

TOWARDS A DESIGN SUPPORT TOOL FOR MULTIMODAL INTERFACE DESIGN

The Taxonomy Workbench and Theory Demonstrator

Niels Ole Bernsen, Shijian Lu, Michael May

Abstract

This paper presents the layout and functionality of Version 2 of the Taxonomy Workbench and Theory Demonstrator which is being developed at the Centre for Cognitive Science. The Theory Demonstrator uses multimedia/multimodal hypertext to present the taxonomy of unimodal output modalities which has been developed as part of addressing the research agenda of Modality Theory. The Workbench functionality of the system is supported by (i) a facility for classifying multimodal objects represented in the media of graphics, acoustics and haptics; and (ii) a search facility which enables flexible search on the basic properties which define the +70 unimodal modalities generated in the taxonomy. The ultimate objective is to develop the Taxonomy Workbench and Theory Demonstrator into an interface design support tool.

1. Introduction

During a two-year period, the multimodal systems group at the Centre for Cognitive Science, Roskilde University, has been working on establishing and implementing parts of the research agenda for Modality Theory. The research agenda for Modality Theory is the following (Bernsen 1993a):

1. To establish sound conceptual and taxonomic foundations for describing and analysing any particular type of unimodal or multimodal output representation relevant to human-computer interaction (HCI);
2. to create a conceptual framework for describing and analysing interactive computer interfaces;
3. to develop a practical methodology for applying the results of steps (1) and (2) above to the problem of information-mapping between work/task domains and human-computer interfaces in information systems design.

Modality Theory thus aims to establish the theoretical and methodological basis for addressing the information-mapping problem in its general form, i.e.:

Given any particular class of task domain information which needs to be exchanged between user and system during task performance, identify the set of input/output modalities which constitute an optimal solution to the representation and exchange of that information.

An ultimate objective is to use results in building computerised tools for the support of interface design.

Work started on the first part of this research agenda, i.e. the development of a taxonomy of output modalities in the media of graphics, sound and touch. It was realised from early on that work progress might benefit from the support of a software tool in which we might represent large numbers of samples of output representations for the purposes of analysing their properties and testing possible taxonomy schemes. This led to the development of Version 1 of the Taxonomy Workbench (May and Bernsen 1993, May and Tobin 1993), which was demonstrated by May at INTERCHI '93 in Amsterdam in the Spring of 1993. Version 1 is a database tool programmed in OMNIS 7 and designed to assist research by (a) setting up a common multimedia/multimodal database of output example representations, (b) assisting the description and classification of these examples according to different assumptions about the modalities involved, and (c) enabling thought experiments such as, e.g., the testing of different hypotheses about features of the modalities and their interrelations.

When we had succeeded in establishing a robust, intuitively plausible and principled taxonomy of output modalities (Bernsen 1994a,b), the Workbench in its current configuration had done its job, proving the usefulness of software support for Modality Theory development (Bernsen 1993b, May 1993a,b,c,d). This gave rise to the idea of re-designing the software tool with four objectives in mind: (1) to create a software demonstrator of the taxonomy of output modalities, (2) to use the demonstrator to further explore the functional properties of different output modalities in order to map out which information a particular modality is suited for representing, (3) to support exploration of the Information-Mapping Methodology (Bernsen and Bertels 1993, Verjans and Bernsen 1994, Verjans 1994), and (4) to move towards turning the Workbench concept into a support tool for multimodal interface design.

The identification of functional properties of modalities is important to the achievement of objectives (2) and (3) above. We view representational modalities as having two broad kinds of property, i.e. declarative properties and functional properties. *Declarative properties* are the properties, basic or otherwise (see below), assigned to a particular modality in order to define or describe what it is. Thus, for instance, linguistic modalities share the property of being syntactic-semantic systems of meaning. *Functional properties* characterise what a certain modality is good or bad, suitable or unsuitable at representing and sometimes under which conditions this is the case. Arbitrary acoustics, for instance, may serve useful alert functions in low-acoustic environments. Successful information mapping must be informed by knowledge of functional properties and not just by declarative knowledge. It turns out that many functional properties of modalities follow more or less directly from their declarative properties, which is why, in the modality document structure to be described below, no distinction has been made between the set of declarative properties and the set of functional properties. This means that many functional properties can be analytically derived from the basic declarative properties of modalities (cf. Bernsen 1994b). However, to capture other sets of functional properties, an empirical

approach is needed in which different modality samples are analysed to identify their functional characteristics, often in conjunction with scenarios of use.

This paper presents Version 2 of the Taxonomy Workbench. The system has now become a combined demonstrator of the taxonomy of output modalities and a Modality Theory workbench. Objective (4) above, i.e. the creation of an interface design support tool, has been approximated, but hardly fully achieved, through these developments. When the tool has been fully implemented in a couple of months, we will evaluate its appropriateness as a design support tool including the amount of further re-design that might be needed to achieve this aim. The overall specification of Version 2 was done jointly by Bernsen, Lu and May. Lu designed the classification and search tools and did most of the implementation, including the search for document illustrations and their inclusion in the workbench. Bernsen and May created the documents to be illustrated.

Version 2 of the workbench represents a complete re-design of Version 1. It has three basic components: a multimedia/-multimodal database, the taxonomy theory demonstrator and the taxonomy workbench. The database serves as a common basis for the two other components. The taxonomy workbench provides tools for manipulating and interacting with the database while the theory demonstrator draws illustrations directly from it. The workbench includes two management tools, a classification tool and a search tool. The most important differences between the two versions of the system are the following:

- 1) *Change of scope*: from covering a variety of taxonomy ideas to concentrating on our current taxonomy of unimodal modalities.
- 2) *Change of focus*: from a declarative taxonomy to a combined declarative/functional taxonomy (cf. above).
- 3) *Upgraded functionality*: the strength of the workbench lies in the kind of support it offers. The workbench database should not just provide a large set of data records but should offer extensive support for tasks such as classifying objects, analysing objects and retrieving relevant records based on sets of properties. Version 2 incorporates revised versions of the analysis and classification facilities of Version 1 as well as a much improved search facility.
- 4) *New dimension added*: to achieve re-design objectives (1) and (2) above, the taxonomy theory demonstrator has been added. The demonstrator provides a structured presentation of the theory, which is simultaneously a platform for closely scrutinizing the theory, testing its analytic power by requiring it to account for complex or difficult cases, etc. Theory consolidation is achieved in this process of analysing, testing, validating and enriching the theory.
- 5) *Interaction with information mapping work*: Modality Theory, as implemented in the workbench, is being applied in the series of case studies currently undertaken to conceptually test the Information Mapping Methodology (Bernsen and Bertels 1993, Verjans and Bernsen 1994, Verjans 1994). This process provides valuable information on the usefulness and scope of Modality Theory as a potential design support tool as well as adding a source of test results on the theory itself.
- 6) *New media and hypertext added*: the output media of sound and touch have been added to the workbench, touch, however, only through graphic presentations, and workbench documents are now linked through hypertext links.

The main elements which remain to be implemented in order to complete Version 2 of the workbench are (i) a complete set of illustrations of the hypertext documents. In particular, Version 1 of the workbench only contained graphic objects and we have only recently begun to include acoustic and haptic/graphic objects (cf. (6) above) in the

database, which is why these media will receive little attention in what follows; (ii) creation of all necessary hypertext links; (iii) a thorough inspection of the close to 150 entries which jointly express Modality Theory; (iv) addition of the sub-atomic level (cf. Sect. 2.1 below); and (v) acquisition of more database sample material in order to support the full scope of the theory.

The plan of the paper is as follows. The taxonomy theory demonstrator is presented in Section 2. This is done by providing an overview of the theoretical framework of (output) Modality Theory, followed by a presentation of the hypertext document structure. The classification tool is introduced in Section 3 in which the interface and its functionality are presented in reasonable procedural detail. The search tool is similarly treated in Section 4. Section 5 concludes by briefly discussing how to develop the current workbench into a design support tool. Appendix A lists the current modality document and lexicon entries in the theory demonstrator.

2. The Taxonomy Theory Demonstrator

The taxonomy is demonstrated in three types of presentation: a taxonomy tree, multimedia/multimodal modality documents and multimedia/multimodal lexicon documents. The taxonomy tree is a graphic representation of the structure of the unimodal taxonomy. The modality documents define, explain and illustrate each unimodal modality. The lexicon documents accessed through modality documents define, explain and illustrate additional key concepts of the theory of output modalities.

2.1 Overview of the theoretical framework

The current taxonomy of unimodal output modalities has been generated from a set of basic properties, such as analogue/non-analogue, arbitrary/non-arbitrary, linguistic/non-linguistic, and static/dynamic as well as distinction between the three media of graphics, acoustics and haptics (Bernsen 1994b). Unimodal modalities are individually defined by their basic properties. The taxonomy is a hierarchical structure having at least three levels, i.e., from the top down: the super level, the generic level and the atomic level. In one sector of the taxonomy, i.e. static analogue graphics, a sub-atomic level has been added at which finer distinctions are needed for the taxonomy to properly serve its purpose. The *super level* is defined by the first three sets of basic properties, i.e. analogue/non-analogue, arbitrary/non-arbitrary, and linguistic/non-linguistic. At the *generic level*, the static/dynamic and media distinctions are introduced. The *atomic level* modalities are defined through an additional, larger set of basic properties. For the purpose of preparing the workbench for use as a design support tool, the atomic level (and sometimes the sub-atomic level) is assumed to constitute the interface designer's 'toolbox' level. From there, through the property inheritance principle (see below), the interface designer should be able to access workbench information at a level of detail sufficient to support design reasoning.

2.2 The taxonomy tree

The taxonomy tree has 70 nodes, i.e. 4 at the super level, 20 at the generic level and 46 at the atomic level (Fig. 1). In the tree structure, colour, being one of the information channels in graphics, is used to carry differential information. The super, generic and atomic levels are differentiated by their background colours, i.e. blue, blue/grey and light green, respectively. Different media are marked by different analogue icons, i.e. graphics by an eye, acoustics by a loudspeaker and haptics by a hand. Static and dynamic graphics are differentiated through the foreground colours of their icons, i.e. green and white, respectively. Property inheritance links are shown as lines connecting different unimodal modalities. Via these links, properties are inherited from the super level down to the atomic and sub-atomic levels, the latter of which is not shown in the tree structure as we are currently integrating sub-atomic level information at the atomic level. The layout of the tree is mainly determined by spatial constraints. At the top right-hand corner, a darker shaded grey explicit structure contains the legend of the taxonomy tree.

