Palette
Pedagogically sustained Adaptive Learning through
the Exploitation of Tacit and Explicit Knowledge

Information Society
Technologies

Project no. FP6-028038

Palette

Pedagogically sustained Adaptive LEarning Through the exploitation of Tacit and Explicit knowledge

Instrument: Integrated Project
Thematic Priority: Technology-enhanced learning

D.PAR.07 – Second analysis of usability

Due date of deliverable: 31 July 2008
Actual submission date: 25 July 2008

Start date of project: 1 February 2006 Duration: 36 months
Organisation name of lead contractor for this deliverable: ULg

<table>
<tr><td colspan="3"><b>Project co-funded by the European Commission within the Sixth Framework Programme</b></td></tr>
<tr><td colspan="3"><b>Dissemination Level</b></td></tr>
<tr><td><b>Nature: R</b></td><td>This deliverable is public with a particular attention paid to provide the developers with useful information to bring to completion the development of their tool or service.</td><td><b>Dissemination: PU</b></td></tr>
</table>

Keyword List: usability, ergonomics, acceptability, CoP members, participatory design

Responsible Partner: Étienne Vandeput, ULg

MODIFICATION CONTROL			
Version	Date	Status	Modifications made by
0	10-06-2008	Plan	Vandeput Étienne
1	13-06-2008	Draft	Malengrez Déborah, Ledent Maryse, Rieppi Stéphane,Vandeput Étienne
2	27-06-2008	Draft	Malengrez Déborah, Ledent Maryse, Rieppi Stéphane,Vandeput Étienne
3	10-07-2008	Sent to the developers	Rieppi Stéphane
4	14-07-2008	Draft including developers comments	Rieppi Stéphane, Vandeput Étienne
5	15-07-2008	Sent to the reviewers	Rieppi Stéphane, Denis Brigitte, Vandeput Étienne
6	23-07-2008	Reviewed	Amaury Daele, Stéphane Sire
6	25-07-2008	Final (sent to the SC)	Rieppi Stéphane, Vandeput Étienne

Deliverable manager

- Étienne, Vandeput, ULg

List of Contributors

- Brigitte, Denis, ULg
- Cécile, Roisin, INRIA
- Déborah, Malengrez, ULg
- Étienne, Vandeput, ULg
- Irène, Vatton, INRIA
- Jean-David, Labails, CRP-HT
- Maryse, Ledent, ULg
- Manolis, Tzagarakis, CTI
- Stéphane, Rieppi, ULg
- Sandy, El Helou, EPFL

List of Evaluators

- Stéphane, Sire, EPFL
- Amaury, Daele, UNIFR

Summary

This deliverable aims at measuring the collaboration between developers and expert users (researchers/CoP mediators) in charge of a usability analysis of the PALETTE tools and services. It compares, from an ergonomic point of view, the development of these tools and services at an early time of the project with the current and future developments.

It is to notice that, because they used all of them in the real framework of their daily job, the team of expert users consider themselves to be good representatives of a CoP members' audience.

As it was a big task generating lots of working documents, the deliverable tries to synthesise where effort has been made and which results it produced.

Supplementary suggestions are made in order to help developers to still move forward on the way of a good acceptability of these tools and services.

Table of contents

1. Introduction

On the main page of the PALETTE project website (section “Objectives”) you can read the following sentence: “*It has been recognised that web-based technologies could support the development of virtual CoPs.*”. This single sentence explains why the project is built on the idea that those technologies can be exploited in order to develop tools and online services helping CoPs growing, organising activities, sharing and capitalising knowledge. It seemed interesting to analyse what can be considered like the first contact between the user (any CoP member) and a tool or service: its user interface. Evidently, the usability of an interface is not the only thing that matters, but user's tenacity is not always very high and to neglect these aspects could be dangerous because it helps his/her motivation going forward.

From the beginning, there has been a sensitive issue to deal with. But in the course of time, this issue has found good responses. When thinking about their software, developers had (and always have) in mind an audience. The team of expert users in charge of the usability analysis had also (and always has) in mind an audience. The issue to face first was to agree on a common user profile and further to try to determine which kind of constraints these users were ready to accept, before leading any usability analysis.

The concern of the team of expert users was really to produce every necessary effort making easier the adoption of the tools and services by CoP members. They considered themselves like representatives of the CoPs. To lead this usability analysis, they decided to use the tools and services on the occasion of their own work to give more credibility to their comments and suggestions.

Developers had also to make their software evolve, not only from an ergonomic point of view, but with the aim of adding new functionalities fitting to generic scenarios. So they had to be at two places at once. That may explain that the challenge was sizeable.

To help developers accomplishing their task, the team of users led the operation in three steps.

- First of all, each tool or service described hereafter has been analysed using a precise method that led to the sending of a detailed feedback to the developers.
- Secondly, developers have made their tool or service evolve.
- Thirdly, progress has been evaluated and remaining issues have been examined aiming at future improvements.

Of course, depending on the tools or services, these three steps have not always been so separated. These differences are detailed in section 2. Moreover, various types of contacts between analysers and developers have allowed interesting discussions leading to useful compromises.

A large part of the analysis was over when the D.PAR.04 (“User centred description of PALETTE tools and services, and first analysis of usability”) was written. That is why only some of its elements have been exploited in this deliverable through the description of the ergonomist's point of view. In these sections (one by tool or service), emphasis has been put on pieces of advice to provide in order to prevent the user with all kinds of useless workload and waste of time.

Lots of usability issues identified at that time have been fixed. This deliverable aims at taking stock of the situation, tool by tool or service by service, but also more globally (see section 4), trying to show how this usability analysis has affected their development by taking care of the user and his/her context of use. **It does not take into account nor evaluate new features**

which are not directly linked to the first analysis except when they are supposed to solve part of some usability issues identified at this moment.

The next section (2) describes how the analysis has been led. Some differences between tools and services are also explained in relation to their current state of development, their relative complexity, the relative stability of the idea of the task they are supposed to model. In section 3, and for each tool or service, main user goals are shortly described. They are examined taking into account the usability issues identified when trying to reach these goals in the version which has been the object of the analysis. Improvements are mentioned, either because they are effective or they are promised by the developers. Section 4 focuses on ergonomic criteria and take a look at the set of services in order to determine on which of these criteria were located most of the situations. This could be useful when looking for guidelines to adopt in order to build some federating tools like the portal, the showroom, CroSSE... (section 5).

This plan shows how usability analysis has contributed to make PALETTE tools and services more acceptable for CoP members.

Developers have suggested some relevant changes in section 3. In each paragraph C of this section, they tell us a bit more about the future development of their tool or service having regard to the results of the first usability analysis. For that reason, their names have been added to the list of contributors.

Finally, a last comment: it has often been a temptation to insert more images because of their evocative power, but we had to keep a cool head. The ones that have been chosen are distributed as fairly as possible on the set of tools and services. They illustrate real improvements of usability through the recognition of ergonomic criteria and can be found in section 4.

2. Methodological considerations

2.1 Leading a first usability analysis

2.1.1 Generalities

A same global process has been used to lead a first usability analysis of the six following tools and services: Amaya, SweetWiki, LimSee3, CoPe_it!, eLogbook and BayFac. The main outcome is a set of six usability reports. Depending on the development progress, each report has been transmitted to the fitting development team as a working document. In agreement with the development teams, the content of this report has been considered like a starting point for discussions. Such discussions aimed at defining optimal solutions regarding the possibly different opinions. In this section, this first process is described as clearly as possible.

Prominence has been given to the user's point of view. Obviously, the user's and the designer's points of view are not often quite different. On one hand, the designer takes into account various implementation constraints and, as it is the case in the PALETTE project, may quite rightly focus on standards use and interoperability objectives. On the other hand, the user's point of view is linked to both his/her mental representation of how the system works and a relatively shared (that means well defined) task model. Taking into account a user's point of view means that the audience to which s/he is presumed to belong is well known. Considering that, to give sense to these reports, the audience had to be qualified as finely as possible, and to make clear in which tasks provided by the services and the tools CoP members are interested.

Audience

A good knowledge of the target audience mainly helps specifying user's experience and his/her habituation when dealing with electronic interfaces. D.PAR.01 “Grid of analysis supporting the participative design methodology” provided general considerations on this issue by collecting data from the CoP. D.PAR.03 “Description of six scenarios and results of six validated trials” also describes the “audience” *i.e.* the CoP members and their technical skills. So we have done the hypothesis that *the experience of most of the CoP members is limited to a “not necessary efficient” use of well known edition, communication and navigation tools*. A reference could also be done to the experience of the members of another CoP: the “non technical” partners of the PALETTE project.

Tasks

Circumscribing the task helps focusing on most of the user's goals when s/he is using the service or the tool. This way of working allows the partners to solve the more frequent problems and to increase significantly the usability of their tool or service. For instance, a symbol on a button may not carry a strong meaning. But if that button is used only when one has to proceed a very special and rare task, the issue is not so critic as if it had to be used to achieve a very frequent goal (let us say, printing, for instance). That is why reports about ergonomic problems must not be considered as exhaustive since focus was on the more frequent goals.

Moreover, very often, a global task can be broken down into different sub-tasks levels and the process can be repeated. So, a kind of goals tree is created and it becomes more convenient to consider the lowest levels, the leaves of the tree as user's goals. For instance, during a task of text edition with Amaya, the user may wish to move a part of a text. Such an action may be considered as a user's goal at a particular moment of use. As another example, debating with

CoPe_it! can be broken up to produce a sub-level “workspace management” and further, a goal “linking a document to a comment”.

It is convenient to carry out usability inspections starting from the most elementary goals. Very precise indications may be provided. On the other hand, it is easy to understand why this deliverable will focus on the main goals.

2.1.2 Method

In order to lead this usability analysis and to take into account the previous remarks, the method described hereafter has been adopted. For each service (or tool) the task presumed has been specified and the goals have been detailed for one or more sub-tasks related to the user's profile. Their usability has been analysed from them. A team of expert users have analysed the services and tools trying to achieve these goals. They have observed a series of usability issues and have treated them in a same way. To complete analysis, they have also proceeded by a “systematic walkthrough” technique based on Nogier (2005). Indeed, some considerations may be done without having in mind a particular goal. Using both techniques, they expected to be as efficient and accurate as possible. They also hoped to provide the developers with a wide analysis of the software acceptability.

One may claim that it would have been interesting to also work with end-users. In fact, it was the case because what is called “a team of expert users” was composed with people, using the tools and services in the framework of their daily job. In addition, nearly all those people were CoPs coordinators and/or CoPs trainers. So, they could complete their analysis with information coming from the observation of CoP members working with the tools and services. Of course, one can argue that this information has been unintentional because no systematic tests have been done.

Each of the six mentioned services and tools has been analysed. Issues identified have been related to Bastien & Scapin (2000) ergonomic criteria or to articulatory distance assessment (Hutchins & al., 1986; Norman, 1986). They have been brought together in a single working document. The framework of this document has evolved throughout the time, but its central part describing the situations did not change. Figure 1 hereafter illustrates an example of situation.

A situation contains notably a description of the issue, a screen shot annotated, suggestions of possible improvement, criteria badly taken into account with regard to theoretical references and goals linked if any. Depending on the tools and services, the number of situation has varied from a big thirty to more than two hundred.

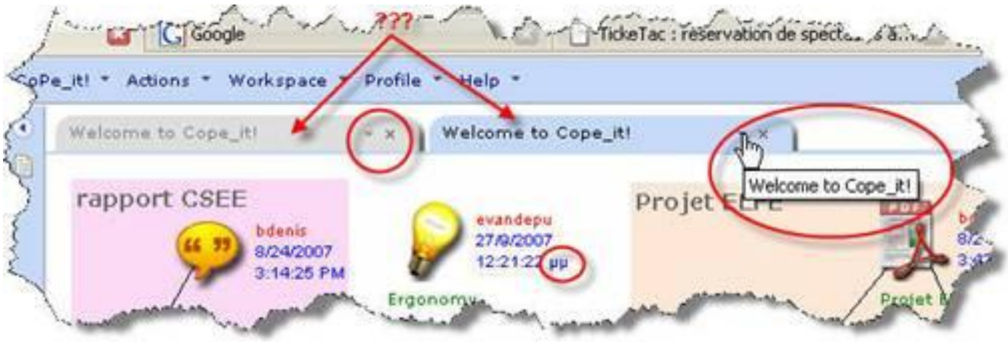
Before sending these working documents to the developers, the different inspectors (whose initials are mentioned) have discussed the relevance of their comments and the reports have been improved. A synthesis has been joined to the list of situations in order to give some global guidelines on the effort to be produced.

Each of the six documents has been a basis for a common work between a team of expert users and developers.

But let us now give more details about the method and some of its underlying concepts: goals, systematic walkthrough, ergonomic criteria and articulatory distance. Let us try to show how they are linked.

