



# THE UNIVERSITY *of* EDINBURGH

## Edinburgh Research Explorer

### The Eco Browser

**Citation for published version:**

Bundy, A, Robertson, D, Muetzelfeldt, R, Plummer, D & Uschold, M 1985, 'The Eco Browser'. in Proc. of the fifth technical conference of the British Computer Society Specialist Group on Expert Systems on Expert systems 85. Cambridge University Press, pp. 143-156.

**Link:**

[Link to publication record in Edinburgh Research Explorer](#)

**Document Version:**

Preprint (usually an early version)

**Published In:**

Proc. of the fifth technical conference of the British Computer Society Specialist Group on Expert Systems on Expert systems 85

**General rights**

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



PLEASE RETURN THIS TO ALAN BUNDY

## THE ECO BROWSER

David Robertson\*  
Robert Muetzelfeldt\*\*  
Dave Plummer\*  
Mike Uschold\*  
Alan Bundy\*

\* Department of Artificial Intelligence,  
University of Edinburgh.

\*\* Department of Forestry and Natural Resources,  
University of Edinburgh.

Keywords : browser

### ABSTRACT

The ECO Browser is the prototype of a Prolog browser for a large knowledge base of loosely structured observational information. Its purpose is to provide a way for users quickly to access the information they require without necessarily having predetermined ideas about the things that are of interest to them. The browsing mechanism is based on simple principles. We adopt an approach which relies on a gradual narrowing of the area of interest, until the desired information is found. Recent small scale tests suggest that this method is natural and easy to use, although a short introductory session is often required. Several extensions to the current system are suggested, including the incorporation of intelligent guidance methods.

### 1. Introduction

ECO [Uschold *et al* 84] is an intelligent front end which helps ecologists construct models of dynamic ecological systems. Its users must have access to a body of ecological information from which they can extract the information they need to build a specific ecological model. Offering facility presents three major problems:

- \* A large variety of ecological information must be represented.
- \* Users must be allowed to search through it in a flexible, efficient manner.

- \* Users cannot be expected to know exactly what type of information they want before using the browser.

This paper describes a method of tackling these problems in an ecological context but these techniques may be useful in other areas where the knowledge is loosely structured.

## 2. Browser Requirements in an Ecological Domain

As ecology covers a broad range of interests, the type of information required will differ widely between individual users. Therefore, the system used to browse around the knowledge base must be based on a mechanism which is largely independent of the ecological context in which it is used. It must operate in a way which seems natural to inexperienced users and must give them control over the selection of information without overburdening them with complex commands. Users may not know exactly what information they need until they have seen part of the knowledge base. The browsing system should lead novice users if necessary, but should allow experienced users to find the information they need quickly. As the task is exploratory in nature, the system should permit users to backtrack to previous states in the browsing session if they have not found what they are looking for in the first instance.

### 2.1. Knowledge representation

Ecological information is often obtained in the form of observational records which may be of varying complexity and may refer to differing aspects of a similar situation. For instance, three ecologists may each make an observation about deer.

1. The weight of male deer in Scotland is, on average, 50 kilos.
2. The body weight of a male deer was 45 kilos.
3. The weight of a deer is about 100 pounds.

Each real world observation corresponds to a *record* in the knowledge base. The bits which make up each record are informally referred to as *items* (eg. deer, Scotland). All the examples above refer to deer but the differences in detail between records means that they may not all be useful to specific users. For example, someone who was interested in deer in Scotland would only want record 1 but someone who was interested in deer weight in general could be interested in

any of the three records. Loosely structured records do not lend themselves directly to efficient browsing but relationships between the items which constitute each record may be used to provide a graph of related items. Such a graph is used by the ECO browser to help users isolate the records which most closely match their requirements. (A separate program which we do not describe is used to convert the set of loosely structured records into this graph.)

## 2.2. Using the browser on large knowledge bases

The current version of the ECO browser represents both the records and relationship graph as Prolog facts. This presents problems of storage space and real time searching speed which has limited the size of our test knowledge base to about 120 records, each with about six constituent items. To overcome this, the prototype system must be re-implemented to provide the following:

- \* Fast access to the records which refer to a selected node on the graph.
- \* A large data storage capacity is essential, and may require secondary storage management.
- \* It will be necessary to tag all the records which have been selected at a particular time and to quickly find which records carry a specific tag, in order to allow previous selection states to be restored. If a large number of records are available, then a large number of tags will need to be applied for each selection. Also, when returning to a previous selection state it will be necessary to check all records for the appropriate tag.

