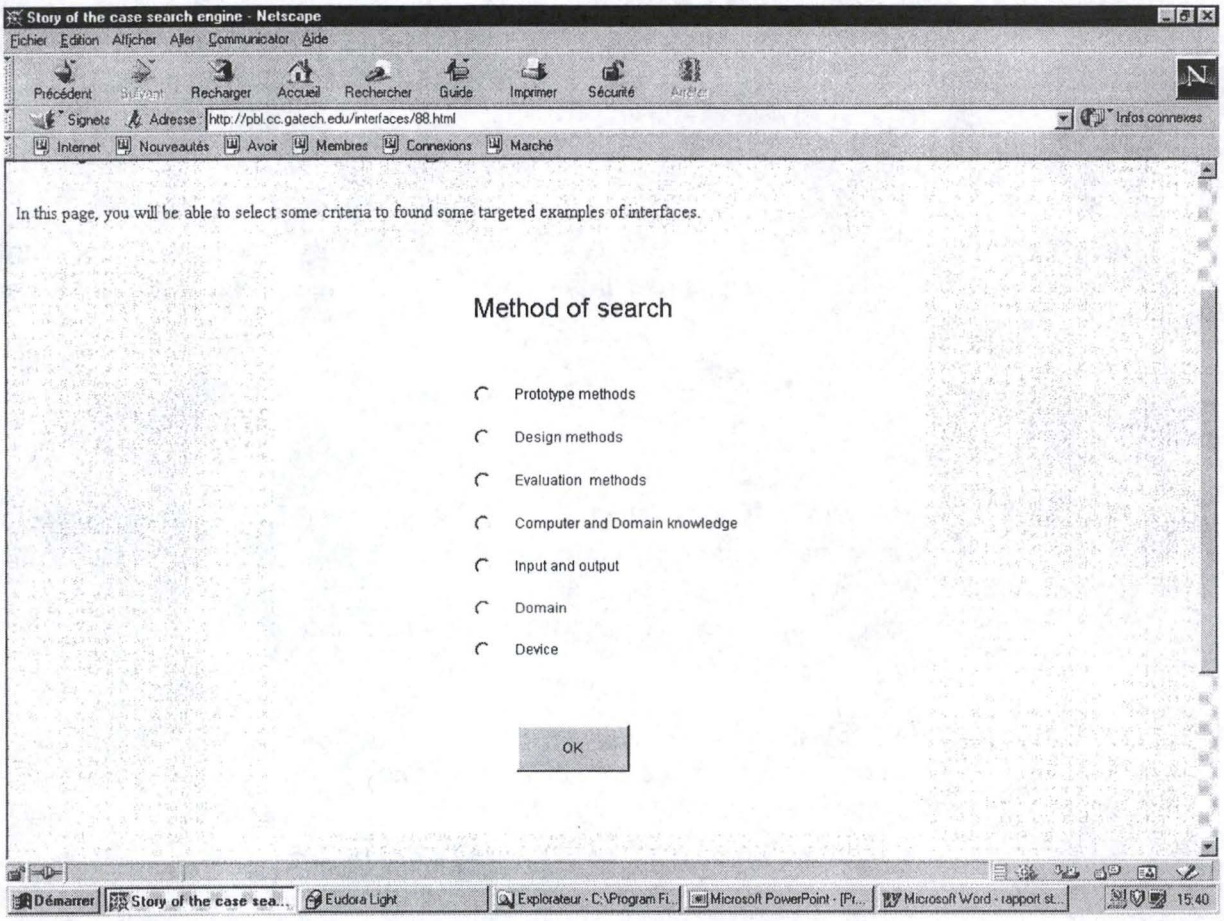


Figure A2-2: Search engine of the history of a case

A2.3. CoWeb features

A2.3.1. Question Page

A2.3.1.1 Goal

This page is independent of the site and serves as a discussion forum between students and also between students and professors.

A2.3.1.2 Functioning

This page is accessible from any other page of the site and opens itself in a new window. So, it's possible to consult a case and to add question in the same time.

To add a question, a reply or a comment, it is very simple: You just need to edit the page.

Users have also the possibility to send us e-mail by clicking on an envelope. A professor can make exactly the same to allow students to ask him questions personally. Nevertheless, this system would break the goal of this forum of discussion.

A2.3.2. General structure of a CoWeb page

Here is a screenshot of a CoWeb page in normal mode.

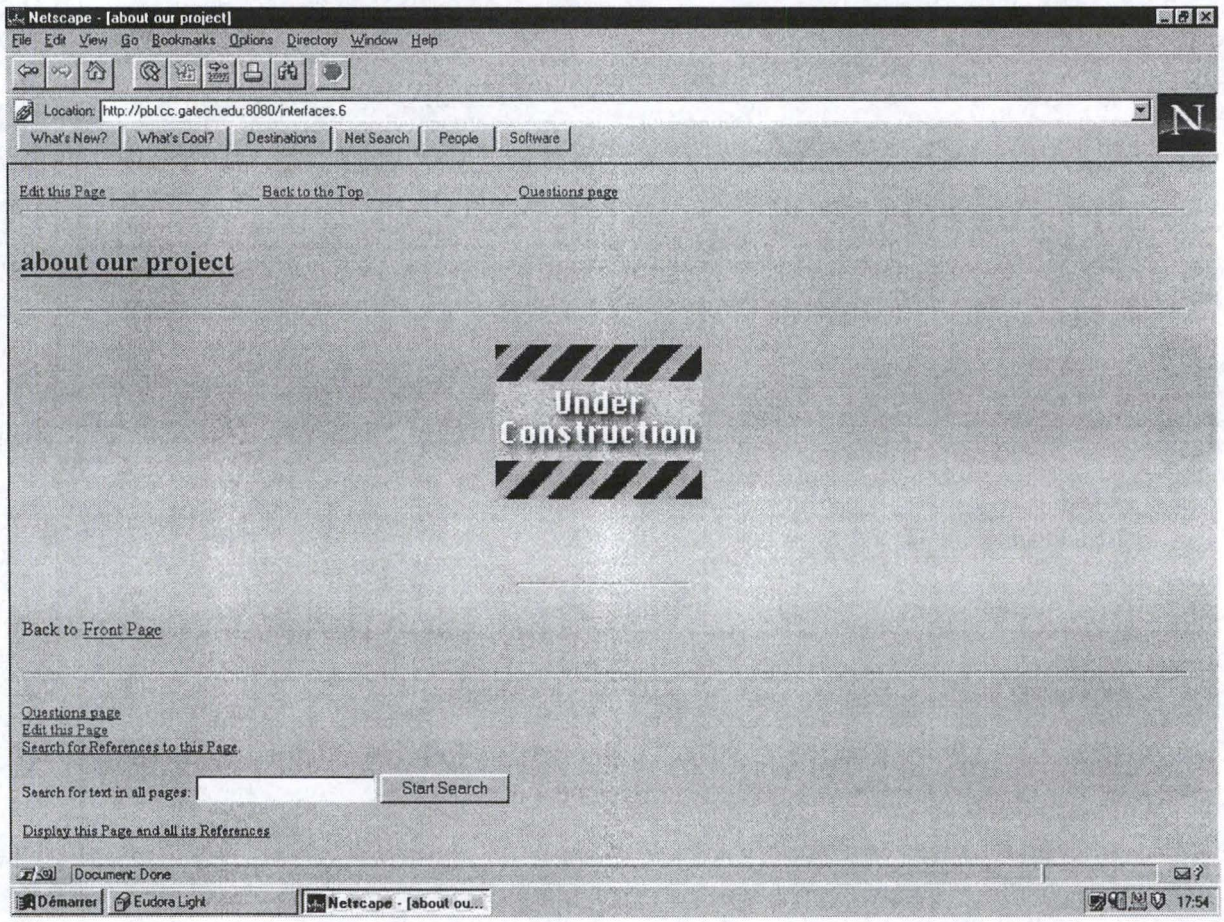


Figure A2-3: General structure of a CoWeb page

In heading, one finds three links:

- "Edit this page " that allows to edit the page
- "Back to the Top " that allows to return to the main page of the site
- "Question page" that opens the question page in a new window

In the bottom of the page, We find four links

- "Question page" that opens the question page in a new window.
- "Edit this page" that allows to edit the page
- "Search for Reference to this page " allows to display all pages making reference to the current page.

- "Display this page and all its references " allows to display all pages to which the current page refers

as well as a search engine allowing to research the text in all pages of the site.

Here is the same page in the edit mode

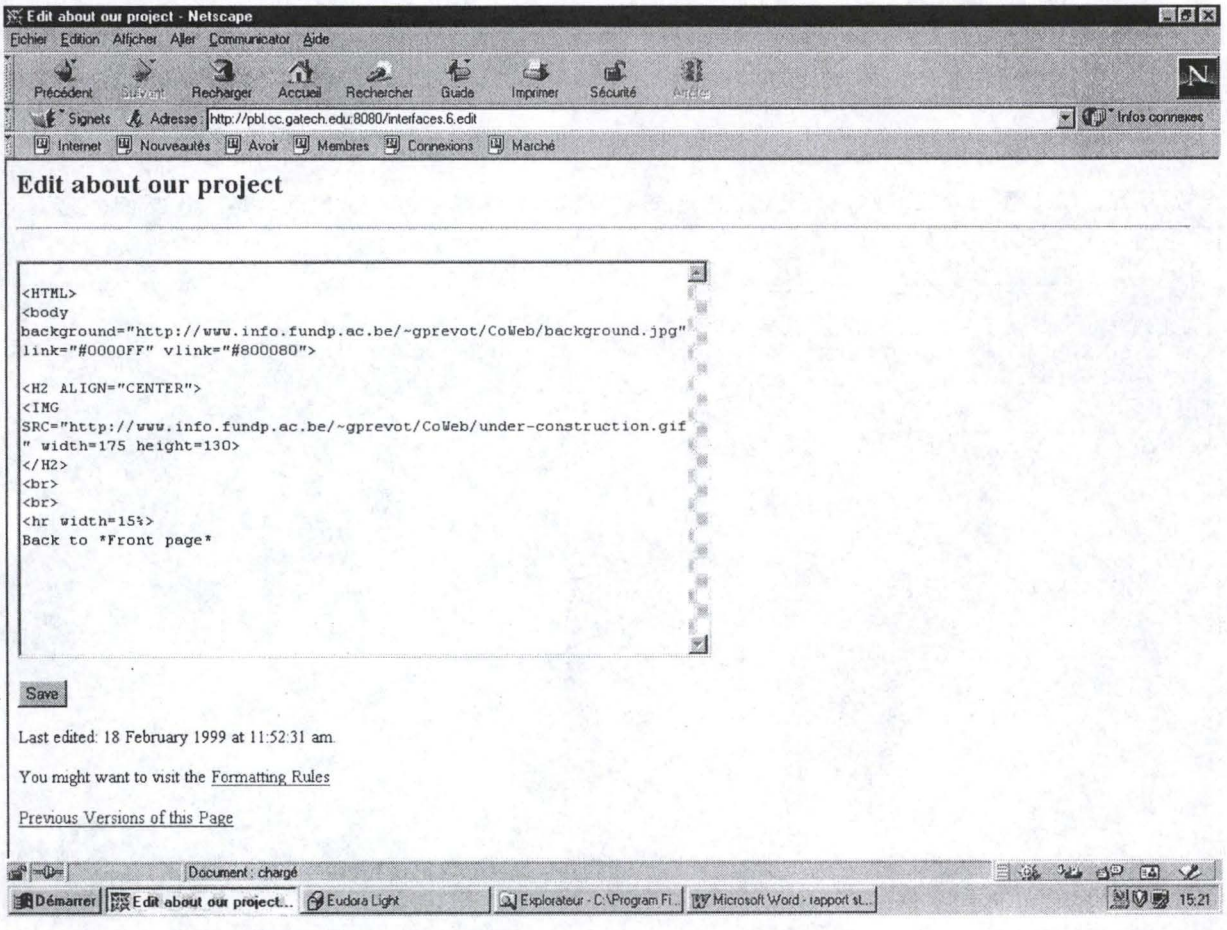


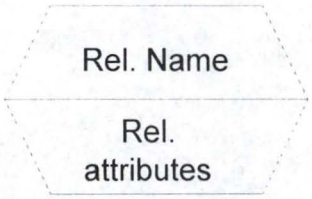
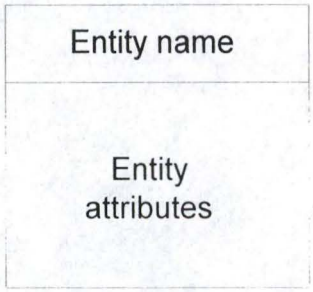
Figure A2-4: CoWeb page in edit mode

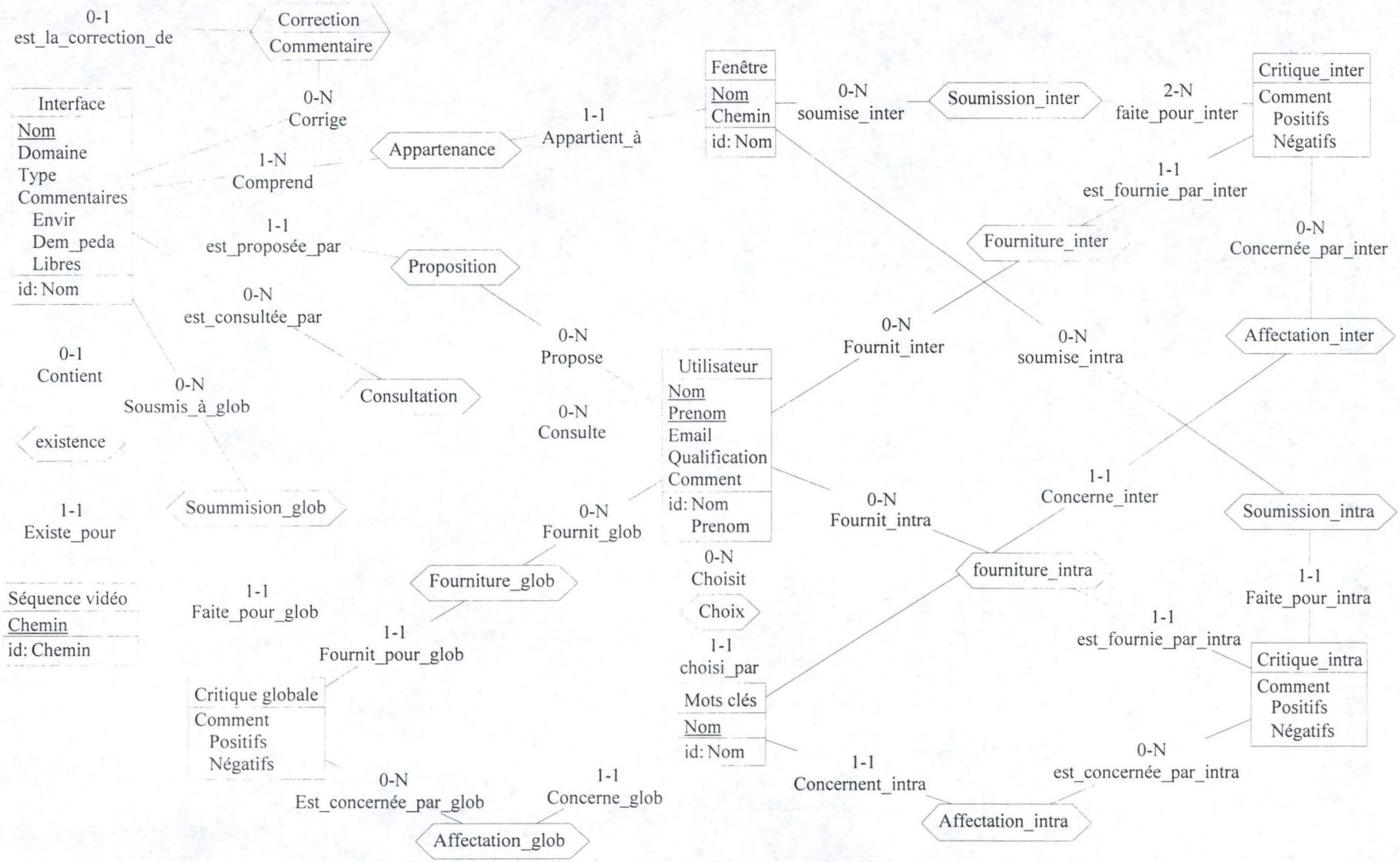
You just have to enter your text in the edit zone. This page allows equally to find the last three versions of the page and allows also to see formatting rules.

Appendix 3:
ER Scheme

A3.1 The entity relationship approach

In this appendix, we will provide the complete ER schema. Before presenting the schema, we make a small summary about legend used.

	This drawing is a relationship between two entities. It must have a name and can have some attributes
	This drawing is an Entity. It must have a name and can have some attributes


0-1
est_la_correction_de
Correction
Commentaire
Interface
Nom
Domaine
Type
Commentaires
Envir
Dem_peda
Libres
id: Nom
0-N
Corrige
1-N
Comprend
Appartenance
1-1
Appartient_à
Fenêtre
Nom
Chemin
id: Nom
0-N
soumise_inter
Soumission_inter
2-N
faite_pour_inter
Critique_inter
Comment
Positifs
Négatifs
1-1
est_fournie_par_inter
1-1
est_proposée_par
Proposition
Fourniture_inter
0-N
Concernée_par_inter
0-N
est_consultée_par
0-N
Propose
0-N
Fournit_inter
0-N
soumise_intra
Affectation_inter
0-1
Contient
0-N
Sousmis_à_glob
Consultation
0-N
Consulte
Utilisateur
Nom
Prenom
Email
Qualification
Comment
id: Nom
Prenom
existence
1-1
Existe_pour
Soummision_glob
0-N
Fournit_glob
0-N
Fournit_intra
1-1
Concerne_inter
Soumission_intra
0-N
Choisit
Séquence vidéo
Chemin
id: Chemin
1-1
Faite_pour_glob
Fourniture_glob
fourniture_intra
1-1
Faite_pour_intra
Choix
1-1
Fournit_pour_glob
1-1
est_fournie_par_intra
1-1
choisi_par
Critique_intra
Comment
Positifs
Négatifs
Critique globale
Comment
Positifs
Négatifs
Mots clés
Nom
id: Nom
0-N
Est_concernée_par_glob
1-1
Concerne_glob
1-1
Concernent_intra
0-N
est_concernée_par_intra
Affectation_glob
Affectation_intra

Appendix 4: **Conversation conception**

A4.1. Introduction

Remind that our chaining graph of function have two UP: UP1 of interface presentation and UP2 of interface critique.

The analysis of the conversation of the application that we present here can be decomposed in three levels:

- Inter-UP level
- Inter-FL level (or intra-UP level)
- Intra-FL level

A4.2. Inter-UP level Conversation

A4.2.1. Objects used

OC-UP1 /* Presentation of the interface */
OC-UP2 /* Critique of the interface */

A4.2.2. Events received

Ev-début-applic	from Ei
Cliquer-suivant	from OC-UP1
Cliquer-précédent	from OC-UP2
Cliquer-quitter	from OC-UP1
Cliquer-terminer	from OC-UP2
Cliquer-terminer	from OC-UP1

Rem : Ei is an initial event that starts the application.

A4.2.2.1 Schema

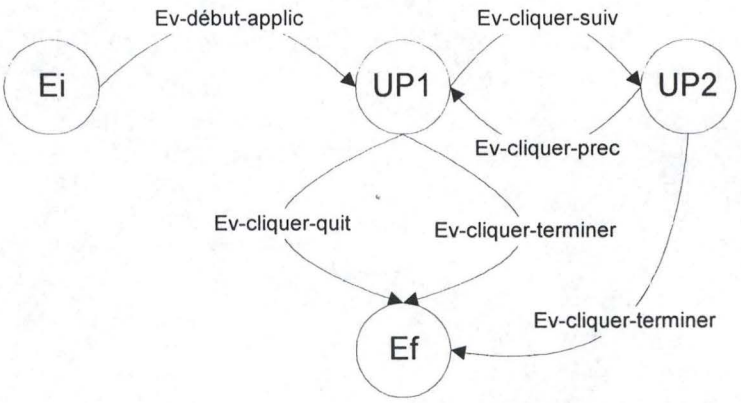


Figure A4-1: Inter UP conversation

A4.2.2.2 Pseudo-code

a) Initial State

```
If Ev-début-applic
Begin
        Activate Script présentation de l'interface
        State = Présentation
End
```

b) Transitions States

```
If State = Présentation and Cliquer-quitter
Begin
        Deactivate Script présentation de l'interface
        State = Application finie
End


If State = Présentation and Cliquer-terminer
Begin
        Deactivate Script présentation de l'interface
        State = Application finie
End


If State = Présentation and Cliquer-suivant
Begin
        Deactivate Script présentation de l'interface
        Activate Script critique de l'interface
        State = Critique
End
```

```
If State = Critique and Cliquer-précédent
Begin
        Deactivate Script critique de l'interface
        Activate Script présentation de l'interface
        State = Présentation
End

If State = Critique and Cliquer-terminer
Begin
        Deactivate Script critique de l'interface
        State = Application finie
End
```

A4.3. Inter-FL level Conversation

At the end of this point, we will give the schemas of the inter-FL conversation from our two UP (with legend).