Description of the situation

It is not easy to find the way to shift from «Desktop view» to «Forum» or «Formal views» even if you know that it is possible. The arrow in the tab is very tiny and when the mouse pointer is over, the balloon help contains only the name of the workspace as information. Moreover, tabs including different views of a same workspace cannot be distinguished.



Suggestion(s)

- To add the functionality in the upper menu, for example by creating a «View» menu
- To add the functionality in the «Quick Links» area as three buttons to be pressed in
- To help distinguishing the different views by adapting the tab (to add a symbol...)
- To keep tabs for the opening of different workspaces but not different views

Notice that menus may be organised differently. A «Communities» menu should gather the items of the current «CoPe_it!» and «Actions» menus. A «View» menu should provide the choice of different views of the current workspace instead of using tabs (see synthesis). A shared space for the «Formal» and «Desktop» views is also desirable.

Reference(s)

Prompting, consistency

Main goal(s)

To get a synthetic view of a discussion

Initials

ML, DM, EV

Figure 1

List of goals

Goals have played an important role in the usability analysis. They have been a kind of reference for the inspectors. Of course, a list of goals depends on the task the service is designed to achieve. That is why, in each case, the description of the tool or service by the developer has been helpful to define these goals. But in some cases, the list has been limited because inspectors considered either that they were presently to far from the CoP members habits or that those functionalities were not acceptable enough (from an ergonomic point of view). As an example, managing the "folksonomy" with SweetWiki has been considered like a very specific goal. Only a few CoP coordinators would try to achieve such a goal. Moreover, whereas this goal is not so frequent, developers did not care much with ergonomics. As a consequence, this goal has not been taken into consideration. Figure 2 shows an example of division of a task in goals and sub-goals.

Here is an excerpt from a list of goals defined with respect to Amaya. It is linked to a document edition task. In this case, the task has been broken up in five sub-tasks.

These sub-tasks are:

- content edition
- layout
- content and layout
- navigation
- information management

For each sub-task, different goals have been selected. Here is an example for the sub-task «content edition»:

- to move the pointer from the current position to...
- to select a block
- to insert a table
- to insert an image
- ...

Figure 2

As an example, considering the goal "to select a block", an inspector tries to achieve this goal when the block is a word, a paragraph, an image... and can observe that increasing the selection (from a word to a paragraph) is possible by pushing the *Esc* key. S/he can conclude that this approach is not compatible with other edition software.

Systematic walkthrough

Some usability issues can be detected without being directly related to a task or because they refer to recommended guidelines (Apple 2007), for instance, a bad screen layout or a long list in a menu. That is why it can be interesting to examine the interface without any goal in mind. Issues identified this way are easier to classify using Nogier's topics (2005). Nogier classifies potential issues or qualify interfaces using topics which do not necessarily refer to user goals. These topics are another particular way to gather the situations. They consist in five categories: user interface, display, user assistance, terminology and others. The first three categories are subdivided (table 1).

Category	Sub-categories
User interface	Menus/Toolboxes Dialog boxes Buttons Hyperlinks Keyboard Direct manipulation (mouse)
Display	Screen organisation Graphics
User assistance	Error treatment Help
Terminology	
Others	

Table 1

The topic “User interface” is related to interaction objects what makes easy any observation of them. “Error treatment” is more ambiguous because it refers to a more dynamic process. Errors do not appear spontaneously by clicking here and there. “Terminology” can be examined exploring screens, menus, dialog boxes, never mind the goal. “Others” is, of course, a catch-all without importance in this context.

Bastien and Scapin's criteria

Bastien & Scapin (2000) defined eight main ergonomic criteria to qualify the interaction between human beings and machine. These criteria seem to be a good synthesis of different previous surveys, theories and heuristics (Nielsen 1994, Shneiderman 1992, Hutchins, Hollan & Norman, 1986). This classification is particularly convenient when analysing a Web interface in a particular context (see also Lompré 2007). That means they characterise the issue “how does the user render the information sent by the system through the interface?”.

These criteria are: guidance, workload, explicit control, adaptability, error management, internal consistency, significance of codes and compatibility. The more the user is inexperienced, the more they have to be optimised. Some of them are broken up on one or two levels. The total number of criteria and sub-criteria is eighteen. The fields covered by the main criteria are shortly explained in section 4.

Several cross references can be done with Nogier's topics, for instance in what error treatment refers to. User assistance refers to user guidance and so on. But our considerations are a bit different. Observations focus on what an observer may find a bit more randomly. Anyway, it is not useless to detect some issues using, from time to time, another technique. It just allows completing the identification.

Articulatory distance

Bastien and Scapin's criteria do not focus on sequences of actions. Sometimes, to achieve a simple goal, the user has to accomplish a long series of actions. That is why in some cases it is spoken about “articulatory distance” issues ((Hutchins & al., 1986; Norman, 1986). They generate systematically a useless cognitive load that may discourage the user in the long run. Articulatory distance can be considered at two levels: execution of the task and evaluation of its achievement. To a certain extent, achievement may be related to the feedback sent by the system which is analysed through the B & S criteria. So, articulatory distance will be considered only at the level of execution (see section 4.2.9).

2.1.3 Results of ergonomic analysis

Outputs

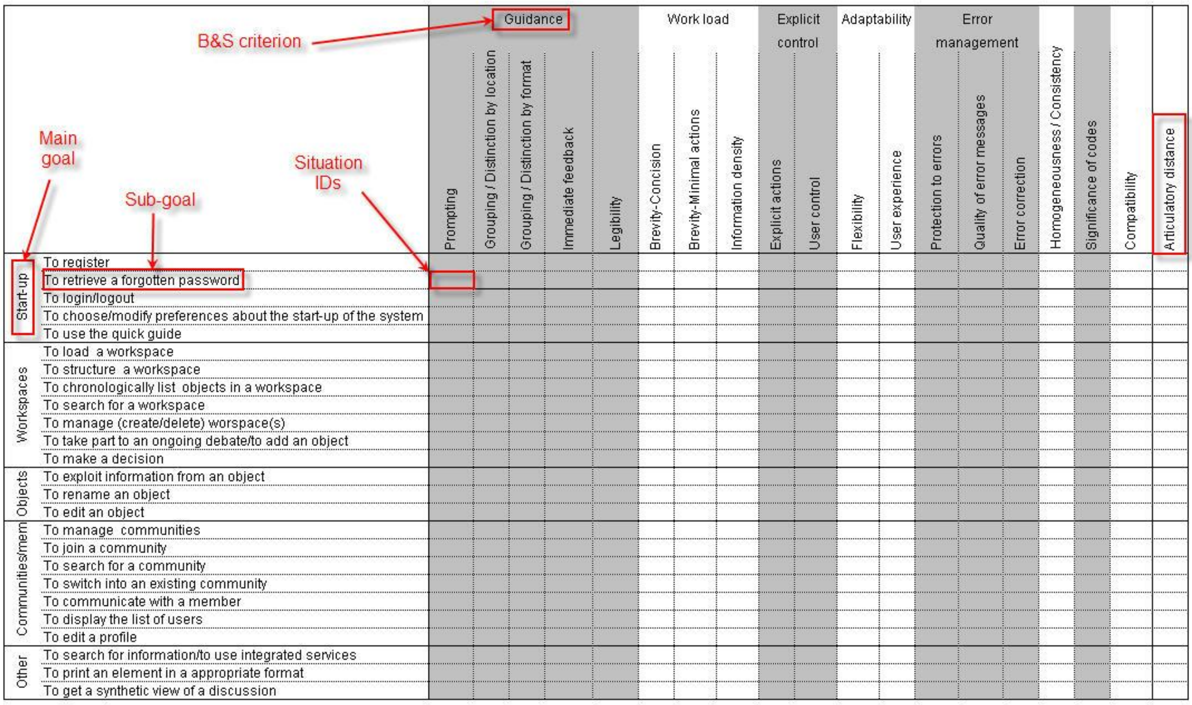
For each of the six tools and services, a report has been established.

The report contains a list of situations. This list is an important output. In spite of the fact that it is a working document, it has allowed, in most of the cases, one or more long and constructive discussions between developers and inspectors to determine which solutions were prior and which solutions could be chosen. The feedback has focused on what inspectors considered as essential goals, *i.e.* covering a large percentage of a normal and presumed user's activity. This feedback is important because it relates to a determining factor for the adoption of the service (or tool) by less convinced users.

The report contains also a synthesis (two or three pages). Issues identified may be numerous and sometimes not very significant when separately taken into account. That is why the synthesis tries to summarise them by giving prominence to general ergonomic guidelines. To give sense to these guidelines and main comments, a double entry table shows the relationship between goals and ergonomic criteria (from Bastien and Scapin, 2000) through the issues identified.

Owing to this table, developers get the opportunity to concretely visualise the nature of the problem: in the report, table cells have been filled in with numbers referring to specific examples fitting to the list. So, through a quick look at these tables, they have got a good idea of the most important critical points of the interface inspection.

As an example, have a look hereafter at the empty table (table 2) drawn up to analyse CoPe_it! Goals and sub-goals are listed vertically. The Bastien & Scapin criteria and sub-criteria are listed horizontally. In the synthesis, each cell contains IDs of issues like they appear in the detailed list. You can also observe that the last column concerns issues of articulatory distance.


B&S criterion
Main goal
Sub-goal
Situation IDs

		Guidance					Work load			Explicit control		Adaptability		Error management						
		Prompting	Grouping / Distinction by location	Grouping / Distinction by format	Immediate feedback	Legibility	Brevity-Concision	Brevity-Minimal actions	Information density	Explicit actions	User control	Flexibility	User experience	Protection to errors	Quality of error messages	Error correction	Homogeneousness / Consistency	Significance of codes	Compatibility	Articulatory distance
Start-up	To register																			
	To retrieve a forgotten password																			
	To login/logout																			
	To choose/modify preferences about the start-up of the system																			
	To use the quick guide																			
Workspaces	To load a workspace																			
	To structure a workspace																			
	To chronologically list objects in a workspace																			
	To search for a workspace																			
	To manage (create/delete) worspace(s)																			
	To take part to an ongoing debate/to add an object																			
	To make a decision																			
Objects	To exploit information from an object																			
	To rename an object																			
	To edit an object																			
Communities/mem	To manage communities																			
	To join a community																			
	To search for a community																			
	To switch into an existing community																			
	To communicate with a member																			
	To display the list of users																			
	To edit a profile																			
Other	To search for information/to use integrated services																			
	To print an element in a appropriate format																			
	To get a synthetic view of a discussion																			

Table 2

Notice that, earlier in the project, the synthesis and the list of situations were separated. The first list was produced for Amaya. Because it was a working document, it has been written in French. The other lists have been written in English. All the synthesis documents have been written in English.

2.2 How changes have been measured

2.2.1 Before, now and after

This deliverable aims at measuring a difference, an evolution. Concrete elements are available to assess that evolution. First of all, of course, there is the list of situations. Focusing on the main goals, observation consists in measuring how many issues corresponding to a given goal have been fixed. And concerning those who have not been fixed, developers may explain what they intend to change in a very close future. It is the reason why some contacts have been taken just before writing this deliverable. It was important to know which supplementary changes were planned by the end of the project. So, the assessment could be refined, taking into account the list from the first usability analysis, the current (or at least recent) versions and the developers' plans for the next months.

2.2.2 Adapting the assessment

Tools and services must be considered differently for several reasons.

Some of them (Amaya, SweetWiki) were developed a couple of years before the beginning of the project. The analysis could start immediately and focus on real usability issues. That means that developers did not have to imagine what kind of task their tool or service was intended to model. They just had to concentrate on real interface issues. That did not prevent them thinking about new features, but they could do it as they wish. On the other hand, inspectors knew precisely what kind of goals they had to assess and they certainly were more demanding. For Amaya and SweetWiki, analyse could take place in one shot. Adjust was made quickly after discussions during one or two conference calls. In the case of Amaya, a new interface adapted to the main audience of the project was even proposed, negotiated and globally accepted.

For more recent services like CoPe_it! and eLogbook, the challenge was bigger. At the time of the development, developers had to precise the task their service was intended to model. Rather than adapting a service to a context, they had to make from the context an important part of the reflection. In such cases, that is as evidence that, (at first stage) ergonomic concerns are generally not the more important thing to deal with. For eLogbook, changes have been done "in the action", by successive touches. Like eLogbook, CoPe_it! had to find its way. So it has evolved from a mediation service to a collaborative one. It is also clear that a usability analysis is more difficult to lead because changes are frequent and substantial.