It is possible that these requirements would be satisfied by using existing knowledge representation tools, such as Loops [Bobrow & Stefik 81], ART [Clayton 84], or NETL [Fahlman 79]. The former two are frame-based and have hash coded patterns and relationships for efficiency. The latter uses a parallel network approach designed for anticipated parallel hardware. Special purpose hardware for high speed information retrieval is already being developed [MacGregor 84] which may also support the ECO browsing strategy. This paper is concerned with the way search should be controlled and displayed to users and does not concentrate on knowledge representation. However, efficiency problems must be overcome if the system is to prove useful for practical applications. Also, the browser, in its current form, is unsuitable for search graphs which are very deep and/or have high branching rates. A discussion of this problem appears in section 3.3.

### 3. The Browsing Mechanism

The ECO browser supports two main types of activity:

1. *Search for items of interest.* This assumes no prior knowledge of what may be contained in the knowledge base. Users are faced with a menu of general items (eg. habitat, nutrients, organism). They may then search for further items which are related to these (eg. organisms subdivide into the categories of plants and animals; animals then subdivide into reptiles, fish, mammals, etc.).
2. *Select records.* At any point during the search, users may select records which pertain only to items in which they express interest.

This section describes these actions and the method for displaying information to users. A variety of additional commands and special features are described in section 4.

#### 3.1. Searching for items of interest

At the start of a session, users are positioned at the most general node in the graph – the root node. From there, they are encouraged to search for more specific items of interest by being shown the nodes connected directly below that node. This process can be repeated until one of two things happen:

- \* They find an item which interests them and select the records to which it applies (see below).
- \* They reach a leaf node of the graph, below which there are no more specific items.

Users can continue to search and select by jumping back up the graph and travelling down different pathways toward various selection points. This process continues until a user has selected as finely as s/he needs.

#### 3.2. Selecting records

One selects records relevant to a particular item by selecting that item. Selecting an item causes the records which contain that item – either explicitly, by containing the item itself, or implicitly, by containing one of its descendants – to be tagged as belonging to the set of records which are available to the user at that time. All other records are discarded as irrelevant.

This process can be illustrated using the examples of records in

section 2.1. If the item "weight" is selected then all three records would be retained, since they all refer to some type of weight. However, if the item "male" was then selected, only records 1 and 2 would be tagged as being available. If further selection of the item "body weight" occurred then only record 2 would remain because this is the only record which contains types of "weight", "male" and "body weight".

A side effect of selecting records is that items which were previously available for the user to select are made unavailable. This occurs because by reducing the number of records in this way, the diversity of information in the remaining records tends to decrease. Using the current example, if record 3 were the only record left then selection of "male" would not be possible. It is undesirable to have this as an irreversible process, so additional features have been added to allow the undoing of selections.

### 3.3. Displaying information to users

At all times, users should be aware of their position in the graph; the number of records which are available and the items which they have selected. This information must be conveyed succinctly and, in our case, within the limitations of a standard VDU display (the potential of graphics facilities will be discussed later). The display which is currently in use addresses the following requirements:

- \* Users should have an indication of their position relative to other nodes in the graph. In particular, displaying a path upwards through the graph via more general nodes provides a context for the current position, while an indication of possible paths downward through the graph is given by displaying the children of the current node.
- \* A record of past selections should be shown, to remind users of the items in which they have shown interest.
- \* The display format should be the same throughout the session so that users become familiar with the layout as soon as possible.
- \* Items which are specific types of the current item should be easily accessed without having to provide their names.
- \* Users should know how many records remain available as they progressively refine their ideas about what interests them.

Taking the example from section 2.1, the display would appear as in 3-1 if a user

had selected "weight", "male" and "body weight" in that order. It is possible that the amount of information shown on this display could overwhelm users - particularly if the graph were deep and with a high branching rate. It would be useful to limit the display in two ways.

\* If a user is positioned at a great depth in the graph, only a limited number of nodes should be shown, as context - perhaps those nearest to the current position on a path back to the most general node.

