To fit a Data Bases Querying software for non-specialist end-users

Bernard DENOEUD
Conservatoire National des Arts et Métiers
Laboratoire d'Ergonomie
41, rue Gay-Lussac
75005. Paris
denoeud@cnam.fr

ABSTRACT
The universal design topic is approached under the angle of fitting a Data Base application designed for specialist to non-specialist end-users. This prospect drives to focus attention on the 1st and occasional uses. This fitting problem questions principally the user expertise level because these users can't be expected to acquire a good mental model of the process for transient uses. We reports the problematic and the solutions implemented to advance in that direction. They concerns the conceptual model and a special type of documentation "advices" that aim at an optimal practice despite the lacks and defects of comprehension.

Keywords
HCI, Data Base consulting, user model, expertise, ease of use, documentation, advice, guides

INTRODUCTION

We want to illustrate the topic of universal design by a case study which consisted in redesigning a software for different end-users from the original people concerned with a first design. As it is an extension of the use of Data Base, the new user people must be of different profiles varying as much about domain familiarity as computer ease. We would say we pass from specialists to non-specialists in a professional framework.

This paper report my own design experience as Human Factor specialist in a small team. It presents the problem data, the ergonomic issues we have dealt and the solutions we have implemented in the actual application. The solutions are drawn as useful outlines for a practice code in HCI.

HCI attended for the issue of fitting the software to the end-users for a long time. The prospect of "User Centred Design" [1] aims at centering the design around the needs and characteristics of the user. But Universal Design questions HCI in that feature that first or occasional users will never be expert in the domain they handle by the application. So the issues we have had to deal with is to assure the conditions of a satisfying result in the case of a transient use, despite the fact that users may have a very different expertise.

0. Problem data

The case study is a software of building and retrieving information in Data Bases. The DB collect the references of computerised resources for Universitary teaching. The products are referenced by indexes defined by the person, author or other who wants to sell products to the market. An initial DB was built for chemistry years ago. The new schedule of conditions remained the same, apart that Data Bases had to be developed for all the academic disciplines, literacy, arts and sciences. The ergonomic issue was to fit the current software to end-users varying so in education as in computer and DB practice. The existing software had been evaluated as unacceptable the end-users of the project. The new end-user population consists in

1. teachers of every great teaching disciplines of University
2. librarians as interfaces with teachers and students
3. sand publishing personals carrying out data input.