For a new tool like LimSee3 and a new service like BayFac, the problem is once again different. The task each of them models is also well known (multimedia authoring, document categorisation and search) but they are developed from scratch. As a consequence, LimSee3 is far to be a finished product and a usability analysis comes probably too early. Situations are numerous and it is difficult to determine which of them are prior. From a user's point of view, the task modelled by BayFac is very simple. The number of goals is very limited. That is why developers could care the interface compound of very few screens.

In a certain sense, evaluation of CoPe_it!, eLogbook and LimSee3 has to be regarded as a formative evaluation rather than a summative one.

3. Description of the ergonomic progress

In this section, the focus is put on goals – or more precisely on issues encountered while trying to reach the most frequent goals – and if they were fixed or not in the most recent version of the tools and services at the time of writing this deliverable. As noticed in the introduction of this document, no screen captures will be used in this section, which is by far the largest of all, in order to keep it as concise and factual as possible.

Changes in the tools and services have been highlighted thanks to various data (number of situations, observable changes, the implementation of new functionalities and developers' intentions). See 2.2, "How changes have been measured". At this point, it is important to notice that whatever the tool or service, the quoted issues and solutions are relevant with respect to their evolution but in no way do they constitute an exhaustive list.

The following table shows, for each tool and service, which versions were used for the first analysis and which one was the last available version at the time of writing this deliverable. Please note however that there were a lot of incremental changes between the versions, at a different pace, depending on the tool or service. For instance, even if eLogbook has been tested from version 1 (October 2007) to version 2 (July 2008), there were incremental changes every other day.

This was a very dynamic process, in the spirit of participatory design: bug reports or enhancement requests were sometimes fixed/enhanced hours after they had been pointed out to developers. Therefore, as often with version numbers, the table 3 seems a bit "artificial" and is to be taken as a simple indication.

Tool/service	First analysed version	Present version at the time of writing this deliverable
Amaya	9.54	10.1pre3
SweetWiki	1.2	1.7
LimSee3	0.9.7	1.1.9
CoPe_it!	2.03	2.2
eLogbook	1	2
BayFac	1.1	1.3

Table 3

3.1. Amaya

3.1.1 Utility and goals

Amaya is the first tool for which an ergonomic analysis for PALETTE has been conduced and is probably the one for which the most time talking between analysers and a development team was spent. Partly for this reason, the methodology used was a bit different. Another reason is that Amaya has been in development for many years (the first beta release dates from 1996): a major difference with other PALETTE's tools who are in their first years of development.

Amaya is presented as a Web browser and editor, although its developers do not value the Web browser aspect too much (support for CSS is lacking and no support for JavaScript and Plug-ins). To fit with the presumed needs of the CoPs, the ergonomic analysis was made on a subset of Amaya's editing abilities, focusing on creating simple, valid and interchangeable Web documents, with a focus on the semantic quality of the code and on the substance of the text, not its form. As such, it was analysed as a simple editor providing the user with functions to define semantics of blocks, this definition being associated to an automatic layout process. This choice was explained by the trial of Amaya in CoPs and the use CoP members thought they can have of the tool. Another argument was to insist on the quality of the content, the relevance of the blocks semantics and to considerably simplify the layout.

Considering Amaya as such a document editor, three main goals were taken into consideration:

- managing a document
- editing the content of a document
- designing the layout of a document

Due to the high number of situations noticed in Amaya, examples that follow are a few concrete examples of some of the most evident issues, with a fair balance of fixed ones and still to be fixed ones. Please also note once again that even if Amaya has more situations than other tools and services, it does not mean that Amaya is weaker than other PALETTE tools and services. On the contrary, its maturity allowed us to analyse it in a very detailed fashion, which would have been difficult or senseless in a prototype tool.

3.1.2 Analysis of goals through PALETTE's time frame

In early 2007, resulting from a first usability analysis, researchers in charge of it had a number of phone discussions with Amaya's development team. An important signal they sent was that Amaya's then current version (9.54) was too hard to understand for laymen not aware of computer intrinsics (particularly the XML language and the way of thinking that XML suggests). They suggested a screen mock-up showing an interface closer to CoP member's habits (*cf.* D.PAR.03 “Description of six scenarios and results of six validated trials”). This mock-up has been commented and following the main recommendations, Amaya's developers, implemented that simpler interface naming the associated version “Amaya Editor Lite”. In February 2008, a brand new Amaya was made available, that had the option of running in “Lite” mode or in full mode.

The following analysis will first compare Amaya 9.54 with the “Lite” version of 10.0 and subsequent versions (10.1pre3 at the time of writing this document).

Managing a document

Toward a less confusing new document creation

In this context, managing documents reduce to creating, opening, closing a document, but also sending a document by email.

The creation process is worth a large comment. The first usability analysis has highlighted several issues. Some of them have been solved. For instance, in the “New document” dialog box, the text field containing the path of the document was redundant and has disappeared. Many other values still remain in the dialog box and they may disturb the user (information density). They are important, but most of the time, they can be provided by default and could appear on demand (flexibility) owing to an “Advanced options” button, for instance. The creation of a new document is still an issue for inexperienced users for the following reason: a document name (path + filename) is automatically generated and, logically, the user should change it immediately. That process is rather different of what usually happens (compatibility with other systems). In a large majority of cases, the document is saved when the user decides to save it or when s/he decides to close the document. What happens is that users are not conscious to save documents whose name is “New.html” and that conflicts may occur. Nevertheless, developers argue that naming a document when it is just created allows to make links to it (for instance, if you are editing several pages in the same time). So, it seems that the relevant moment to name a document does not exist, alas.

Other short details could be solved like the message “Title is mandatory” which is not visible enough. In version 10.1pre3, the title is automatically generated from the filename. It is a good solution if the page is not destined to be indexed by a search engine.

In all fairness, the “New document” dialog box is not a good example of what collaboration between researchers and developers has really been. Amaya's developers hesitate to modify it thinking about expert users familiar with older Amaya versions. It remains that there is a very big interest in modifying this dialog box because the creation of a document is a very frequent goal whatever the audience.

With Amaya 9.54, it was not possible to open a recent document through the menu bar. That functionality is now available.

Sending a document by email was not directly possible from Amaya and risky out of Amaya because of the nature of a web page. Nevertheless, it was a strong requirement from CoP members. Editing of XHTML document was useless when they wish to share their documents by mail (if lacking of any available Web server). This functionality is now available. Amaya can either send the XHTML document or create a zip file before sending it. In the second case, the receiver unzips the file and the page can be directly displayed. Links are well managed.

Editing the content of a document

Toward a more standard keyboard and mouse handling

Content editing often consists in simple but frequent handlings: move the cursor, select a block of text, cut and paste a block of text, insert an item in a list, etc. These handlings are made either with the mouse (touchpad) or with the keyboard. Keeping the same concern (to avoid the users changing their habits when they are in a particular program), analysis has focused on which handlings could be standardised.

Moving the cursor uses the classical shortcuts. Selection of a block is also made using the usual shortcuts (*e.g.* click – Shift + click). Notice that when the user clicks on a text to select a part of it by drag and drop, Amaya sometimes 'forgets' to notice when the button is being released and, as a consequence, the selection acts crazy. This is probably a bug limited to a particular OS that developers promised to solve.

Otherwise, users often try to handle some objects dynamically (images, tables). It is possible since very recent versions to resize an image dynamically and to choose the way the text will run around it. For the cells of a table, it still does not work but is under development.

Automatic selection is a bit more particular: standard for a “word” (double click) but not for the extension of the selection (F2 or Esc). Esc should be forgotten as for the command F2, it is well worth a “tip of the year”.

Some of Amaya's keyboard shortcuts are standard (copy, cut, paste, save, print) and others are not (new, open). The developers of Amaya are in a dilemma: either they keep the shortcuts from old versions and not disturb their usual audience, either they modify these shortcuts in order to extend the standardisation. The answer is not so simple, considering for example that it is not sure that inexperienced users actually use shortcuts.

Notice that Amaya's help explains how to customise shortcuts through the edition of a text file. But can anybody carry out such a procedure without being an expert?

Insertion of an item in a list works now like in a word processor if you decide to disable the XML mode (to be done in the preferences). There is another interesting evolution of the Amaya interface. Context menus are now available. These two improvements definitely increase the compatibility with other systems.

Designing the layout of a document

Toward a more human-oriented human-machine dialog

Users like CoP members are generally interested by any tool allowing them to produce text. But very quickly, because of their (bad) habits of text editing, they have new requirements. Many experiences with several groups show that these first requirements are insertion of images and insertion/management of tables. More, they need to design the layout of their document in the same time they insert new content.

The way HTML language is rendered by browsers is a kind of encouragement to show them the good way of editing: focus on the content, on the semantics of its parts and simply choose a global style for the layout *a priori* or *a posteriori*. Version 10.x of Amaya tries to promote this view of editing. To understand the progress made, it is interesting to compare Amaya 9.54 with Amaya 10.x.

The human-machine dialogue in Amaya 9.54 was written with XHTML/XML experts in mind. It was not adapted to CoP members' habits (often using word processing programs). For instance, as well in the tool box as in some long menus, human concepts as different as adding an image, an hypertext link, a table or even putting a text in bold or italics were mixed without apparent logic for anybody not very HTML conscious. The menus, instead of the usual “File”, “Edit”, “View”, “Insert”, “Format”, etc., were: “XHTML”, “XML”, “Style”, “Annotations”, “Cooperation”, etc. The toolbox icons for bold and italics were respectively S and E, which only make sense if you know the XHTML semantic tags <strong> and <em> (emphasis).

For Amaya 10.x, menus and toolboxes were completely rewritten with the described concern in mind. The strategy to design these new menus was to find invariants in multiple other similar text editing applications (since the focus was to reproduce some of their editing features). The semantic marking is shown through the “Elements” box but with the help of more conventional icons. The layout may be carried out very simply through the choice of a “theme” in a list. Nothing is comparable with the previous version.

Table management remains heavy and is a possible and sizeable reason to give up for some CoP members who get used to resize objects dynamically. This observation has been done many times. Obviously, a particular effort in this direction should improve Amaya as much as the new toolboxes did. For the management of images, very recent and noticeable improvements have been made. Users are looking forward the same improvements for table management.

3.1.3 Expected future

Insisting on the idea to focus on the content and its semantic structuring, the theme system asks to be developed further. Indeed, only two (Classic and Modern) themes are available, and they are very minimalistic. They could benefit from some more design work. Notice that an expert user/CoP animator has the opportunity to create customised style sheets to replace the Classic and Modern ones, but it should be interesting to be able to create supplementary style sheets.

Developers pull out all the stops to give the users the opportunity to define, resize tables dynamically. For the same reason themes are very useful, templates will add value to the software. Nevertheless, what is possible is that relevant templates suggestions will be made by CoP members only when they will use Amaya intensively.

3.2. SweetWiki

3.2.1 Utility and goals

As its name suggests, SweetWiki is a wiki engine. It is different from other wikis because it includes some pieces of functionality which are usually not provided. The main functions amongst them are:

- the opportunity to edit a page in a WYSIWYG environment (no particular syntax is needed except if you want to work on the HTML code of the page);
- a page tagging system which generates a folksonomy (this folksonomy can afterwards be organised hierarchically);
- automatic creation of a home page which is related to the login of the user.

The first function is not directly dedicated to CoP members, but it is a bonus when you think about the audience's technical competencies (generally not accustomed to write any formal syntaxes). But page tagging can be directly related to knowledge management through the concept of ontology because folksonomies may be organised into a hierarchy and the home page encourages members to explain who they are, which roles they may play in the CoP, etc.

Like any other wiki, SweetWiki is very well adapted for tasks aiming at sharing, capitalizing knowledge and less convenient for discussions and debates. The more users there are, the more pages are written. Linking pages through hyperlinks is not always relevant. A tagging system proves to be useful. Others functions like recent changes, keyword or other advanced searches complete a very efficient tag search (if the user uses to tag the pages referring to a personal – or collective – logic).

Notice that the organisation of the tags into a hierarchy increases the chance to find the searched page. The “tag search” results page includes links to the pages so tagged but also links to other “tag results” pages, those of related tags.

The main goals of a CoP member using SweetWiki are:

- editing the content of an article (including the tagging which is a form of editing)
- designing the layout of an article
- browsing the wiki
- managing documents

3.2.2 Analysis of goals through PALETTE's time frame

Editing a page content

Toward an editing process clear and without bad surprise

As explained in the next paragraph, SweetWiki uses the Kupu editor that is not implemented by SweetWiki's developers. A Kupu-independent issue though, is that too long editing sessions are not well managed: if the user tries to save the page after more than thirty minutes, Java exceptions are generated and the process to get the previous page version is heavy. You may say that editing a wiki article should not take more time, but curiously, the editor is so user-friendly that you feel like being in front of a word processing interface. The delay has now been increased to one hour. It is more reasonable, but still insufficient is some cases.