We analyze in details the conversation between the two windows :

- F1 : Données sur l'insérant
- F2 : Présentation de l'interface

A4.3.1. Objets used

OC-F1 /* Données sur l'insérant */
OC-F2 /* Présentation de l'interface */

A4.3.2. Received events

Ev-début-applic	from Ei
Cliquer-suivant	from OC-F1
Cliquer-précédent	from OC-F2

A4.3.3. Generated Events

Cliquer-suivant	to OC-F2
Cliquer-précédent	to OC-F1

A4.3.3.1 Pseudo-code

a) Initial State

```
If Ev-début-applic
Begin
        Activate Script données sur l'insérant
        State = Insérant
End
```

b) Transitions states

```
If State = Insérant and Cliquer-suivant
Begin
        Deactivate Script données sur l'insérant
        Activate Script présentation de l'interface
        State = Présentation
End


If State = Présentation and Cliquer-précédent
Begin
        Deactivate Script présentation de l'interface
        Activate Script données sur l'insérant
        State = Insérant
End
```

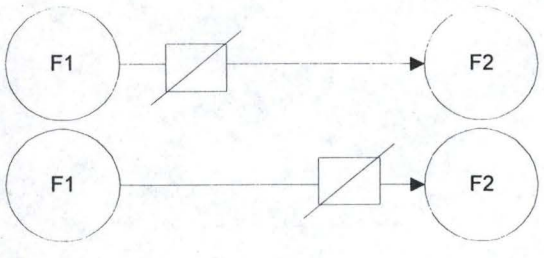
A4.3.3.2 Schemas of the conversations of the two UP

a) General legend elements

Opening of a window

Closing of a window

Remark that the window concerning by the opening or by the closing is the closest window to the shape. For example:



Closing of F1

Closing of F2

b) Legend of the schema of the UP1 "Présentation de l'interface"

F1 : Données sur l'insérant
F2 : Présentation de l'interface
F3 : Correction
F4 : Chemin de la séquence vidéo
F5 : Fenêtres composant l'interface
F6 : Résumé de la présentation

1 : Ev-cliquer-quitter
2 : Ev-cliquer-suivant
3 : Ev-cliquer-précédent
4 : Ev-cliquer-treminer
5 : Ev-cliquer-suivant and Séq-vidéo = oui and Correction = oui
6 : Ev-cliquer-précédent and Séq-vidéo = oui and Correction = oui
7 : Ev-cliquer-suivant and Séq-vidéo = non and Correction = oui
8 : Ev-cliquer-précédent and Séq-vidéo = non and Correction = oui
9 : Ev-cliquer-suivant and Séq-vidéo = oui and Correction = non
10 : Ev-cliquer-précédent and Séq-vidéo = oui and Correction = non
11 : Ev-cliquer-suivant and Séq-vidéo = non and Correction = non
12 : Ev-cliquer-précédent and Séq-vidéo = non and Correction = non
13 : Ev-critique

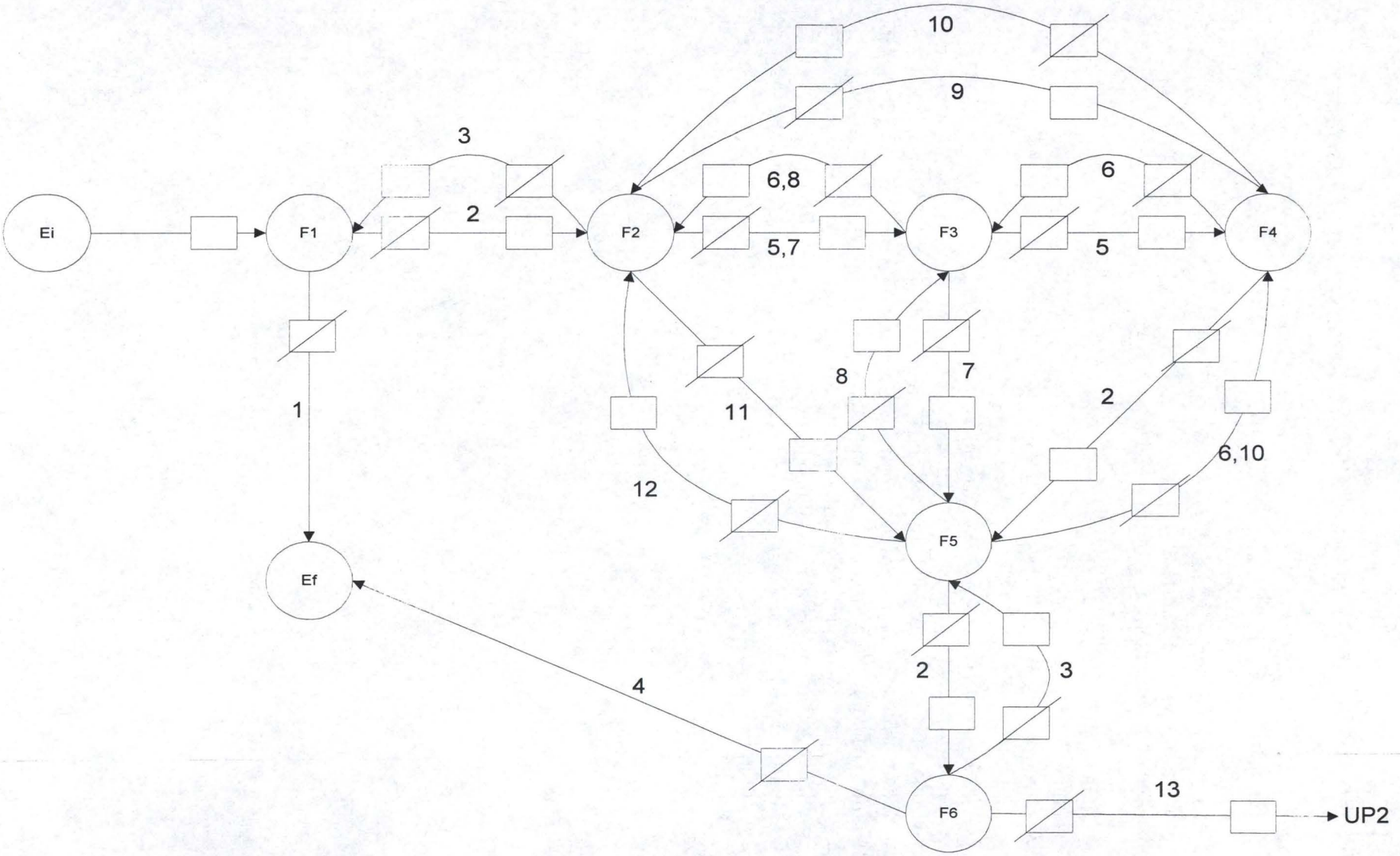


Figure A4-2: UP1 conversation

c) Legend of the schema of the UP2 "Critique de l'interface"

F7 : Choix de la critique
F8 : Critique globale
F9 : Choix des deux fenêtres à critiquer
F10 : Critique inter-fenêtre
F11 : Choix de la fenêtre à critiquer
F12 : Critique intra-fenêtre
F13 : Mots clés
F14 : Résumé de la critique

1 : Ev-cliquer-précédent
2 : Ev-cliquer-suivant
3 : Ev-cliquer-critique-globale
4 : Ev-cliquer-critique-inter
5 : Ev-cliquer-critique-intra
6 : Ev-cliquer-OK
7 : Ev-cliquer-Annuler
8 : Ev-cliquer-nouvelle
9 : Ev-cliquer-terminer

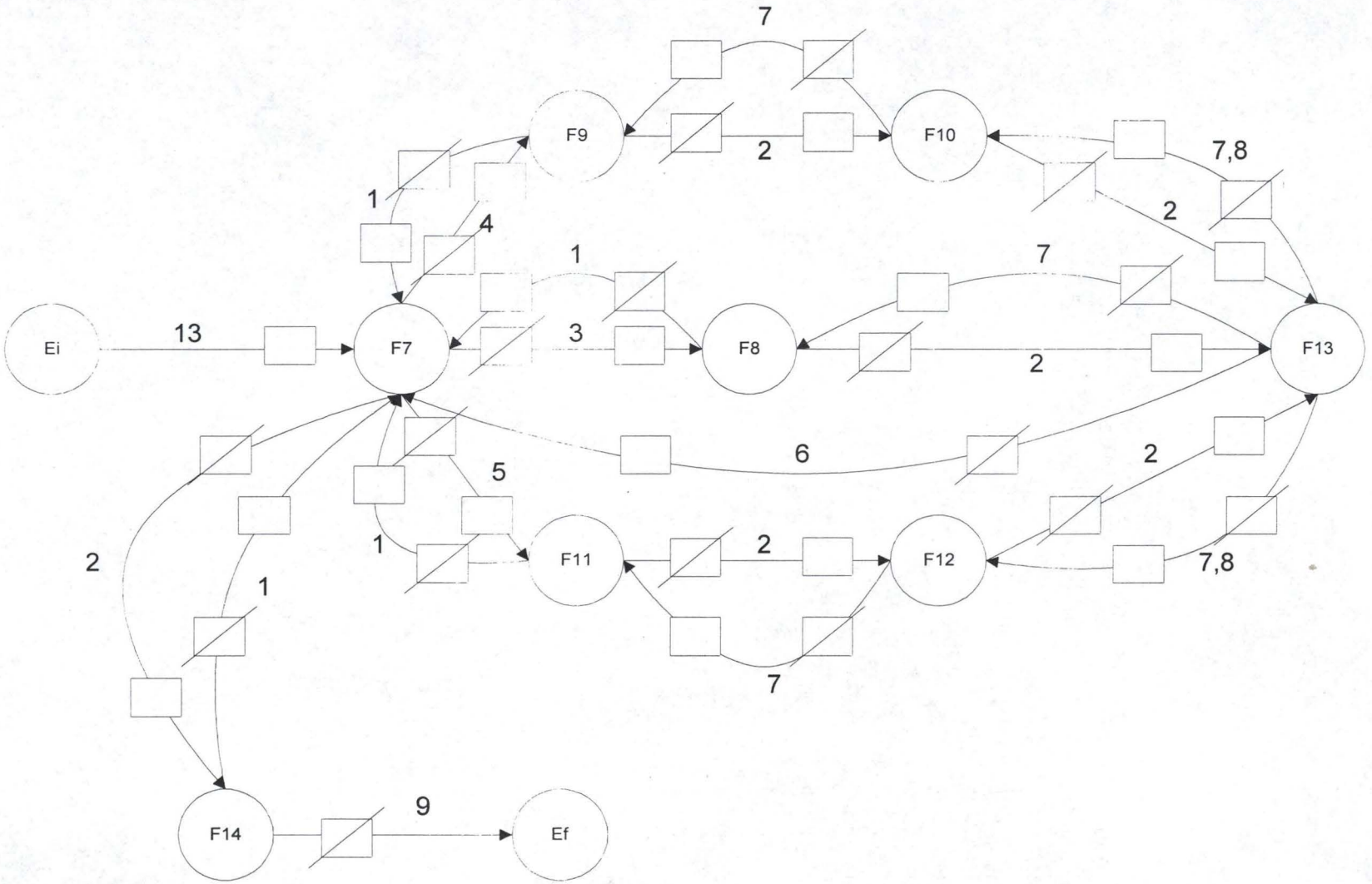


Figure A4-3: UP 2 conversation

A4.4. Intra-FL level Conversation

We describe here the interactions in the F2 window « Présentation de l'interface ».

A4.4.1. Objets used

Icône
Labels
EditBox
ListBox-type-util
Boîte à onglet
Bt-précédent
Bt-suivant

A4.4.2. Used by

OC-UP1

A4.4.3. Received events

Ev-select-listbox-type-util
Ev-select-listbox-type-cas
Ev-select-listbox-séq-vidéo
Ev-select-listbox-correction
Ev-select-btprécédent
Ev-select-btsuivant
Ev-select-boîte-environnement
Ev-select-boîte-démarche
Ev-select-boîte-libres

A4.4.4. Generated events

Ev-listbox-type-util
Ev-listbox-type-cas
Ev-listbox-séq-vidéo
Ev-listbox-correction
Ev-btsuivant
Ev-btprécédent
Ev-boîte-environnement
Ev-boîte-démarche

Ev-boîte-libres

A4.4.4.1 Pseudo-code

a) Initial State

```
Begin
        Show Fenêtre F2
        All the buttons are active
        Listbox initialized
        State = *
End
```

b) Transitions States

```
If State = * and Ev-select-listbox-type-util
Begin
        Generate Ev-listbox-type-util
End


If State = * and Ev-select-listbox-type-cas
Begin
        Generate Ev-listbox-type-cas
End


If State = * and Ev-select-listbox-séq-vidéo
Begin
        Generate Ev-listbox-séq-vidéo
End


If State = * and Ev-select-listbox-correction
Begin
        Generate Ev-listbox-correction
End


If State = * and Ev-select-btprécédent
Begin
        Generate Ev-btprécédent
End


If State = * and Ev-btsuivant
 Begin
        Generate Ev-btsuivant
```

End

If State = * and Ev-select-boîte-environnement
Begin
    Generate Ev-boîte-environnement
End

If State = * and Ev-select-boîte-démarche
Begin
    Generate Ev-boîte-démarche
End

If State = * and Ev-select-boîte-libres
Begin
    Generate Ev-boîte-libres
End

Appendix 5:
Detailed example

A5.1. Detailed example

A5.1.1. Context of the example

Different uses of our case base can be made. It will be useless to illustrate each one. So, we decided to take the most complex or rather the most complete and to illustrate it by providing screenshots of the different stages of the case insertion. Let's begin with the context.

Patrick Kreps is student in last year of his Master degree in Computer Science in the "Facultés Universitaires Notre-Dame de la Paix" in Namur. He is very interested in HCI and he heard that, complementary to the HCI course, it exists a case base that can help students to practice interface critiques. So, he decides to use it by inserting a new case, by criticizing it and by asking to others to see and to comment his work.

A5.1.2. Example

A5.1.2.1 Patrick identification

The first window that Patrick can see is the identification window. He has just to fill in the different fields and can go on by clicking on "Suivant".

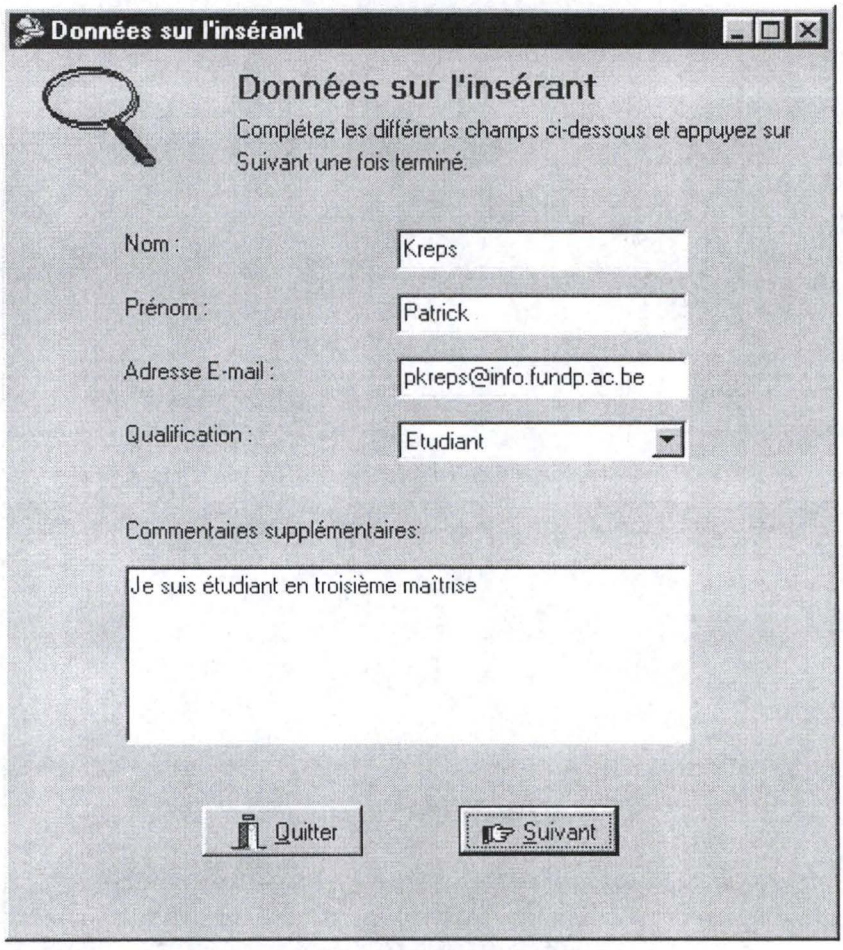


Figure A5-1: Patrick identification

A5.1.2.2 Interface presentation

Now that we know who he is, we need to know some more about the interface he plans to add. He fills in the different fields and he decides also to add some free comments concerning the environment of the interface.

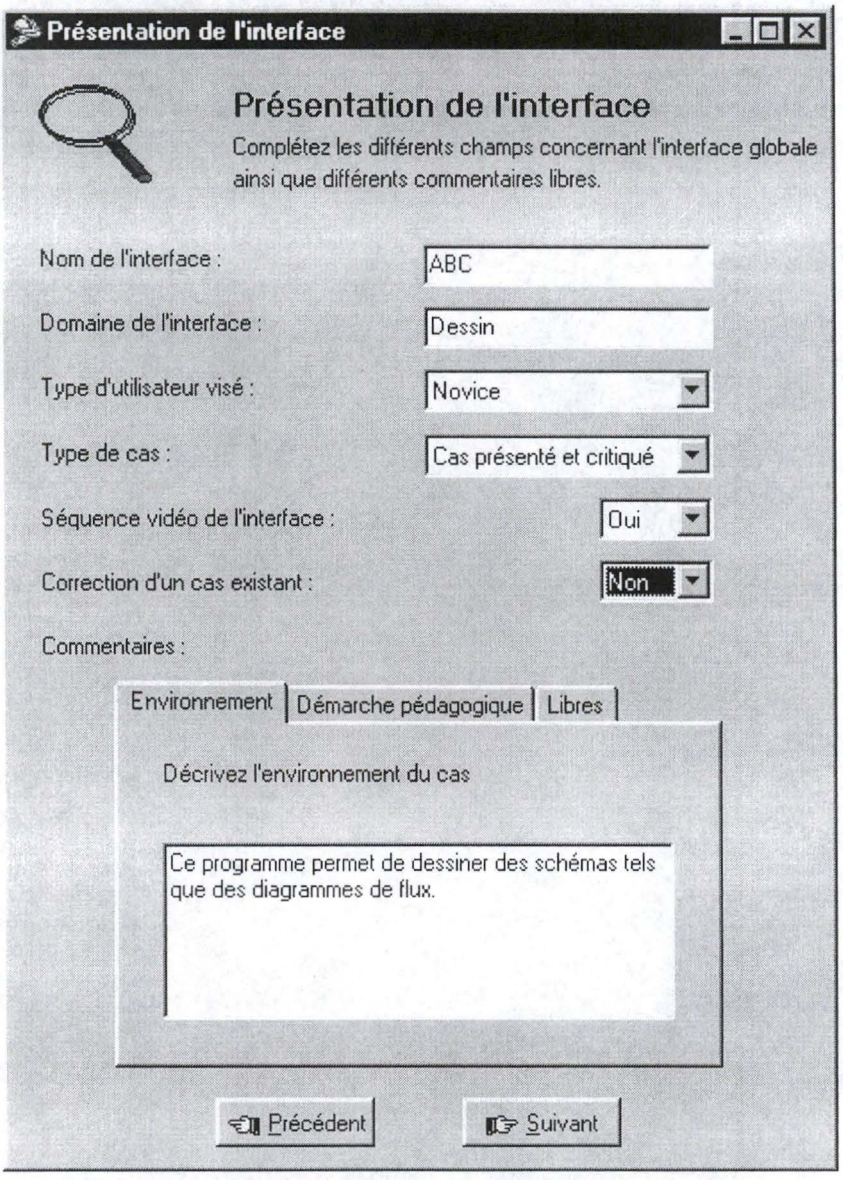


Figure A5-2: Interface presentation

A5.1.2.3 Path of the video sequence

The interface that Patrick wants to add includes a video sequence. To be able to transfer it from Patrick's disk to Vesale database, he must indicate where we can found it.

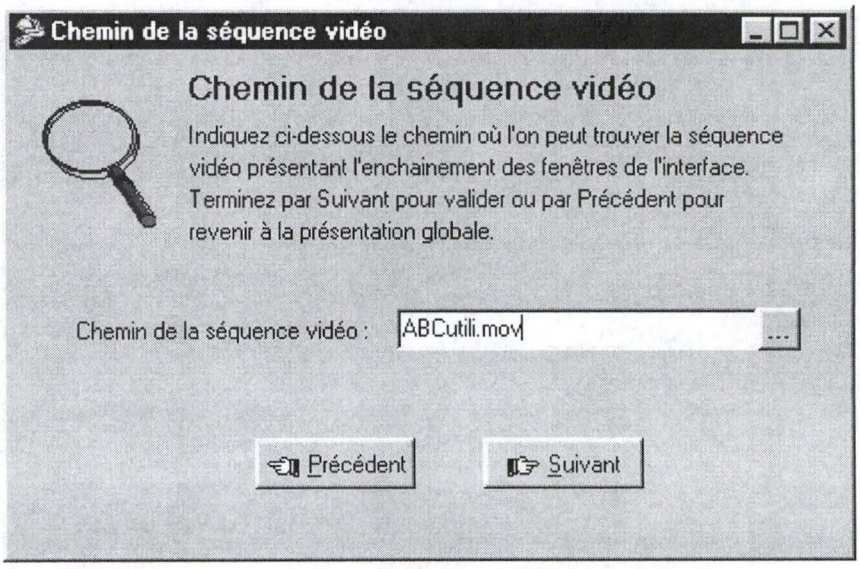


Figure A5-3: Path of video sequence

A5.1.2.4 Windows composing interface

Patrick must now "composed" his interface with the different windows. At this purpose, he disposes of the following window. He must indicate the name of the window and the path of its screenshot. He can then validate this window by clicking on the "Ajouter" button.