\* If the user is positioned at a node which has a large number of children, there should be some way of restricting the number of children displayed. This suggests a need for some form of user guidance, which will be discussed in section 8.3.

```

ONLY ONE RECORD LEFT  Number of records left
SELECTED : .         relevant to the selected items.
a - weight           The items selected, in order of
b - male             selection, were "weight", "male"
c - body weight      and "body weight".
-----
t - general          The path back to the most general
  -l - weight        level goes through "weight"
-----> body weight <----- Current position ---
                        1 - 45 kilos   Specific types

```

Figure 3-1: Sample of a standard display

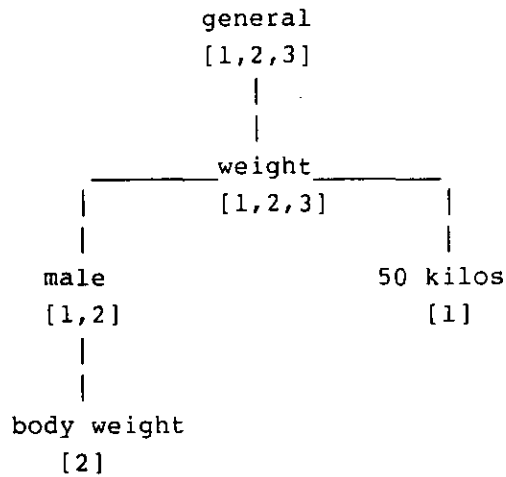
#### 4. Additional Features

Simple searching and selecting operations were found to be insufficient for easy operation of the system and a number of extra features were added. These options allow more sophisticated use of the mechanisms described above.

##### 4.1. Undoing selections

Users may change their mind about selections they have made and want to undo the effects of their actions back to a specified point. The Browser provides a command which returns the records which were available when some previous item was selected. This is possible because the system maintains a tree of selection states, with each node representing a list of records relating to a particular selected item. For example, if someone selected "weight" followed by "male" and "body weight" and then undid the selections back to "weight", finally selecting "50 kilos", the selection tree would have the structure shown in figure 4-1.

Numbers in brackets are available records at each node.



**Figure 4-1:** Example of a Selection Tree

The path back to "general" through the selection tree from the most recent selection is shown to the user every time s/he moves through the graph of items. Additionally, the whole selection tree can be displayed so that the states obtained during unrelated selection sequences can be retrieved. Another related command is "clear", which restores all the records that were available when the session started.

#### 4.2. Displaying selected records

It is necessary to show the items contained within selected records. The ECO browser allows both the printing of complete records and the printing of selected information in records. For example, if someone wanted all the available information about weight in the three example records, the following would be displayed:

```
RECORD 1
weight = 50:kilos
RECORD 2
body weight = 45:kilos
RECORD 3
weight = 100:pounds
```



#### 4.3. Help facility

At any point in the session, a help package can be accessed which describes how to operate the various components of the system. It has been found that the operation of the Browser can be explained using simple conventions – for example, that items shown at a higher level in the path upwards from the current position are more general than those at lower levels – so an inexperienced user need not become bogged down in technical detail when trying to understand how the system operates.

### 5. An Example of the Browser in Action

A simple test run of the system is shown in figure 5-1. Only the most fundamental operations of the system are demonstrated in this example because of space limitations. The text appears as seen on the screen of a VDU, with italicised annotations at strategic points. User input is shown in boldface.

Note the difference between “searching” and “selecting”. One may *search* through the relationship graph by simply typing the choice number in the list of items. This is simply browsing around to see what is there, and has no effect on the records available. One *selects* an item by prefixing the choice number with an “s” (eg. **s 3**). Also, when searching or selecting, you may use the name of an item instead of its number. This option may be used to search for or select items which are *not* in currently displayed list of choices (eg. **s sheep**).

At the end of the session, the user has isolated two records, both of which happen to contain parameter values. If judged suitable, these may be then added to the specification of the ecological model using the ECO program. (Alternatively, the records could have contained mathematical relationships or other information which could be used in the model.)