Otherwise, the editing process has considerably been improved. Editing a page implies that the user has to connect him(her)self. The login elements have been moved to the top of the screen and are prominently displayed. Once connected, the “Register” link is replaced by another one: “Edit this page”. The not very significant “Administration” zone (which appeared in that case) has disappeared, but the very useful “New Page” section remains.

Designing the layout of an article

Toward a faithful code generation

Unfortunately, SweetWiki does not manage the process of content editing by itself. The editor Kupu takes charge of editing and one may regret that some bugs are resulting of the transfer between the used version of Kupu and SweetWiki. For instance, some style changes (font weight, colour, size...) produce curious results (the code is modified during the transfer) when the page is browsed by SweetWiki. The SweetWiki developers are not responsible for these bugs and so, the best thing to do is to warn users of this kind of issues. Notice that the advantage of Kupu is to provide a graphical interface as complete as possible, preventing the users to write any syntactical expression when s/he wants to assign semantics to the blocks of text.

Browsing the wiki

Toward another way of browsing

Varied search functionalities already existed: tags, keywords, recent changes, links related to the hierarchy, advanced search (by author, by web, by date...). The display of values (semi-automatic data entry) in the tag search process has been improved.

Globally, the interface has been simplified. Useless commands (useless in the sense that they were only useful for developers – such as debugging functions or a SPARQL query system) have disappeared. That makes links more visible either if you want to browse other pages (*i.e.* after a tag search) or you search for a function command.

Managing documents

Toward more flexibility in documents management

No particular roles are defined in SweetWiki. Everybody has the opportunity to modify any page, even the homepage of somebody else. Namely because everybody can tag any page as his/her wishes (but not only), it is not possible to modify the name of a page or to delete a page. As a consequence, each web contains lots of useless pages. Most of the users find it is a shame.

From the beginning, the different versions of a page are listed chronologically. The date and hour format is now clear, but it is very difficult (impossible) to construe the values.

The request from the users to be automatically aware (let us say by mail) of changes in the wiki has not been completely satisfied, but they have the opportunity to subscribe to RSS feeds in order to be aware of last updates (name of and link to the pages, list of tags, etc.). It is to be noted that the RSS concept is still hard to grasp from the point of view of a layman. Even if an e-mail can be seen as more intrusive, it has the advantage of not requiring any action from the user for it to be sent. Moreover, experience has shown that e-mail is the only tool that (almost) every CoP member use in a daily fashion.

3.2.3 Expected future

From a usability point of view, and because of its stage of development, SweetWiki will probably not evolve much in the future. Wiki functionalities were well known at the moment of adapting the service to make it fit with the CoPs needs: creation of different webs linked to knowledge capitalisation activities of CoPs, management of folksonomies to help them creating their own identity and manage information, possible increase of members activity through RSS feed, etc. Nevertheless, SweetWiki developers plan to program an ergonomic folksonomy editor very soon. Such an editor will help any CoP coordinator or involved member in organising the tags to improve the “search by tags” system.

Otherwise, the effort is now concentrated on the development of functions allowing import to and export from other PALETTE services (notably Amaya and CoPe_it!).

Flashbacking to the first usability analysis, the user may keep some requirements in mind. For instance, the interface of Kupu does not indicate clearly enough that the editing process must end (developers announce they could work on the WYSIWYG editor by the end of the project). A message could be sent to the user when the editing time is just about to exceed the limit. At least, the system might not generate an exception in that case because users despair and become discouraged.

3.3. LimSee3

3.3.1 Utility and goals

LimSee3 is a tool whose utility can be hard to grasp at first, a fact that is easily explainable by the fact that it is an applied research tool. Of course, most of the other PALETTE tools analysed here are too, but the specificity of LimSee3 is that what it aims to accomplish has no direct equivalent and that the development has been done from scratch at the beginning of PALETTE (as for CoPe_it! and eLogbook). This applied research aspect implies that, from the words of the developers themselves, using LimSee3 is next to impossible if the user does not first get a bit familiar with the tool thanks to a reading of the manual.

That being said, one can define LimSee3 as an authoring tool, somewhere between a presentation tool (Microsoft PowerPoint OpenOffice.org Impress) and a multimedia tool (Adobe Director, Adobe Flash, Apple iMovie), dedicated to the creation of multimedia presentations targeted to several standard formats like SMIL and XHTML. SMIL (Synchronized Multimedia Integration Language) is a W3C XML-based mark-up language used to describe multimedia presentations.

The tool was not really ready for the D.PAR.04. A subsequent version (1.0.5) was ergonomically analysed after the D.PAR.04 and there still were a lot of issues with the tool.

At the time of writing this document though, the current version is 1.1.9 and lots of bugs are gone. A lot of ergonomic feedback has been taken into account too.

What makes the originality of LimSee3, compared to other multimedia editors (*i.e.* SMIL, Flash) is that it completely abstracts the presentation language for the user, thanks to a system of templates that allows him/her to never encounter even one line of code.

Here is a simplified list of goals LimSee3 users will have to achieve, for each of which an example detailing the tool's progress will de given in the next section.

- Install the application
- Manage a library
- Manage a multimedia document
- Publish a multimedia document

3.3.2 Analysis of goals through PALETTE's timeframe

Install the application

Toward a simplified installation process

In current versions, the installation process suffers for some issues mostly due to two facts.

- The installation system works only online: it is not possible for now to install LimSee3 on an offline PC (or it is, but with the help of a technician). LimSee3's team plans to buy a commercial installer in order to resolve this issue.
- The installation system opens an alarming pop-up telling that the application is not signed and therefore maybe not secure. The application, in fact, is as secure as can be, but a signature certificate costs half a thousand euros and has not been judged relevant to be taken for this application.

Both these issues are explained in the online manual, but experience shows that people rarely read application manuals, leading to a major guidance issue.

Budgetary reasons are the root of the issue in both cases, not technical ones. The first one will be solved in the future when the license will be bought and a better advice in the interface should be provided for the second one.

Manage a library

Toward a bookmarks-like library

LimSee3 works with an optional library system, which acts very much like a bookmark window. It is possible to put both local files and Web files in this library. The files are not added themselves to the library, only links to the files are. Older versions had some issues with the library: it did not work if the path included accented characters and it was not possible to remove files from a library. A major source of confusion came from the fact that files which were not accessible (either missing on the hard disk, on the web, or with an error in the path) were italicized, which did not mean a lot to the end user. Newer versions use a red background for not accessible files, which is much clearer. The other issues (accented characters and removing a file) were also fixed.

Another issue with the library was that, even if it use was optional, the library frame was opened when first launching the program. It is now closed by default, with an easy to find tab at the right of the screen for the user who wants to use the functionality.

Manage a multimedia document

Toward a fluid edition of documents

Editing documents has some major issues and a few bugs. Most of the bugs were fixed in recent versions (they will not be here enumerated; bugs are not ergonomic issues *per se*, but more of an indication of a software in the early stage of its development). The issues were of different nature, for instance:

- The first slide had to be the splash screen that had to be renamed in order to become a new slide. This has been fixed.
- Adding slides was not intuitive. They were only designed by their name (plus a short, not very descriptive text, only in the English version of the tool). The new versions include not only a better description but particularly a new 'thumbnail' view of the layout of the new slide.

When adding images to slides, it was not possible to resize them, and they sometimes appeared horribly stretched. Even if it was more a feature than a bug (the areas where the stretching occurred were typically title areas, where a large and not high image is expected), it was disappointing. The new version proposes a resizing function, which, even if it still has its share of usability issues, has the merit to be there. Moreover, as this feature is provided thanks to the template mechanism, it will be easy to change the images default parameters in the different document placeholders if the users request it.

Publish a multimedia document

Toward exportation filters

Publishing multimedia documents, the final goal of LimSee3, can produce several standard multimedia formats. The first target has been the SMIL format. The main issue with that fact is that SMIL readers are not very spread and that most users have to install a SMIL reader in order to manage SMIL documents. The developer team is putting a heavy effort to implement a second exportation system to export to XHTML/JavaScript. This service will be provided in the next version, but at the present time it is quite buggy and still a work in progress.

Both these facts should be made clearer in LimSee3, as users will instinctively try to export their documents to XHTML/JavaScript. For laymen not accustomed with SMIL, compatible SMIL readers will be suggested at save time.

3.3.3 Expected future

As of today, LimSee3 is far from perfect and still feels young, but the progresses in the last versions have been tremendous from a usability point of view. The TODO list is still huge, especially in user guidance, and some functions would really help, like a working exportation to XHTML/JavaScript.

Future versions should make the manual less and less needed, since this is a major block in the acceptance of the tool by laymen.

The main services that will be provided in the near future, taking into account LimSee3 development plans and user needs are the following:

- as mentioned, a full export service in XHTML/JavaScript;
- new editing services for the authoring of synchronization between media thanks to the timeline tool;

- authoring services for a new multimedia applications: a third template for registering and annotating, “mock oral exam”, will be added;
- a rich “packaging service” that will allow users to easily publish their multimedia documents taking into account their requirements (for instance, with or without the aggregation of the media of the presentation).

3.4. CoPe_it!

3.4.1 Utility and goals

CoPe_it! was first announced as an online mediation system, allowing online argumentation and decision making. In more recent versions, its developers emphasized its function as a collaboration and learning service destined to CoPs. Its vast scope covers various interaction types, as giving an opinion, qualifying it, adding comments to a debate, producing documents or even conducting a project.

Whatever the task that the CoP wants to lead, four goals are omnipresent. In other words, in order to reach one's goal, members will have to:

- Access the service
- Join a community
- Manage workspaces
- Handle objects supporting interactions

3.4.2 Analysis of goals through PALETTE's time frame

When the user accesses the service and joins a community, s/he meets a few difficulties. However, they are negligible when faced to the two other main goals, which suffer from major guidance, readability and compatibility issues. For those very reasons, the focus has been made on workspace management and the handling of objects supporting interactions. Those two goals are intrinsically linked: if workspaces are badly managed, structured or organized, they become hard to exploit, to understand and to read by users.

Participate in workspaces

Toward more intuitive approaches and a more easy participation

In earlier versions, when the user created his/her workspace, the procedure to choose the public of private character of a workspace was uneasy, mostly because it lacked the visual clues that were added in subsequent versions. These little changes helped users greatly in their understanding of the process.

In the “desktop view”, the user faced a confusing whiteboard. The actions s/he was supposed to perform were not easy to figure without having recourse to the online help. Once s/he had understood how to add objects to the whiteboard, s/he faced another issue: there were various object types and the difference between them was not intuitive. S/he would then have a hard time understanding how to link objects with the help of arrows and frames. In parallel, the user had to face more practical issues (where to add the URL for an image, how to insert a video, how to modify a created object, etc.). As an example of common issues, the icon for adding a document, whatever its kind (image, sound, video), was a Microsoft Word Document icon.

Efforts have been accomplished by the developers to solve these issues. A new, well visible “menu” option appeared on the whiteboard and guides the user in selecting the right kind of object. In the current version of the tool, adding a document (file or URI address) has lost its confusing character (for instance, the Microsoft Word icon has vanished and was replaced by a better-thought menu). Other new and helpful icons and functions were added too, showing that CoPe_it! evolves in the right way and has better guidance functions for the users, resulting in a better management of workspaces and a better collaboration.

Structure a workspace

Toward a quicker and easier to understand display of workspaces

A first annoyance was the slow loading of workspaces and the interface's slow reaction time. For instance, if a user would click on an object, or wanted to move a workspace or an object, there was no immediate feedback to his/her action. This slowness had bad consequences on the perception of the service by the user. Newer versions fixed most of these issues, enhancing the attractiveness of the service.

A second issue is that it was not easy to give a structure to a workspace: the user had difficulties in understanding the chronological order of events and the links between objects. The very large workspaces (far more large than a standard screen) coerced users in scrolling both horizontally and vertically. It was not possible to select multiple objects and move them together. The readability and usability was not optimal because of layers, which tended to superimpose one over the other.

From another point of view, the chosen schematic layout is a model of thought that some user feel very uncomfortable with. Even if it was possible to switch to other views, their features were not 1:1 matches. For instance, it was impossible to interact in the “formal view”. To work around this issue, developers have implemented a new function showing an ensemble view of the workspace: using the “mini map”, the user now has a global view of interactions but can also access various areas more easily and directly. A lot of functionalities have been added in order to make the feeling of rigidity disappear. The new “time-ordered” view helps the user by showing him the history of the various created objects and partly resolves the issues of understanding a workspace in progress. All this work is directly geared towards helping the user to work the way s/he thinks, not the way the service wanted him to think in earlier versions. The direct result is that members of a community can now better collaborate.