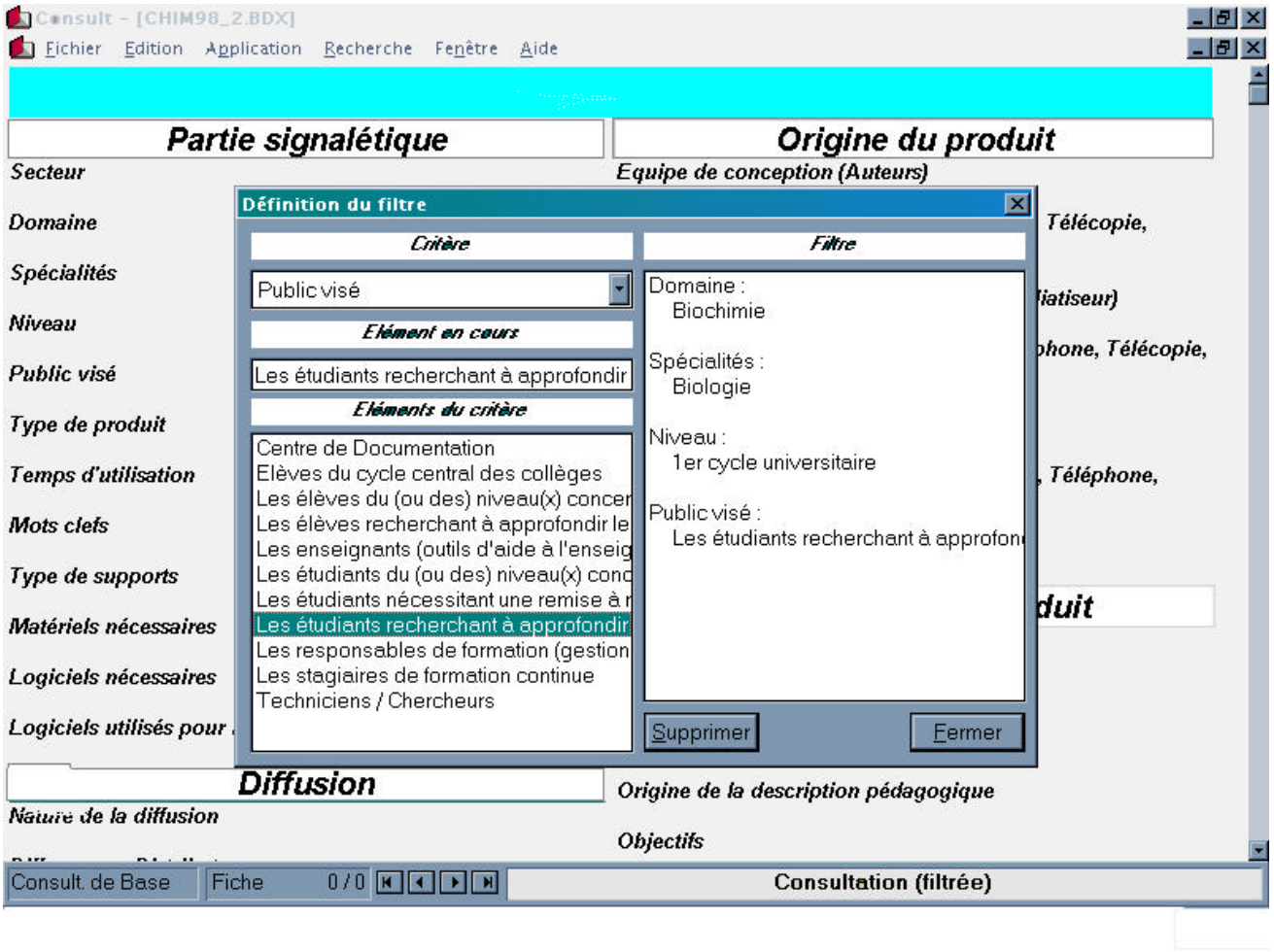

Figure 1. Interface of the Data Base consulting software original version.
This application has been redesigned to fit for non-specialist and non scientific end-users.

The major target users of redesign are the teachers who have to bring the resources into play. We will be principally concerned with them in this paper. Librarians are secondary users, as they had to handle information to inform every possible user. Publishing personals have a temporary part but they ensure the quality of collected information.

In the 1st version designed for scientific users, the querying interface was made of a single form displaying information headings and the relative indexes (figure 1) . This form was linked directly to the 1st sorted card.

The project financed by the French Education Ministry fitted into a voluntarist program of promoting computer technologies in education and developing multimedia market. The project has been worked out as there are now eight Data Bases actuated by a specialist of each discipline, and accessed by the Net.

We will present our redesign issues in 4 points:

1. Implementing a common sense conceptual model.
2. Assisting the User invisible activity
3. Helping the User in his difficulties
4. Considering the software as a tool in an extended context.

1. Implementing a common sense conceptual model

To be accessed by varying people, the software must be based on a common sense conceptual model. This conveys the figure of the whole system, of its function and partially how to use it. It is of major importance for our issue because the transient user is not usually ready to undertake a great cognitive effort for an unpredictable gain.

The concept of metaphor is pertinent as metaphor is an analogy used to convey pre-existing knowledge about the application that has not to be learned. This inexpensive process is worthy as the learning situations stand for heavy cognitive load. Designing for non-specialists requires metaphors understood by a great majority of people.

As an example, the "filter" concept is the metaphor of research process in the 1st release. We have found the metaphor badly fitted to the process working which consists in looking for wordings (precise and exact wordings) in resource descriptive files. We have preferred the metaphor of "search engine" that is now much used on the Internet that is the environment of the application.

But a rich metaphor is only a starting point, as its content is not precise, operative and equivalent among the users. So this concept must be explained for everyone might build an appropriate and effective model and take advantage of his/her commitment. Every means of teaching method are invited to convey knowledge and to prompt a convenient use and control : terminology, textual explanation, iconography.