### 6. Trial Runs With Inexperienced Users

The browser has been tested by five people who had no previous contact with the system. Two of these were familiar with AI techniques; the remainder were final year ecology undergraduates. Before using the browser, they were given a five minute introductory talk and a ten minute demonstration of the system in operation. This was sufficient introduction and they quickly became

WELCOME TO THE ECO BROWSER

If you are unfamiliar with this program, or become confused at any stage, type "h" for information about permitted commands

YOUR STARTING POINT IS AT THE HIGHEST LEVEL (GENERAL)  
YOU HAVE 124 RECORDS TO BROWSE THROUGH

1 - age  
2 - sex  
3 - author  
4 - process  
5 - equation  
6 - nutrients  
7 - habitat  
8 - geographical location  
9 - organism  
10- numeric variable I am interested in "organism"  
SELECTION :9 so I move to choice 9

THERE ARE 124 RECORDS LEFT All the observations remain  
NO SELECTIONS MADE I haven't selected anything

t - general I have moved down to "organism"  
-----> organism (<-----  
1 - animal  
2 - plant

SELECTION :1 I move down to "animal"

THERE ARE 124 RECORDS LEFT  
NO SELECTIONS MADE

t - general I am shown a path back to the most  
-1 - organism general level so I know where I am  
-----> animal (<-----  
1 - insect  
2 - bird  
3 - mammal  
4 - fish

SELECTION :3 I want information about "mammal"  
so I select it

THERE ARE 20 RECORDS LEFT

SELECTED :  
a - mammal I can see that I have selected "mammal"

t - general  
-2 - organism  
-1 - animal  
-----> mammal (<-----  
1 - shetland pony  
2 - dog  
3 - human  
4 - pig  
5 - guinea pig  
6 - cow  
7 - sheep

SELECTION :a female I know that "female" exists  
so I select it directly

THERE ARE 8 RECORDS LEFT

SELECTED :  
a - mammal  
b - female

t - general  
-1 - sex  
-----> female (<-----

NO CHOICES AVAILABLE BELOW THIS POINT

SELECTION :a sheep I remember that "sheep" was  
available so I select it

ONLY ONE RECORD LEFT

SELECTED :  
a - mammal  
b - female  
c - sheep

t - general  
-3 - organism  
-2 - animal  
-1 - mammal  
-----> sheep (<-----

NO CHOICES AVAILABLE BELOW THIS POINT

SELECTION :p I want to see all the information in  
the remaining observation

THE RECORDS AVAILABLE AT THIS POINT ARE :

RECORD 100  
[standard,metabolic rate] is 25.7:cal/kg<sup>-1</sup>day<sup>-1</sup>  
sex is female  
age is adult  
mammal is sheep

SELECTION :u b I don't want this information so I  
undo my selections back to "female"

THERE ARE 8 RECORDS LEFT

SELECTED :  
a - mammal The "sheep" selection is lost from the  
b - female current selection history

t - general  
-1 - sex I have been automatically moved to "female"  
-----> female (<-----

NO CHOICES AVAILABLE BELOW THIS POINT

SELECTION :s cow I decide to select "cow"  
instead of "sheep".

THERE ARE 2 RECORDS LEFT

SELECTED :  
a - mammal  
b - female  
c - cow

t - general  
-3 - organism  
-2 - animal  
-1 - mammal  
-----> cow (<-----

NO CHOICES AVAILABLE BELOW THIS POINT

SELECTION :p

THE RECORDS AVAILABLE AT THIS POINT ARE :

RECORD 102  
[standard,metabolic rate] is 15.2:cal/kg<sup>-1</sup>day<sup>-1</sup>  
sex is female  
age is adult  
mammal is cow

RECORD 96

sex is female  
[activity,metabolic rate] is 21.:cal/kg<sup>-1</sup>day<sup>-1</sup>  
age is adult  
mammal is cow

SELECTION :quit I decide to stop at this stage

Figure 5-1: Example Session

accustomed to the display format and effects of basic commands. The only source of confusion was between searching to an item and selecting records which refer to an item. Some of the subjects expected records to be selected automatically whenever a new item was visited but the difference between these actions was accepted without apparent difficulty, when explained by example.

## **7. Comparison With Other Browsing Systems**

Other systems have been constructed which tackle similar browsing problems for different domains and using different approaches. A browser is being developed for the United Kingdom Department of Health and Social Security (as an Alvey Large Scale Demonstrator Project) which allows users to provide a description of the required information by providing values for one or more fields in a standard form [Clark & Pettit 84]. This would be impractical for our application because the number of types of ecological information is extremely large.