Handle objects supporting interactions

Toward better organized and exploited objects

Once objects are created, the user had numerous issues related to the way they were displayed, a fact that made them hard to exploit and even understand. Available widgets were not acting the same as what is commonly accepted in mainstream operating systems and applications, or, when they were similar, they did not work as their common counterparts. This used to lead users to make bad manipulations. For instance, widgets looking like they would maximize or minimize a window did in fact open or close it. Copy and paste functions were dysfunctional as was the printing subsystem.

To solve these issues, the developers changed their widgets to more commonly accepted ones: the confusion between opening/closing a window and maximizing/minimizing it is gone. They also added new widgets, such as very useful scroll bars in objects. A lot of issues were resolved, but there is still work to do about certain functions: copy and paste functions, as well as the printing subsystem and superimposed objects still cause issues.

3.4.3 Expected future

Looking at the various enhancements, it is easy to say that developers are actively working to make the CoPe_it! service evolve. Numerous issues have been at least partly resolved. Regarding the future of the service, the team plans to go further and improve a number of existing aspects as well as add new functionality. In particular, these improvements and additions include:

- user interface improvements related to opening workspaces and visualisation of the online workspace users;
- the ability to multiple-select items on workspaces;
- the ability to aggregate and cluster workspaces items and handling them – under the circumstances of some operations such as moving – as one entity;
- the ability to perform synchronous collaboration within a workspace. Synchronous collaboration options have been extended with new operations such as signalling;
- the ability to chat, permitting participants to exchange messages during their collaboration within the context of a workspace;
- the ability to replay the collaboration in a workspace, so that participants can see how the workspace evolved over time;
- additional awareness mechanisms such as a list of currently connected users to a workspace;
- additional options for filtering the items of a workspace when it gets transformed into a formal object (users can select which item should be transformed);
- synchronisation of formal and informal projections. Adding an item in the formal projection adds it also to the informal projection;
- the ability to specify filters during the transformation of a workspace into its formal view.

3.5. eLogbook

3.5.1 Utility and goals

Even if the name "eLogbook" itself literally suggests an electronic diary (log book), the "shared desktop" metaphor will probably give the user a better idea of what it makes possible. Developed by the Swiss Federal Institute of Technology in Lausanne (EPFL), eLogbook allows the creation of workspaces where users can register CoPs (**actors**), work on tasks (activities, recently referred to in the views, by "**spaces**"), collaborate by the means of **assets** sharing and be kept aware of the evolution of the project(s) where they are directly implied, *via an awareness feature.*

In the goal of "*Collaborate, share and be aware*" (the service's slogan), eLogbook offers different functionalities, that can be grouped around four usage goals.

Each actor can:

- access the service;
- share documents and join workspaces;
- navigate, search and make use of information inside the service;
- be kept aware of the evolution of workspaces.

3.5.2 Analysis of goals through PALETTE's timeframe

Access the service

Toward a more intuitive interface

As a first task, the user has to register and log into the service. There were several issues with these functions: some were not visible enough on the screen and ignored usual conventions. Some functionalities did not work. The confirmation e-mail had several issues. The interface itself was sometimes hard to understand and forced the user to scroll far too often (articulatory distance). Most of these issues were addressed thanks to a better screen layout, where various elements are placed at more intuitive places. Useful and easy to understand warning messages were added and the e-mail issues fixed.

When using the service for the first time, the user sees a screen showing a mixture of coloured areas, which is disappointing at a first look. In older versions of the service, these colour-coded areas had no title, with few clues for the user about the purpose of these areas. The information density on this screen was huge, with a lot of hard to understand widgets.

In more recent versions, developers fixed some of these issues. For instance, the colour-coded areas now have a title. A new, more attractive logo was designed. Features that were not useful on an initial screen were removed and better terms were chosen for ambiguously named functions.

Therefore, the most observable evolution was the clean up of the interface at identification time. However, even if these changes are a step in the good direction, there is still a lot of work to do in order to guide the user in the carrying out of his tasks, in order to promote the use of a more intuitive method. These enhancements will give the user a better autonomy and help him to become quickly familiar with the service.

With all those issues occurring as soon as the user enters the service, the help system is needed quite often. In older versions, the online help was only available through a “getting started” document which was hard to find, as it was lost in the assets list (and so required the user to have understood the logic behind the colour-coding before accessing it). An effort was made for this document to be easier to find as an alias was placed outside the assets list. However, the document is still too static and incomplete to be really helpful. Moreover, at the time of writing, the document is referring to an obsolete version of the service. A new, up-to-date help system would be very useful for the new user.

Share documents and take part in workspaces

Toward a more concise and efficient role management

As a member of a community, the actor wants to create a space for his CoP inside the service. However, the lexical choices do not help the user to understand the very fact that a new community is in fact a new activity, in eLogbook terms. The developers changed the name “activity” by “space”, which better insinuates the inner working of a CoP (exchange of practice, common interests, etc.). However and in spite of these efforts, the notion of creating a “space” for hosting a CoP's work is still fuzzy for the layman.

A major issue is that the granularity of roles in eLogbook is very hard to understand and is still a major hassle. Roles, privileges, public and private activities, member status, once mixed, give an enormous configurability of roles, but is almost impossible to handle for the layman. The process of creating roles is confusing, not intuitive and lacks homogeneity.

Granted, creating new roles is not mandatory when using eLogbook and basic predefined roles are available, members and administrators. There is a major guidance issue though: when using eLogbook, the fact that creating new roles is optional is not clear at all. Moreover, just two roles may be a bit too restrictive. The service would gain a lot if clear, predefined roles were available. A solution to these issues that the EPFL is working on would be to have a basic view showing only role names and there associated right type, and a more advanced one – which shows granular action rights associated with each role: can manage, can link, can tag, etc.

eLogbook lacks a good error and notification management system. For instance, if a member rejects a role in an activity, the administrator is not notified of the rejection. The only way for him to find out is to look at a colour-code next to the member's icon. Another example is that warning messages are often too general, not contextualized enough. From screen to screen, the user cannot easily find his marks, because of a lack of homogeneity and coherency between pages. The fact that a same function is represented in different ways from screen to screen often leads the user to search where s/he can find a given function.

Depending of the type of use of the service by the CoP, eLogbook's flexibility and complexity can have the inverse consequence of empowering the user, by surcharging him with information and options. A better user's guidance in the service's settings would help, as would a simplification of access right granularity. This necessary evolution was not really worked out as of today, but the developers are aware of the issue and one can hope that in future versions of the services, the issue will be addressed.

Navigate, search and make use of information inside the service

Toward more consistency

When accessing the service, the first screen mixes all activities, resources and actors. The user must perform an action before viewing the contextualization (links between activities, resources and actors). Sometimes though, the user just needs to find a given object directly. This is possible thanks to the "search by keywords" module. At the present time, this module is working: it was introduced between the two versions analysed, a proof that the service is constantly evolving.

The tag manager has been reworked several times, but still is not ergonomically satisfying. Nevertheless, a few interesting tagging functions have been added in recent versions, such as the distinction between private and public tags, the ability to see the number of times a tag has been used by another actor, etc. In the near future, using tags will be far easier, consistent and stable.

Be aware

Toward a more controlled, more concise awareness system

According to the analysis, the awareness does not introduce numerous situations. However, this principle generates some rather heavy fallout for the user in term of utilization and working. In fact, the member of the community is really aware of the workspaces evolution. These warning messages diminish the user's risk of forgetting or not being regular in its interventions. Within eLogbook, the mails are too numerous and drown the mailboxes when a lot of modifications are accomplished by the actors. It causes the inverse result because the user's reflex becomes to delete the mails in his box without having them read. In that case, the awareness has not the expected effect.

In a tentative to remedy to this issue, the developers have implemented profile preferences where the user can choose to accept or not member's mail invitations. Even if it does not solve the plethoric character of awareness mails, this solution allows the user to keep control on the flow of mails.

3.5.3 Expected future

Generally speaking, we can say that using eLogbook takes a bit of adaptation time, notably because its configuration and its numerous features. The problem is that CoP members rarely are in a position where they can invest much time for learning the service. Expected improvements are an increased intuitiveness of the service, an increased simplicity for accessing it, a better-studied granularity of roles, a better stability and consistency in the search and exploitation of workspaces and finally, a better, more controlled awareness system. These changes will greatly enhance the service learning curve.

Several problematic goals that were pointed in the first analysis were already – or are in the process of being – fixed, such as creating an account and logging in, understanding the general interface, managing tags and search by keywords. Others have not yet been tackled, such as the help system, the community building system, the mail flood of the awareness system and, particularly, a more intuitive presentation of the invitation mechanisms and the granularity of roles. These issues are harder to fix and are more time-consuming. Moreover, some of them have been noticed only recently, in the second analysis, or are related to recently developed features.

eLogbook is a continuously evolving product and their developers are listening to all ergonomic feedback. Therefore, one can expect the service to evolve a lot. A lot of innovating features are constantly added in the service, such as OpenID (as recommended for all of PALETTE's service, *cf.* WP5) or an RSS-based awareness system. In the near future, the developers plan to implement new features within eLogbook, such as chat mechanism, and the possibility for users to define spaces and asset types. For instance, a user will be able to specify that a space is reserved for a given community and that another one is reserved for a project with the community. For instance, one space could consist of a “work package”. This gives more meaning to the collaboration medium used by CoP members, and helps them relate to their community and understand the purpose behind the creation of every space or subspace. Other features and future improvements include augmenting the interface with more guidance cues (*e.g.* emphasizing new items), improving the notification mechanism (adding more notifications for significant events and give the user more control over these notifications) and a multilingual interface.

3.6 BayFac

3.6.1 Utility and goals

In the PALETTE project, the BayFac service is dedicated to search and classify documents, share and exchange resources and provides a documents repository. Its name comes from the fact that it is based both on a Bayesian approach of classification and on a faceted classification.

Initially, the BayFac service was dedicated to simplify the exploitation of an ontology. An ontology could be very complex according to the granularity level of the domain description of a CoP (or any kind of organization/community). A given user group would only exploit a part of this ontology daily, and they could not be familiar with ontology concepts. Moreover, usual tools for creating and exploiting ontologies are not very user-friendly. BayFac was born from those observations: it allows isolating a part of the ontology, the one which is used daily by the user group. It also allows populating it and search through it.

Back to the PALETTE project, the idea is that the system will be in charge of the major part of the classification work by the way of facets, just leaving the user a role of controller *a posteriori*. Parallel to the classification system, BayFac offers a search interface through which the user determines the values of the facets according to which s/he wants the search to take place.

In BayFac, CoP users are going through various goals, as register/login, find/search for a document, classify documents, download a document and administrate the system. These goals also determine various kinds of users. Here is a brief description of the roles identified in BayFac:

- “end users” are the persons who enter BayFac without access rights, simply to consult the database and retrieve public documents,
- “end users with rights” log themselves before querying the database in order to also be allowed to retrieve private documents,
- “classification managers” are the persons who are responsible for the addition and the classification of the documents,
- “facet managers” are also allowed to add and classify documents and also to manage facets,
- “ontology managers” have the same rights as the facet managers, but are also in charge of the ontology management,
- the “administrator” has every right described above and also some tasks of maintenance as checking the integrity of the database, clearing the cache, but also the management of users. At present, to ease the process, the administrator of the Form@HETICE CoP (for example) is a member of the BayFac development team, but eventually, this role will be assigned to a CoP member.

3.6.2 Analysis of goals through PALETTE's time frame

A new release of BayFac has been launched on the Internet in the end of April. In this report, issues encountered in the previous version – the one that has been submitted to the usability analysis – will be firstly taken into account. However, some new features and improvements of the current version will also be talked about. Let us note that improvements that occur in the current version are not already the fact of our usability analysis report, but come from previous discussions our team has kept up with the CRP-Tudor team and by end users' feedback. However, the usability report will be used for the future improvements of BayFac, expected in the next few months. The process is on its way.

In general, the interface of the BayFac service is relatively complete. It is fluid and easy to take in hand. The features are clearly visible and usable. There are no useless buttons or functions. From an ergonomic point of view, very few major issues have been reported. Most issues are not of major importance, but their resolution could enhance the acceptance of BayFac.

Access the service

Toward a more autonomous and collaborative service

For a newcomer, it is really simple to access BayFac, especially for a user who wants to find a document: s/he only needs to enter keyword(s) and facet values in the left menu and s/he directly accesses a list of results. Of course, without logging in, this user will not have rights to access private documents. However, s/he will have an idea of the work done in the CoP and the capabilities of BayFac.

Other users need to log themselves. It was mentioned in the usability analysis that the ID and password boxes were placed at the same level (upper menu) as the BayFac functions, which is not usual. In the current version, developers already move them in the upper right, which is a good place. However, there is no "lost password" link, nor a "register" link. The attribution of ID and password is done by a contact with the administrator, *i.e.* nowadays a member of the development team.