In our example, we have developed the search engine concept in 4 displays for giving it consistency:

1. definition of search criteria
2. building of a criteria table where logic connectors AND/OR are inserted
3. chart of the number of cards sorting by every criteria
4. display of search results by card titles and summaries

before showing the searched cards.

Despite the call to metaphor and explanations, understanding errors will remain. This looks significant of the major challenge of learning for transient use, and for non-specialists.

2. Assisting User invisible activity

While consulting a indexed Documentary Data Base the most difficult and abstract task is constructing queries to retrieve information from the DB (Data Base). Query languages [2] had been developed since the beginning of Computing era, but these languages are intended to be used by professional programmers. For the public to access DB, Data Base Managing Systems has been developed that consist in key-word systems, or more recently plain-text motor engines.

Starting from a concise and incomplete expression, the subject must elaborate his query interactively with its DB application until he forms a query acceptable for the system.

As the domain of academic teaching resources is very sharp if disciplines and topics are distinguished, we have added a specific constraint to search process, that is completeness of results. The DBMS (Data Base Manager System) must sort every card on a specific query despite indexation hazards.

DB querying task must solve multiple undeterminations sources: number and choice of indexes, logical combination according to syntax, domain structuration in headings and indexes, working of search engine.

How do DBMS support querying task?

For indexed DB, headings and indexes are predefined. So they may be showed to the User to make selections. Documentation usually describes the Boolean syntax, acknowledged as difficult for non-specialists. The query is supposed to be iterative or refined after a 1st request..

For his querying task, the User is always left alone, and the constrain that the search be complete is ignored.

The querying activity is invisible, transparent for the system. The inputs and selections at the interface are not significant of reasoning or problem-solving activity to process undeterminations evoked above.

Designers may back out of the responsibility about querying errors since they don't question software running, but they are use errors. Let us give examples of use errors. In a test where subjects were presented indexes as "CD-Rom", "CDRom" and "Cédérom" under holder heading, they carry the behaviour of choosing as representative of the whole class. A good conceptual model command to select them all to reach search completeness.

It is recommended to avoid such types of errors by excluding synonyms among indexes. But for headings like "learning activities" or "covered subjects" it is impossible to avoid items to overlap, or index to differ from user domain architecture.

The observed errors don't result from unattention or unthinking, but on the contrary they are the product of sophisticated reasoning, as we find about human error and fiability topic [3].

In short universal design call us to design over the traditional user model that considers user as always competent for his own activity. This issue prompts to

assist the transient user in all his activity, and to support even invisible activity for the software.

3. Helping the User in task difficulties

Universal design doesn't mean to rub and to conceal difficulties in the objective to yield software easy to learn and to use. On the opposite the thing is helping him all along his task and considering his actual skill. The naïve user is not an expert who knows his domain well and is able to avoid difficulties and traps as we have observed.

Verbal protocols and interactions analysis have brought Users opinions about the supplied services and their quality. Generally users were very confident with consulting quality. They carry out a single querying and they confide to outcome completeness. Surprised by system confidence, we have paid a great attention to use errors made by different subjects of experiments. They result from wrong inferences about search engine running or about fiability of DB indexation.

These erroneous inferences consist in:

- believing that system is able of intelligent operations on indexes (equivalence of wordings near at semantic level, but differing in the lexicon plan, class inclusions among wordings)
- believing the derived forms of a wording (singular, plural, capital or lower-case, varying spelling) are equivalent for search engine
- believing that indexation strategies are steady and consistent for the whole DB.

These wrong inferences show that users "add" something to the conceptual models explained in documentation. They are prompt to attribute to search engine human intelligence capacities that pass beyond the technical functionalities. Hence they have not a critic attitude about the quality of indexation in DB. One would expect naïve users to be suspicious about unknown mechanism, or to multiply querying exercises varying number and kinds of indexes. It is not observed. Some reasons may be put forward: the practice is too short and the explanations provided are insufficient to prevent these slips on the background of complex knowledge.