The taxonomy tree structure is a reduced version of the taxonomy of unimodal output modalities. For instance, the tree structure does not manifest the distinctions between static and dynamic acoustics and between static and dynamic haptics. Instead, these distinctions are being invoked in the documents on acoustics and haptics. Similarly, some nodes in the tree do in fact have daughter nodes although these have not been shown. For instance, the generic level modality *Static analogue graphic language*, i.e. static graphic language using analogue signs, does have a set of atomic level daughter nodes for representing hieroglyphic (or iconographic) writing in the modality types *Text*, *Labels/keywords* and *Notation*. However, this information has been incorporated into the presentation of *Static non-analogue graphic language*, i.e. static graphic language using non-analogue signs such as those which the reader is currently reading. The reason for these purely pragmatic reductions which have been made without losing important information, are (i) to reduce the number and nature of unimodal atoms to those which are expected to be important to interface design; and (ii) to avoid proliferation of - sometimes even useful - atoms in the acoustic and haptic media.

The taxonomy tree provides the main organising principle for the modality documents which demonstrate the taxonomy. Moreover, the tree is the starting-point for navigation in the theory demonstrator and hence for accessing lexicon entries. All the nodes in the tree are active. If any one of them is mouse-clicked, the relevant modality document opens. Mouse-clicking on any of the legend items opens the relevant lexicon document.

2.3 Hypertext document structure

There are currently a total of 134 modality and lexicon documents comprising some 150 Kb of (non-illustrated) text. The number of hypertext links is well above two thousand. The super level and generic level documents, the atomic level documents on linguistic, arbitrary and explicit structure modalities as well as the related lexicon entries have been done by Bernsen. The atomic level documents on analogue modalities and related lexicon entries are being done by May.

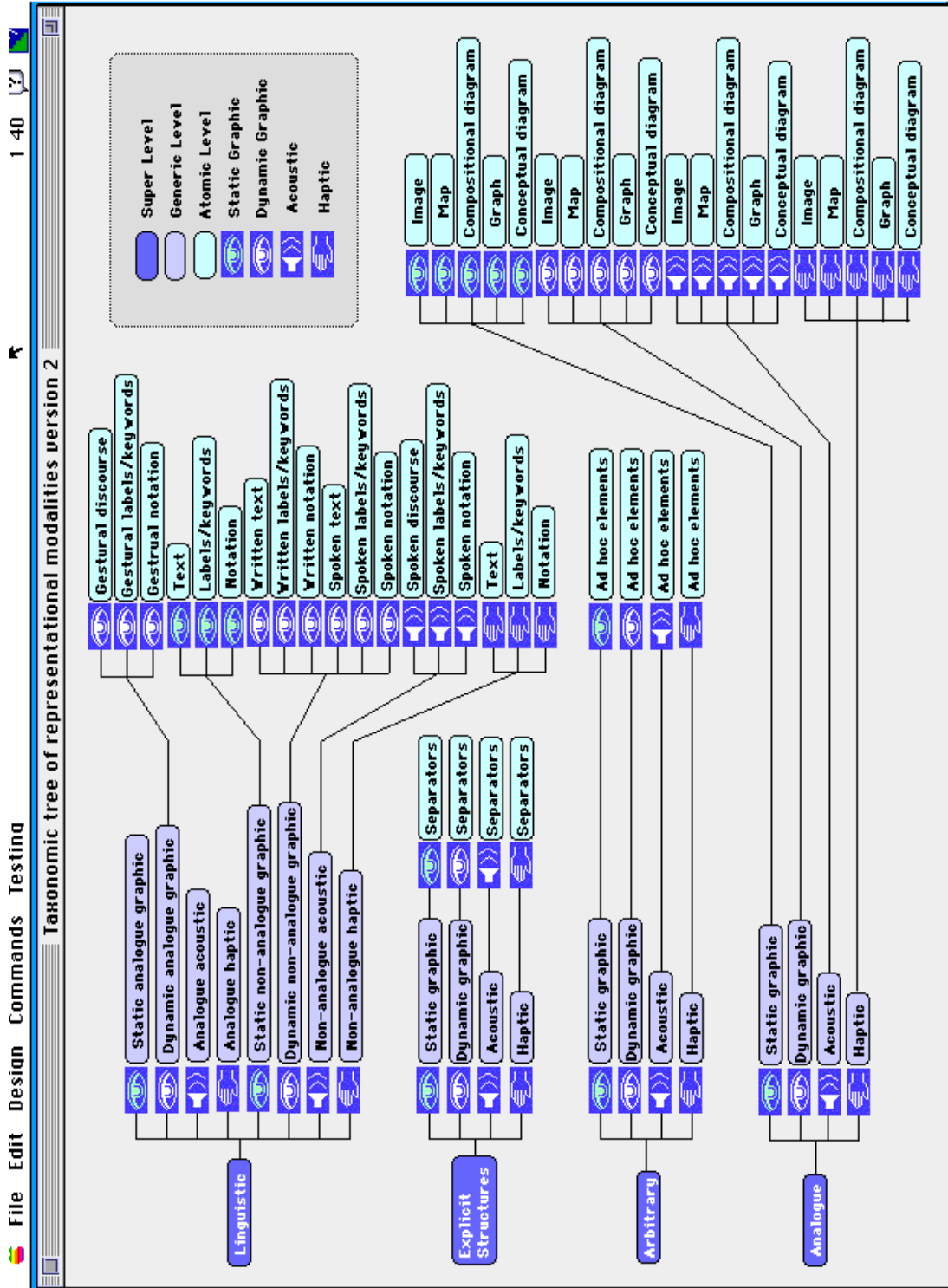


Figure 1. Workbench main screen. The taxonomy tree.

2.3.1 Modality documents

As stated above, modality documents are those documents which define, explain and illustrate unimodal modalities. These documents have a common format which includes the following entries (cf. Fig. 2):

1) *a relevant branch of the taxonomy tree*

The active modality node is highlighted on green background. Taking the active modality node as reference, the taxonomy tree branch normally includes all the levels above and one level down.

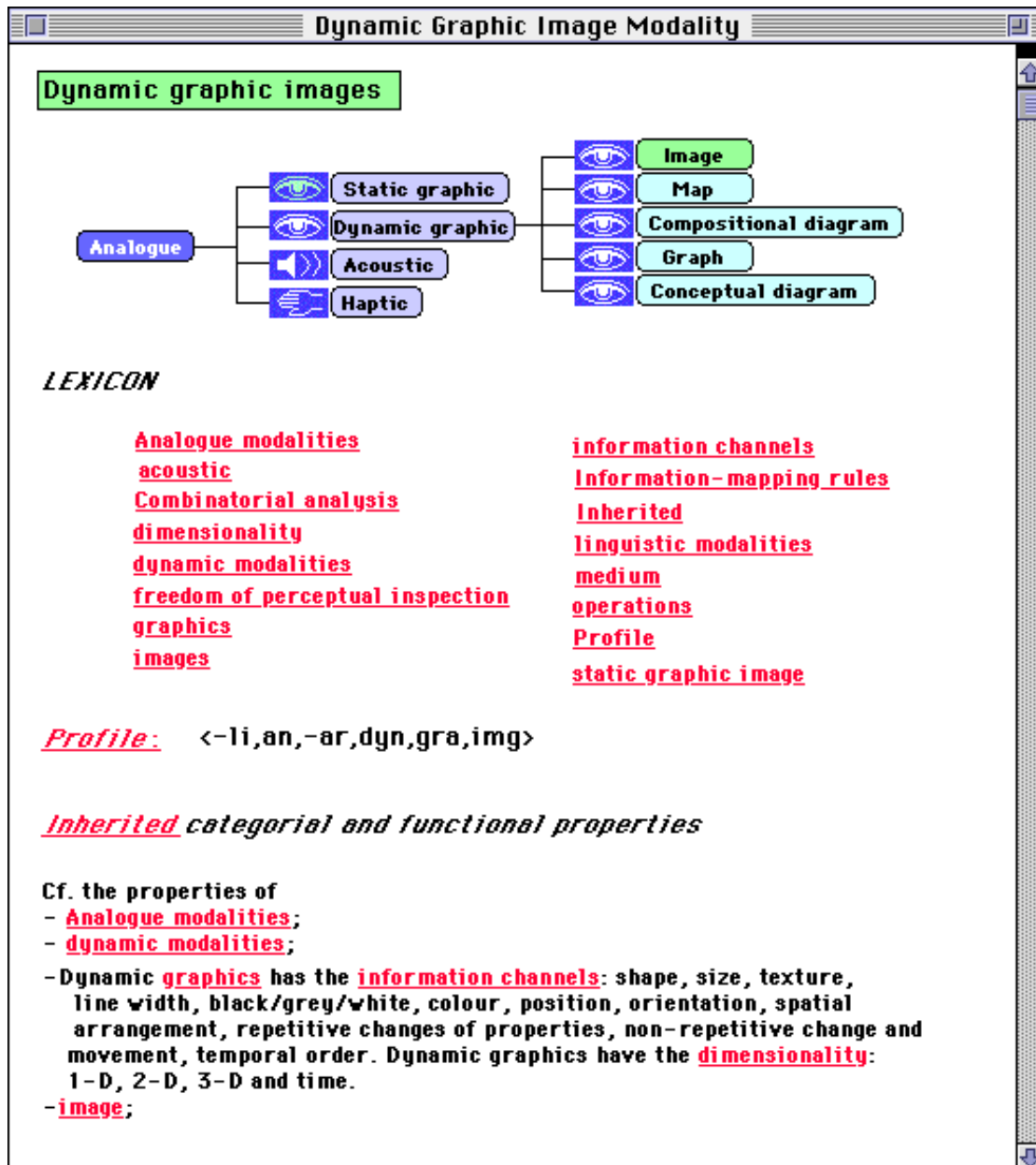


Figure 2. Modality document. The dynamic graphic image modality (to be continued).

2) a list of anchors

I.e., a list of all the active words in the present document.

3) profile

A notation is used to express the profile of the modality, i.e. the set of basic properties which suffices to specify the modality as distinct in kind from other modalities at the same level.

Dynamic Graphic Image Modality

Specific categorial and functional properties

Dynamic graphic image
- is a derived, dynamic version of **static graphic image**.
Dynamic graphic images, however, does not simply differ from static ones by their involvement of time rather by their lack of **freedom of perceptual inspection**. Therefore, a simple repetition of an images will be considered as static despite of its temporal dimension, for example, a blinking cursor.
- being dynamic, puts heavy cognitive load onto users.

- Exists in at least two forms (Fig.1), sequence or real-time animation. The former refers to the case where the presentation is a montage of very different static images which are inspectable over a limited duration of time. Figure 2 illustrates this kind of dynamic images.

Dynamic { Sequence
Real-time animation

Fig.1




Fig.2 Dynamic sequence of labelled still pictures, Static graphic display images with superimposed static graphic labels: A tour of San Francisco.

Real-time animation may be animated sketches(Fig.3) or video display.(Fig.4)

Figure 2. (Continued).