A comment has to be made at this point: BayFac in the PALETTE project is dedicated to CoPs that are already well organised. It requires inviting members that will fulfil the different roles in BayFac. It is worthy to wonder how the developers are imagining the future of this procedure when BayFac takes its flight.

In comparison with the version analysed in the usability analysis, roles have been redefined in the current version of BayFac. For example, the task of adding a value to a facet is not anymore in the hands of the classification manager. It is currently the role of facet or ontology managers. The aim of the developers behind this redefinition is to maintain a high level of coherence in the ontology, which is, of course, a very good idea. The main difficulty will be to find the CoP members able to fulfil the different tasks and to make them collaborate as closely as possible. It is a key point to the use of BayFac.

Find/search a document

Toward a more fluid document searching and retrieving

The goal of finding/searching for a document is the one where most of the issues occur. It is not because of the accessibility of this goal (the procedure of entering keywords and facet values in a well visible and intuitive menu on the left was discussed before). In fact, the execution of this goal is simply not smooth enough to please a user: brevity and adaptability issues were described, meaning some kind of heaviness of the manipulations. For example, the results are displayed ten by ten. If the list of documents is very long, the user has to go from one page to another to find the document of interest. It is tedious as the documents in the list are only displayed in an alphabetical order. It is not necessarily the more useful order, as users do not necessarily know the name of the document of interest. Developers have already been asked to sort documents chronologically, but it is not implemented yet, because it needs a heavier work of development.

Problems in brevity and adaptability result in an ineffective workload by multiplying the number of needed actions before achieving a task or by not allowing another way to achieve them. As they are very frequent when accomplishing the task supposed, a particular effort should be done to improve their execution.

Some problems also concern user control: for example, as previously mentioned, if a user clicks on the link to reach the second screen of the list of the retrieved documents, s/he will be redirected to the complete list of documents, not only the ones selected by the search. It is still the case in the current interface. This issue is only a bug and will be fixed really soon, according to a member of the development team. It shows the value of the usability analysis:

point this kind of little issues, which could not be seen by the developers anymore, because of their focus on the major functionalities of the service.

Classify documents

Toward a more intuitive classification manager

When classifying documents, a majority of issues encountered are linked to brevity. Here is an example of a step of the procedure of classification that is time-consuming: when a user edits a document (listed in the fifth page, for example) and clicks on “validate”, s/he is automatically redirected to the first page. Both the workload and user control criteria are encountered in this situation.

In the version analysed, developers have paid attention to the visual recognition of classified (or not) documents. However, there was not any difference between the visual status of a partly or entirely classified document. In the current version, developers have added for the classification managers a percentage of classification for each document. Also, they had anticipated the need of taking into account the automatic fillings of facets by the Bayesian engine. In fact, documents automatically classified (partly or totally) are distinguished from documents classified by the human hand: a light bulb icon is placed next to documents automatically classified.

It is important to keep the distinction between the automatic and the human classification in order to test the capabilities of the algorithm.

Administration

Toward simplified administrative procedures

Concerning the goal of administration in general (including facet and ontology management), it is to be noted that a good knowledge of computer grounds is an essential prerequisite. Also, the administrator and his/her colleague(s) responsible of the ontology management will need to get familiar with programming environment, for example to delete an improper value. Administration tasks are presumably not aimed at being opened by every CoP member. In fact, in the version of BayFac analysed it would have been impossible for a layman not really accustomed with computer grounds to cope with these tasks by himself.

An important point was the absence of any help function in the version analysed, even if it is true that quite an intuitive interface often makes for it. However, for example in administration tasks, contextual help would be helpful.

An online help system resolves the issue in the current version.

As a conclusion, BayFac will be able to help users classifying and finding documents of interest the day when a fully operational Bayesian algorithm and a more flexible system of classification (e.g. deleting of facet values) will be available.

3.6.3 Expected future of BayFac

The discussion has already been opened with the developers some months ago and both developers and expert users' teams are eager to continue the collaboration in order to improve the usability of the service. Expected developments in the near future, as discussed with a member of the development team, are listed hereafter.

On the short term, here are some improvements planned by the developers:

- improvements of the interface according to the usability analysis report. A meeting is planned in the next weeks in order to continue the collaborative work between CRP-Tudor and ULg teams,
- insertion of the description of the different user roles in the help,
- improvements of the facet management. Developers are planning to update the interface in order to better take users by hand in the facet management task. This will be possible through the BayFac interface soon.

In the long term, the priorities of the developers are to develop a more efficient Bayesian engine and to make the tool quicker.

4. Transverse analysis of the progress considering B&S criteria

4.1 Global view and reference table

In this section, PALETTE tools and services adherence to the main Bastien & Scapin ergonomic criteria will be examined. In order to achieve this goal, data coming from the synthesis tables have been merged into a single table, in order to give a precise idea about which criteria had to be focussed in order to make PALETTE tools and services more acceptable to CoPs members. Their global evolution will also be considered. In addition to these criteria, articulatory distance is also taken into consideration because it is a critical element in the acceptance process.

A bit less than seven hundred situations have been described in the set of reports. Once again, this figure gives an idea of the size of the task but hardly anything more. Analysis was limited to (what was considered like) the main goals. Notice the difference between a situation and a problem. Several situations may correspond to the same problem because they appear when trying to achieve different goals. As a consequence, there are generally a bit less problems than situations. In the case of Amaya, it is quite different because a problem appears often related to several goals. So, 210 situations correspond to only 101 problems. It is important to keep this information in mind to avoid a bad rendering of the figures.

Our main comments were dragged out from table 4. The figures in this table must be commented and cannot always be rendered in the same way, notably the total number of situations by tool or service. The more important remark concerns the total number of situations for Amaya. It is explained in section 2 that, with respect to the available functions, Amaya was successfully completed. So, ergonomic inspection could finely focus on usability issues. That is why the analysis has been very detailed in the report we sent to the developers. On the other hand, the task was well defined and goals and sub-goals were very numerous. The analysis led to the design of a new and simpler interface. That is also why it would be an error to compare the number of situations with the ones of the other services.

Depending on the tool or service, the most interesting figures to render are:

- the percentage, for each tool or service, of situations in respect to the criteria
- the number of situations by tool or service
- the number of situations in respect to the criteria, taking all tools and services together.

Sometimes, figures have to be considered simultaneously. For instance, a high percentage is maybe not significant if the total number of situations is weak.

Percentage linked to a criterion

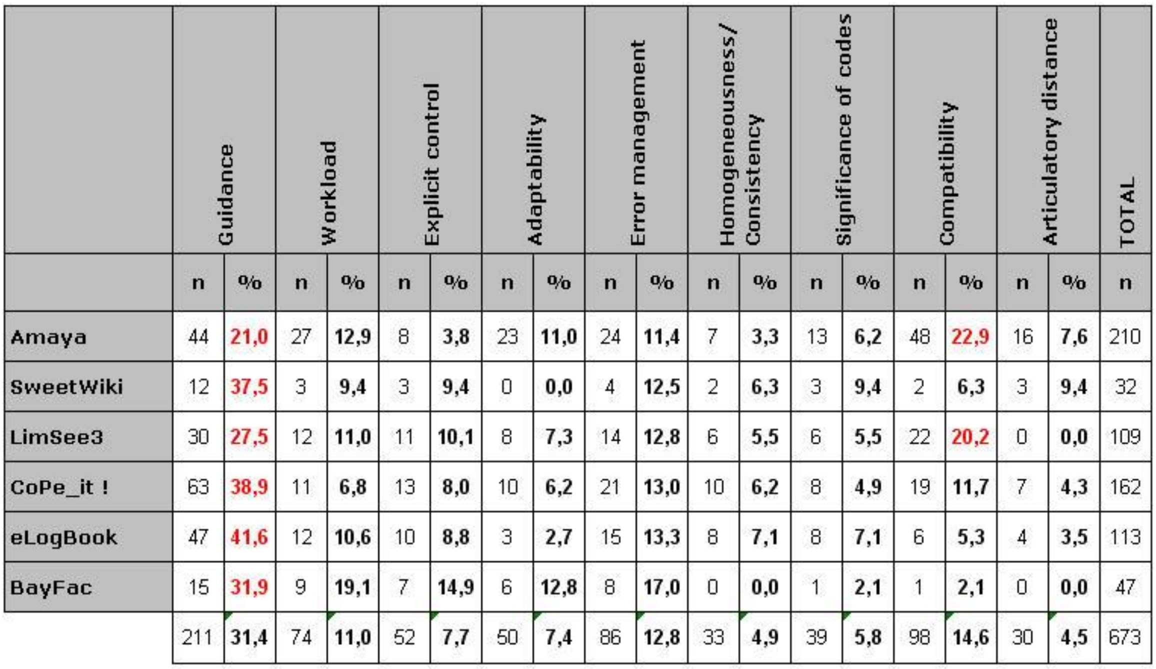
This is probably the easiest figure to render. It shows, in a certain sense, where is localised the weakness of the product, from a user interface point of view.

Number of situations by tool or service

As explained before, these figures must be carefully rendered. When it is high, it might mean that numerous details have to be fixed or that the development is ongoing and the ergonomic concerns no prior.

Total number of situations by criterion

It is an indicator of the effort developers should produce when developing new and/or crossed services. So it can be a good source to provide guidelines for the development of federative tools.



	Guidance		Workload		Explicit control		Adaptability		Error management		Homogeneousness/ Consistency		Significance of codes		Compatibility		Articulatory distance		TOTAL
	n	%	n	%	n	%	n	%	n	%	n	%	n	%	n	%	n	%	n
Amaya	44	21,0	27	12,9	8	3,8	23	11,0	24	11,4	7	3,3	13	6,2	48	22,9	16	7,6	210
SweetWiki	12	37,5	3	9,4	3	9,4	0	0,0	4	12,5	2	6,3	3	9,4	2	6,3	3	9,4	32
LimSee3	30	27,5	12	11,0	11	10,1	8	7,3	14	12,8	6	5,5	6	5,5	22	20,2	0	0,0	109
CoPe_it !	63	38,9	11	6,8	13	8,0	10	6,2	21	13,0	10	6,2	8	4,9	19	11,7	7	4,3	162
eLogBook	47	41,6	12	10,6	10	8,8	3	2,7	15	13,3	8	7,1	8	7,1	6	5,3	4	3,5	113
BayFac	15	31,9	9	19,1	7	14,9	6	12,8	8	17,0	0	0,0	1	2,1	1	2,1	0	0,0	47
	211	31,4	74	11,0	52	7,7	50	7,4	86	12,8	33	4,9	39	5,8	98	14,6	30	4,5	673

Table 4

This table exists because it was possible to take a snapshot of each service or tool at a given moment. On the other hand, to draw up a second table seemed not relevant for two reasons. Firstly, a second snapshot would have quickly become obsolete. Secondly, a summative evaluation proves not to be useful in any case, notably when the development of the service or tool is ongoing.

The table and the resulting changes are commented in the next sections.

4.2 Comments on each criteria

Most of the criteria are sensitive but differently. For instance, guidance is crucial for inexperienced users. Many CoP members belong to that category. On the long run, the importance of the problem decreases (cf. the well-known example of the “Start” button that must be clicked to shut down a Microsoft Windows system). On the contrary, workload is more acceptable at the beginning when the user is starting to use the service but not afterwards. It may become disheartening. One can get used to curious error messages or unusual symbols on buttons, but it takes time, etc. That is the reason why each of them is shortly analysed hereafter.

Figures can help us to try some comparisons between the tools and services. Let us comment how the different criteria have been taken into account from the first analysis until now.

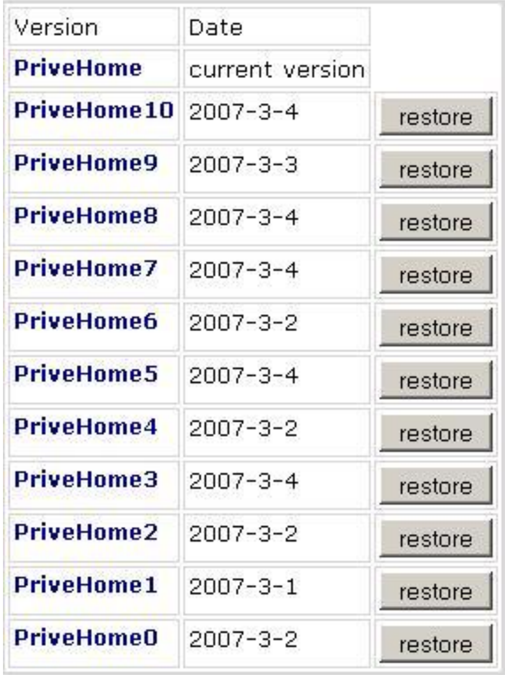
4.2.1 Guidance

Guidance can be defined as all what can help the user to know what the system (through the interface) is expecting him (her) to do. Sub-criteria are: prompting, grouping or distinction by format or by location, immediate feedback and legibility.