For a public of non-specialists the software must be an instructor of the domain concerned; the thing is guiding the users to an optimal use, despite the gaps and mistakes in understanding that are common for a first or infrequent use.

In a long paper we shall develop the tentative solutions to advance in that direction. They consist in a functionality of Advising the User on the optimal actions at the convenient times. The style of advices contrasts with the practice, because their contents would display the actual capacities of the system, make explicit its limitations. This attitude of "speaking the truth" is not very common in the usual documentation, but the limited learning time makes it of major importance for non-specialists.

4. Considering the software as a tool in an extended context.

The critics in the HCI literature may contribute to the definition of universal design. This literature blames computer systems for not considering the user as an actual actor, striving after professional aims, while collaborating with other persons in a working environment.

Remaining in the original prospect of placing information about multimedia resources at disposal of Universitary Teaching, we have spread the view around documentation search and resources using in teaching. We only report here what concerns the major user-group, teachers.

As source of document references, a DB is only a start point for teacher to spot resources and to put them into play. Instrumentalisation view [4] drives to offer different storage forms for this information: files HTML, printed texts, on-line or downloaded DB. The contents and quality of information must serve different concerns about resources (teachers, documentalists, librarians, students). The service "Applications and documentation" allows to extend the facility of acceding directly to resources that needn't any payment.

As university teachers are very scattered geographically, an DB application designed on a disciplinary base offers the opportunity of creating contacts and relations among persons concerned by the same topics. The analysis of occupational conditions of Universitary Teaching shows up the potential interest to facilitate the exchanges between beginners and intensive users, or even just interested colleagues. So we have implemented functionalities of Forum, either about each resource, or on a general level, in the hope that persons might contact and collaborate for the community profit. This prospect of creating and animating interest communities leads us to develop a "News" Service to broadcast general interest news that are not easily available to spread and young people.

As an evidence, the very opened prospect is linked to the origin and funding of the project, the French National Education Ministry that as an employer looked for promoting organisation and cooperation in the Universitary circles. This particular context has given us a laboratory to experiment a more universal perspective apart from commercial concerns.

Conclusion

The project of fitting a Data Base application to non specialists has allowed us to examine how HCI and cognitive ergonomy literature could serve a "universal design" in the area of software. The concerned topics are named conceptual model, user model or competency model. In our field study, as we have had got the whole running application and an actual DB since the beginning, we have focused attention on ergonomic criteria fitted to "non specialists", i.e. a population varying in Data Base familiarity and Computer ease.

While the HCI literature is concerned with the fitting for the user, universal design questions it about user expertise model. Designing for non specialists and for transient users means that users will never be expert in a sophisticated cognitive task, and they need assistance as much about the interaction procedures as about the expert rules specific to that field.

Finally from our case study we can draw some useful rules to design in a universal prospect.

1. The expertise model must be build on a very basic level, as much about software interaction as on the domain controlled. Conceptual model and user model are to be based on this minimal competence level.
2. The universal design drives to consider the use as a part of the design. Designers can confine to create an environment where only an expert user can achieve an optimal result.
3. The system must be an instructor about the domain of the task, as much as about the application.
   The available solutions remain to be found out. They consists in intelligent assistance, guiding. We have worked on the documentation that must be and dead-ends. In short, the think is that the user be guided to the good practice, even if he hasn't acquired the convenient mental models about the domain and computer system.
4. Consider the application as a tool among others (of different types and supports), and facilitate its instrumentalisation by the user. As the ordinary things in common life are learned without effort, a cognitive tool well introduced in the user needs may be learned more easily. The major point here is that functionalities form a coherent set around the user tasks and they constitute an enrichment.

REFERENCES

1. Norman, D.A., Draper, S.W.. *User-centered system design.* Erlebaum, N.J., 1986.
2. Reisner, P.. Query languages. *Handbook of Human Computer Interaction.* Helander, M., (Ed). Elsevier, North Holland, 1988, 257-276.
3. De Keyser, V., Van Daele, A. *L'ergonomie de conception*. Ed Universitaires, Paris, 1989.
4. Rabardel, P.. Les hommes et les technologies. Approches cognitives des instruments contemporains. Colin, Paris, 1995.

.