Dynamic Graphic Image Modality

Real-time animation may be animated sketches(Fig.3) or video display.(Fig.4)
This type of dynamic images is intrinsically dynamic.




Fig.3 An animation, in this case a dynamic graphical sketch with a superimposed sound sketch, depicting a running Alsatian dog




Fig.4 A unimodal dynamic graphical display image: A cinematographic movie rendering the Hindenburg disaster in 1937.

Information-mapping rules

Dynamic graphic image is a very interesting output modality. Static images are good at representing structure, pattern or trend, and an object realistically, whereas the strength of dynamic graphic images lies in providing a realistic-feel of processes or events.

Dynamically express processes or events <->
Consider using dynamic graphic image.

Combinatorial analysis

Dynamic graphic image combines badly with dynamic graphic **linguistic modalities** because of the risk of cognitive overload.

Dynamic graphic image combines well with **acoustic** linguistic modalities. There are at least two reasons for that. Firstly, the two kinds of modalities rely on different sensors to be perceived, therefore their combination does not impose threat of cognitive overload. Secondly, dynamic graphic image, as other analogue modalities, has the virtue of specificity and lacks the power of focusedness which is the strength of linguistic modalities. Consequently, the forementioned combination is optimal in the sense that they complementary to each other. Figure 5 gives a good example.




Fig.5 The same graphical display image as in Fig.4 combined with dynamic spoken discourse: A cinematographic movie rendering the Hindenburg disaster in 1937.

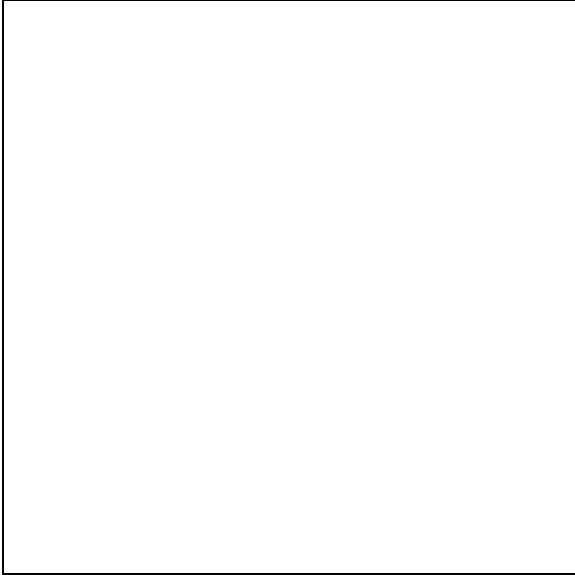
Figure 2. (Continued).

4) *inherited categorical and functional properties*

These are the properties (if any) which the modality inherits from higher levels. To keep documents short, these properties must be retrieved through the hypertext links.

5) *specific categorical and functional properties*

These are the properties which characterise the modality as being specifically different from its sister modalities with which it shares a common ancestry through inheritance from higher levels.



Relevant operations

As with figurative static graphics we can specify dynamic figurative images as object-oriented sketches or as viewer-oriented display. The difference between dynamic sketches and dynamic displays are of the same nature as we saw with regard to static graphics. Dynamic displays will be viewer-oriented in their presentation of real or virtual objects and events, that will (appear to) look like they would look like, seen from a particular angle, a particular distance etc. This can be illustrated by a photographic video sequence, that presents some objects and sequence of events in real-time (thereby supporting the viewers "realist" attitude of observation), and by a computer generated animation sequence, with so called photo-realistic rendering techniques, that attempts to reconstruct the visual information, that would be given to an observer, if he was observing virtual scene in reality and not in the constructed virtual reality.

Identified types-of

- Dynamic graphic sketches
- Dynamic graphic display

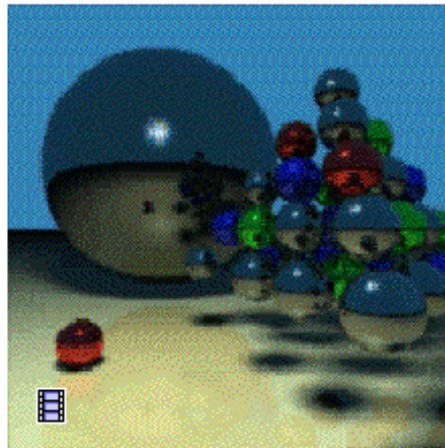


Fig.6 A unimodal dynamic graphical display image: A "Fly over"-animation of objects and events in a virtual world, generated by a ray-tracer program for photorealistic rendering (MacRTrace).

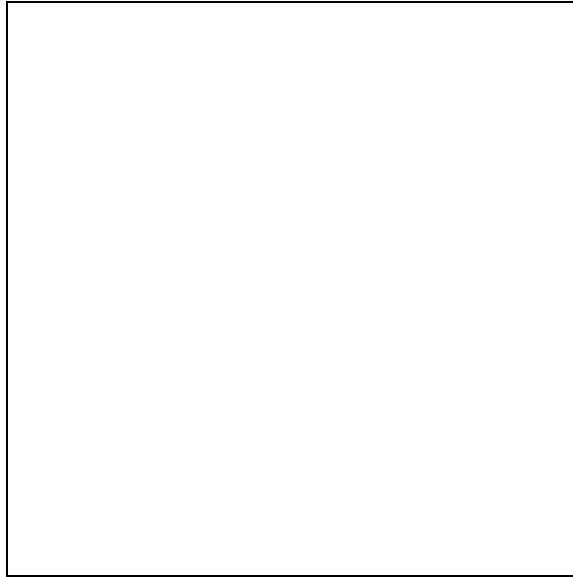


Figure 2. (Continued).

6) *information mapping rules*

Information mapping rules are similar to production rules. They express aspects of which information the current unimodal modality is good at, or unsuited for, representing and sometimes under which conditions this is the case.

7) *combinatorial analysis*

This analysis addresses compatibilities and incompatibilities between the present modality and other unimodal modalities.

8) *relevant operations*

Are operations applicable to the current unimodal modality.

9) *identified types*

Are the types of the present unimodal modality as expressed one level down in the taxonomy hierarchy.

10) *illustrations*

Modality illustrations may be found anywhere in a modality document.

Fig. 2 shows the *dynamic graphic image* modality draft document which can be accessed by clicking on the appropriate atomic node in the tree. Active words in all documents are in red and underlined.

2.3.2 Lexicon documents

Lexicon documents are documents which define, explain and illustrate additional key concepts of the theory of output modalities. Lexicon documents have no common format. They are normally accessed through the modality documents. Fig. 3 shows the lexicon document on *information channels*.

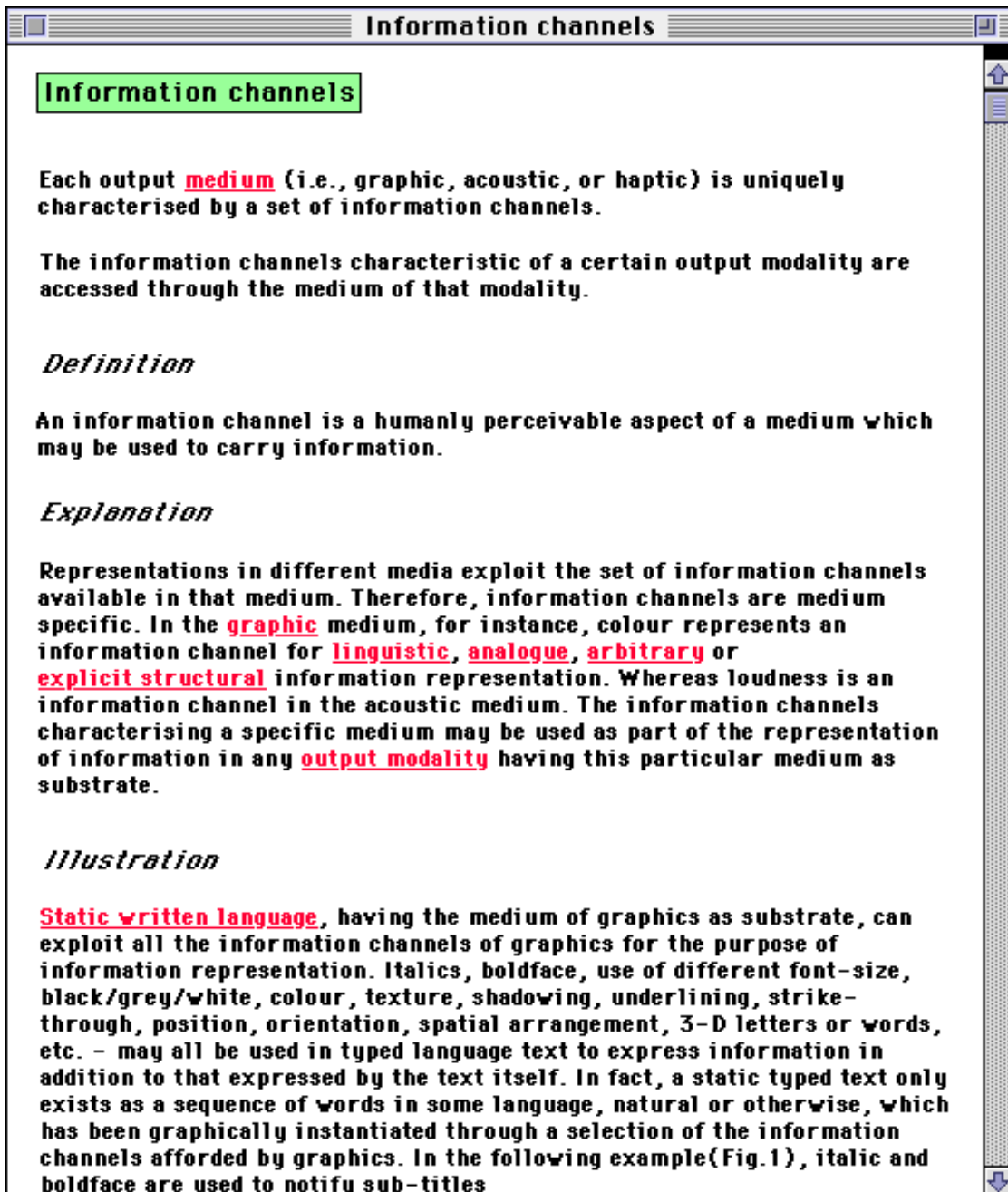


Figure 3. Lexicon document. Information channels (to be continued).