Guidance is a sensitive criterion, mainly when the user is a layman. His/her motivation may rapidly drop when s/he does not find how to achieve his/her goal. A quick look at the table figures indicates that all the PALETTE tools and services have to improve their interface with respect to this criterion which collects the higher percentages (211/673). Nevertheless, and following our hypothesis on the user's motivation, the number of situations shows that SweetWiki (only 12) and BayFac (only 15) could be more rapidly adopted by CoP members than the others.

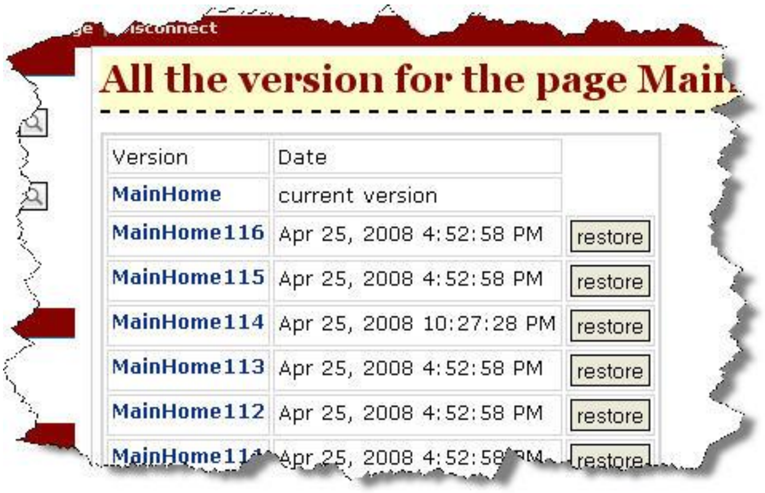
eLogbook, CoPe_it! and SweetWiki have the highest percentages. For SweetWiki, it represents only 12 situations. So, the percentage is not very relevant. Notice that the interface of Kupu (the editor associated) has been analysed, but figures concerning Kupu do not appear in the table because Kupu is not developed by PALETTE partners. For eLogbook and CoPe_it! it is understandable because, as explained in section 2, their development directly started from needs of the CoPs. So, what counts firstly is to define the right or more appropriate functions.

Here is an example from SweetWiki. Links to page versions can be displayed. When trying to restore an older version (managing documents), information about time and hour are of course very relevant. On the first illustration (figure 3) it is difficult to understand which format is used. The last figure does certainly not match to the day. So, very often, users hesitate to restore a version or another because it is confusing.



Version	Date	
PriveHome	current version	
PriveHome10	2007-3-4	restore
PriveHome9	2007-3-3	restore
PriveHome8	2007-3-4	restore
PriveHome7	2007-3-4	restore
PriveHome6	2007-3-2	restore
PriveHome5	2007-3-4	restore
PriveHome4	2007-3-2	restore
PriveHome3	2007-3-4	restore
PriveHome2	2007-3-2	restore
PriveHome1	2007-3-1	restore
PriveHome0	2007-3-2	restore

Figure 3



All the version for the page Mai

Version	Date	
MainHome	current version	
MainHome116	Apr 25, 2008 4:52:58 PM	restore
MainHome115	Apr 25, 2008 4:52:58 PM	restore
MainHome114	Apr 25, 2008 10:27:28 PM	restore
MainHome113	Apr 25, 2008 4:52:58 PM	restore
MainHome112	Apr 25, 2008 4:52:58 PM	restore
MainHome111	Apr 25, 2008 4:52:58 PM	restore

Figure 4

The second illustration (figure 4) shows a more significant format. Unfortunately, the date is always the same. But it should not be difficult to correct.

Generally, suggestions about guidance are easy to argue because users spontaneously express their distress when they do not know what to do. That means that a lot of suggestions have been taken into consideration by the developers during the adjustments period.

4.2.2 Workload

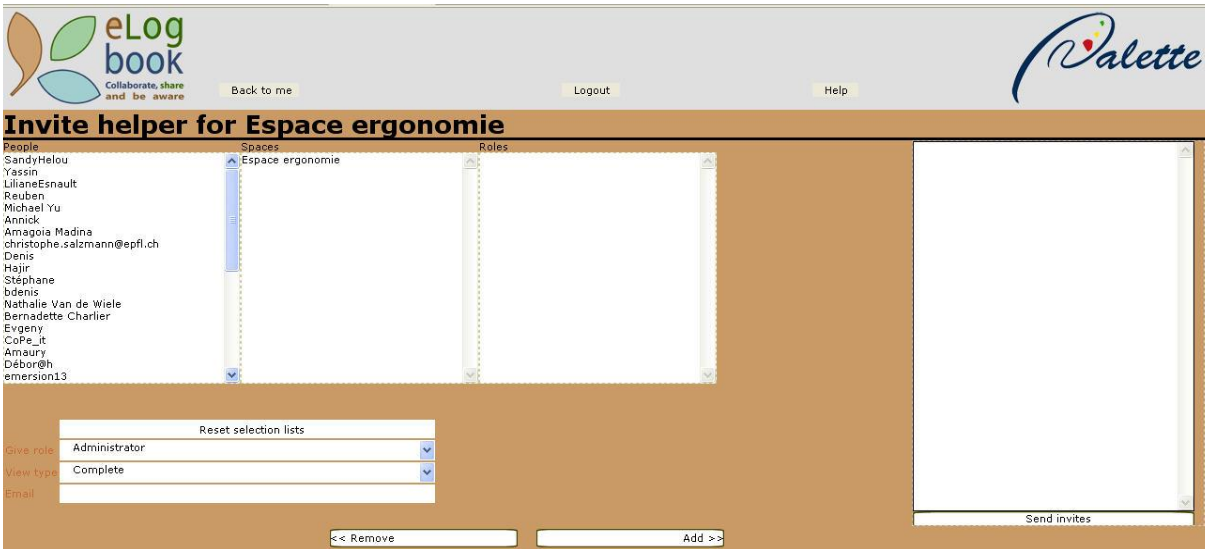
Workload has to be as weak as possible because, on the short-term, it is a reason for demotivation. Bastien and Scapin mention two sub-criteria: brevity divided into concision and minimal actions and information density. In a certain sense, minimal actions may be related to articulatory distance, but the former does not necessary concern a long sequence of actions. Information density does not help the layman to comprehend a new service, on the contrary. A very important fact is that, for a CoP user, every interface must be simple and intuitive.

A weaker percentage of situations concerns workload. With Amaya, for instance, 27 situations have been identified, what may be amazing. An explanation can be found through the fact that the audience for previous Amaya versions was composed with experienced users. For such an audience, experience compensates for workload. That is also why, on the long run, such issues are less crucial.

The PALETTE project has allowed to fit a CoP members audience by decreasing the number of situations, namely through the design of a new interface and the development of a “lite” version.

A comment about BayFac: it is interesting to notice that all the nine situations are linked to the “minimum actions” criterion. Users had often to start again a manipulation because the system did not keep in mind the parameters. Most of these situations are now obsolete.

Here is an example (figure 5) showing how workload can be reduced. In the eLogbook “Invite helper” (share documents and take part in workspaces), the names of possible members are already displayed. Notice that in the eLogbook interface, a special effort could be done to improve guidance (see previous paragraph). For instance, grouping and distinction by the format is very poor. That does not help the user to quickly become familiar with the service.



Invite helper for Espace ergonomie

People
SandyHelou
Yassin
LilianeEsnault
Reuben
Michael Yu
Annick
Amagoia Madina
christophe.salzmann@epfl.ch
Denis
Hajir
Stéphane
bdenis
Nathalie Van de Wiele
Bernadette Charlier
Evgeny
CoPe_it
Amaury
Débor@h
emersion13

Spaces
Espace ergonomie

Roles

Give role: Administrator
View type: Complete
Email:

Figure 5

4.2.3 Explicit control

Does the user control the system or does the system take some “initiatives”? That is, shortly said, the question associated to this criterion. If such situations are frequent, the user is tempted to give up using the software.

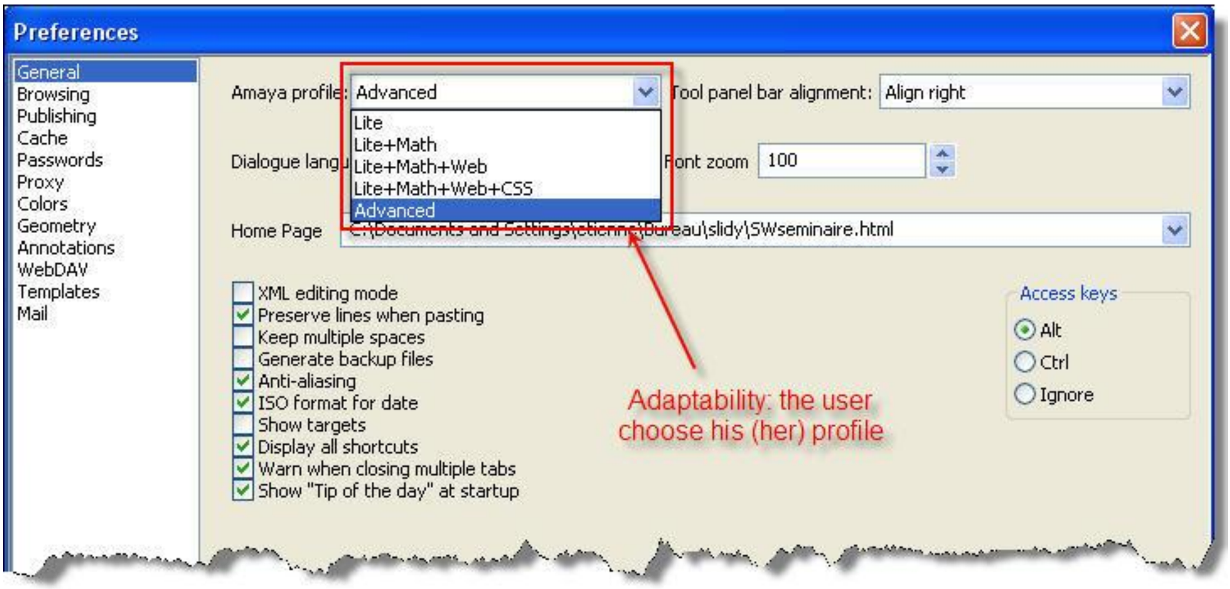
Fortunately, the global percentage is weak. BayFac collect seven situations. It is quite normal while the service is developed from scratch. Some functions must still be refined.

Notice that BayFac accumulates a large majority of situations in the first criteria. The last ones (compatibility, consistency, significance of codes...) are well assessed, what can be easily explained by the same reason.

4.2.4 Adaptability

Adaptability is the quality of any software designed for a very large audience. Does the user get the opportunity to shape the interface as s/he wants? Does the system provide him (her) with solutions to shorten some well known handlings?

Such questions begin to have an interest when the software is really adopted and frequently used. It is not still the case with the PALETTE tools and service. That is why only few situations have been identified. Nevertheless, as already mentioned, version 9.54 of Amaya was designed for experienced users. Version 10 gives the user the opportunity to choose the level of use (figure 6) with a consequence on the execution of any of the goals: managing a document, editing the content of a document, designing the layout of a document.



Preferences

General
Browsing
Publishing
Cache
Passwords
Proxy
Colors
Geometry
Annotations
WebDAV
Templates
Mail

Amaya profile: Advanced

Lite
Lite+Math
Lite+Math+Web
Lite+Math+Web+CSS
Advanced

Tool panel bar alignment: Align right

Dialogue langu

Font zoom 100

Home Page C:\Documents and Settings\etienne\Bureau\slidy\SWseminaire.html

□ XML editing mode
☑ Preserve lines when pasting
□ Keep multiple spaces
□ Generate backup files
☑ Anti-aliasing
☑ ISO format for date
□ Show targets
☑ Display all shortcuts
☑ Warn when closing multiple tabs
☑ Show "Tip of the day" at startup

Access keys
● Alt
○ Ctrl
○ Ignore

Adaptability: the user choose his (her) profile

Figure 6

This is very good example that shows how a large class of problems may be solved.

4.2.5 Error management

Error management groups together: protection to error, quality of error messages and error correction. They are some constants linked to this criterion. Percentages are very close (about 12%). Situations are by a majority related to protection to error. Here is our interpretation. When a service is under development, developers can prevent the system to generate errors, but it is more difficult to imagine lots of particular situations resulting from users errors.

The quality of error messages is something easy to deal with. For instance, SweetWiki could be programmed in order to replace the display of the error page generated when the user does not save quickly enough the page currently edited by a message explaining clearly the source of the problem.