The RABBIT browser [Tou et al 82] addresses problems which are more similar to those which the ECO browser faces. It assumes that users will have an incomplete knowledge of the names which they can use in the system to specify their requirements. It also uses "perspectives" which correspond approximately to the "context" information supplied by the ECO browser. However, there are major differences between the approaches used in the RABBIT and ECO browsers. RABBIT relies on finding examples from the knowledge base which match users' specifications and uses these as a means of prompting for further specification details. In other words, the "records" in RABBIT's knowledge base are shown as a prompt for further selection criteria, rather than ECO's method of using relationships between items to drive the selection process and displaying records only upon specific request. This requires users of RABBIT to have some idea, however vague, of what they are looking for when entering the system, in order to access an initial set of examples. The ECO browser starts the user off with a display of the most general items in the knowledge base and offers these as possibilities which may be explored in more detail, allowing the system to be used by people with no preconceptions of the things they are interested in.

The Smalltalk-80 system provides a browser for examining definitions of object classes [Tesler 81]. It distinguishes 4 levels within its classification, each of which is represented within a separate "pane" on a graphics screen. Smalltalk browsing operates on the following principle. First, one item is selected from the list of items at the most general level. The system then displays, in the adjoining pane, a list of children of the selected item. An item from the list of options in the new pane can then be selected. This process is repeated until one item from each level has been chosen, thus defining an individual object description.

The ECO browser differs from the Smalltalk approach in four important ways.

- \* The Smalltalk browser operates on a comparatively shallow hierarchy, comprising 5 levels. This allows it to display, simultaneously, all the items at each point in the user's path toward a specific object description. The ECO browser must accommodate hierarchies of, potentially, much greater depth. Therefore, it cannot provide all this information at once, for fear of overwhelming the user (see section 3.3).
- \* Items at a particular level of the Smalltalk classification appear to be mutually exclusive. Thus, selecting one item from a level automatically makes other items at that level unavailable. The ECO browser accommodates records which may contain several items from the same level so the process of selection may entail multiple selections from items of similar status (see section 3.2).
- \* The only way to move down the Smalltalk classification is by selecting an item from the level above. The ECO browser distinguishes between simply moving around the relationship graph and selecting items of particular interest (see section 3). This allows users to look around before committing themselves to a particular set of choices - an essential feature when browsing through a knowledge base which has connections to observational records at varying levels in the relationship graph (see section 3.1).
- \* The ECO browser provides a means of returning to previous selection states, thereby freeing users from the tedium of remembering and re-implementing a cancelled selection sequence which they want to retry.

The browser used in Smalltalk is capable of operating successfully, without using complex mechanisms, because it is applied to a comparatively small, rigidly structured knowledge base. The added complexity of the ECO browsing mechanism stems from the need to accommodate a large volume of loosely structured data.

## **8. Future Additions**

### **8.1. Graphics display**

The ECO browser relies heavily on communicating to users their position in the graph. The use of bit-map displays via a suitable graphics interface would allow the meaning of displays to be more obvious to the user. For example, the tree of previous selections could be drawn on the screen using lines for arcs and boxes for nodes, rather than the current method of indenting the name on each

line according to its depth in the tree.

## 8.2. Permitting complex choices

Currently, the browser operates on the assumption that users will be capable of making definite selection choices. This was the case in our short trial sessions but some users may want to make imprecise selections. For instance, in the example in figure 5-1 one might have wanted to say "I am probably interested in cows but might be interested in sheep" instead of selecting cows alone or sheep alone. A mechanism is needed for allowing selections with associated degrees of certainty, combined with the existing definite choice mechanism.

Another browser prototype is being developed which selects records by scoring each one according to how closely its constituent items are to nodes in the graph specified by the user as being of interest. In this way, an arbitrary number of criteria, some of which may be exclusive of each other, may be used to find the most suitable records. This scoring system could be extended to allow the user to specify the importance of each item selected and the system could weight the scores accordingly. However, such complicated mechanisms are likely to reduce the speed of selection significantly. Our policy has been to maintain a simple control system which is maximally efficient and easy to understand.