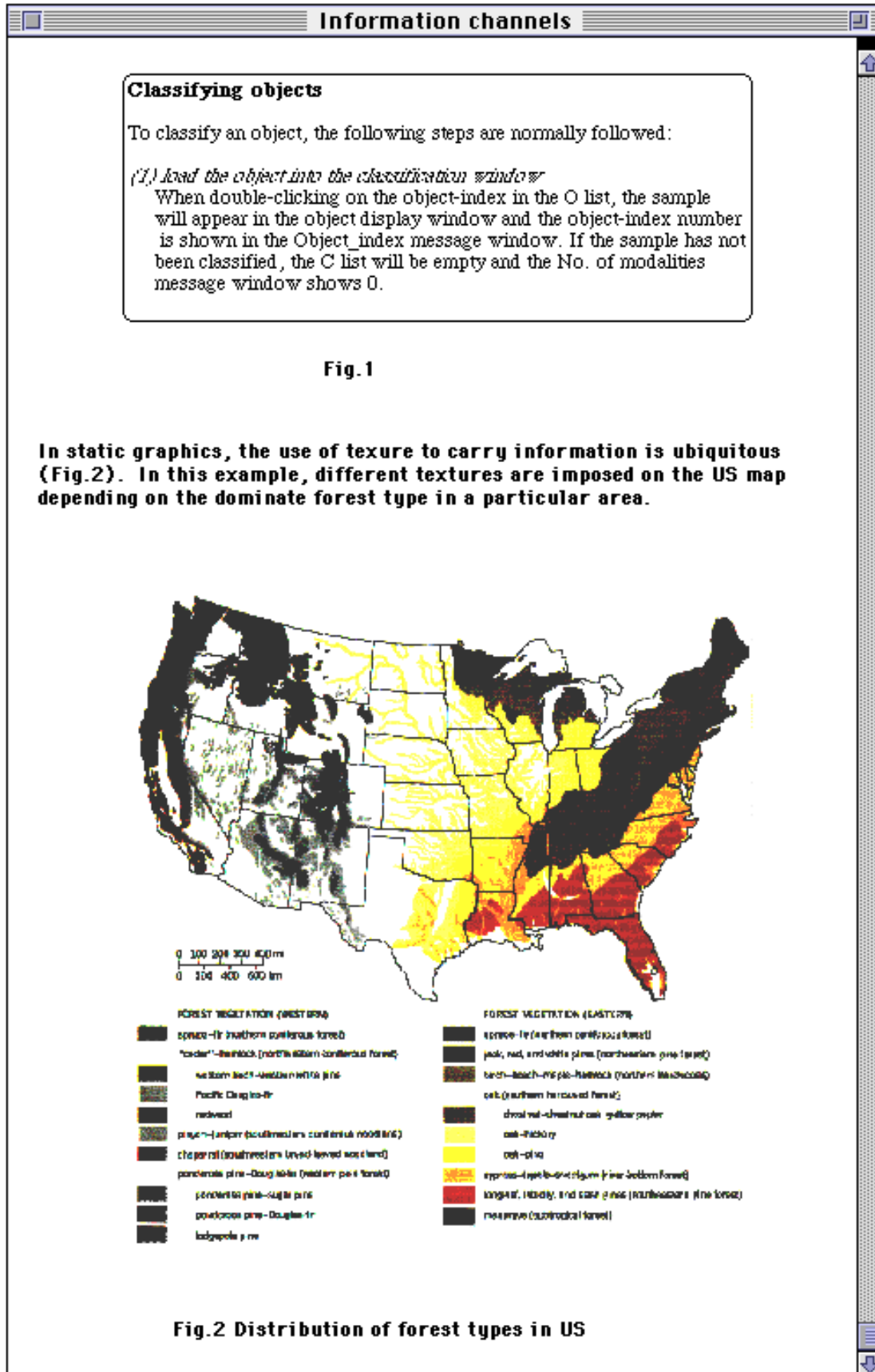


Figure 3.

3. Classifying Objects

For the workbench to serve its purpose, all the samples in the database, whether unimodal or multimodal, should be classified in accordance with the taxonomy scheme. A classification tool has been designed for this purpose.

3.1 The interface

The classification tool interface is shown in Fig. 4. In the top left-hand corner of the window, four pulldown menus are grouped together. They represent classification information on the *Super level*, *Generic level*, *Atomic level* and *Sub-atomic level*, respectively. The sub-atomic level information has not been implemented yet. Items in the pulldown menus are properties which are introduced at that particular level (see Fig. 5). The pulldown menus are used to narrow down choice options in the T list. There are three lists: T, C, and O. All 70 unimodal modalities are listed in the *T list*. In the *O list*, all object indices in the database are listed. The *C list* shows all the unimodal modalities which jointly constitute a particular sample. Four push buttons, labeled *Reset T list*, *Edit*, *OK*, and *Cancel*, are grouped together in the bottom right-hand corner of the window. To the left of the push buttons is an *Object display* window. Above the object display area are two message windows. The *Object_index* message window shows the object index of the object displayed in the object display window. The *No. of modalities* message window shows the number of unimodal modalities which constitute the classified object.

3.2 Functionality

3.2.1 Classifying objects

To classify an object, the following steps are normally followed:

(1) *load the object into the classification window*

When double-clicking on the object-index in the *O list*, the sample will appear in the *Object display window* and the object-index number is shown in the *Object_index* message window. If the sample has not been classified, the *C list* will be empty and the *No. of modalities* message window shows 0.

(2) *enter classification mode*

Press the *Edit* push button to enter classification mode.

(3) use pulldown menus to narrow down options available in the T list (optional, see also step (4))

The pulldown menus are designed for naive users who have difficulty in identifying a valid unimodal modality and locating it in the T list as well as for objects which are difficult to classify. The pulldown menus provide a way of focusing on one property at a time so that, step by step, the appropriate modality will be eventually identified. The menus operate directly on the T list. Initially, the entire set of unimodal modalities is present in the T list. When an item is selected from a menu, the T list is updated such that all and only the unimodal modalities which contain the chosen property are presented.

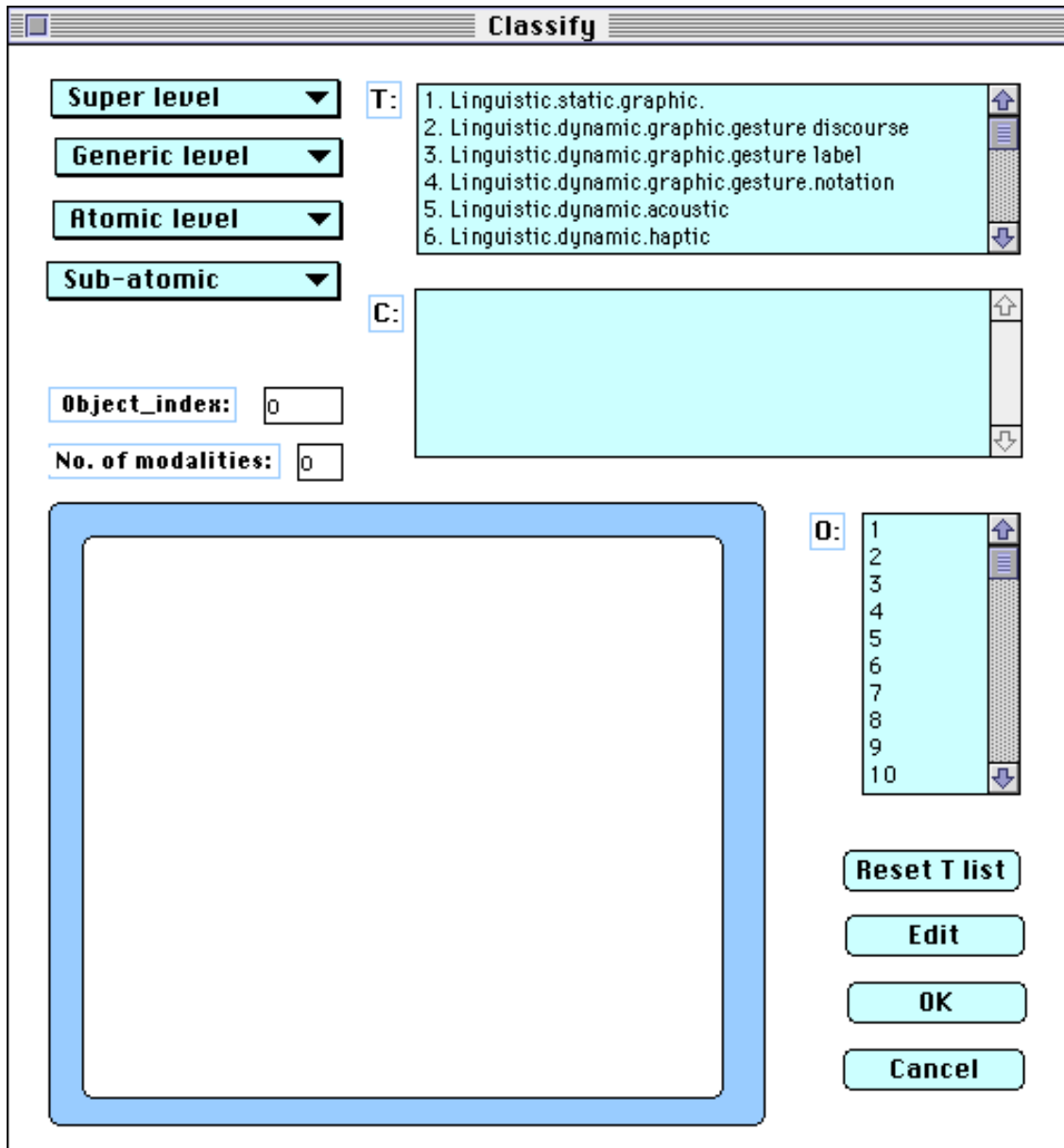


Figure 4. The classification tool interface.