4.2.6 Homogeneousness/Consistency

Simply said, homogeneousness and consistency concern the interface of the software in itself. Do screens, layout, position and look of interaction objects (buttons, hyperlinks...) contribute to a certain consistency? All these questions help to know once again if the user will quickly find his/her habits, if s/he will be at ease with the interface.

The table shows that this criterion is relatively well respected (minus than 7%). It can be easily explained by the fact that developers also have their own habits. Once the design and layout choices are made, they are kept during the development and they often hesitate to adopt all kinds of other suggestions (see the comments about compatibility). That is probably why few issues of consistency have been detected.

4.2.7 Significance of codes

Significance of codes concerns the correlation between the “object” (text, icon...) displayed and what it refers to.

Have a look on this old Amaya toolbox (figure 7).

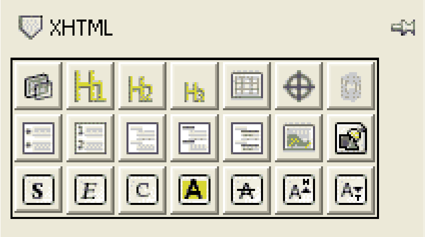


Figure 7

In a HTML editing context (audience of HTML developers) it fits. But in a simple and more classical context (editing the content of a document, designing the layout of a document) a problem rise. The next illustration (figure 8) is more significant. H1, H2, H3 become T1, T2, T3 ("T" for Title) and S and E (Strong and Emphasized) take a more graphic coloration what makes more sense. Notice that owing to the help balloons, the semantic concern is kept.

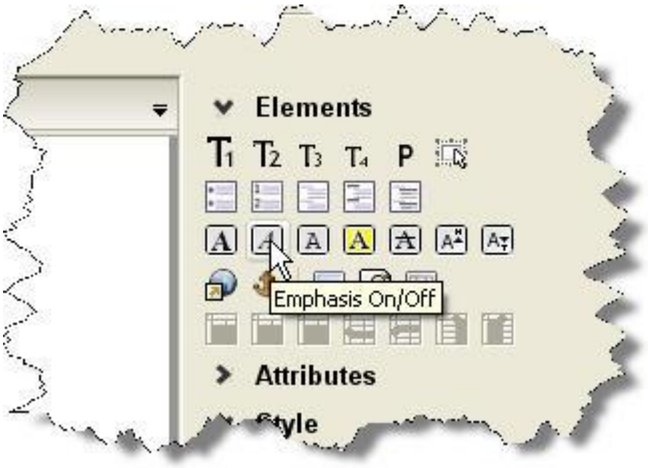


Figure 8

4.2.8 Compatibility

Here is a criterion that is worth a longer comment. The table shows that three of the six tools or services collect a large percentage of situations on it. Amaya and LimSee3, and to a lesser extent CoPe_it! are these three. A part of the explanation is maybe that Amaya and the previous version of LimSee (LimSee2) are tools firstly developed more for experienced users than for laymen. Compatibility mistakes are easy to correct, what has been done for most of them. The interface of Amaya 10 provides lots of good examples of such corrections.

Here is another illustration (figure 9). In CoPe_it! (managing workspaces), the following icons were used to resize a window.

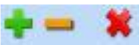


Figure 9

More compatible icons (figure 10) are now used deleting all ambiguity.

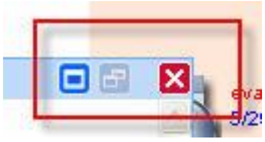


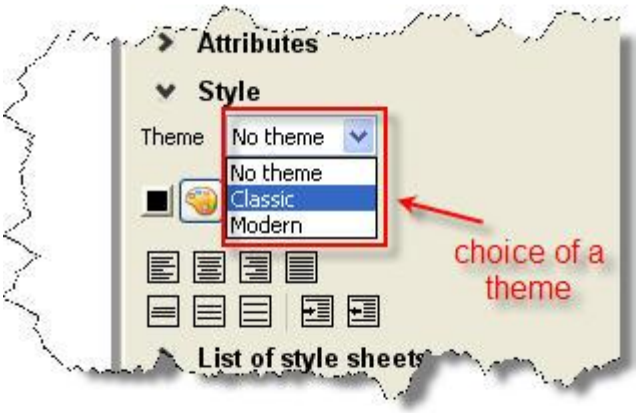
Figure 10

Many examples could be shown for each tool or service because a very large number of situations are related to compatibility.

4.2.9 Articulatory distance

Articulatory distance (Hutchins) refers to the effort the user must accomplish to reach his/her goal. It implies to take into account the length of the sequence and/or the complexity of the actions needed to achieve the goal owing to the system.

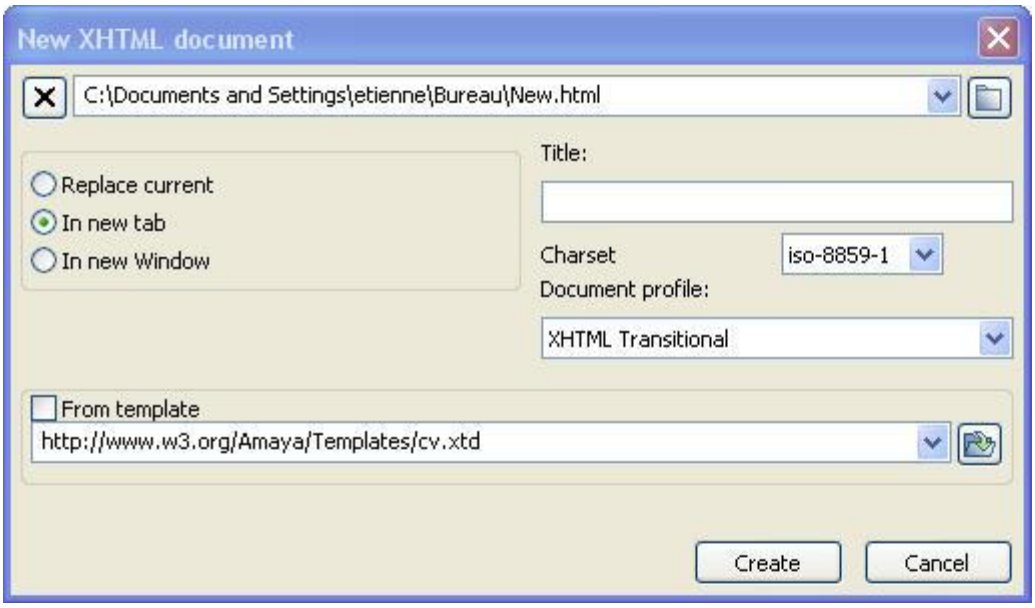
This kind of situation is rather scarce except for Amaya. Sixteen situations were detected, probably for the same reason linked to the situations of the previous criterion. The design of the “lite” interface forces us to minimise these considerations. For instance, from now on, in order to design the layout of a document, users can apply a style to an Amaya document almost in a single click (figure 11).



choice of a theme

Figure 11

Nevertheless, some other processes should still be shortened. An example is the waited simplification of the “New document” creation (editing the content of a document). Have a look on the dialog box hereafter (figure 12).



C:\Documents and Settings\etienne\Bureau\New.html

○ Replace current
● In new tab
○ In new Window

Title:

Charset iso-8859-1

Document profile: XHTML Transitional

□ From template http://www.w3.org/Amaya/Templates/cv.xtd

Figure 12

5. New services usability

Let us make two important recommendations for the improving of the current tools and services, but also for the building of the interface of global services. People who may be interested by these guidelines are thus the designers of the showroom on the website, the portal, the on-line training and CroSSE services.

The usability analysis of the six tools and services allows us to insist on the importance to pay attention to guidance and compatibility. The second criterion is probably easier to respect than the former. Observation of common features and usual interaction objects in various environments is not so difficult and may help the developers to make good choices. These choices concern all kinds of conventions: icons, drag and drop functions, shortcuts, etc.

Guidance is not easy to improve. It forces the developers to take the place of the user and to imagine how the interface can boost his/her activity. That also implies a good knowledge of his (her) habits, competencies, abilities, usual activities... All what may help him (her) at any moment should be suggested (visible) and the rest should be invisible to avoid all kind of information density. Of course, those recommendations are very global. But it should be interesting to control if these two elementary principles are respected. That should help the development of an intuitive interface making useless any help or training system (an ideal to follow). A special effort should be made to improve prompting and legibility. Experience shows that a “newly designed” software often has these two shortcomings. But it is not a general rule. The example of BayFac proves the contrary, without regards to the simplicity of its interface (very few different screens and dialog boxes).

The analysis which has been led tries to prove that building the interface of a (new) service for CoP members implies that the task related to it must be more or less defined. That means that, at least a couple of relevant and frequent scenarios must be identified. These scenarios can be the starting point to draw up a list of goals. For example, when the user will be in front of the CroSSE interface s/he will intend to reach some precise goals. A scenario should be: “I want to have a direct access to a resource about CoPs life. I know that this resource is connected to an eLogbook activity.”. To know a bit more about (to imagine) these scenarios before listing the goals seems to be a necessity. The design of the interface should take them into account. Trying to achieve these goals, expert users will be able to determine which usability issues come to the fore and so, make some relevant comments to improve the usability of the service.

6. Conclusion

As a conclusion, one may say that analysis of usability was not so evident because of the very different development levels of tools and services. In some cases, usability analysis was used for preparing a real summative evaluation, allowing developers to solve most of the usability issues, one by one, without altering the nature of their tool or service (BayFac, SweetWiki). Sometimes, although issues were easy to solve (from a development point of view), difficulties appear because of the existence of an audience predating the project (Amaya).

In other instances (CoPe_it!, eLogbook, LimSee3), one may feel that usability analysis arrived too early to conduct an efficient job. But in that case, usability analysis was relevant as formative evaluation. Even if developers could not always take into account all the situations mentioned, those they have solved have contributed to improve the tool or service in order to make it more acceptable or at least, acceptable for a larger audience. Evidently, when the development of software is ongoing, some problems raise. For instance, some situations are identified and a report is sent, but those situations may have changed in the meantime. Some other situations may sound like details with regard to the programming of some important pieces of functionality.

Nevertheless, the task was, in a certain sense, an instantiation of the participatory design. Through the team of users testing them, more CoPs needs have been met. Dialogue was not always easy but developers and expert users have learnt to meet halfway. You will find more information about the various ways of collaboration in the future D.PAR.05 ("*Participatory design methodological instruments and good practices*"). Globally, many changes occur in the right direction, whatever the tool or the service.

Once again, remember that the audience we had in mind was an audience of "not so experienced users", which led the team to be sometimes very hard to please. But our first concern was to make sure tools and services become really acceptable for CoP members.

7. Main references

Apple guidelines (2007). Retrieved from the Web on May 13th 2007

http://developer.apple.com/documentation/UserExperience/Conceptual/AppleHIGuidelines/XHIGIntro/chapter_1_section_1.html

Bastien et Scapin (2000). A framework for organising web usability guidelines. Retrieved from the Web on May 13th 2007 http://www.isys.ucl.ac.be/bchi/publications/2000/Scapin-HFWeb2000.htm

Hutchins, E. L., Hollan, J. D., and Norman, D. A. (1986). «Direct Manipulation Interfaces,» in D. A. Norman and S. W. Draper (Eds.), User Centred System Design --- New Perspectives on Human-Computer Interaction, Hillsdale, NJ: Lawrence Erlbaum and Associates, pp. 87-124.

Lompré, N. (2007). Comparaison entre norme AFNOR et norme ISO 9241. Laboratoire Société Environnement Territoire, UMR 5603 du CNRS et Université de Pau.
Retrieved from the Web on May 13th 2007
http://web.univ-pau.fr/~lompre/documents/Comparaison_Normes_AFNOR_ISO.pdf

Nielsen, J. (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY.

Nogier, JF. (2005). Ergonomie du logiciel et design web : Le manuel des interfaces utilisateurs (3e édition). Paris: Dunod.

Norman, D. A. (1986). Cognitive engineering. In Norman, D. A., & Draper, S. W. (Eds.), *User centred system design: New perspectives on human-computer interaction*, 32-65. Hillsdale, NJ: Lawrence Erlbaum Associates.

Shneiderman, B. (1992). Designing the User Interface: Strategies for Effective Human-Computer Interaction, 2nd edition. Addison-Wesley Publishing Company, Reading, MA, 1992.

WP1 (2007) Description of six scenarios and results of six validated trials (D.PAR.03). PALETTE project report, IST Project no. FP6-028038.

WP1 (2007) Grid of analysis supporting the participative design methodology (D.PAR.01). PALETTE project report, IST Project no. FP6-028038.

WP1 (2007). User Centred Description of PALETTE tools and services and first analysis of usability (D.PAR.04). PALETTE project report, IST Project no. FP6-028038.