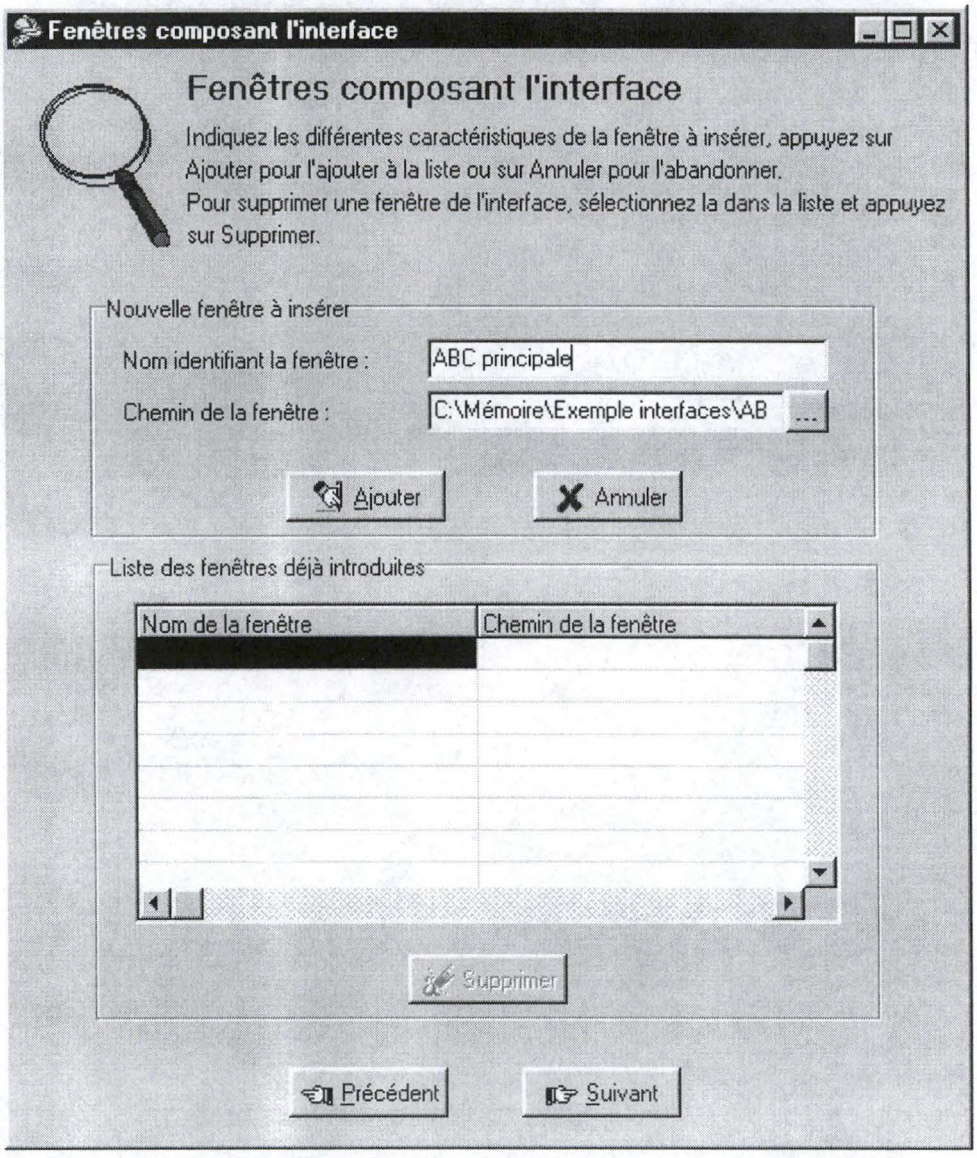


Figure A5-4: Windows composing interface (1)

Here is the result after validation of the window

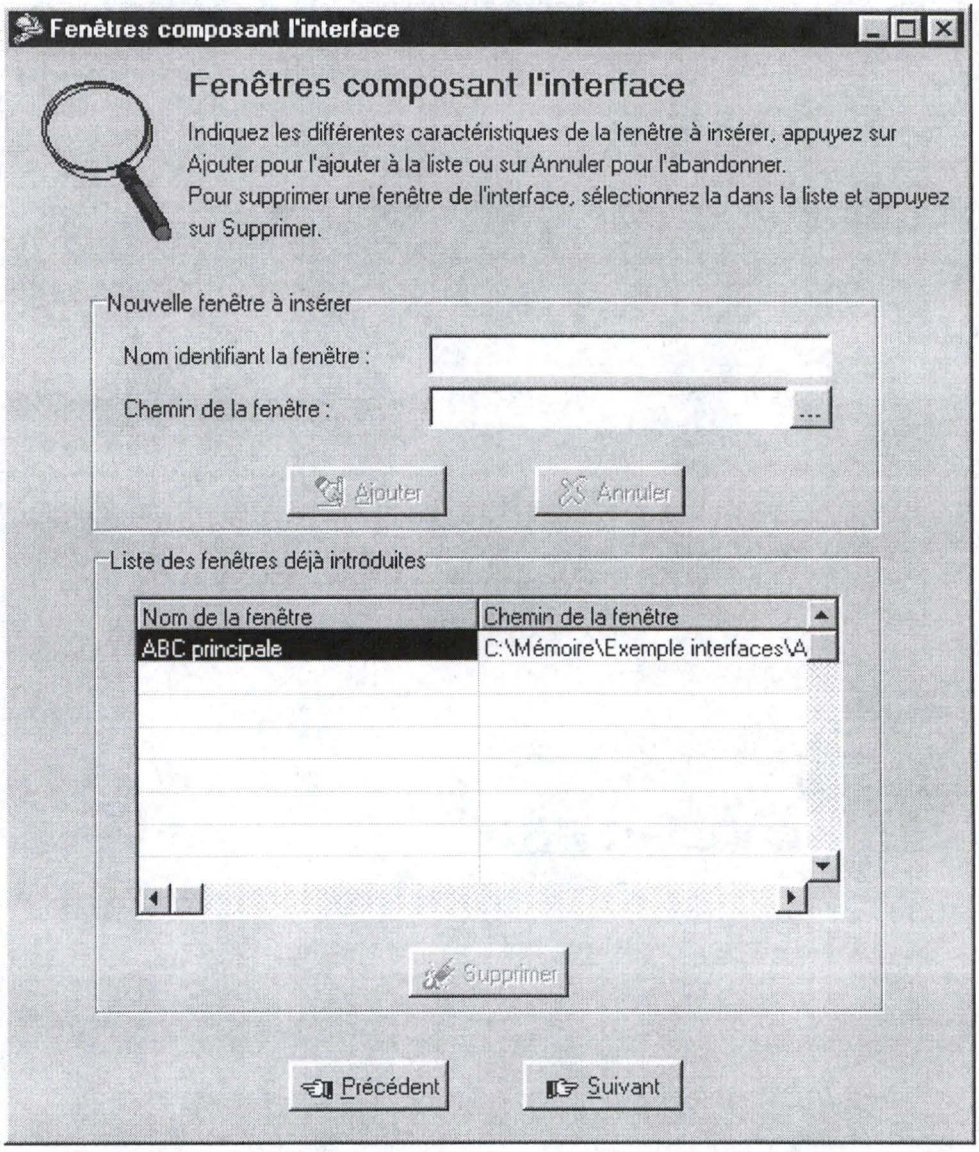


Figure A5-5: Windows composing interface (2)

Three windows compose Patrick interface. So, he has to repeat the process three times. Here is the final result

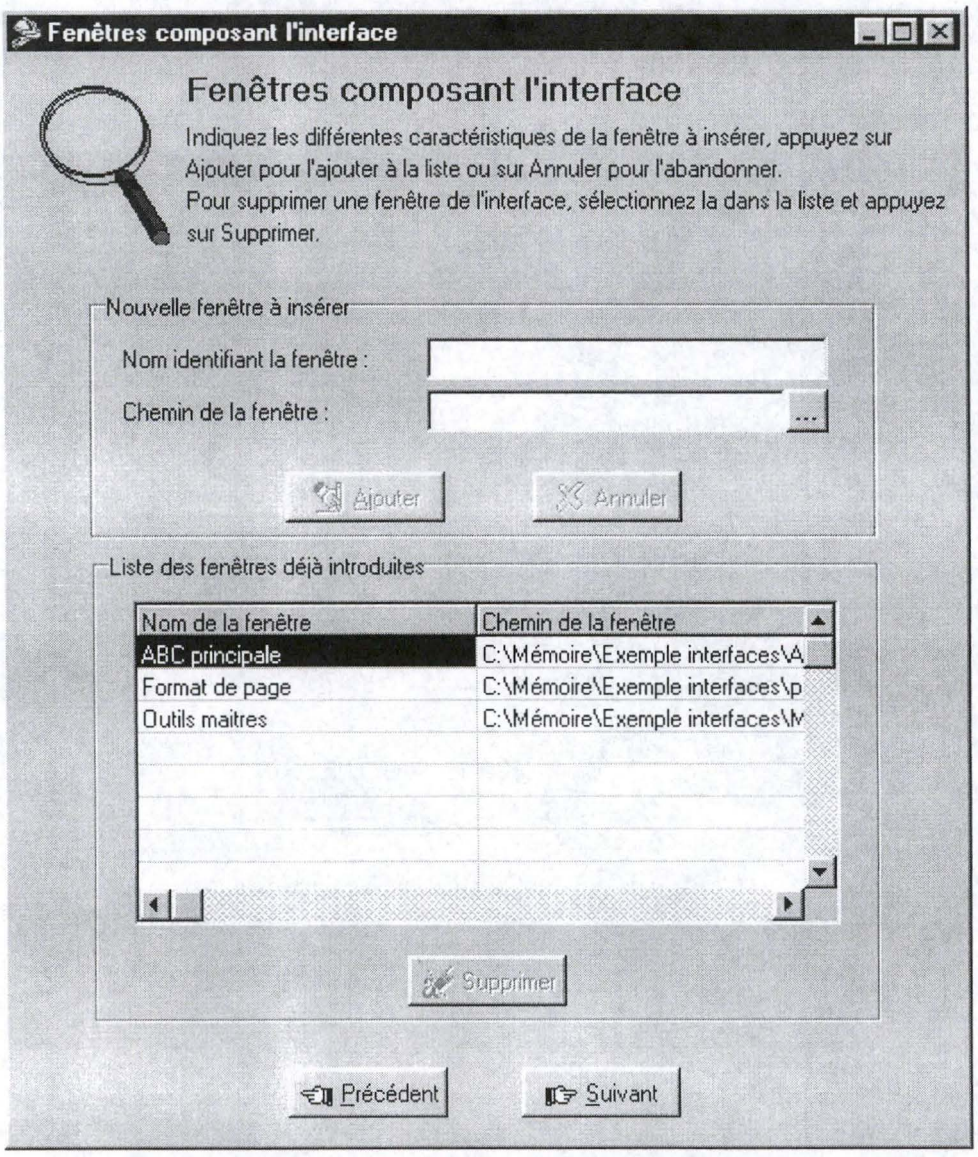


Figure A5-6: Windows composing interface (3)

A5.1.2.5 Choice of the critique (1)

The next window appearing is the window allowing Patrick to make his different critiques. He can make a global, inter or inta-windows critique. Patrick wants first to make a global critique. He clicks then on the "Critique globale" button.

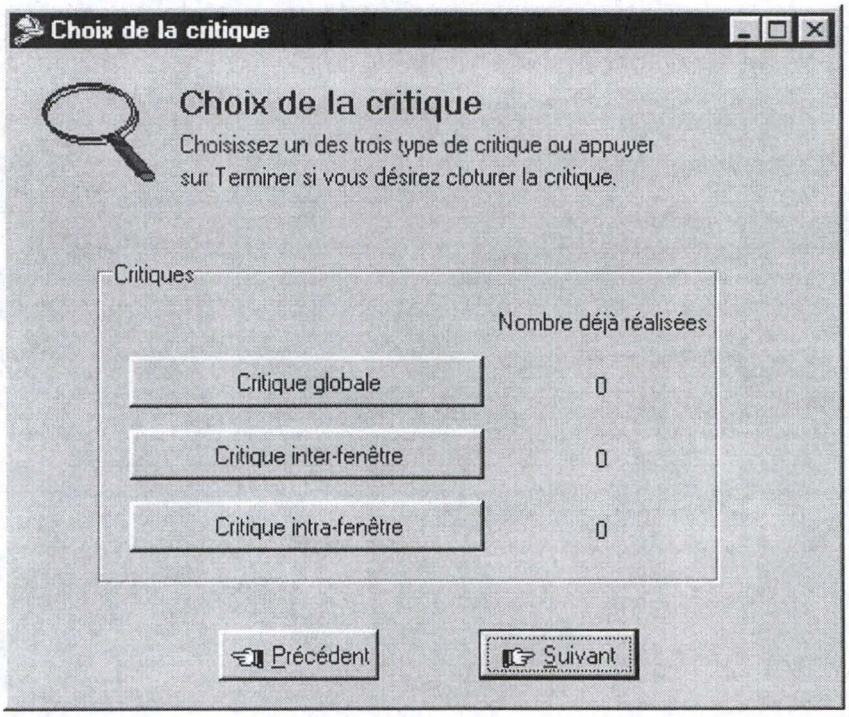


Figure A5-7: Choice of the critique

A5.1.2.6 Global critique

He can see the first window of his interface and can click on both arrows to see other windows. He decides first to make a comment about the interface as compared to task environment. He finds that it is not very good. He clicks then on right arrow to see the next window.

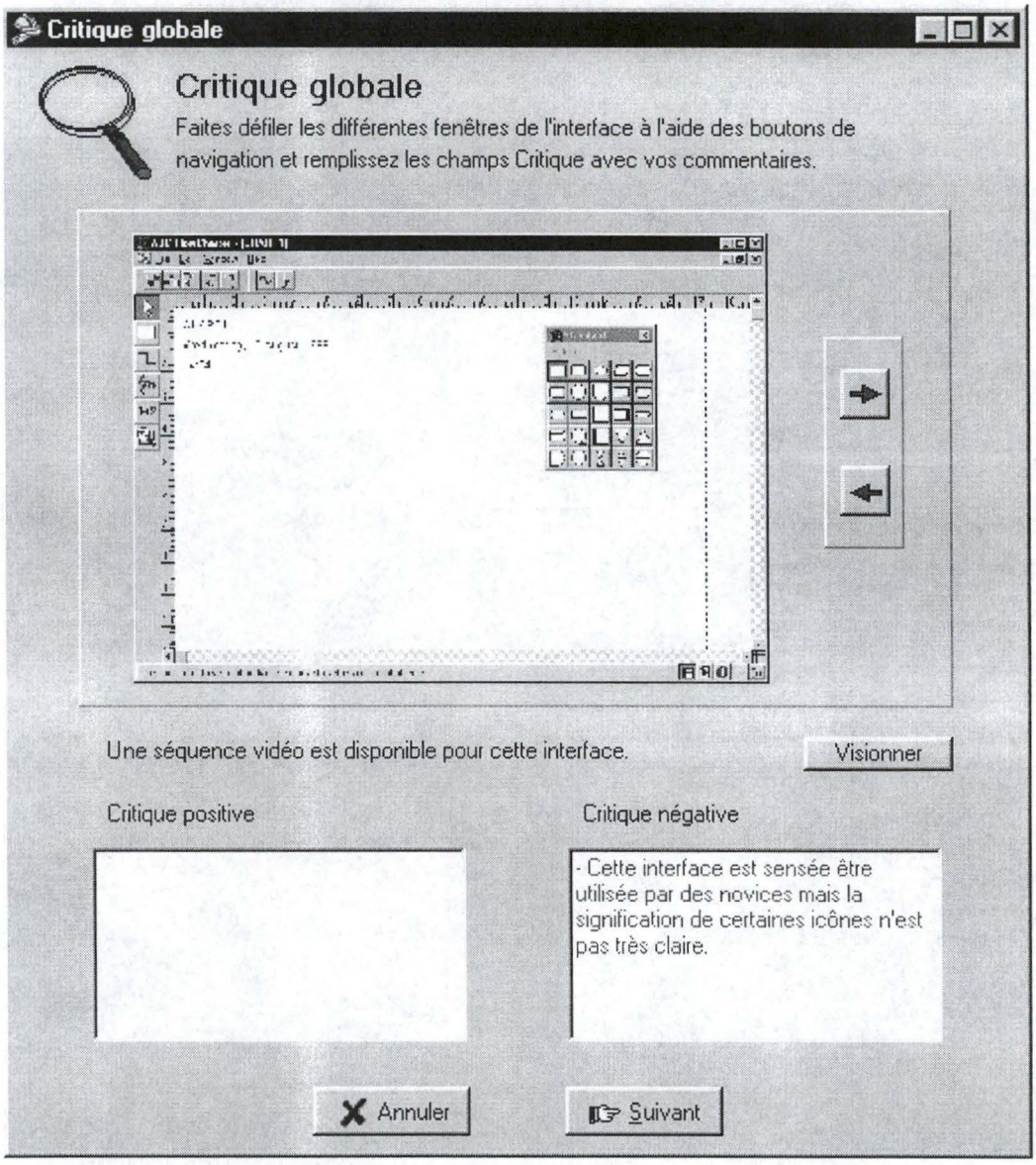


Figure A5-8: Global critique (1)

He makes here a positive critique because he found that fonts used are good. He doesn't see anything to add at this time and he clicks on "Suivant".

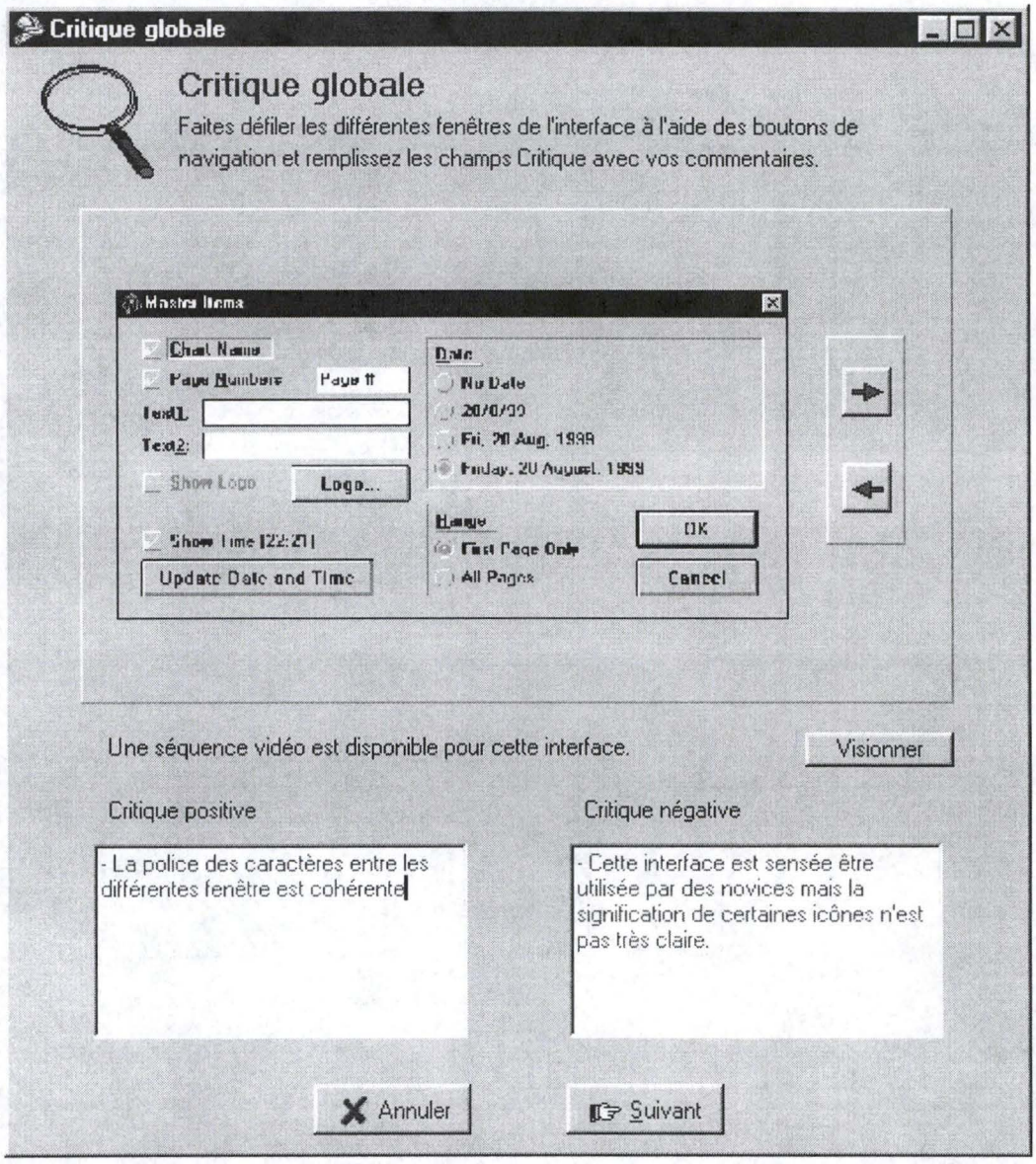


Figure A5-9: Global critique (2)

Windows with keywords concerning global critique appears then. In one of his comment, Patrick speaks about consistency (cohérence). This keyword is then preselected. He doesn't see any other keyword corresponding to his critique and he validates then his global critique.