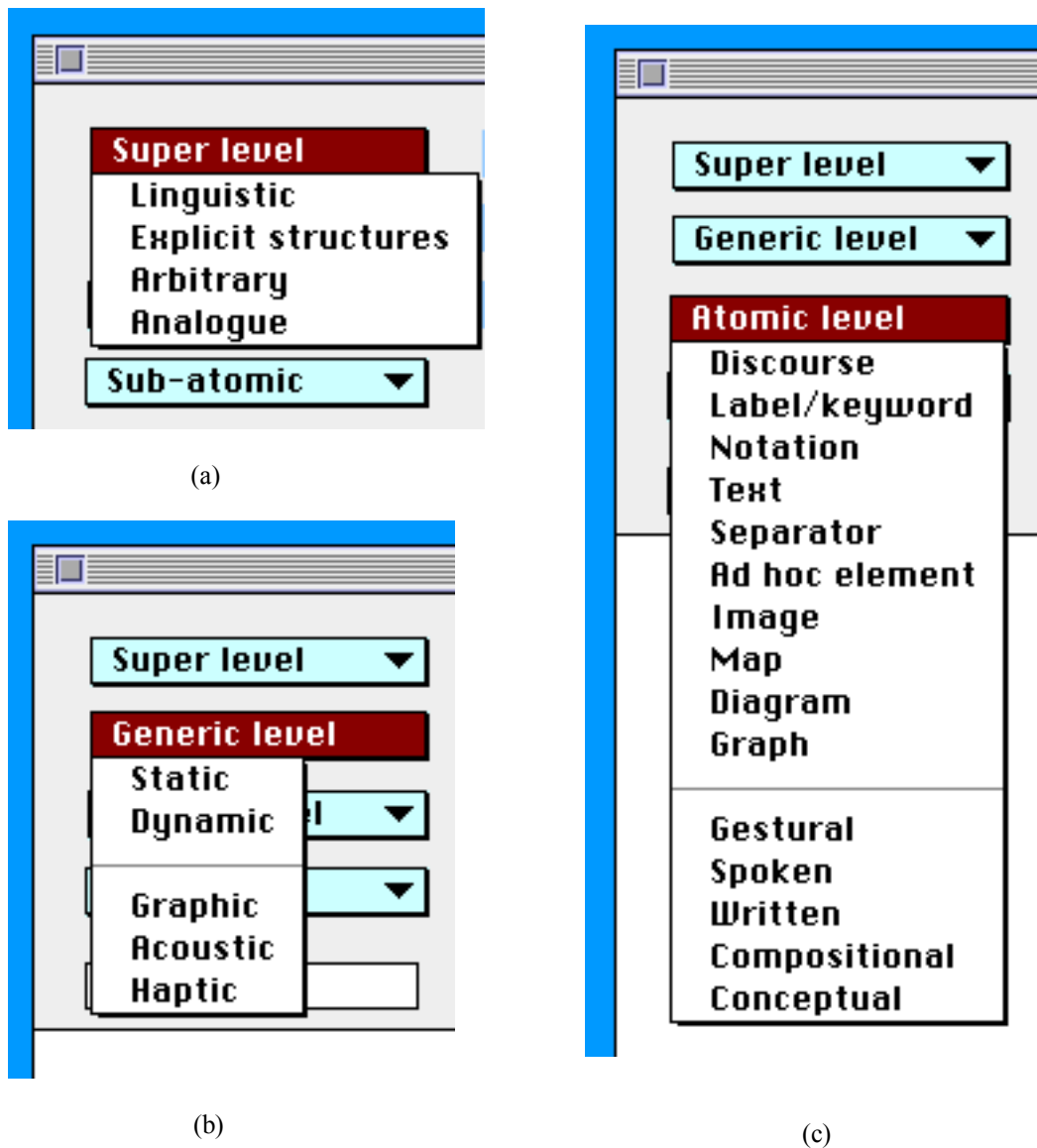


Figure 5. The classification pulldown menus. (a) Super level,

(b) Generic level, (c) Atomic level.

For example, consider the object numbered 422 in the database (Fig. 6). This object is a video clip imported from a piece of electronic educational material. As it is dynamic, one may inspect how it works by pressing the film-clip icon in the lower left-hand corner. The object is multimodal. To classify it, we focus on one component of the representation at a time. Let us focus on the mathematical formula immediately below the title. An experienced user will easily classify the formula as linguistic.static.graphic.notation and effortlessly locate it in the T list. Doing so, however, may be hard for novice users. Instead, the novice user may first

choose *Notation* from the atomic level. As a result, the *T list* will list all and only the notational unimodal modalities as shown in Fig. 7(a). The user may go on to choose *Static* from the generic level menu. In this case, the *T list* will contain all and only the unimodal modalities which are both notational and static as shown in Fig. 7(b). Now it has become much easier to classify this aspect of the multimodal object as linguistic.static.graphic.notation.

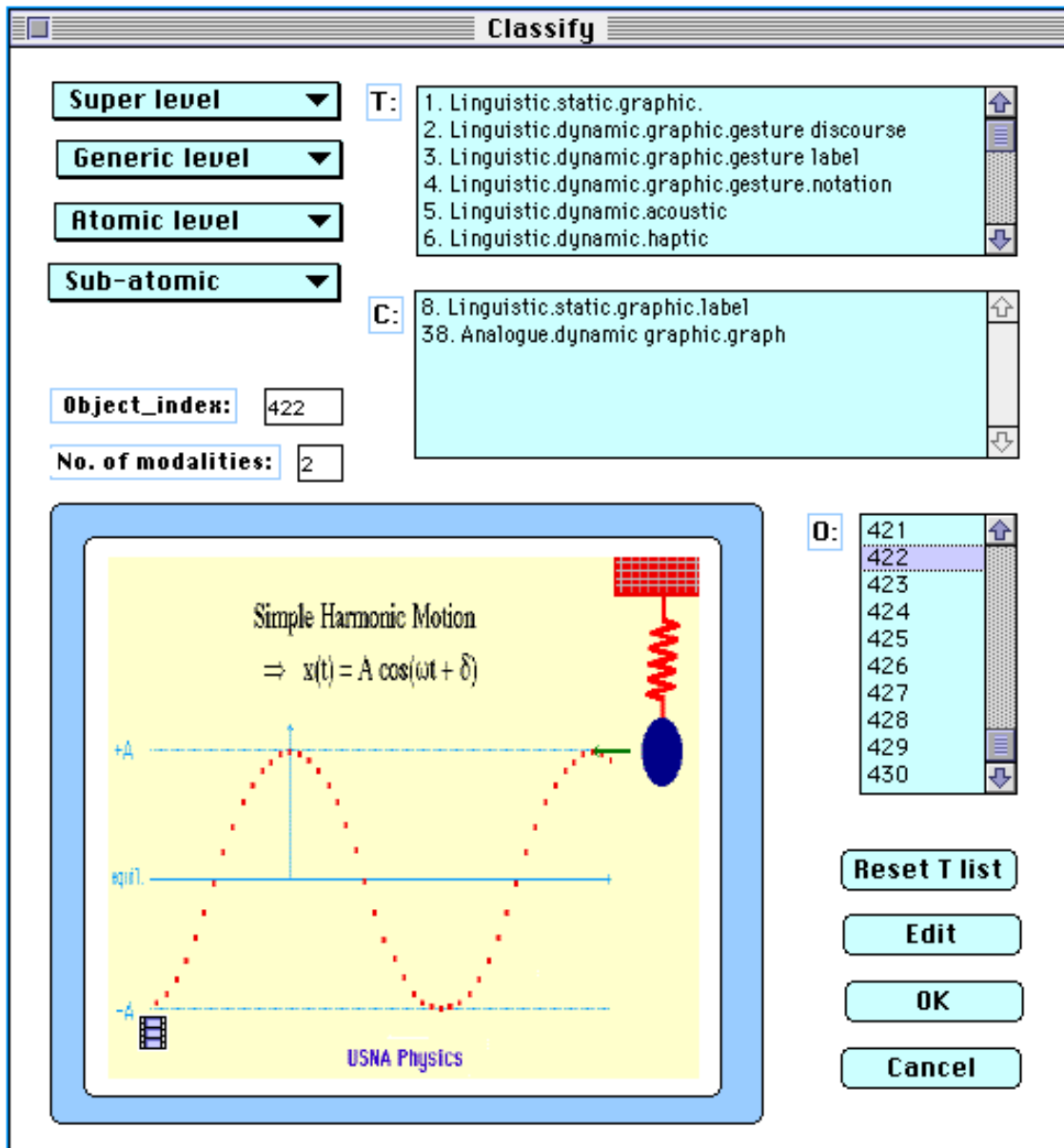
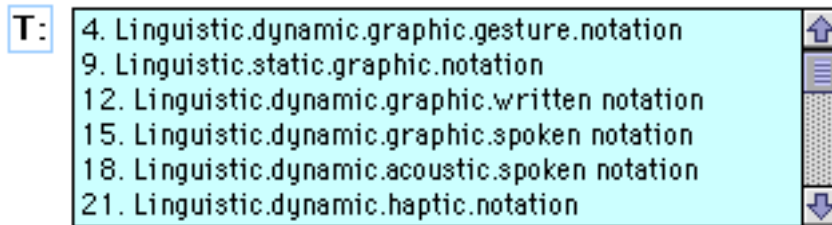
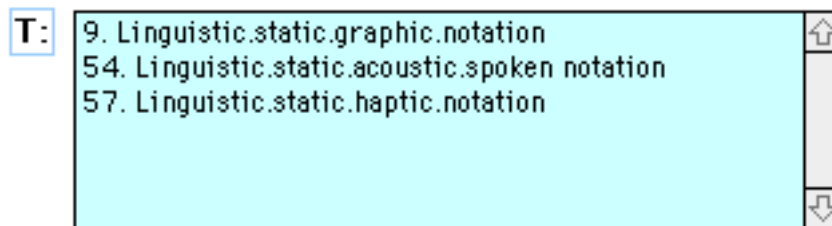


Figure 6. Object classification exemplified.



(a)



(b)

Figure 7. Response of List T to pulldown menu operations.

(a) List of notational modalities. (b) List of static and notational modalities.

(4) choose the right unimodal modality from the T list

Users may either navigate directly in the T list or use pulldown menus to assist in finding the right unimodal modality (cf. Fig. 7). Once the proper unimodal modality has been identified, double-clicking on it in the *T list* will copy it into the *C list*.

(5) modify classification (optional)

If an incorrect unimodal modality has been copied into the C list, users can easily undo the result by double-clicking on it.

(6) save classification

To save the classification listed in the *C list*, press the *OK* push button. In addition to the save action, clicking on the *OK* button causes two more system actions, i.e., exit of editing mode and reset of the list T to its initial state.

(7) reset list T (optional)

When classifying multimodal representations, the pulldown menus can be used iteratively when focusing on different unimodal aspects of the same object. Before shifting focus to a novel aspect, the *T list* should be reset to its initial state by clicking on the *Reset T list* push button.

(8) abort classification and exit edit mode (optional)

At any stage of the classification process, the user may abort and exit editing mode by clicking on the *Cancel* push button.

3.2.2 Viewing classification

When double-clicking on the object index in the *O list*, the object will appear in the *Object display window* and the object-index number is shown in the *Object_index* message window. If the object has already been classified, the *No. of modality* message window will show the number of its constituent unimodal modalities and the *C list* will present the constituent unimodal modalities.

In the example of Fig. 6, the object index number is 422. The object has been classified as a multimodal representation consisting of 5 unimodal atoms. These atoms are:

linguistic.static.graphic.label
 linguistic static.graphic.notation
 explicit.structure.static.graphic.separator
 analogue.dynamic.graphic.image
 analogue.dynamic.graphic.graph

4. Search in the Workbench

Once objects have been classified, search on any number of properties will greatly assist the study of individual unimodal or multimodal representations. The search tool interface was designed to the following commitments:

- (1) be as error-proof as possible. For example, the keying in of search specifications was discarded because it is considered an error-prone input technique which, in addition, puts an extra cognitive load on users. The keying in of search specifications requires users to be familiar with the notation in advance.
- (2) be easy to use. Users should not be required to know file structures and connections nor the search syntax.
- (3) be flexible. Users may initiate search at any stage in the search specification process.