## 8.3. Guidance for users

At present, the onus is on the user to decide which items s/he is interested in, with no intelligent advice available from the system. This does not seem to be a problem in the context of the ECO program, since the users are assumed to be prepared to search the knowledge base, testing out various selection options. The people who tested the browser found the method quite easy to understand. However, in some circumstances it would be desirable for the Browser to provide advice about which options to choose; shield some of the possible options from the user or make some decisions automatically. Guidance of this sort would depend on the previous choices made by the user, the current structure of the simulation model which is being built, and/or information about the user's purpose in entering the system. This implies an initial phase of determining the user's needs. It seems likely that this type of control could be built on top of the current system and work along these lines may take place soon.

## **9. Conclusions**

The ECO browser uses a simple search strategy and display format which is designed to facilitate easy operation of the system. Users with no notion of the type of information they require should be able to select at a general level at first, progressively refining their description as they move through the relationship graph. This approach differs from that of existing large knowledge base browsers, such as RABBIT, which rely on matching examples to users' initial specifications and use these examples as cues for further, detailed specification. Small scale experimental trials with inexperienced users tentatively suggest that our specific requirements of flexibility plus ease of understanding have been achieved by our browser. However, the problems of implementing the prototypical browser to suit large knowledge bases have not been resolved.

### **Acknowledgements**

This work was funded by SERC grant GR/C/06226. Austin Tate provided information about knowledge representation methods and helpful comments on earlier drafts. The browser was tested by Diana Bental, Paul Brna and three anonymous Ecology students.

## References

- [Bobrow & Stefik 81]  
Bobrow D. & Stefik M.  
*The LOOPS Manual*.  
Technical Report KB-VLSI-81-13, Knowledge Systems Area, Xerox  
Palo Alto Research Center, 1981.
- [Clark & Pettit 84]  
Clark M., & Pettit, P.  
The Alvey Large Scale Demonstrator Project for the Department of  
Health and Social Security.  
In *The Proceedings of the Joint Alvey MMI/IKBS Workshop on  
Applications-Driven Research*. 1984.
- [Clayton 84]  
Clayton B.  
*Programming Primer: ART (Version 1.1)*  
Los Angeles, CA, 1984.
- [Fahlman 79]  
Fahlman, S.E.  
*NETL: A System for Representing and Using Real-World Knowledge*.  
The MIT Press, 1979.
- [MacGregor 84]  
MacGregor D.R.  
An Integrated High Performance, Hardware Assisted, Intelligent  
Data Base System for Large-Scale Knowledge Bases.  
In *Proceedings of the First Alvey Workshop on Architecture for  
Large Knowledge Bases*. 1984.
- [Tesler 81]  
Tesler, L.  
The Smalltalk Environment.  
*Byte* 6(3):90 - 147, 1981.
- [Tou et al 82]  
Tou F., Williams M., Fikes R., Henderson A., & Malone T.  
RABBIT: An Intelligent Database Assistant.  
In *Proceedings of AAAI-82*, pages 314-318. 1982.  
Xerox Palo Alto Research Center.
- [Uschold et al 84]  
Uschold, M., Harding, N., Muetzelfeldt, R. and Bundy, A.  
An Intelligent Front End for Ecological Modelling.  
In O'Shea, T. (editor), *Proceedings of ECAI-84*. ECAI, 1984.  
Available from Edinburgh as Research Paper 223.

## Table of Contents

1. Introduction	0
2. Browser Requirements in an Ecological Domain	1
2.1. Knowledge representation	1
2.2. Using the browser on large knowledge bases	2
3. The Browsing Mechanism	3
3.1. Searching for items of interest	3
3.2. Selecting records	3
3.3. Displaying information to users	4
4. Additional Features	5
4.1. Undoing selections	5
4.2. Displaying selected records	6
4.3. Help facility	7
5. An Example of the Browser in Action	7
6. Trial Runs With Inexperienced Users	7
7. Comparison With Other Browsing Systems	9
8. Future Additions	10
8.1. Graphics display	10
8.2. Permitting complex choices	11
8.3. Guidance for users	11
9. Conclusions	12



## **List of Figures**

<b>Figure 3-1:</b>	Sample of a standard display	5
<b>Figure 4-1:</b>	Example of a Selection Tree	6
<b>Figure 5-1:</b>	Example Session	8