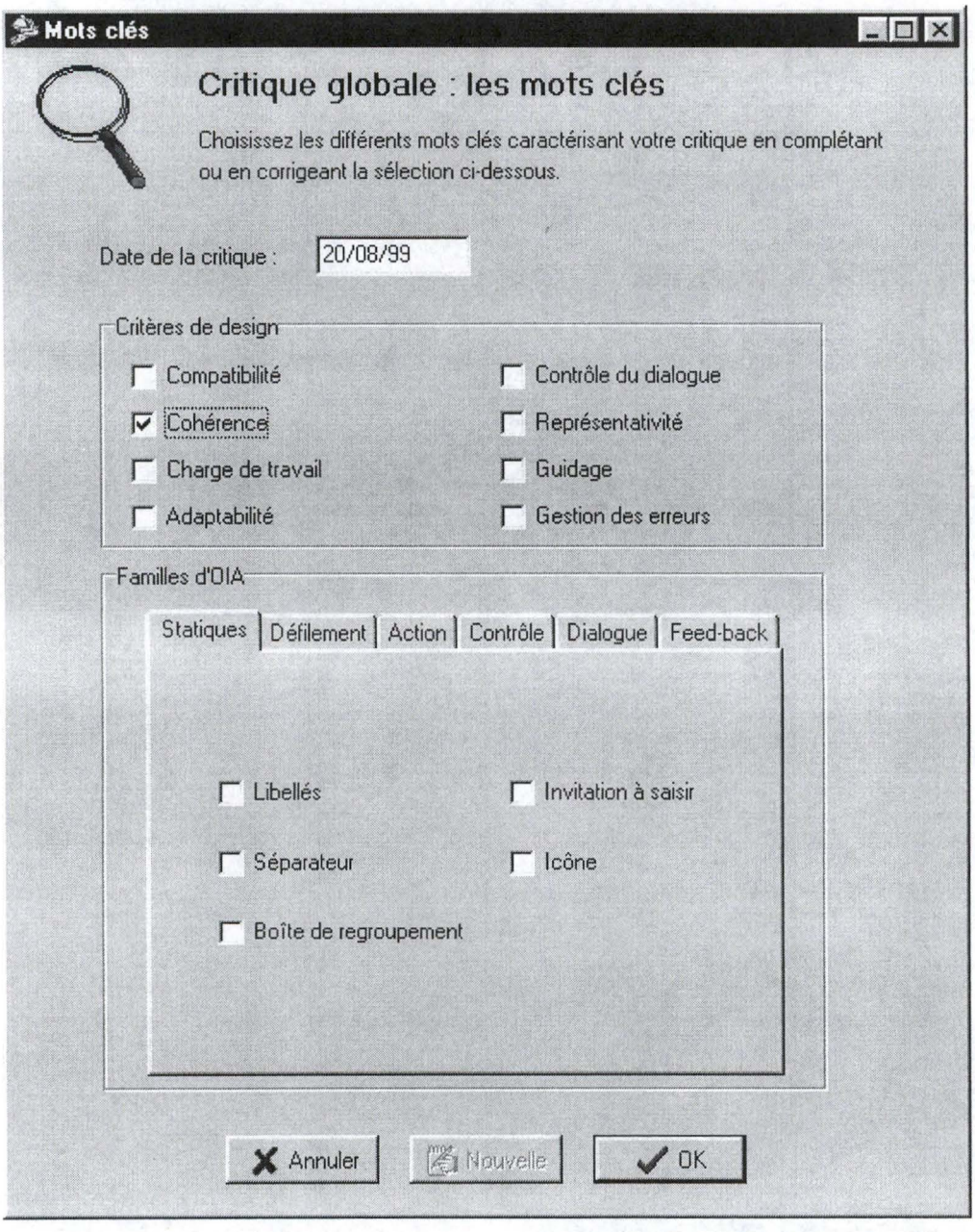


Figure A5-10: Global critique, keywords

A5.1.2.7 Critique choice (2)

Patrick comes again to the critique choice window. He can see that his global critique has been validated. He wants now to make an inter-windows critique and click then on corresponding button.

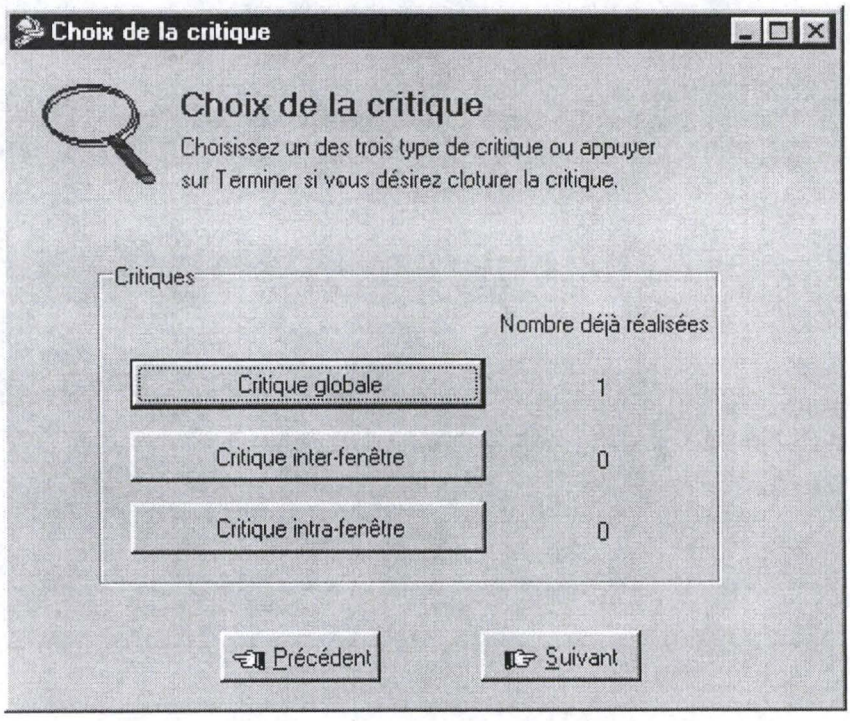


Figure A5-11: Critique choice (2)

A5.1.2.8 Inter-windows critique

He has first to choose the two windows he wants to compare. He decides to compare "Format de page" with "Outils maitres". Note that each time he selects an item in one of the two lists, he can see its illustration. In this case, the last chosen item is "Outils maitres"

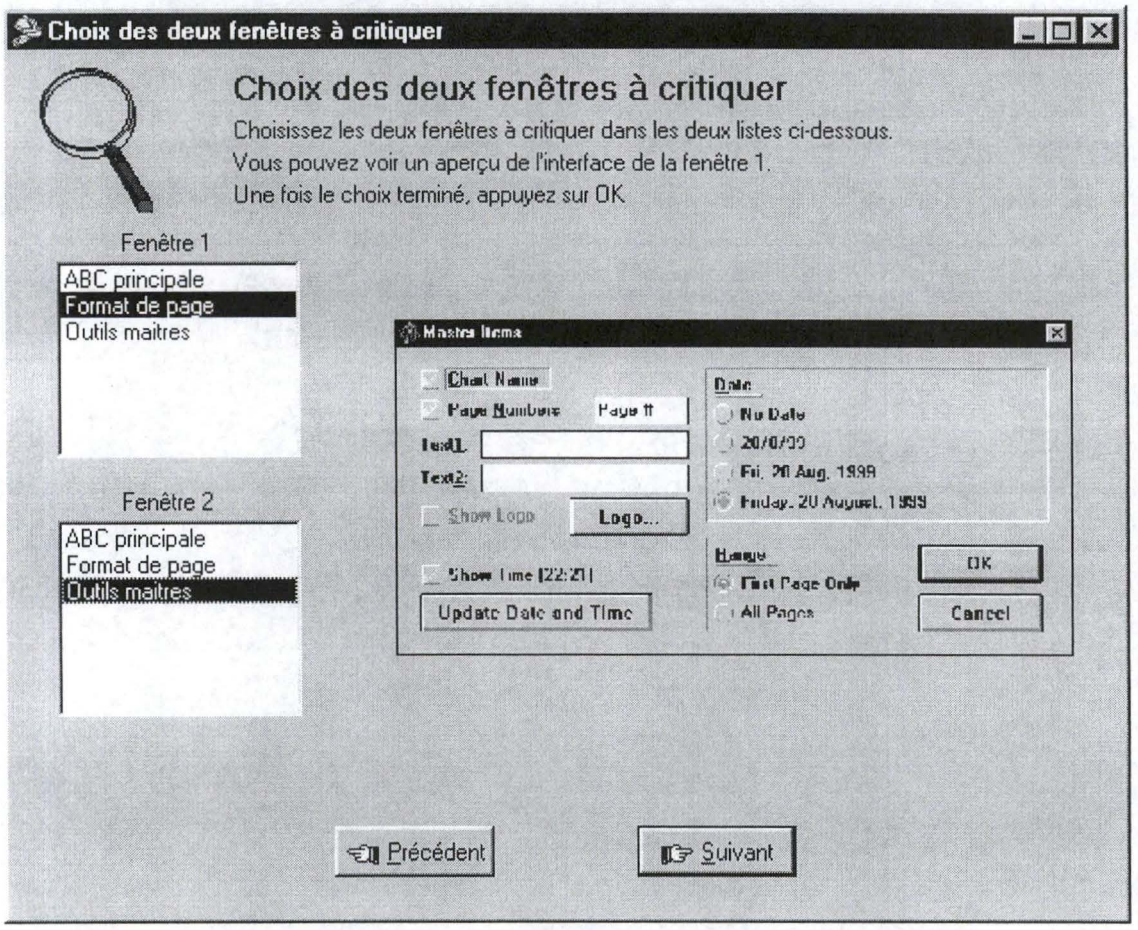


Figure A5-12: Inter-windows critique, the choice

He comes to the inter-windows critique where he can see the two chosen windows and the graphical tool. He would like to mark interfaces but it's not easy because illustrations are small. He decides then to use zoom function for "Format de page".

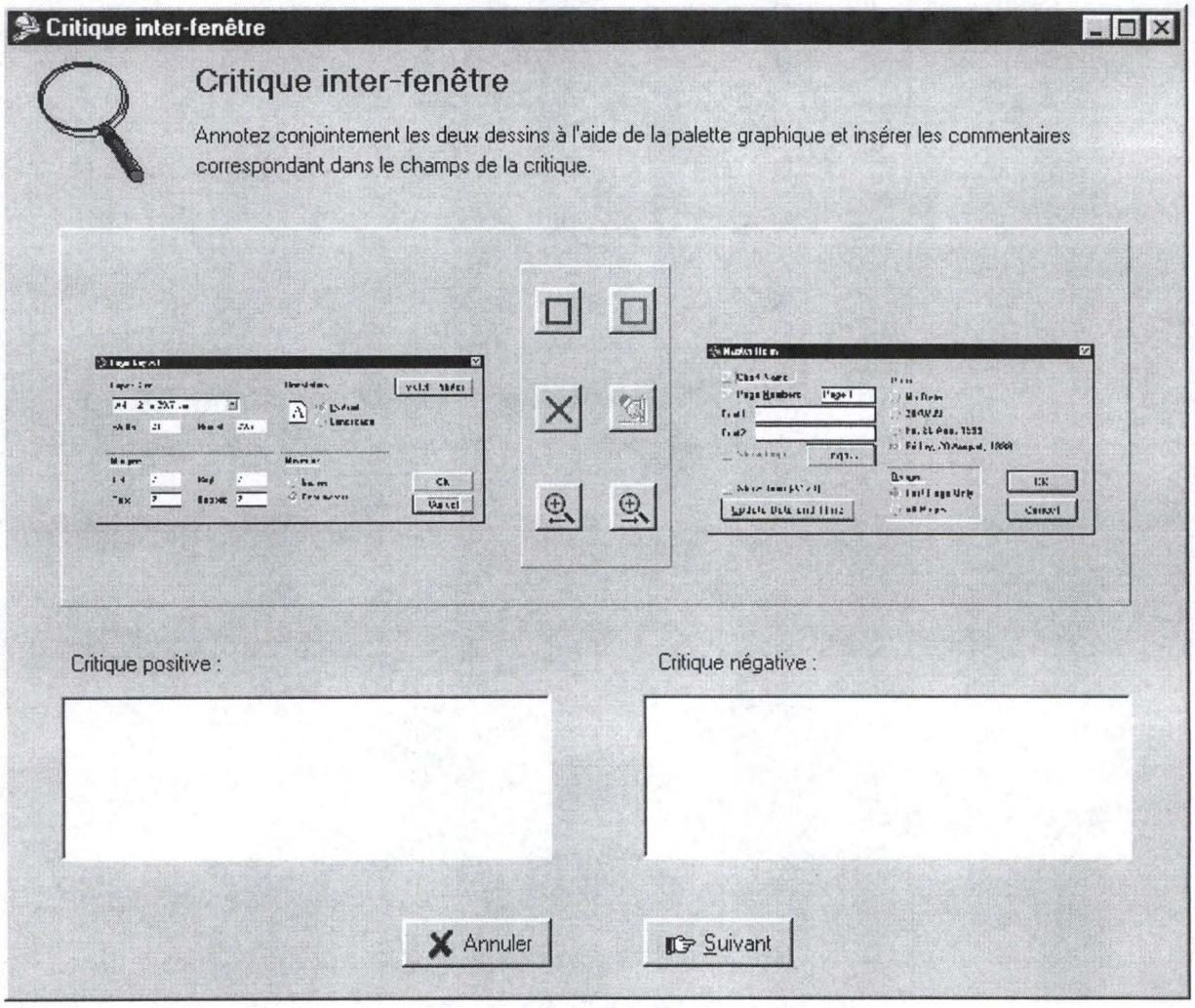


Figure A5-13: Inter-windows critique (1)

He can then see illustration in real size and can draw his first rectangle. Because he wants to make a negative critique, he uses red rectangle.

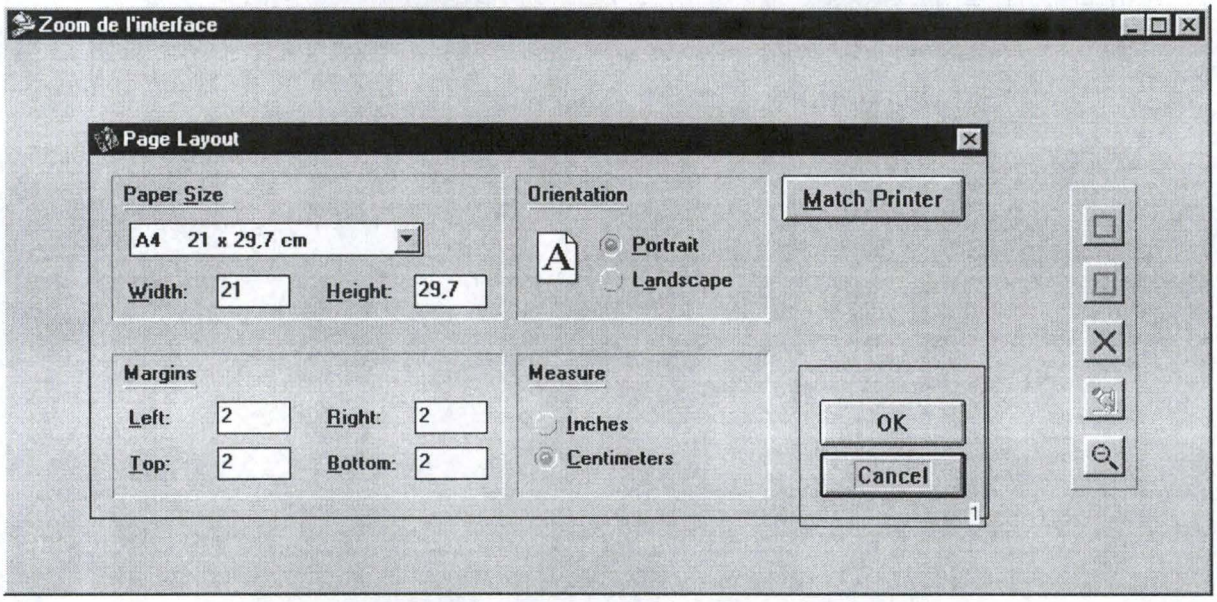


Figure A5-14: Inter-windows critique, zoom

He makes the same for the second window and can add his critique. He has nothing more to add and click on "Suivant".

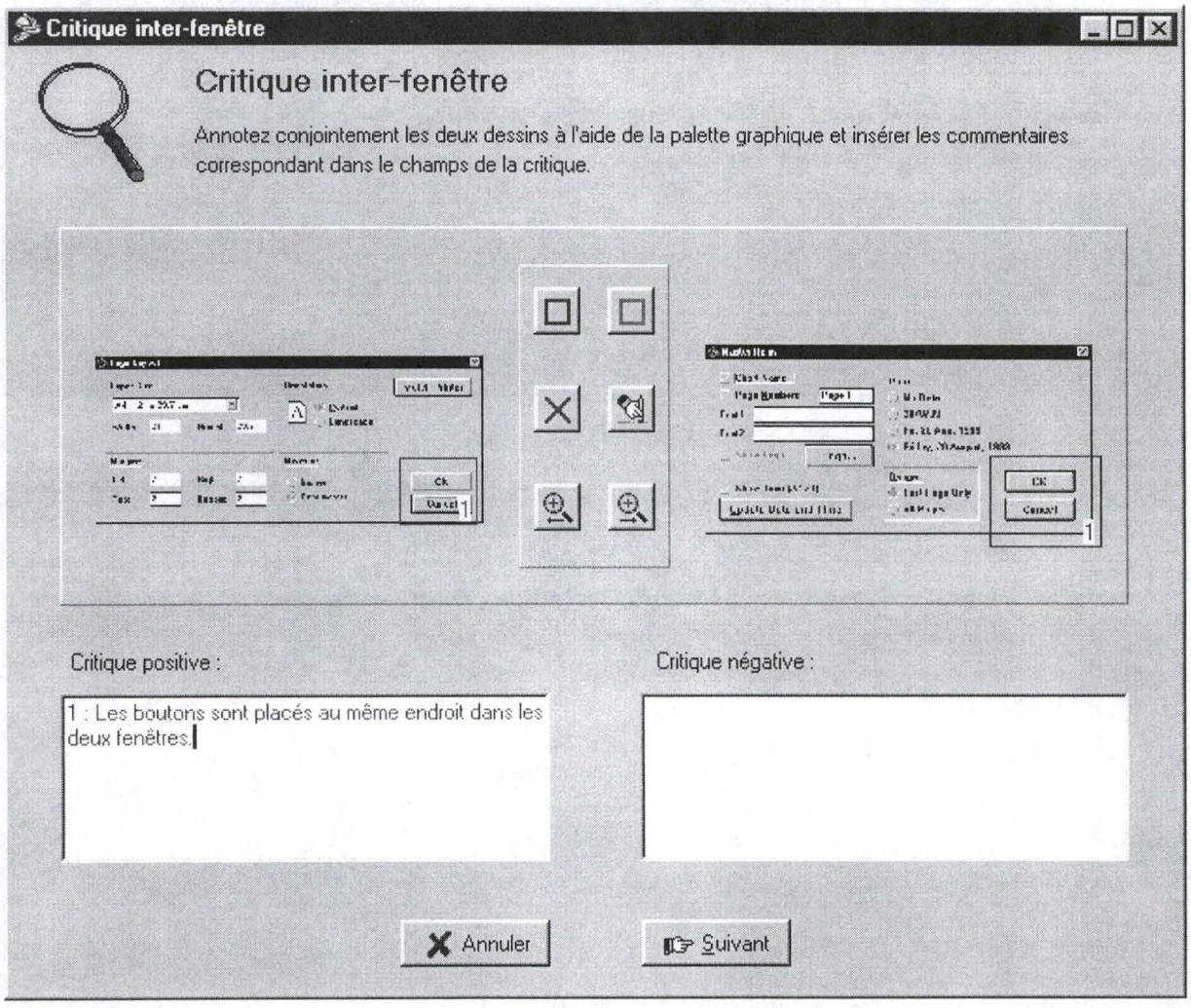


Figure A5-15: Inter-windows critique (2)

He arrives on keyword window where "fenêtre" is preselected. He agrees with it and doesn't want to insert a new inter-window critique. He validates then critique one by clicking on "OK".