4.1 What to search for

- (1) *Search by classification* (including atomic level distinctions).

This is to enable users to retrieve relevant objects by selecting any number of combinations of properties in the taxonomy. For instance, a user could search for objects which are *Linguistic* only, or objects which are both *Linguistic* and *Dynamic*, etc.

- (2) *Search by negation*.

This is search for objects which lack a particular property.

(3) *Exact match or inclusive search.*

Any classified object or record in the database will be characterised by a set of unimodal modalities. Let X denote the sub-set of unimodal modalities which characterises an object in the database and let Y denote the sub-set which have been specified in a particular search (both X and Y can be either unimodal or multimodal). In *exact match search*, only objects where $X=Y$ will be returned. *Inclusive search* will return all objects where $X=Y$ or $X \supset Y$ (i.e., Y is a sub-set of X).

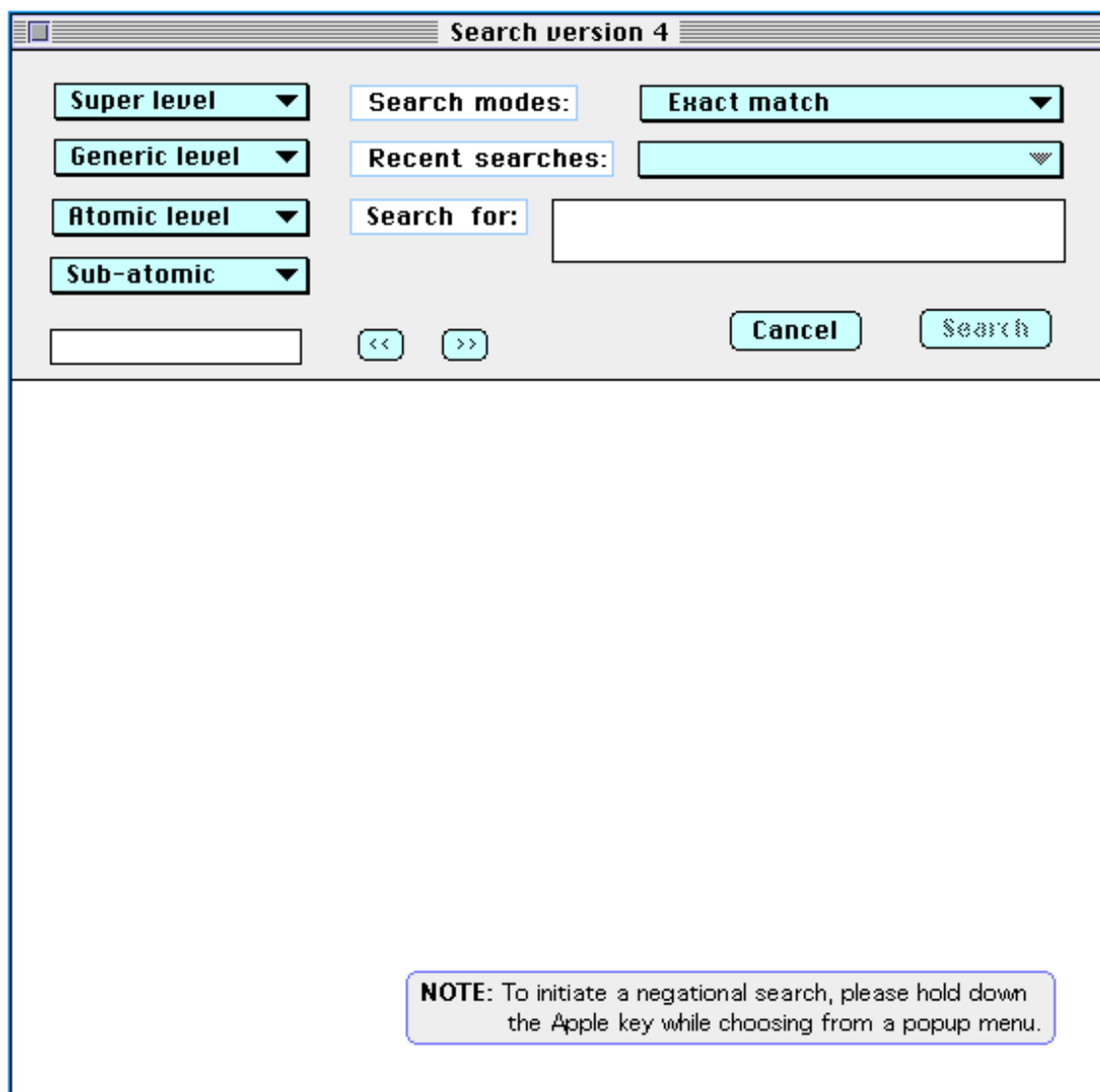


Figure 8. The search interface.

4.2 The search interface

Fig. 8 shows the search window which has two parts. The upper part with grey background is a control area. The lower part is a display area where the search result is represented as an iconic list. In the control area, there is a group of pulldown menus, two pulldown lists, two small message windows and four push buttons.

4.2.1 The pulldown menus

Each pulldown menu lists the properties introduced at that particular level of the taxonomy (cf. Fig. 5). Users specify their search from these menus. Search may be initiated from any of the menus using any number of valid combinations of properties. At the generic and atomic levels, a horizontal line separates the items in two parts, an upper and a lower part. Items from these two parts may be combined in specifying search whereas no items may be combined with other items from the same part of the same menu nor may the items at the super level be combined with each other.

4.2.2 The search specification message window

The small *Search for* message window (Fig. 9) keeps tracking the selections a user has made from the pulldown menus.



Figure 9. The search specification message window.

4.2.3 The cancel and search buttons

On opening the search window, the *Search* button is inactive. This button becomes active as soon as a menu item has been chosen. When the *Search* button is clicked, the specified search is initiated and the search specification is simultaneously stored in the *Recent searches* pulldown list. The *Cancel* button is used to clear the two message windows and the display area.

4.2.4 Search modes

There are two options in the *Search modes* list (Fig. 10), i.e. *Exact match* and *Inclusive*. If *Exact match* is set (done by default), only unimodal modality objects are being searched. If the user wants to consider multimodal objects which have some particular unimodal modality as constituent, then *Inclusive* should be set prior to initiating search.

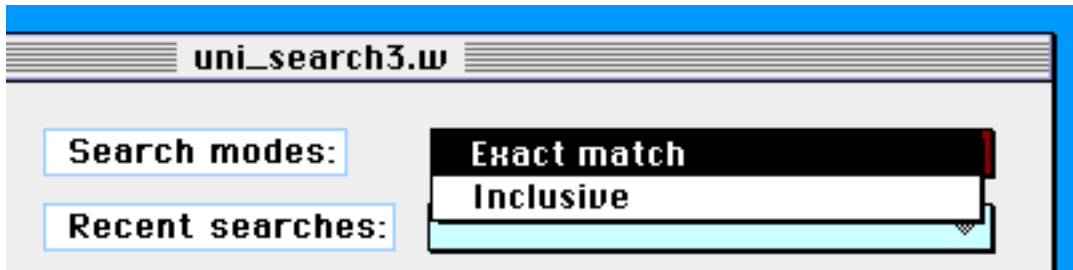


Figure 10. The Search modes pull-down list.

4.2.5 The iconic list

The search result is, firstly, presented in the small message window immediately beneath the pull-down menus. This window presents the number of matched objects. If at least one item was found, the items are, secondly, presented in the display area in the form of small icons which are scaled-down versions of the matched objects. If more than 20 items were found, only the first 20 items are displayed. The advantage of using icons is rather obvious, as they are much more informative than, e.g., a list of object-index numbers.

4.2.6 Recent searches

The *Recent searches* pull-down list (Fig. 11) is a temporary buffer where the 10 most recent search specifications are stored in the order of entry. The most recent specification ranks top in the list. Users can directly initiate a previous search from this list by selecting it. However, if the current search mode is different from the one in which the previous search was initiated, a different result will be produced. The *Recent searches* pull-down list is cleared when the search application is closed.

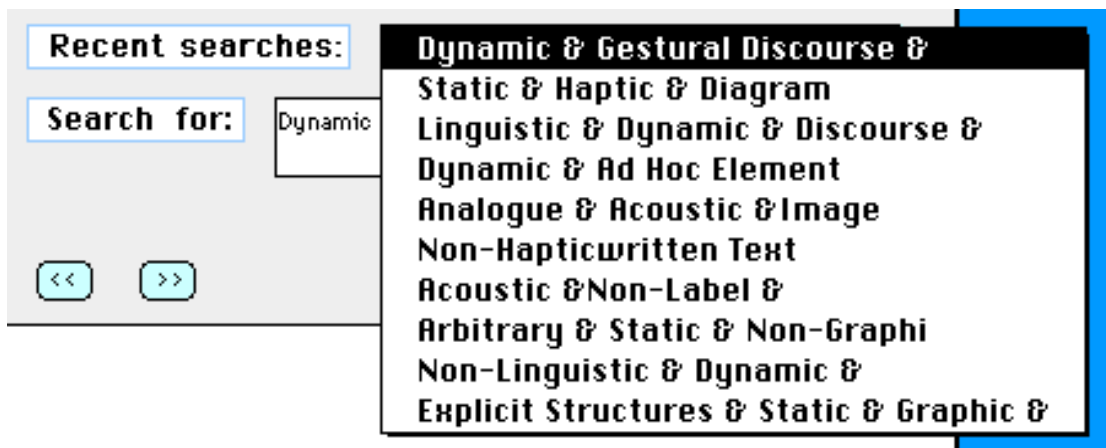


Figure 11. The Recent searches pull-down list.

4.3 How to search

4.3.1 Setting search mode

By default, *Exact match* is set on opening the search application. Users can always switch to the desired search mode by directly selecting it from the *Search modes* pulldown list. If only unimodal representation samples are of interest, *Exact match* should be selected. Otherwise, *Inclusive* is selected.

4.3.2 Initiating search specification

Search specifications can only be made through the *levels* pulldown menus. To search for objects having a certain property, users only need to choose that property from the menus. There is no need to follow any specific order nor to specify the number of properties in advance. Users may start specification from the *Super level*, *Generic level* or *Atomic level* menu and start search at any time. Whatever is selected from these menus, is presented in the *Search for* message window. For example, if a user would like to specify *Linguistic* and *Discourse*, s/he could either first select *Linguistic* from the *Super level* menu and then *Discourse* from the *Atomic level* menu or vice versa.