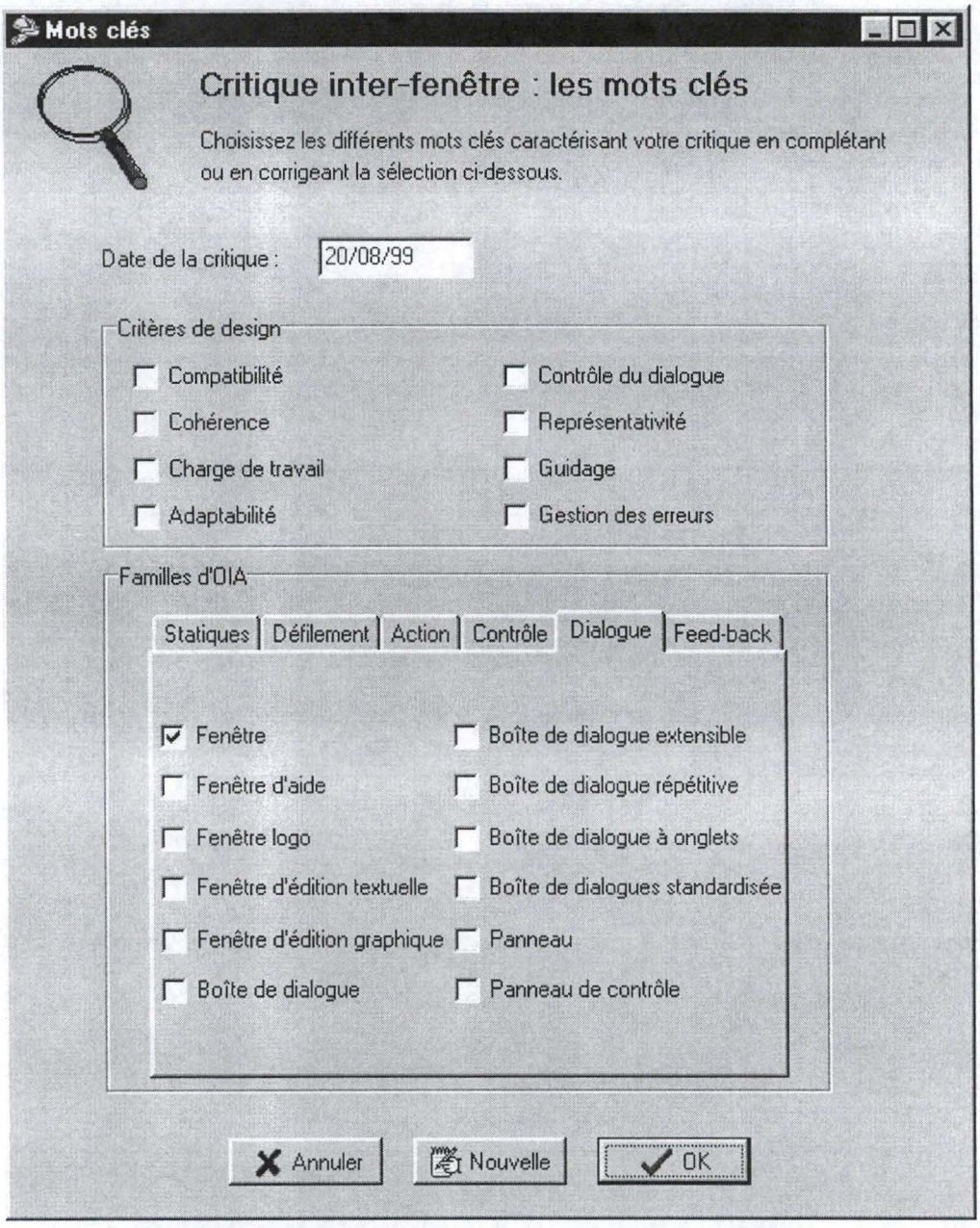


Figure A5-16: Inter-windows critique, keywords

A5.1.2.9 Critique choice (3)

Patrick wants now to add intra-windows critiques and click then on the "Critique intra-fenêtre" button

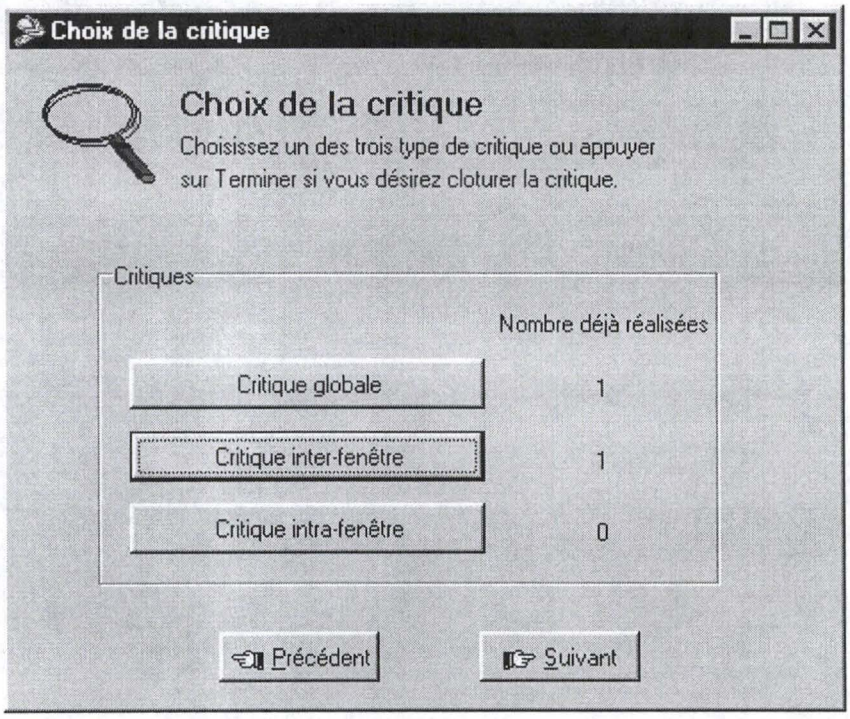


Figure A5-17: Critique choice (3)

A5.1.2.10 Intra-window critique

As for then inter-window critique, Patrick has to select the window he wants to criticize. He chooses the main window of application.

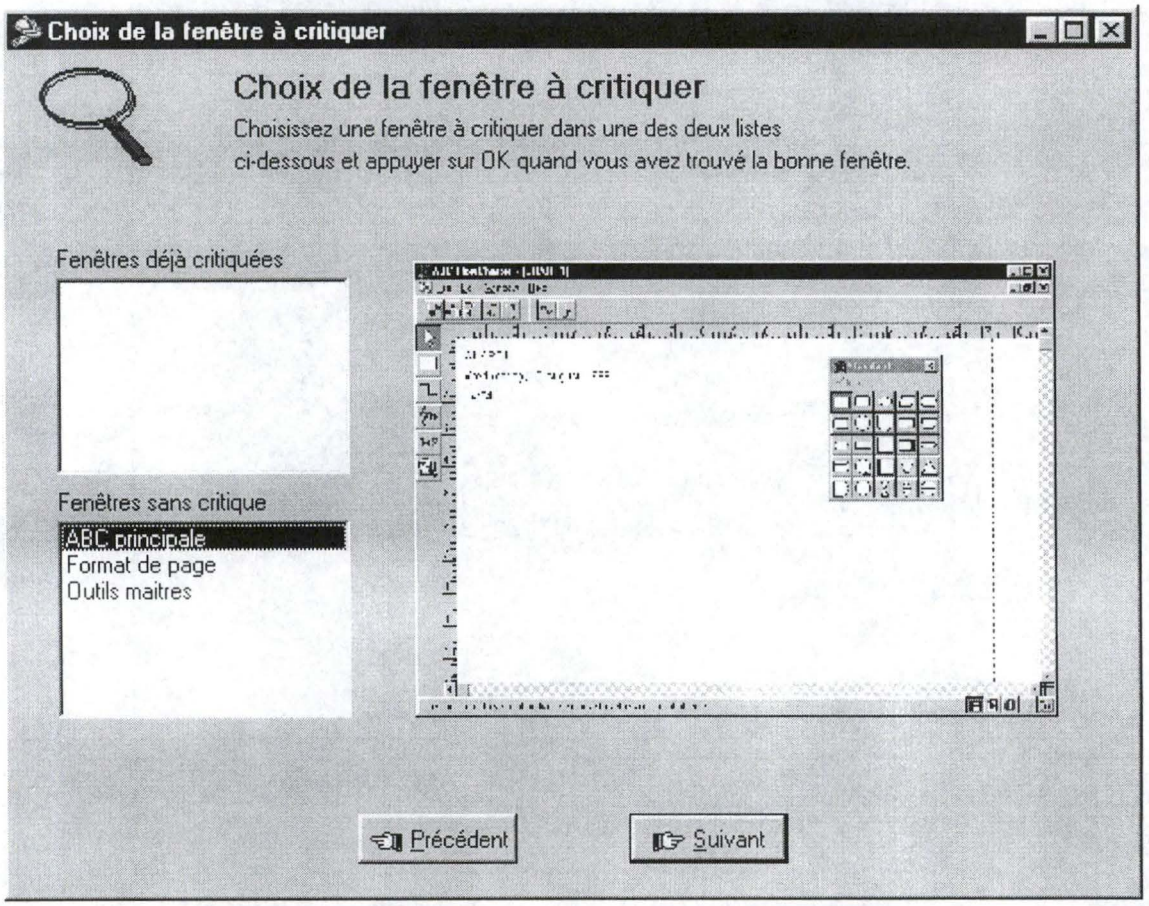


Figure A5-18: Intra-windows critique, the choice (1)

He inserts a positive and a negative critique and click on "Suivant" buttton.

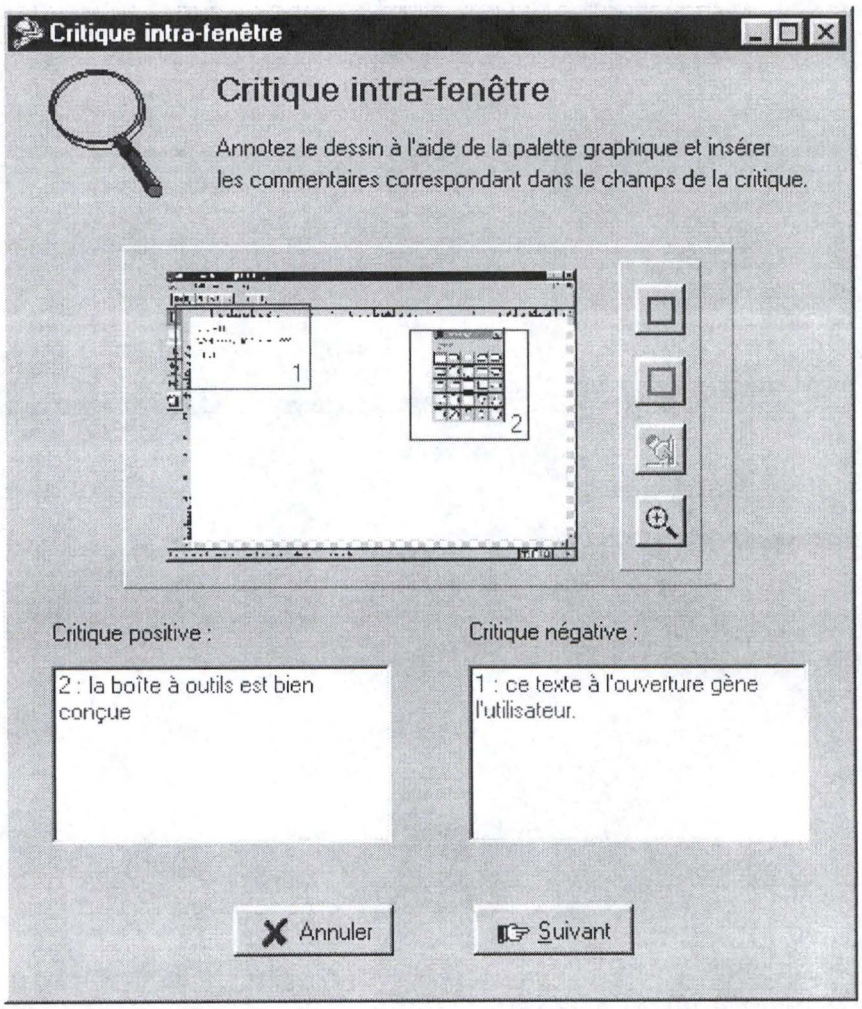


Figure A5-19: Intra-windows critique (1)

In this case, no keywords have been preselected and Patrick decides that it's a good choice. He has other window to criticize and click then on "Nouvelle".

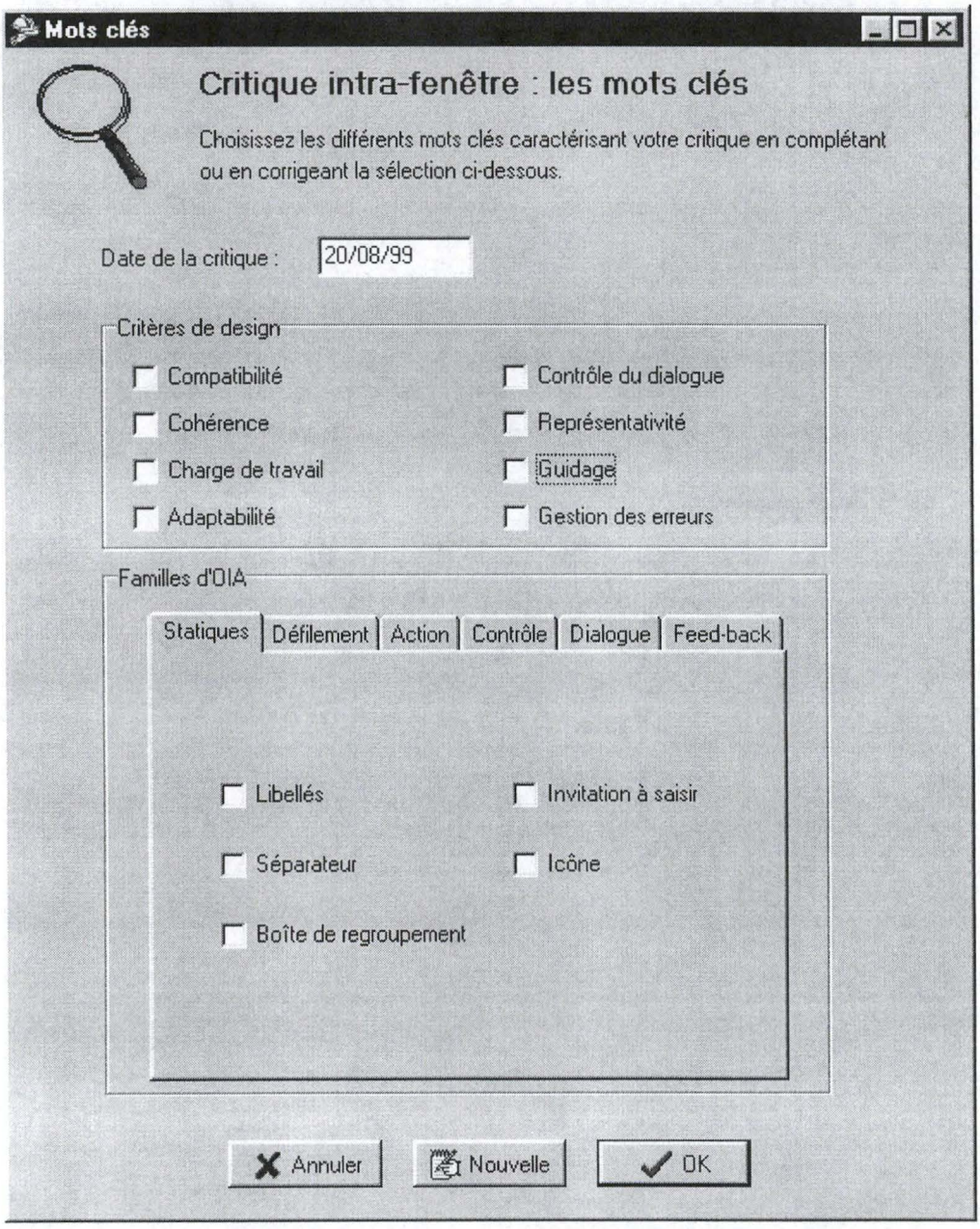


Figure A5-20: Intra-windows critique, keywords (1)

He has to choice again the windows he wants to criticize.

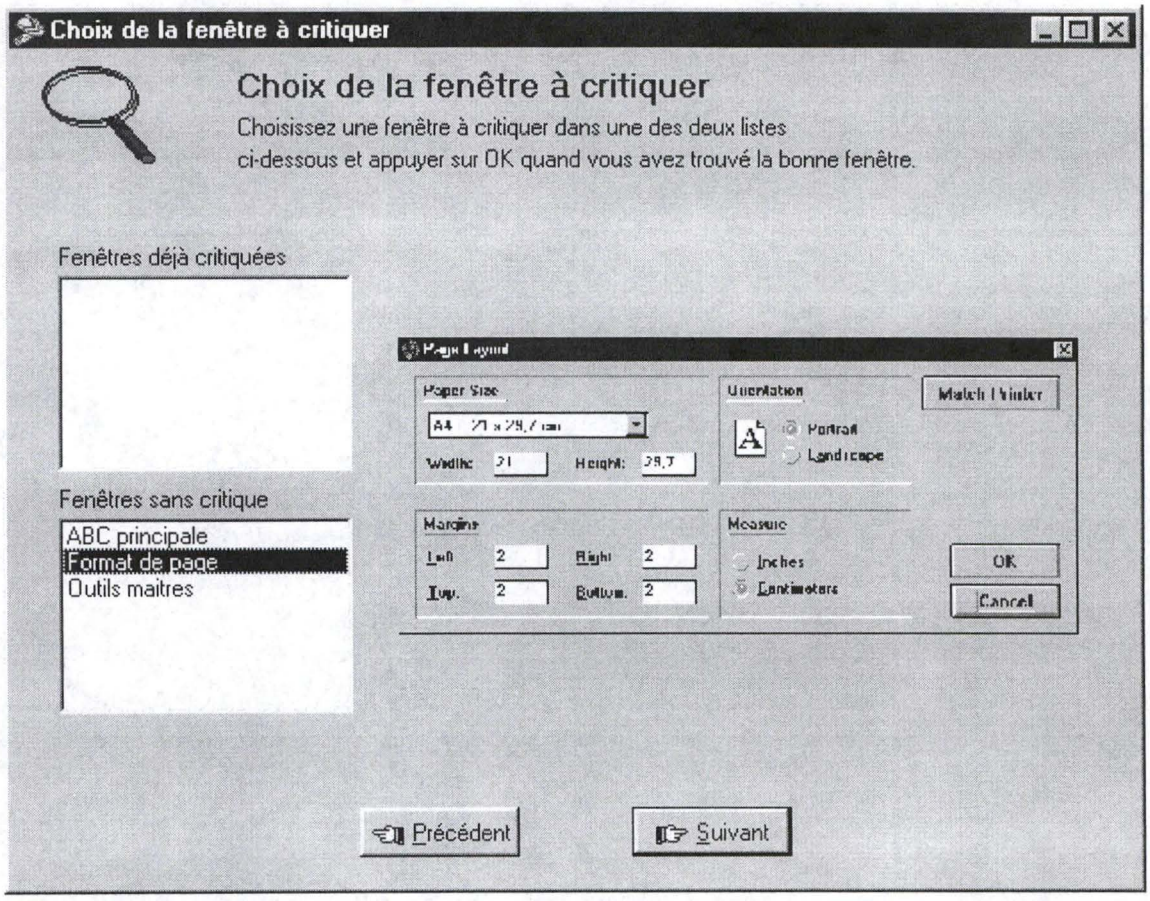


Figure A5-21: Intra-windows critique, the choice (2)

He criticizes it and click on the "suivant" button

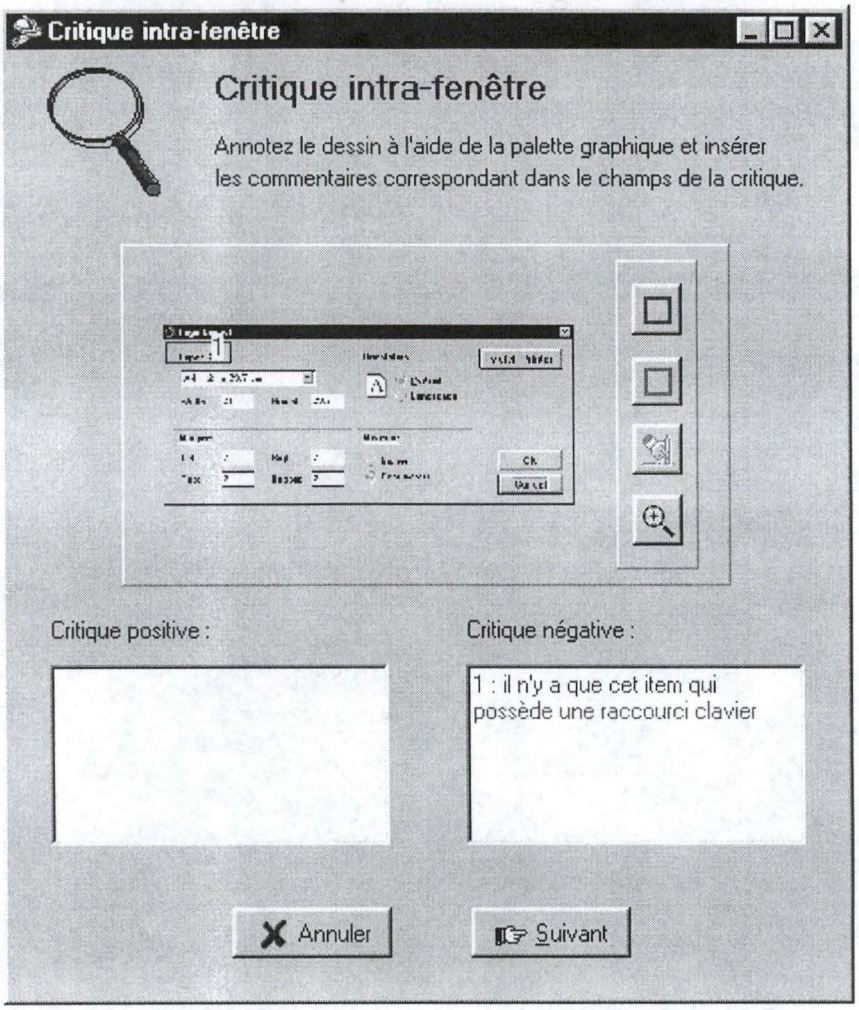


Figure A5-22: Intra-windows critique (3)

No keywords have been preselected but in this case, Patrick wants to add one. He clicks then on "cohérence". He validates his critique by clicking on the "OK" button.

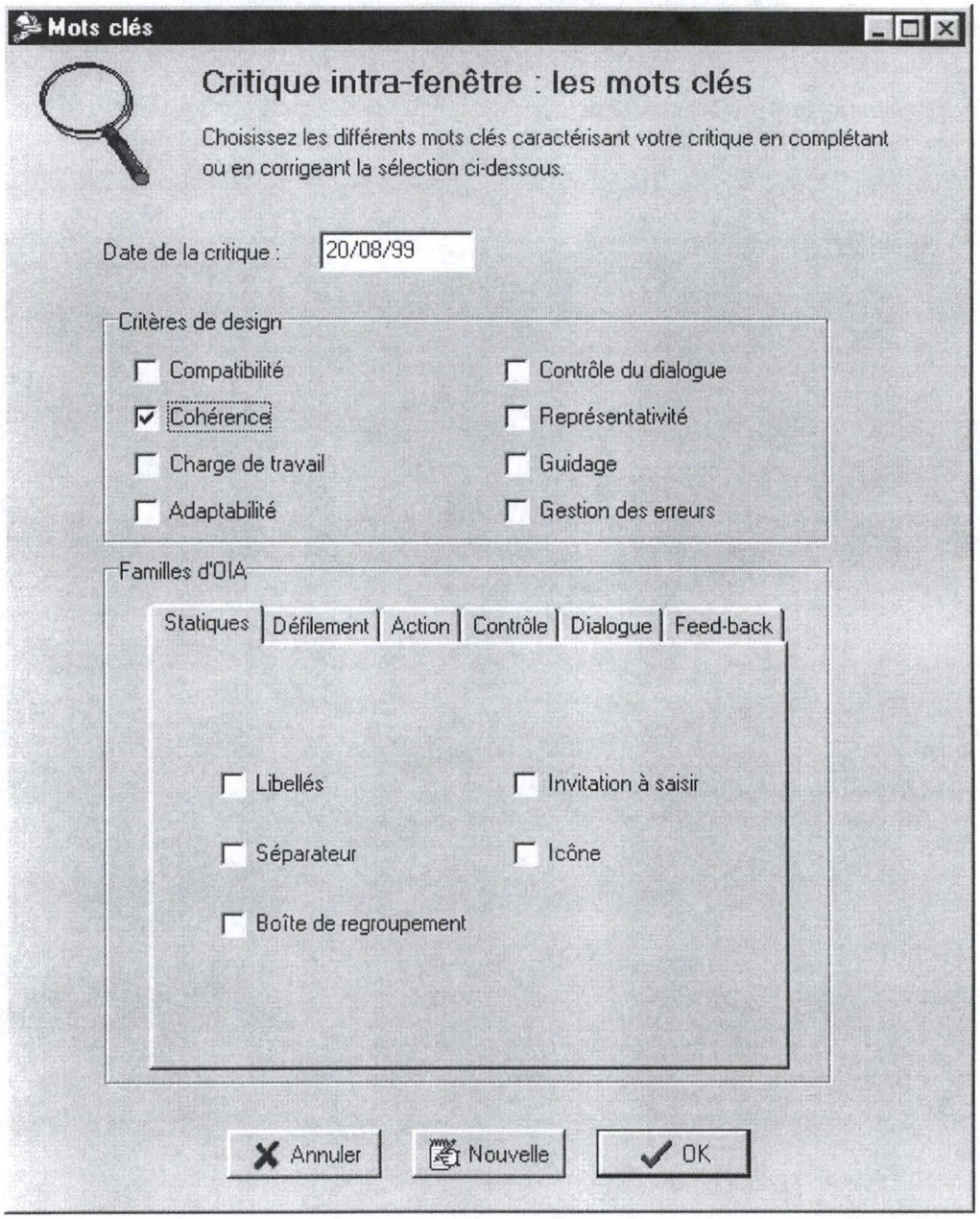


Figure A5-23: Intra-windows critique, keywords (2)

A5.1.2.11 Critique choice (4)

He comes back to the critique choice window but he doesn't want to insert more critique and he decides then to continue.

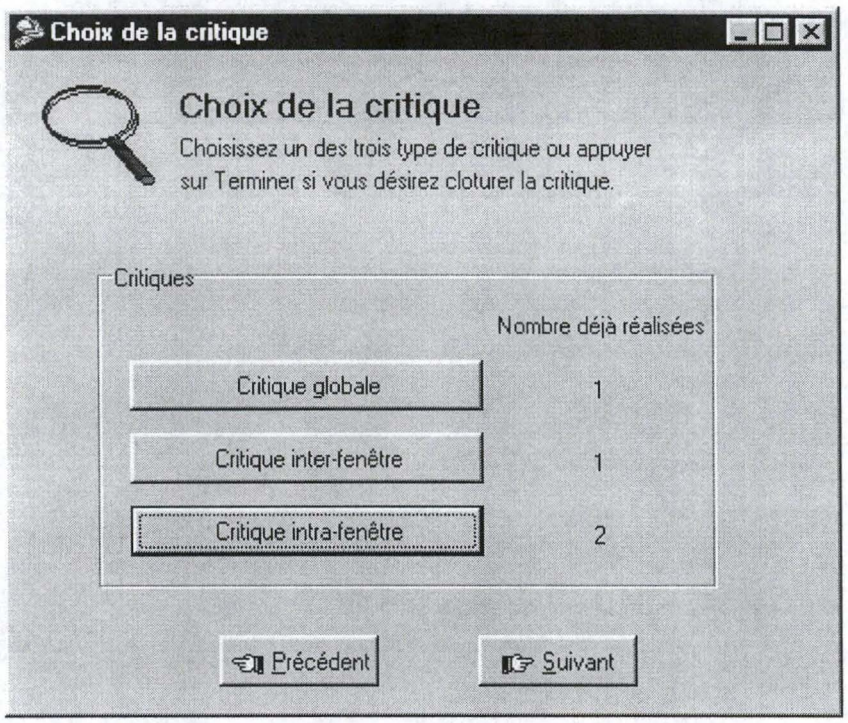


Figure A5-24: Critique choice (4)

A5.1.2.12 Summary

He can now see a summary of the inserted interface.

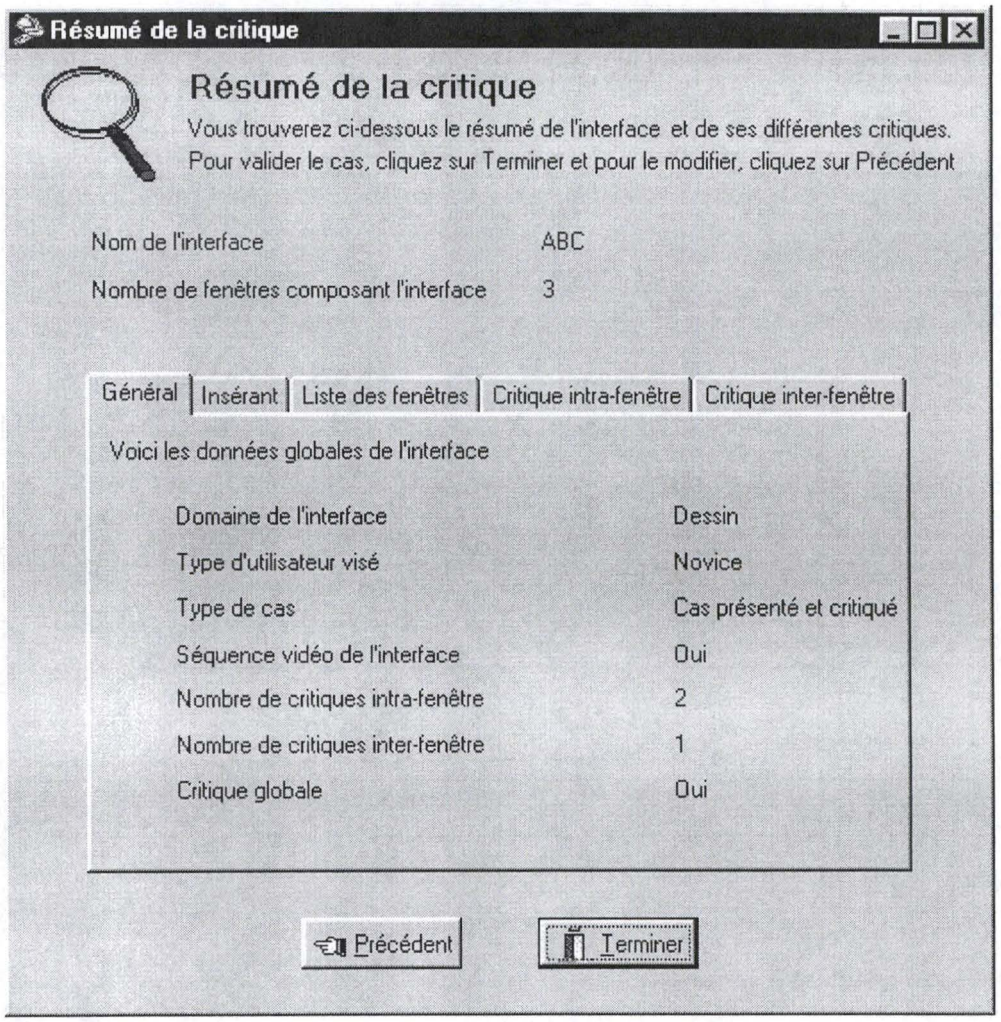


Figure A5-25: Summary (1)

He agrees with it and concludes his insertion

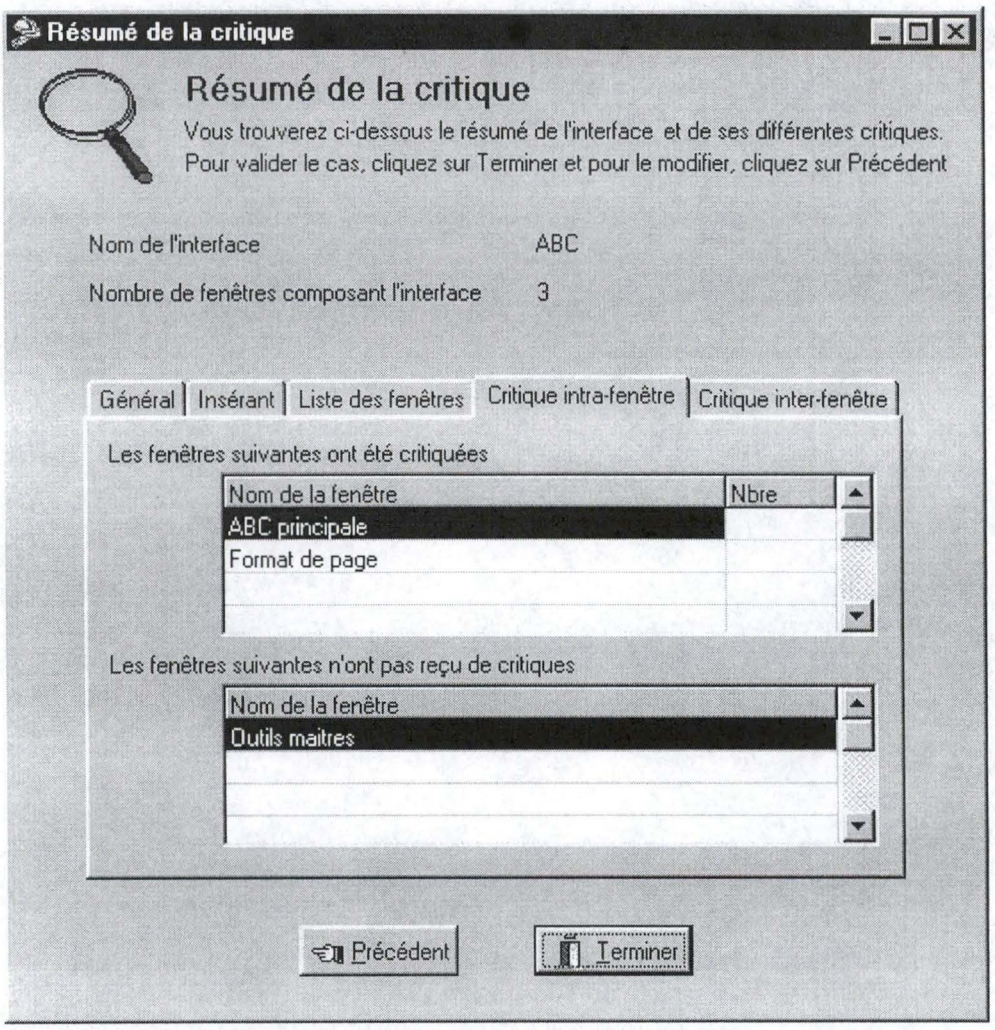


Figure A5-26: Summary (2)

Appendix 6:
Presentation of windows composing our interface

A6.1. Introduction

In chapter 4, we described the « Critique intra-fenêtre » in details. We will know present the other windows of the interface but with fewer details. We will provide a screenshot of them and a description of the different components.

A6.2. Windows description

A6.2.1 Window «Renseignements sur l'insérant»

(Information about people who insert the case)

A6.2.1.1 Screenshot

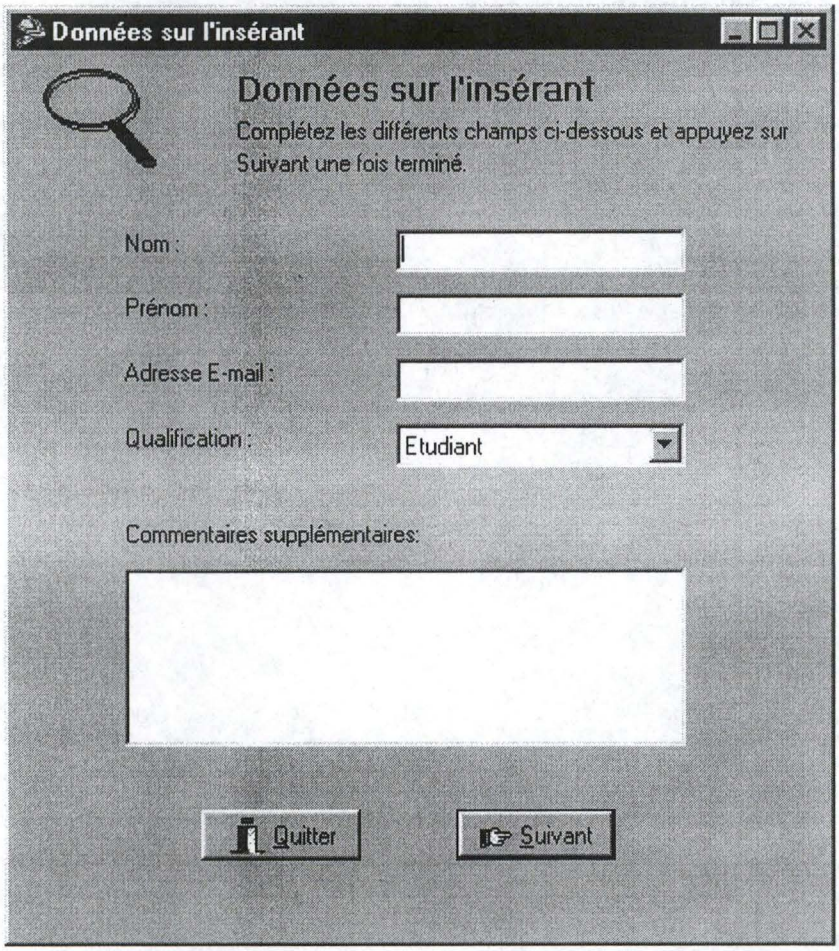


Figure A6-1: Information concerning inserant

A6.2.1.2 Components description

This window helps to collect information about the people who insert a case. One asks following information:

- First name: thanks to an EditBox. This field is mandatory.
- Last name: thanks to an EditBox. Again, this field is mandatory.
- Address Email: thanks to an EditBox, the user provides an email. This field is optional.
- Qualification: thanks to a ComboBox, the user provides its qualification. He has the choice between Student, incumbent Professor or Other (expert or novice). This field is important because it informs us on the knowledge of the person who inserts the case concerning Human-Computer Interaction. It is therefore mandatory.
- Comments: thanks to a Memo, the user provides all the other information that he wants concerning him. This field is optional.

The window includes finally two buttons:

- Button «Quitter» (Leave): this button leaves the application.
- Button «Suivant» (Following): this button allows to go to the next window in the application that is to say the window "Présentation de l'interface".

A6.2.2. Window «Présentation de l'interface» (interface presentation)

A6.2.2.1 Screenshot

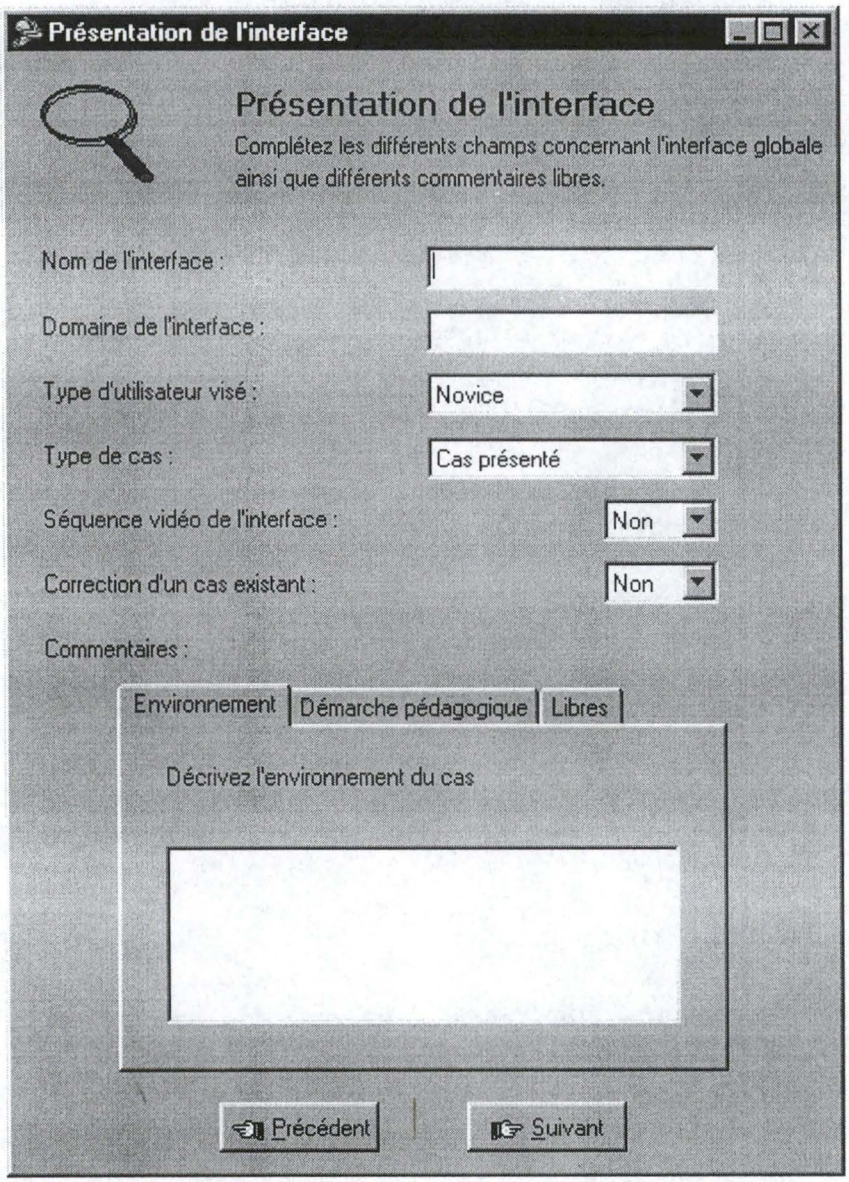


Figure A6-2: Interface presentation

A6.2.2.2 Components description

This window allows the user to provide the different useful information about the interface. Asked information are the following:

- Interface name: thanks to an EditBox, the user provides the name that he wants to give to his interface. This field is mandatory.

- Interface domain: thanks to an EditBox, the user provides the field of activity aimed by the interface. This field is mandatory.

- Type of user aimed: thanks to a ComboBox, the user provides the type of user aimed by the interface. He has the choice between «Novice», «Intermédaire» and «Professionnel». The default value is «Novice».

- Type of case: thanks to a ComboBox, the user provides the type of case that he is going to present. He has the choice between «Presented Case» (he will not have the possibility to criticize), «Criticized and Presented Case» (he will have automatically the possibility to provide a critic of its choice) or again «Reference case» provided that he is the holder of the course. The default value is «Presented Case».

- Video Sequence of the interface: thanks to a ComboBox, the user informs us about the existence of a video sequence that completes its interface. If a video sequence exists, the user will have then to provide the path to find this sequence (to see window " Chemin de la séquence vidéo "). The default value is that there is no video sequence.

- Correction of an existent interface: thanks to a ComboBox, the user indicates us if the case that he inserts is the correction of an existing case already present in the database. If it is the case, the user has to indicate (seeing window " Renseignement supplémentaires concernant l'interface ") of which interface the interface he introduces is the correction and he is able, in a free comment to tell us why. The default value is no.

- Comments: thanks to a mitre box, the user has the possibility to add comments about the interface. Thus, he can add comments concerning the environment of the case, the pedagogical step used or others free comments. These fields are optional.

The window includes finally two buttons:

- Button «Précédent» (preceding): this button serves to return to the window " Données sur l'insérant ".

- Button «Suivant»: this button allows to go to the next window of the application that is, the window "Renseignements complémentaires concernant l'interface" if the inserted interface is a correction, the window "Chemin de la séquence vidéo" if the user provides a video sequence linked to his interface and that this last one is not a correction, or the window " Fenêtres composant l'interface ".