There is an interesting feature in the *levels* pulldown menus which is used to block illegitimate combinations of properties. Initially, all the menu items are active. As soon as an item has been chosen, those items which are incompatible with it are disabled. This may be illustrated by the following example. Having chosen *Linguistic* from the *Super level* menu, all items in the *Atomic level* menu which are incompatible with *Linguistic* are disabled (cf. the grey items in Fig. 12). This mechanism leaves users with only valid combinations to select from.

4.3.3 Specifying negational search (optional)

In addition to search by specified properties, users may search for objects which lack certain properties. To specify a negational search, hold down the *Apple* key while choosing an item from a pulldown menu. Negated properties can be combined with other negated properties as well as with positive properties. Consider the following example. Having selected *Linguistic* and *Static* from the *Super level* and *Generic level* menus, respectively, the user holds down the *Apple* key while selecting *Text* from the *Atomic level* menu. In effect, *Linguistic & Static & Non-Text* has been specified. The implication of the search specification is the following. In *Exact match* mode, all those unimodal objects which are linguistic, static and non-textual will be matched, i.e. those unimodal objects which belong to one of the following categories: the static discourse, label/keyword, and notation modalities. In *Inclusive* mode, all those objects (whether unimodal or multimodal) having a constituent modality that is linguistic, static and non-textual will be matched.

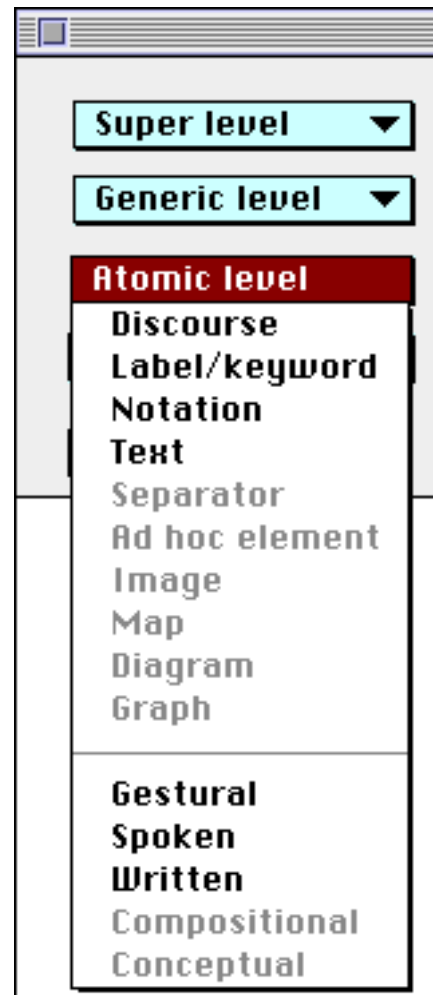


Figure 12. Items incompatible with *Linguistic*

4.3.4 Modifying a search specification (optional)

A search specification is normally modifiable provided that the substituting items are not disabled in the pulldown menus. In this case, modification is simply done by re-selecting the intended items from these menus. Re-selection will cancel the previous selection. If the substituting items are disabled, however, the search specification has to be canceled by pressing the *Cancel* push button. Then steps 4.3.2 and/or 4.3.3 above should be repeated.

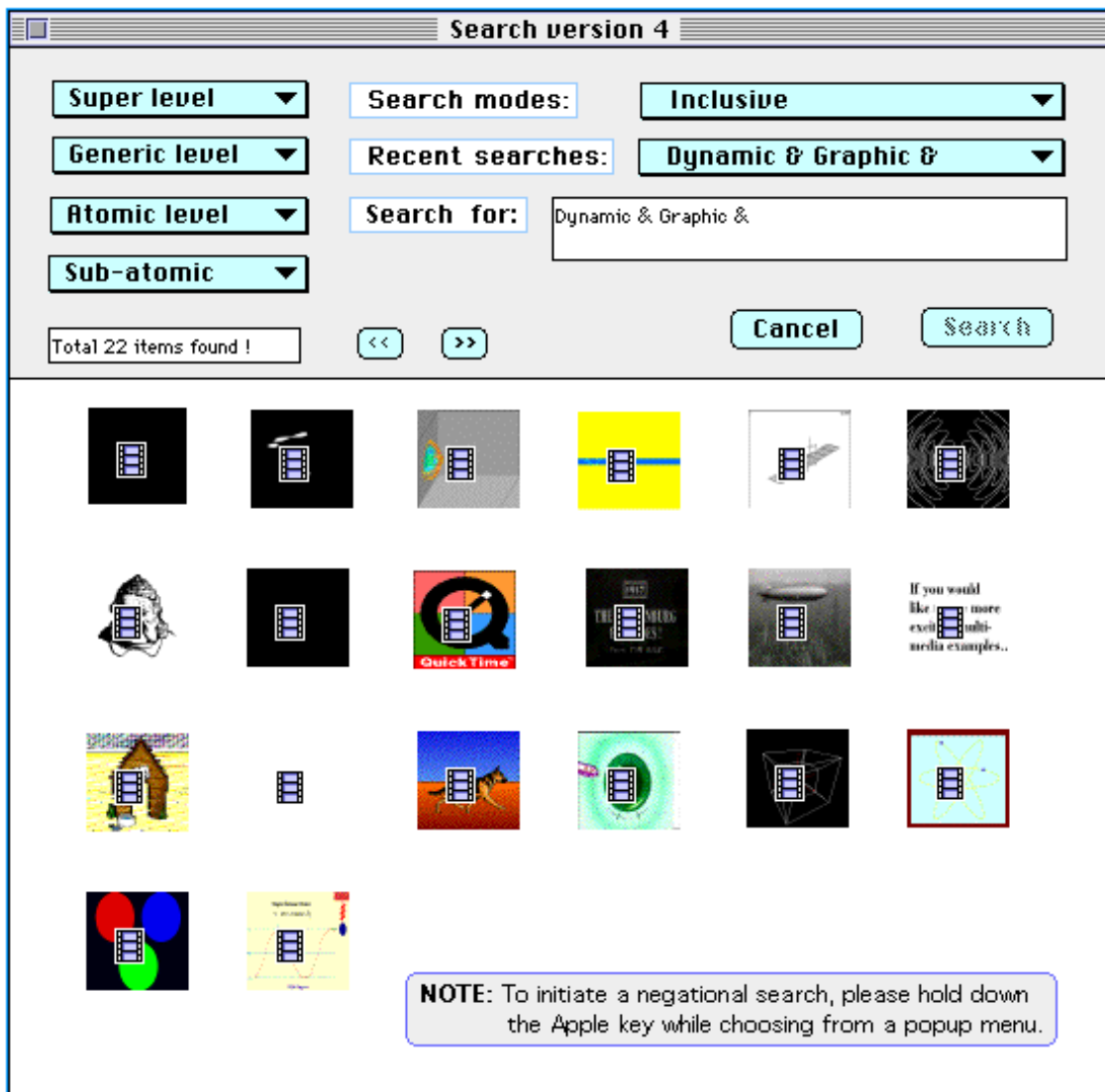


Figure 13. An example search result.

4.3.5 Initiating search

Once a search has been specified, it can be carried out immediately by clicking the *Search* button. When this button is clicked, four things will happen: (a) search begins according to the search specification shown in the *Search for* message window; (b) the search specification is copied to the *Recent searches* message window; (c) the pulldown menus are restored to their initial states; and (d) the result of the search is displayed.

4.3.6 Browsing through the search result list (optional)

Immediately after a search has been executed, the number of items found and the first 20 items (if more than 20 items were found) are iconically displayed (see Fig. 13). If more than 20 items were found, users may use the *Increment* and *Decrement* buttons to browse through the iconic list in steps of 20 items. The *Increment* and *Decrement* buttons are only enabled when an appropriate action can be done through them. Both buttons are disabled before initiating search. The *Increment* button is enabled when search has resulted in more than 20 matched objects. Having browsed through to the end of the result list, the *Increment* button will be disabled. The *Decrement* button is disabled when the first item in the result list is being displayed in the display area and otherwise enabled.

4.3.7 Inspecting objects

An object iconically listed in the display area may be inspected in detail by transferring it to the *Inspection* window (see Fig. 14). This is done by mouse-clicking on the item. An object in the *Inspection* window is re-sized to fill the screen so that minute details are clearly visible. Acoustic and haptic objects as well as multimodal objects including acoustics and haptics will similarly be presented on clicking their icons.

4.3.8 Initiating search from the Recent search pulldown list

Any one among the last 10 search specifications can be re-initiated directly from the *Recent search* pulldown list by selecting it. One should be aware, however, that the search mode notified in the *Search modes* pulldown list takes effect.

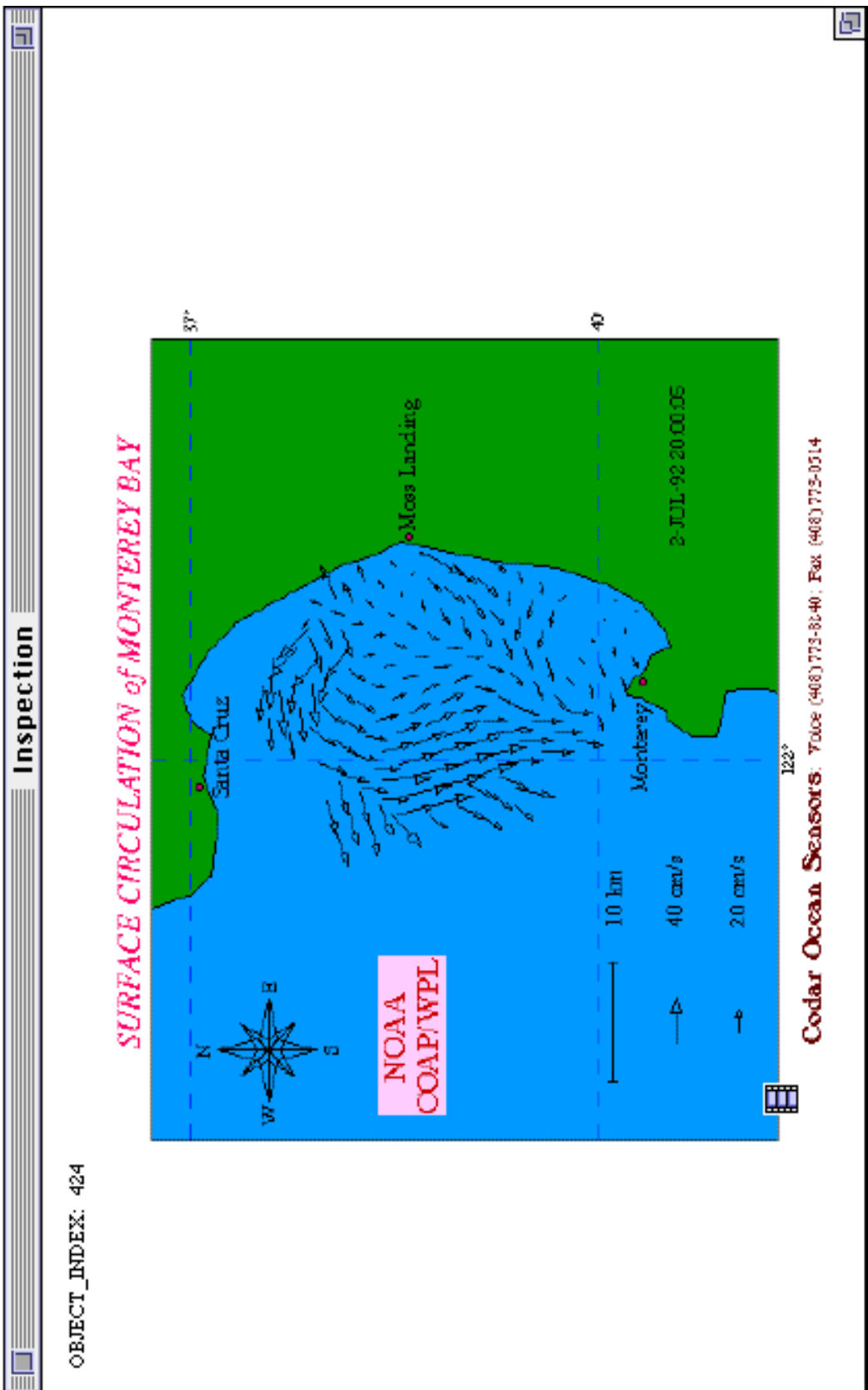


Figure 14. A search result displayed in the Inspection window.