A6.2.3. Window «Renseignements complémentaires concernant l'interface» (complementary Information about the interface)

A6.2.3.1 Screenshot

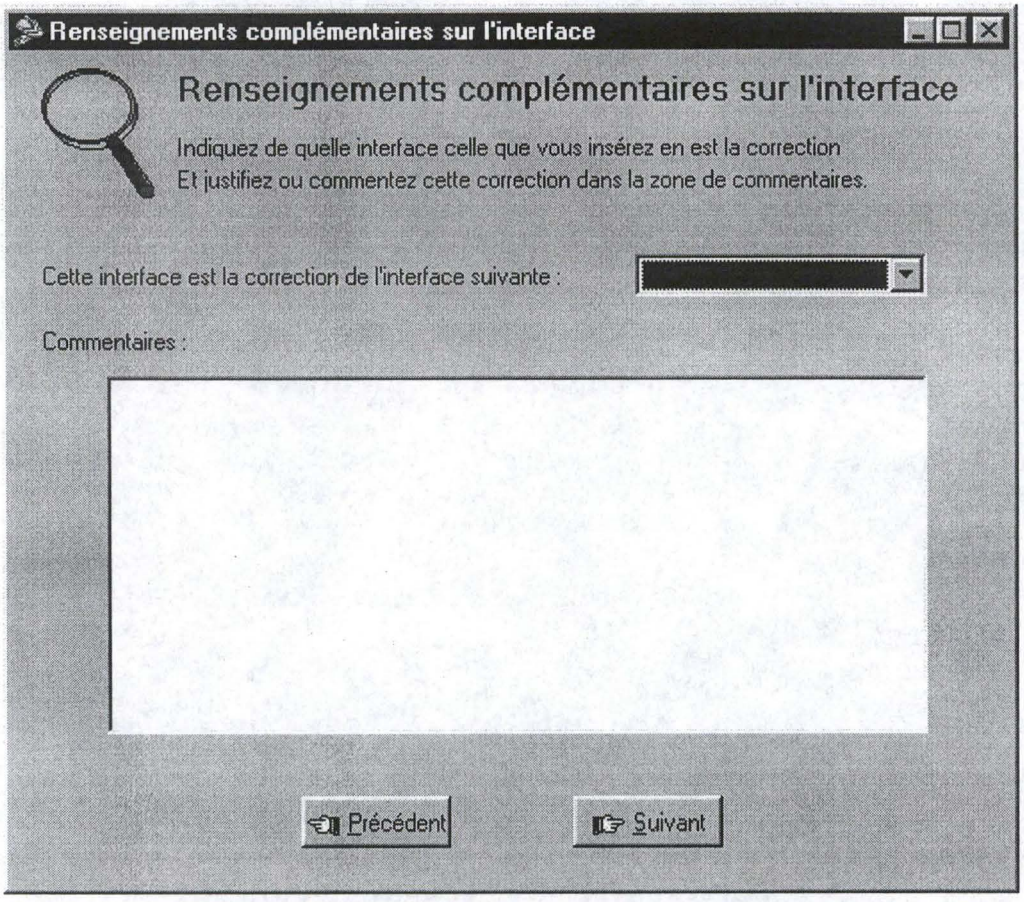


Figure A6-3: More information about the interface

A6.2.3.2 Components description

This window allows the user to indicate which window is corrected by the current and to tell why. Asked information are the following:

- This interface is the correction of the next interface: thanks to a ComboBox, the user chooses in the list of already introduced interfaces which one is corrected by current. This field is mandatory.

- Comments: thanks to a memo, the user can explain in what the interface that he introduces is the correction of another interface. This field is optional.

The window understands finally two buttons:

- Button «Précédent»: this button serves to return to the window " Présentation de l'interface ".

- Button «Suivant»: this button allows to go to the next window of the application that is to say the window " Chemin de la séquence vidéo" if it exists or to the window " Fenêtres composant l'interface " in the opposite case.

A6.2.4. Window «Chemin de la séquence vidéo» (Path of the video sequence)

A6.2.4.1 Screenshot

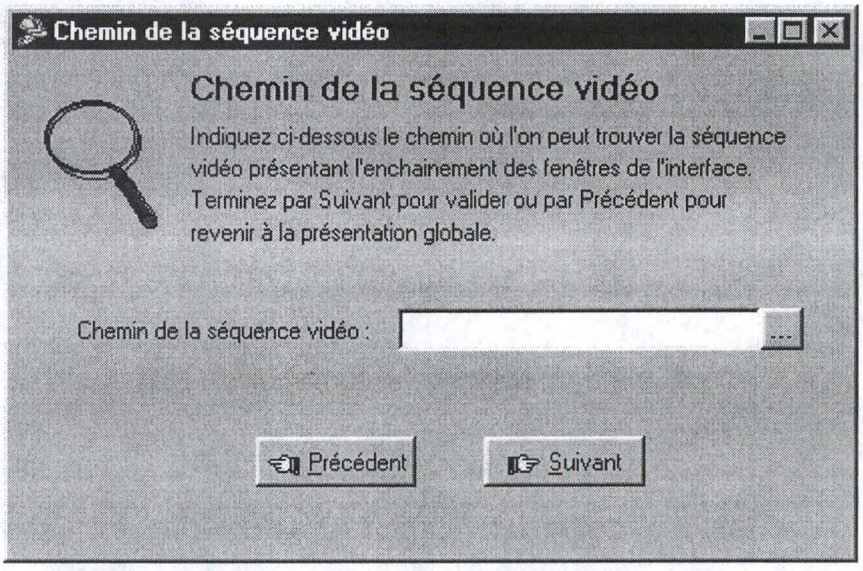


Figure A6-4: path of the video sequence

A6.2.4.2 Components description

This window allows the user to specify the path of the video sequence that it puts at the disposal users with its interface. Asked information are following:

- Path of the video sequence: thanks to an EditBox, the user provides the path allowing to find the video sequence. User has allowing it to check discs to find this path. This field is mandatory.

The window understands finally two buttons:

- Button «Précédent»: this button serves to return to the window "Présentation de l'interface" or "Informations complémentaires concernant l'interface" according to the path followed.

- Button «Suivant»: this button allows go to the next window of the application that is to say the window «Fenêtres composant l'interface».

A6.2.5. Window «Fenêtres composant l'interface»

(Windows composing the interface)

A6.2.5.1 Screenshot

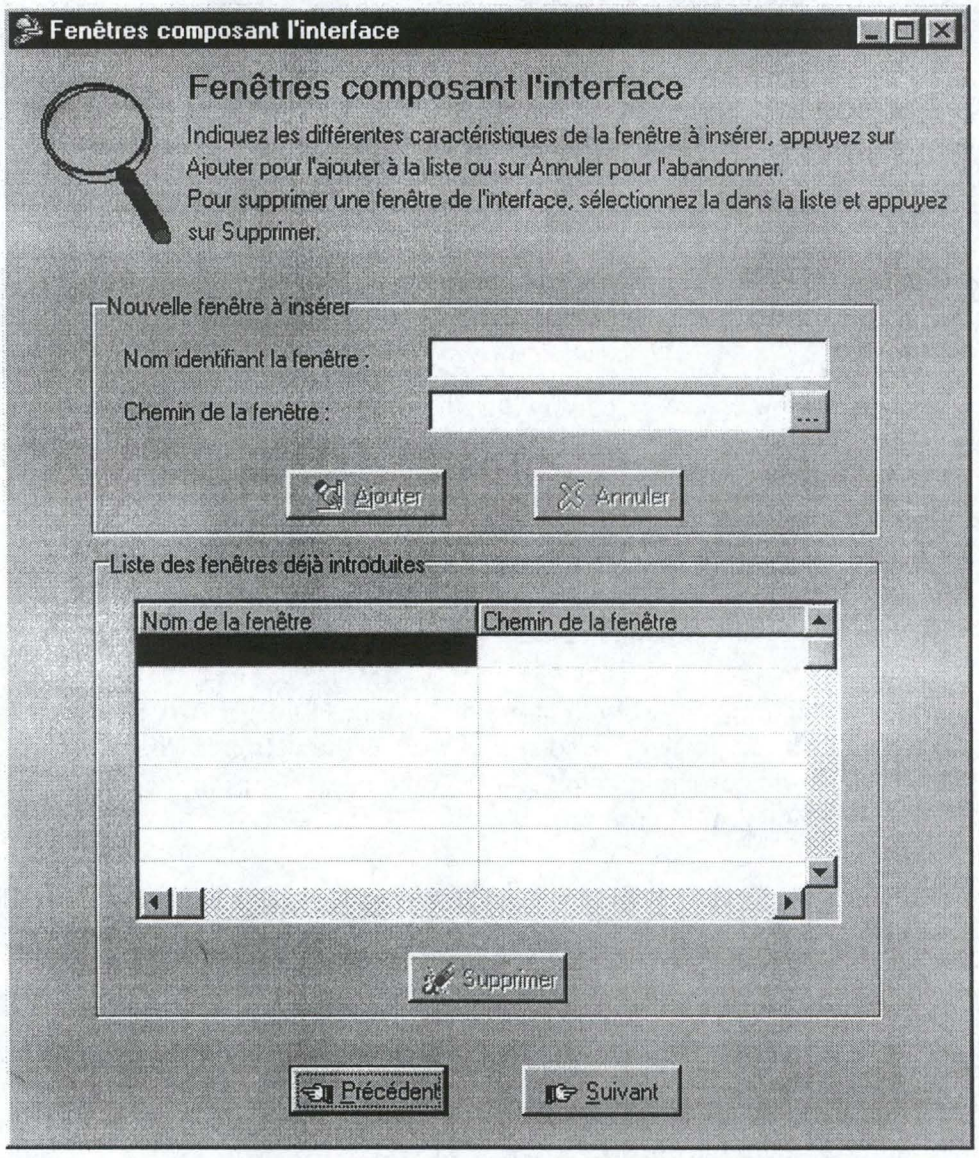


Figure A6-5: windows composing the interface

A6.2.5.2 Components description

This window allows the user to provide the path by which we can access to the different windows of the interface proposed to the other users and to have an overview the different windows that it has introduced for its interface. Request information are:

- Name of the window: for each window of the proposed interface, the user has to choose a name. An EditBox is aimed to receive the name of the window. This field is mandatory.

- Path of the window: for each window of the proposed interface, the user has to provide the path that allows to access to this window. An EditBox is aimed to collect this path. The user can browse his discs thanks to a button. This field is mandatory.

These two elements are gathered in a GroupBox named: "Nouvelle fenêtre à insérer» (New window to insert) with

- Button «Ajouter» (Add) allowing the validation of the new window for the insertion

- Button «Annuler» (Cancel) deleting the contents of both fields.

The window also includes other elements gathered in a GroupBox named "Liste des fenêtres déjà introduites» (List of already introduced window) :

- Tableau récapitulatif (Summary Table): this table on two columns and n lines summarizes the name of windows as well as the path allowing to access them. No modification is possible to this place. If the user wishes to make changes, he has to delete the entry and to reinsert it.

- Button «Suppression» (Suppress) that allows to suppress a selected window.

Finally, we can find the following elements:

- Button «Précédent»: this button allows user to return to the former window according to the source.

- Button «Suivant»: this button allows to go to the next window of the application that is to say the window «Résumé du cas présenté» (Summary of the Presented case) if the interface was a «Presented case» and to the window " Choix de la critique " otherwise.

A6.2.6. Window «Résumé du cas présenté» (Summary of the presented case)

A6.2.6.1 Screenshot

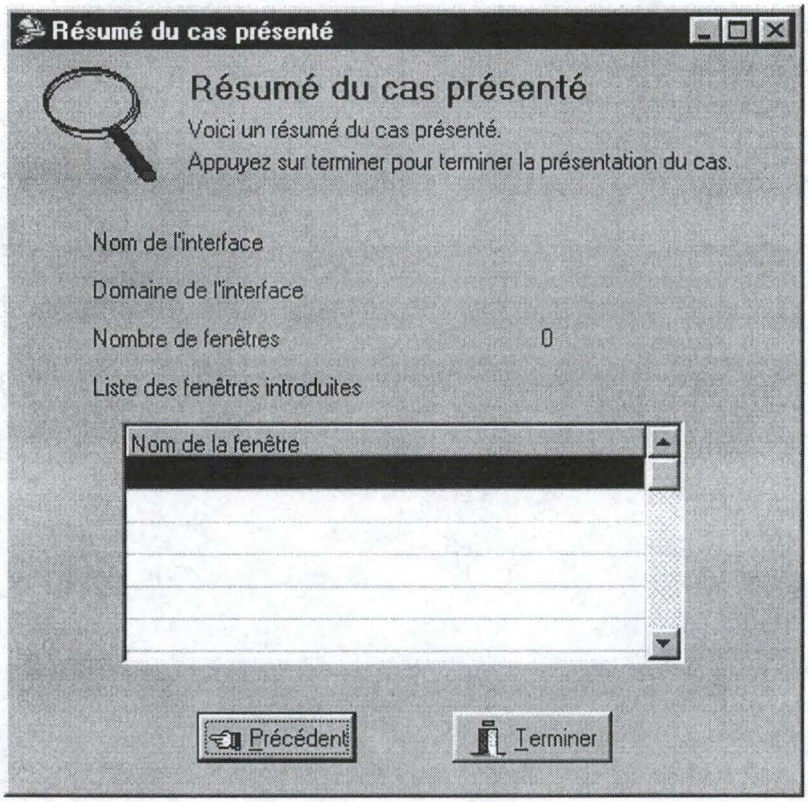


Figure A6-6: Summary of the case

A6.2.6.2 Components description

This window presents a summary of the case the user has inserted. So, there are no fields to fill.

The window contains next elements:

- Interface name: Label with the name of the interface
- Domain of the interface: Label with the domain of the interface
- Number of windows introduced: Label with the number of windows composing the interface
- List of windows introduced: StringGrid with the name of the different windows introduced.

A6.2.7. Window «Choix de la critique» (critique choice)

A6.2.7.1 Screenshot

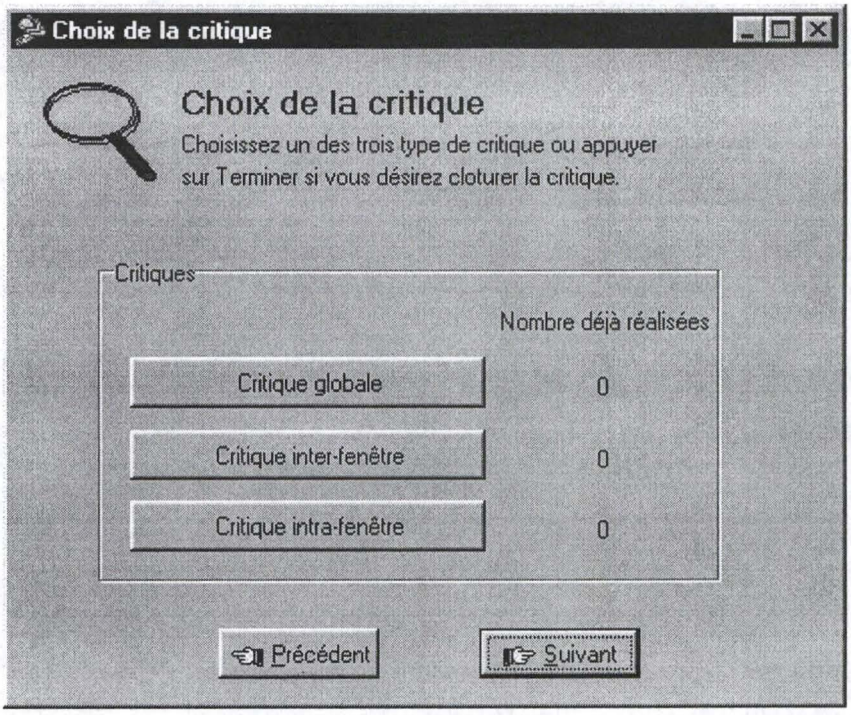


Figure A6-7: Critique choice

A6.2.7.2 Components description

This window allows the user to begin the interface critique. he will have the choice between a global critique, inter- and/or intra-windows critique.

The window includes next elements:

- Button «Critique globale» (Global critique): this button allows the user to begin a global interface critique. Then he goes to the window "Critique globale".

- Button «Critique inter-fenêtre» (inter-window critique): this button allows the user to begin a inter-window interface critique. Then he goes to the window "Choix des deux fenêtres à critiquer".

- Button «Critique intra-fenêtre» (intra-windows critique): this button allows the user to begin a intra-window interface critique. Then he goes to the window "Choix de la fenêtre à critiquer".

Notice that a table indicates to the user the number of critiques of each kind already realized.

- Button «Précédent»: this button allows to return to the window "Fenêtres composant l'interface".
- Button «Suivant»: this button allows to go to the window "résumé de la critique".

A6.2.8. Window «Critique globale» (global critique)

A6.2.8.1 Screenshot

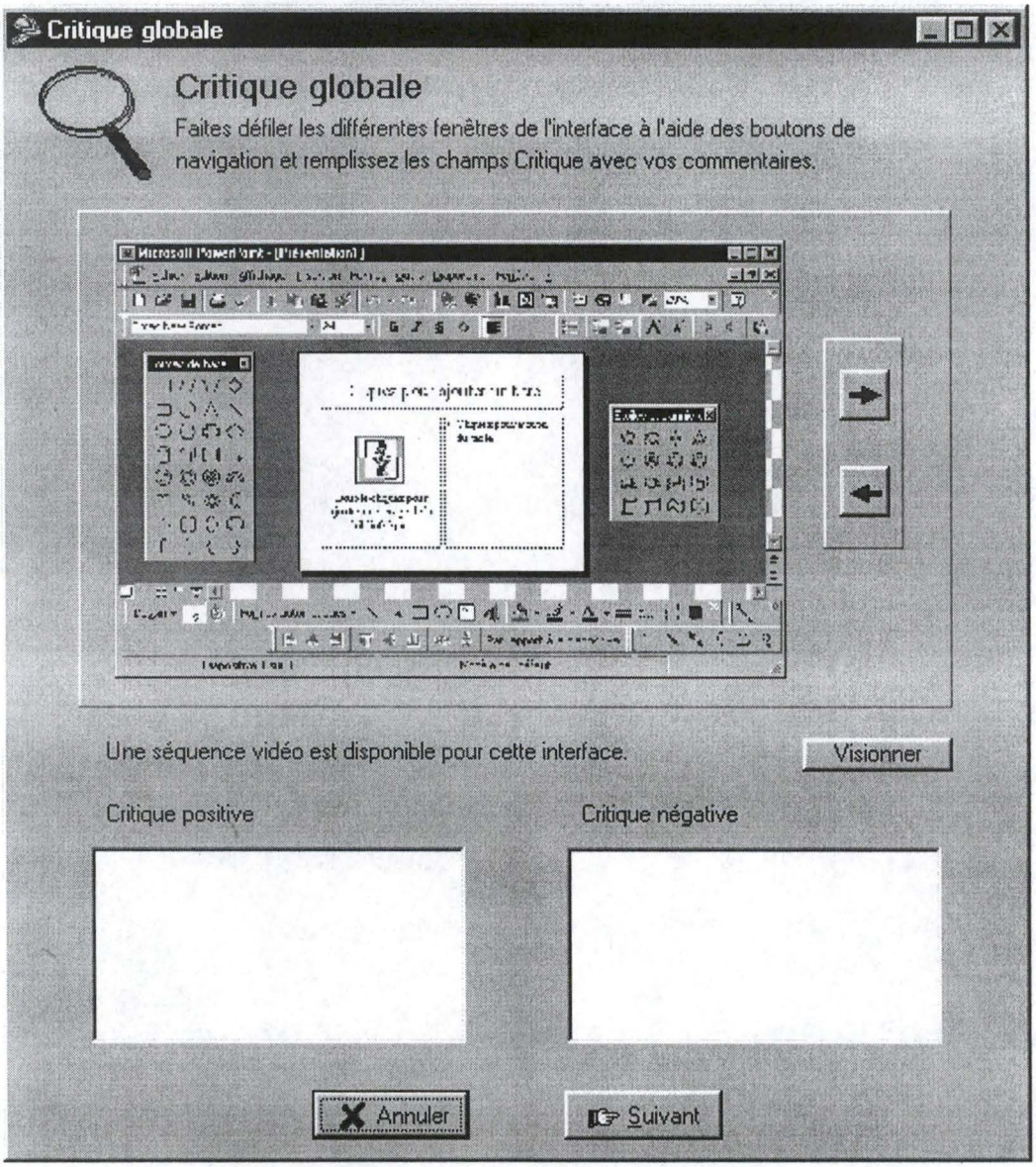


Figure A6-8: Global critique

A6.2.8.2 Components description

This window allows the user to make a global interface critique that he has proposed and includes next elements:

- Scrolling tools: this tool allows the user to see the different windows composing the interface. This manner, it is easier to make a global critique since one can have constantly each window under eyes.

- Video Sequence: Label indicating the existence of a video sequence. In the affirmative, a button allows to see the video sequence in a separate window.

- Text zones "Positive critique " and "Negative critique": thanks to two Memos, the user is able to criticize the interface. This field is mandatory.

- Button «Suivant»: this button allows to go to the window " keywords ".

- Button «Annuler» (cancel): this button allows to return to the window " Choix de la critique ". Information contained in the window « global Critique « are not then saved.

A6.2.9. Window «Choix des deux fenêtres à critiquer» (Choice of the two windows to criticize)

A6.2.9.1 Screenshot

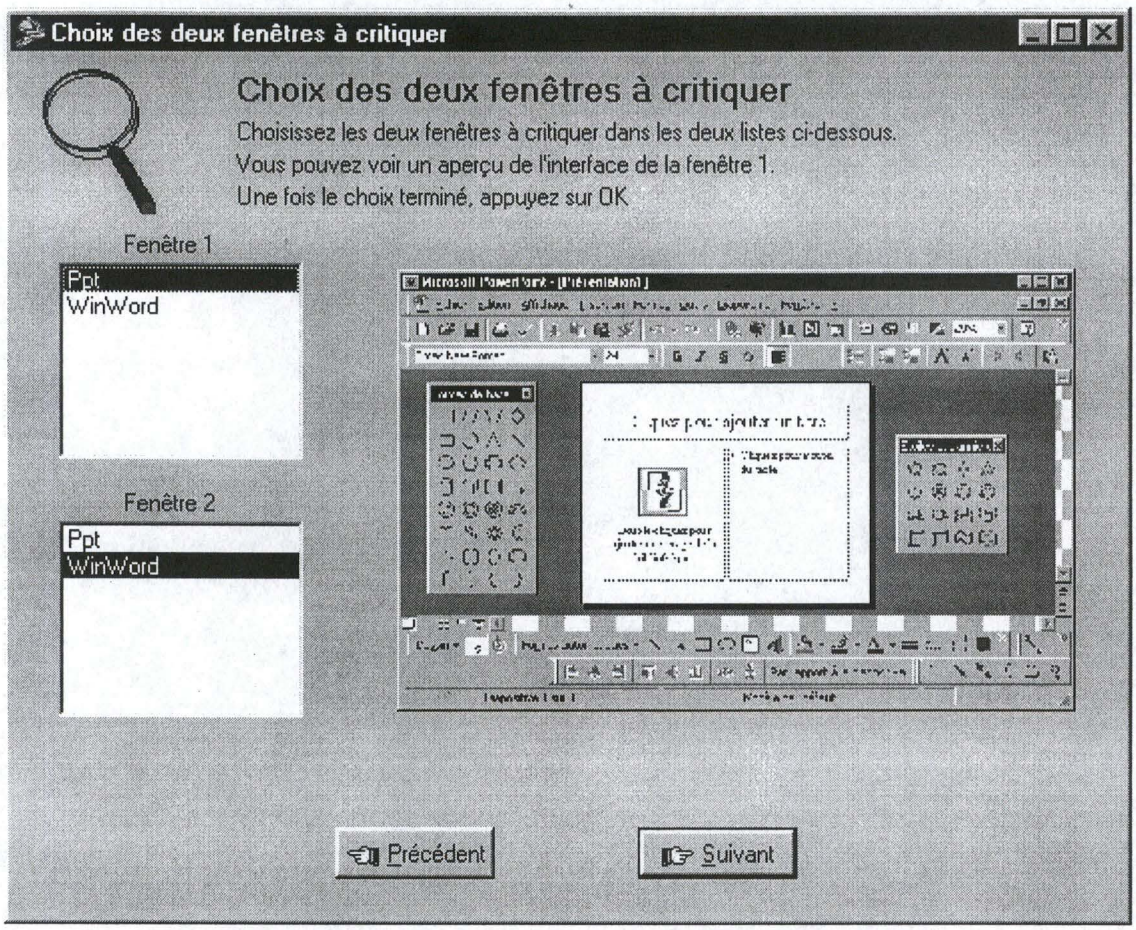


Figure A6-9: Choice of the two windows to criticize

A6.2.9.2 Components description

This window allows the user to select the two windows that he will compare to make his inter-windows critique.

The window includes next elements:

- List of windows I: this list includes all windows of the interface. The user selects in this list the first of the two windows that he wants to compare in his inter-windows critique.

- List of windows II: this list is identical to the precedent. The user selects in this second list the second of the two windows that he wants to compare in his critique.

- Illustration: When the user selects a window in one of the two lists, an illustration of this interface appears.

Remark: when a window has been selected in one of the two lists, the window with the same name in the other list can no longer be selected. Indeed, two distinct windows are necessary to make an inter-windows critique. This is why, if the user chooses in the two lists the same window, a message of error appears in the bottom of the window and the user can not continue his his critique (the button «Suivant» is unusabled).

- Button «Suivant»: this button allows to continue the inter-windows critique and to go to the next window of the application, the window "Critique inter-fenêtre".

- Button «Précédent»: this button allows the user to return to the preceding window, that is to say the window " Choix de la critique ".

A6.2.10. Window «Critique inter-fenêtres» (Inter-windows critique)

A6.2.10.1 Screenshot

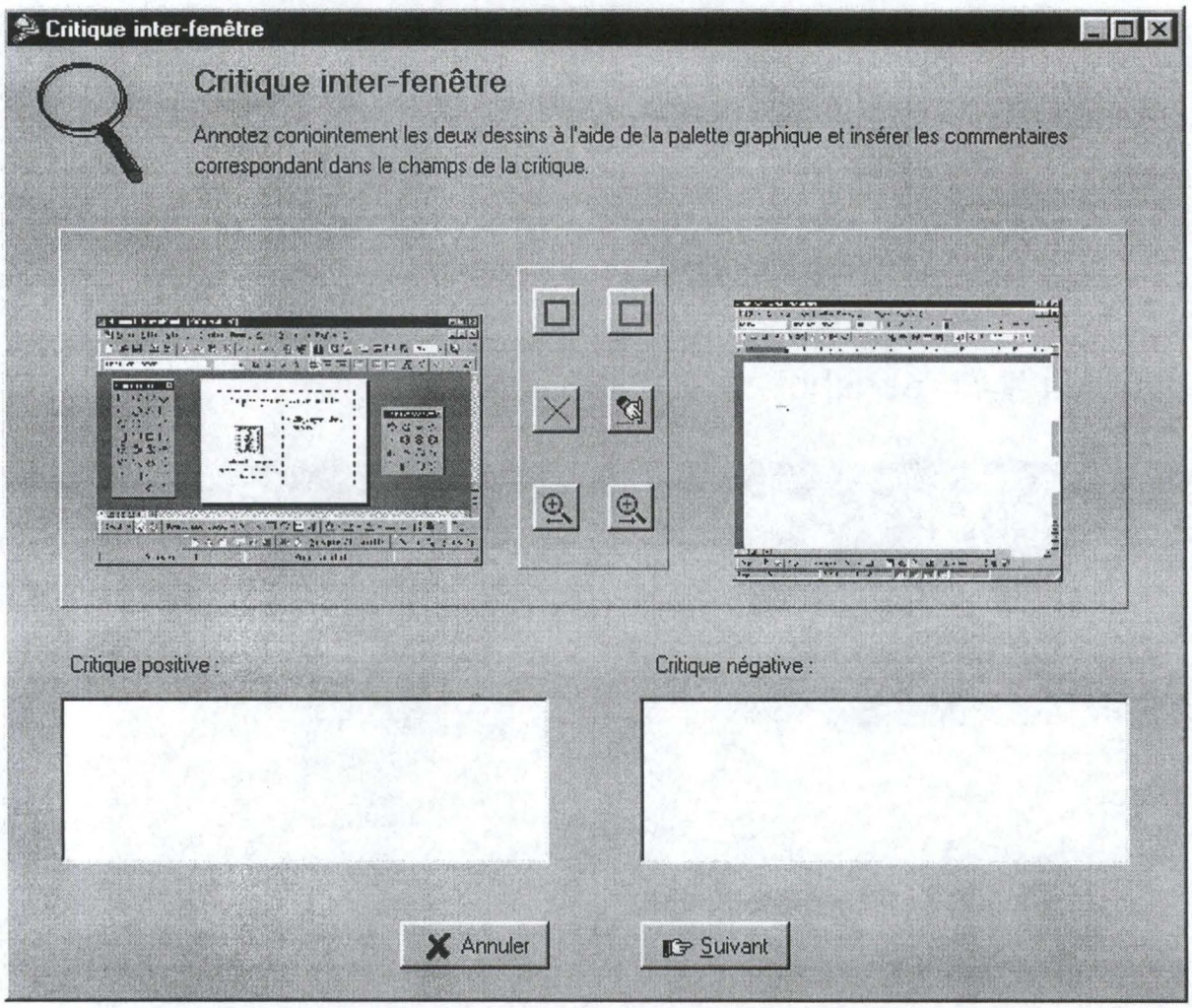


Figure A6-10: Inter-windows critique

A6.2.10.2 Components description

This window allows the user to make a inter-windows critique of the two selected windows that he has selected.

The window contains next elements:

- Graphical Tool: this tool allows the user to draw rectangles on illustrations, enlarging these illustrations, and so on. The utilization of this graphical tool is explained in a section of this report.

- Illustrations: the user can see an illustration of the two windows that he is comparing.
- Text zones (positive Critique and negative Critique): thanks to two Memos, the user can comment his inter-windows critique. This field is mandatory.
- Button «Suivant»: this button allows to go to the window "Mots clés".
- Button «Annuler»: this button cancels the inter-windows critique and allows the user to return to the preceding window, that is to say the window " Choix de la critique ". Information concerning this critique are not then saved.

A6.2.11. Window «Choix de la fenêtre à critiquer»

(Choice of the window to criticize)

A6.2.11.1 Screenshot

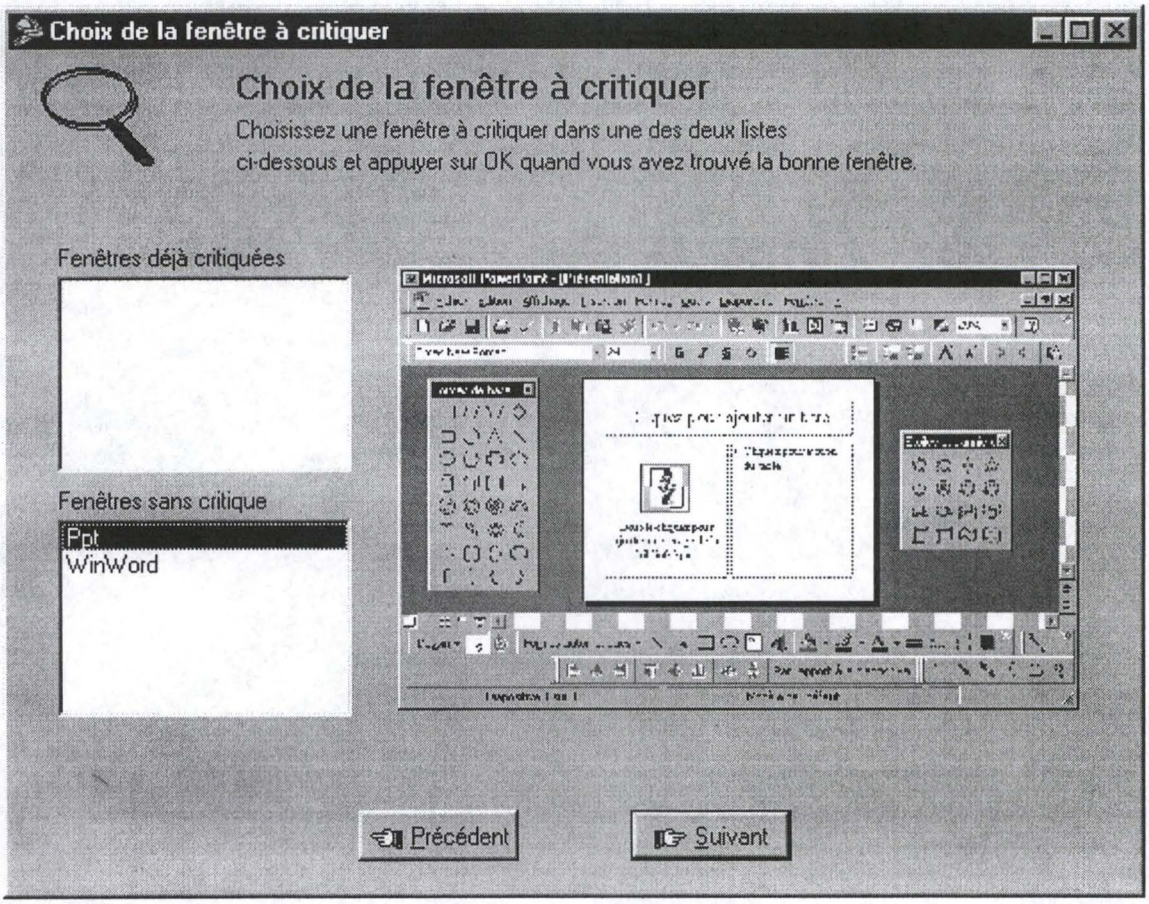


Figure A6-11: Choice of the window to criticize

A6.2.11.2 Components description

This window allows the user to select the window he will criticize and includes next elements:

- List of already criticized windows: this list includes all windows of the interface already criticized. The user can even so select one of these windows to make some new criticizes.

- List of windows without critique: this list contains all the windows of the interface that have not been criticized. The user can select one of these windows to make some critique.

- Illustration: When the user selects a window in one of the two described lists, an illustration of this interface appears.

- Button «Suivant»: this button allows to continue the intra-windows critique and to go to the next window of the application, the window «Critique intra-fenêtre».

- Button «Précédent»: this button cancels the intra-windows critique and allows the user to return to the preceding window, that is to say the window " Choix de la critique ".

A6.2.12. Window «Critique intra-fenêtre» (inter-windows critique)

This window is criticized in Chapter 4.

A6.2.13. Window «Mots clés» (Keywords)

A6.2.13.1 Screenshot

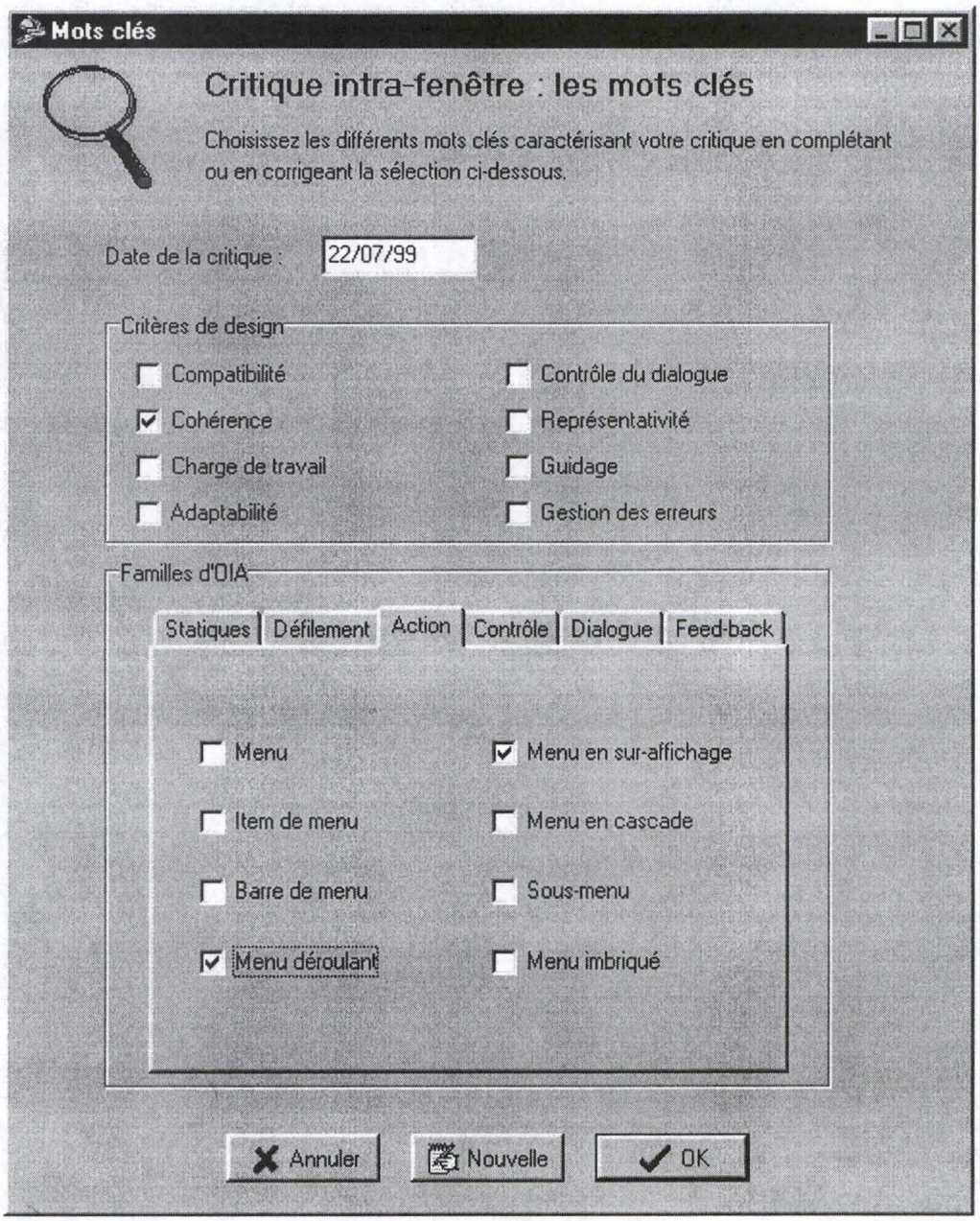


Figure A6-12: Keywords

A6.2.13.2 Components description

This window allows the user to choose keywords that define his critique. These keywords are therefore linked to the critique rather than to the interface or to the window.

The window contains next elements:

- Date of the day: thanks to an EditBox, user provides the date of the day. Thanks to this field, it will be possible to a user to make a search according to a date of insertion. Note also that this field is filled automatically without the assistance of user (the former has nevertheless the possibility to modify the value of the field).

- List of design criteria: thanks to CheckBoxes, the user can choose the different design criteria that characterize his critique. These criteria are gathered in a GroupBox.

- Family of AIO : thanks to a PageControl, the user can select different AIO among 6 families of AIO.

- Button «Annuler» : this button allows to return to the window corresponding to the critique made.

- Button «Nouveau»: this button allows to validate the critique and its keywords and introducing a new one. It opens again the window "Choix de la fenêtre à critiquer" if keywords are associated with an intra-windows critique and "Choix des deux fenêtres à critiquer" if it is to an inter-windows critique. In case of a global critique, this button is unusable because it can exist only one global critique for a specific interface critique.

- Button «OK»: this button validates the critique and its keywords and returns to the « Choix de la critique » window .

A6.2.14. Window «Résumé» (Summary)

A6.2.14.1 Screenshot

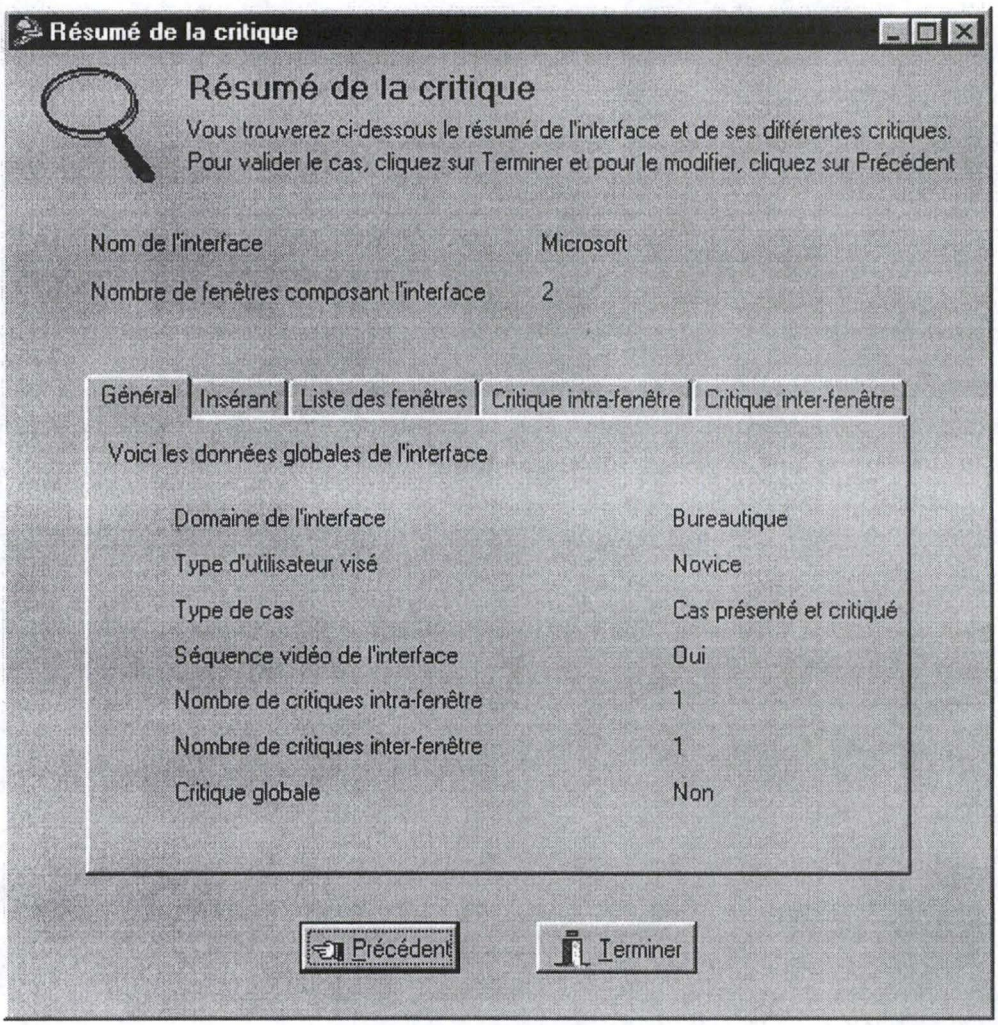


Figure A6-13: Summary

A6.2.14.2 Components description

This window provides the user a summary of the interface and includes next elements:

- Interface name: Label with the name of the interface.
- Number of windows composing the interface: Label indicating the number of windows composing the interface.
- Summary of the interface: PageControl of composed next pages :

- General: domain of the interface, aimed user type, type of case, video sequence of the interface, number of intra-windows critiques, number of inter-windows critiques and global critique.
- User who inserts: name, Given name, email address and qualification
- List of windows: list of windows composing the interface
- Intra-windows critique: list of windows having received a critique and list of windows that have not.
- Inter-windows critique: list of windows criticized.

- Button «Précédent»: this button allows to return to the « Choix de la critique » window

- Button «Terminer»: this button allows to end the application and validating the interface.