5. Conclusion and Future Work

As remarked in the introduction, it remains an open question to what extent the current version of the workbench, when completed, will need to be further re-designed in order to function as a design support tool. A key question concerns automation. The Information Mapping Methodology assumes that practical information mapping is done in two broad iterative phases (Bernsen 1994a). In the first phase, information is collected and succinctly represented on the interface requirements to do with the information to be represented and exchanged between user and system during task performance on the artifact to be designed. In the second phase, this information is ‘put through’ a design tool based on Modality Theory, which will map the information onto a set of input/output modalities which could optimise the interface to the artifact. The workbench should be developed into such a tool. An apparent solution would be to fully automatize the workbench by developing its current set of information mapping rules into an expert system rule set which could support interface design at any level of detail. However, a recent case study (Verjans and Bernsen 1994) confirms our hypothesis that this would be infeasible. The real world of IT artifacts and their various work domains, tasks to be supported, user types, etc. is quite simply too complex and unmanageable to make such an endeavour a realistic one. At the opposite extreme, the workbench should not be automatized at all but make its information easily accessible to interface designers who would use their ‘natural intelligence’ to let the information constrain their design decisions. Furthermore, the workbench information should be perfected down to a certain level of detail only, leaving the lowest levels of interface design detail to designer craft skills.

We consider the latter of the options just mentioned to be feasible and prudent and propose to explore it further together with the issues it raises. For instance, might the expert systems approach still be feasible at higher levels of abstraction? And, whether or not this be the case, to what extent might such a system be made useful to interface designers and benefit their work? How to integrate *Input Modality Theory* into the workbench described in this paper? Issues such as these will be addressed in parallel with the completion of Version 2 of the workbench.

References

- Bernsen, N.O. (1993a): A research agenda for modality theory. In Cox, R., Petre, M., Brna, P. and Lee, J., Eds. *Proceedings of the Workshop on Graphical Representations, Reasoning and Communication*. World Conference on Artificial Intelligence in Education, Edinburgh, August, 43-46.
- Bernsen, N.O. (1993b): Matching Information and Interface Modalities. An Example Study. *Esprit Basic Research project GRACE Deliverable 2.1.1*.
- Bernsen, N.O. (1994a): Modality Theory in support of multimodal interface design. In *Proceedings of the AAAI Spring Symposium on Intelligent Multi-Media Multi-Modal Systems*, Stanford, March.
- Bernsen, N.O., (1994b): Foundations of multimodal representations. A taxonomy of representational modalities. To appear in *Interacting with Computers*.

Bernsen, N.O. and Bertels, A. (1993): A methodology for mapping information from task domains to interactive modalities. *Esprit Basic Research project GRACE Deliverable 10.1.3.*

May, M. (1993a): Representations and homomorphisms. A taxonomy of representations for HCI, part 1. *Esprit Basic Research project GRACE Deliverable 2.1.5.*

May, M. (1993b): Levels of representations and mappings. A taxonomy of representations for HCI, part 2. *Esprit Basic Research project GRACE Deliverable 2.1.6.*

May, M. (1993c): From semantic types to multimodal presentations: Case-studies. A taxonomy of representations for HCI, part 3. *Esprit Basic Research project GRACE Deliverable 2.1.7.*

May, M. (1993d): Taxonomy and levels of mappings in the construction of multimodal interfaces. *Workshop Notes from AI-ED '93: World Conference on Artificial Intelligence in Education, Edinburgh, August. Workshop on Graphical Representations, Reasoning and Communication.*

May, M. (1994): A Taxonomy of Analogue Representational Modalities. *Esprit Basic Research project AMODEUS Working Paper.*

May, M. and Bernsen, N.O. (1993): The Taxonomy Workbench (software). *Esprit Basic Research project GRACE Deliverable 2.1.8.*

May, M. and Tobin, R. (1993): The Taxonomy Workbench. Purpose, functionality and layout. *Esprit Basic Research project GRACE Deliverable 2.1.9.*

Verjans, S. (1994): EuroCODE: Preliminary information mapping analysis of an AMODEUS common exemplar. *Esprit Basic Research project AMODEUS Working Paper.*

Verjans, S. and Bernsen, N.O. (1994): PaTerm: A Case Study in Information-Mapping. *Esprit Basic Research project AMODEUS Working Paper.*

Acknowledgements. The work reported in this paper was done on grants from the Esprit Basic Research projects GRACE (1992-1993) and AMODEUS II and the Danish Research Council for the Natural Sciences whose support is gratefully acknowledged.

Appendix A.

List of Document Entries in the Theory Demonstrator

(AL = Atomic Level, GL = Generic Level, SL = Super Level,
MM = Michael May, NOB = Niels Ole Bernsen)

Acoustic compositional diagrams (AL) MM
 Acoustic conceptual diagrams (AL) MM
 Acoustic graphs (AL) MM
 Acoustic images (AL) MM
 Acoustic maps (AL) MM
 Acoustic separators (AL) NOB
 Acoustics NOB
 Ad hoc acoustic elements (AL) NOB
 Ad hoc dynamic graphic elements (AL) NOB
 Ad hoc elements NOB
 Ad hoc haptic elements (AL) NOB
 Ad hoc static graphic elements (AL) NOB
 Analogue acoustic language (GL) NOB
 Analogue acoustics (GL) NOB MM
 Analogue dynamic graphics (GL) NOB MM
 Analogue haptic language (GL) NOB
 Analogue haptics (GL) NOB MM
 Analogue modalities (SL) NOB MM
 Analogue static graphics (GL) NOB MM
 Annotation NOB
 Arbitrary acoustics (GL) NOB
 Arbitrary dynamic graphics (GL) NOB
 Arbitrary haptics (GL) NOB
 Arbitrary modalities (SL) NOB
 Arbitrary static graphics (GL) NOB
 Atomic level NOB

Basic Property NOB

Combinatorial analysis NOB
 Compositional diagrams MM
 Conceptual diagrams MM

Degree of resolution NOB
 Diagrams MM
 Dimensionality NOB
 Dimensionality reduction NOB
 Discourse NOB
 Dynamic acoustics NOB
 Dynamic analogue graphic language (GL) NOB
 Dynamic graphic compositional diagrams (AL) MM
 Dynamic graphic conceptual diagrams (AL) MM
 Dynamic graphic graphs (AL) MM

Dynamic graphic images (AL) MM
Dynamic graphic maps (AL) MM
Dynamic graphic separators (AL) NOB
Dynamic graphics NOB
Dynamic haptics NOB
Dynamic modalities NOB
Dynamic non-analogue graphic language (GL) NOB
Dynamic written labels/keywords (AL) NOB
Dynamic written notation (AL) NOB
Dynamic written text (AL) NOB

Explicit acoustic structures (GL) NOB
Explicit dynamic graphic structures (GL) NOB
Explicit haptic structures (GL) NOB
Explicit static graphic structures (GL) NOB
Explicit structure modalities (SL) NOB

Focus NOB
Freedom of perceptual inspection NOB

Generation NOB
Generic level NOB
Gestural discourse (AL) NOB
Gestural labels/keywords (AL) NOB
Gestural notation (AL) NOB
Graphic spoken discourse (AL) NOB
Graphic spoken labels/keywords (AL) NOB
Graphic spoken notation (AL) NOB
Graphics NOB
Graphs MM
Groups NOB

Haptic compositional diagrams (AL) MM
Haptic conceptual diagrams (AL) MM
Haptic graphs (AL) MM
Haptic images (AL) MM
Haptic labels/keywords (AL) NOB
Haptic maps (AL) MM
Haptic notation (AL) NOB
Haptic separators (AL) NOB
Haptic text (AL) NOB
Haptics NOB

Icons NOB
Illustration NOB
Images MM
Independence NOB
Information channels NOB
Information-mapping rules NOB

Inheritance NOB
 Input modality NOB
 Interpretational scope NOB

Labels/keywords NOB
 Level NOB
 Level of abstraction NOB
 Linguistic modalities (SL) NOB
 Lists NOB

Maps MM
 Medium NOB
 Modality NOB
 Multimodal modalities NOB

Non-analogue acoustic language (GL) NOB
 Non-analogue haptic language (GL) NOB
 Non-analogue modalities NOB
 Non-arbitrary modalities NOB
 Non-explicit-structure modalities NOB
 Notation NOB

Operations NOB
 Output modality NOB

Profile NOB

Saliency NOB
 Separators NOB
 Simplicity NOB
 Specificity NOB
 Spoken discourse (AL) NOB
 Spoken labels/keywords (AL) NOB
 Spoken notation (AL) NOB
 Static acoustics NOB
 Static analogue graphic language (GL) NOB
 Static graphic compositional diagrams (AL) MM
 Static graphic conceptual diagrams (AL) MM
 Static graphic graphs (AL) MM
 Static graphic images (AL) MM
 Static graphic maps (AL) MM
 Static graphic separators (AL) NOB
 Static graphics NOB
 Static graphics NOB
 Static haptics NOB
 Static modalities NOB
 Static non-analogue graphic language (GL) NOB
 Static written labels/keywords (AL) NOB
 Static written notation (AL) NOB

Static written text (AL) NOB
Sub-atomic level MM, NOB
Super Level NOB

Tables NOB
Taxonomy NOB
Text NOB

Unimodal modalities NOB