Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

CODE4: Applications for Managing Classification Schemes

Lynne Bowker
Centre for Computational Linguistics
University of Manchester Institute of Science and Technology (UMIST)
P.O. Box 88, Sackville Street
Manchester
M60 1QD United Kingdom
lynne@ccl.umist.ac.uk
bowker@csi.uottawa.ca

Timothy C. Lethbridge
Artificial Intelligence Laboratory
Department of Computer Science
University of Ottawa
150 Louis Pasteur
Ottawa, Ontario
Canada K1N 6N5
tcl@csi.uottawa.ca

0 INTRODUCTION

Classification research is a well-established area of investigation which is integral to many disciplines, including Linguistics, Philosophy, Biology, Information Science, Cognitive Psychology and Computer Science. To be truly effective, classification schemes in any discipline must be carefully managed; however, since these schemes can often be complex and unwieldy, careful management is not always easy.

Our paper discusses a software tool called CODE4 (*Conceptually Oriented Design/Description Environment, version 4*), which is a general purpose *knowledge management system* (KMS) designed to assist any user who would like to create, modify, store or retrieve (i.e., *manage*) knowledge in the form of networks of interrelated concepts.

This paper will be divided into three main sections. The first section will give an overview of CODE4, and discuss some of the general design principles of the system. Section 2 will review the various types of display formats and features offered by CODE4. Finally, Section 3 will focus on those features of CODE4which make it highly conducive to managing classification schemes. Examples are taken from a knowledge base in the subject field of *optical storage technology* that was created using CODE4.

1 OVERVIEW OF CODE4

CODE4 is a KMS intended to help users create and manipulate knowledge in the form of concept networks. KMSs are primarily being developed by researchers in a subfield of Artificial Intelligence (AI) known as Knowledge Engineering. CODE4 is the current version of a KMS that has been under development at the Artificial Intelligence Laboratory of the University of Ottawa, Canada, since 1987. It is programmed in ObjectWorks/Smalltalk and runs on a Unix, Macintosh or 386 platform (8M of RAM required).

ISSN: 2324-9773

Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

CODE4 differs from other KMSs in two main ways [Skuce and Lethbridge 1994:2]: 1) it is generic by design, and 2) it can be used by non-computer experts. While some KMSs are only able to handle special-purpose knowledge representations, CODE4 (and its predecessors) have been used for knowledge management in such diverse disciplines as Terminology, Software Engineering, Technical Documentation, Philosophy and Metallurgy. Although CODE4 is designed as a generic KMS, it *can* be customized for specialized applications if desired. As well as being generic, CODE4 is very user-friendly. Although some KMSs require users to understand computer languages and mathematical techniques, CODE4 features a very versatile yet carefully thought out user interface that allows inexperienced users and non-computer experts to manage knowledge quickly and easily. In fact, the main design objective of CODE4 is to create a practical tool for the effective display and organization of various types of knowledge.

2 KNOWLEDGE DISPLAY AND MANIPULATION IN CODE4

Users can display and manipulate knowledge in several different formats within CODE4. Conditions (in the form of *masks*) can be applied to restrict the information seen at any one time.

2.1 Browsers

In CODE4, knowledge is entered, displayed and manipulated in windows called *browsers,* which provide users with a great deal of power and flexibility for accessing and viewing this knowledge. All browsers can be edited, scrolled, resized and repositioned on the screen at any time. An unlimited number of browsers may be open at one time; a new browser may be opened as a separate window, or as a new pane in an existing window. Changes made in one browser are *automatically* updated in related browsers, and all types of browsers can be dynamically chained together so that the selection of an item in one browser (referred to as the *driving browser*) determines the contents of another (referred to as the *driven browser*).

Browsers permit direct manipulation of nodes (i.e., concepts) and links (i.e., relationships between concepts). Commands for adding, locating, modifying, rearranging and deleting knowledge in the browsers can be selected from mouse-driven menus or activated using hot-keys. There are three different types of CODE4 browsers: outline browsers, graphical browsers, and property comparison matrix (PCM) browsers.

2.1.1 Outline browsers. Outline or textual browsers, such as the one shown in Figure 1, are generally the most commonly used CODE4 browsers. Knowledge is displayed here as lines of text; hierarchical relations between concepts are shown by indentation. Users may restrict the information shown in the list either by "hiding" subhierarchies found in the main list, by indicating that only a specified subhierarchy should appear within the list, or by opening another window showing a subtree of the main hierarchy. Concepts can also be displayed in flat alphabetical lists if desired.

2.1.2 Graphical browsers. In our experience, graphical displays have proved exceptionally useful when dealing with classification issues. CODE4 has highly developed graphical browsers which display knowledge in the form of concept networks composed of nodes and links, such as that shown in Figure 2. Restricted portions of a concept network can be seen by

ISSN: 2324-9773

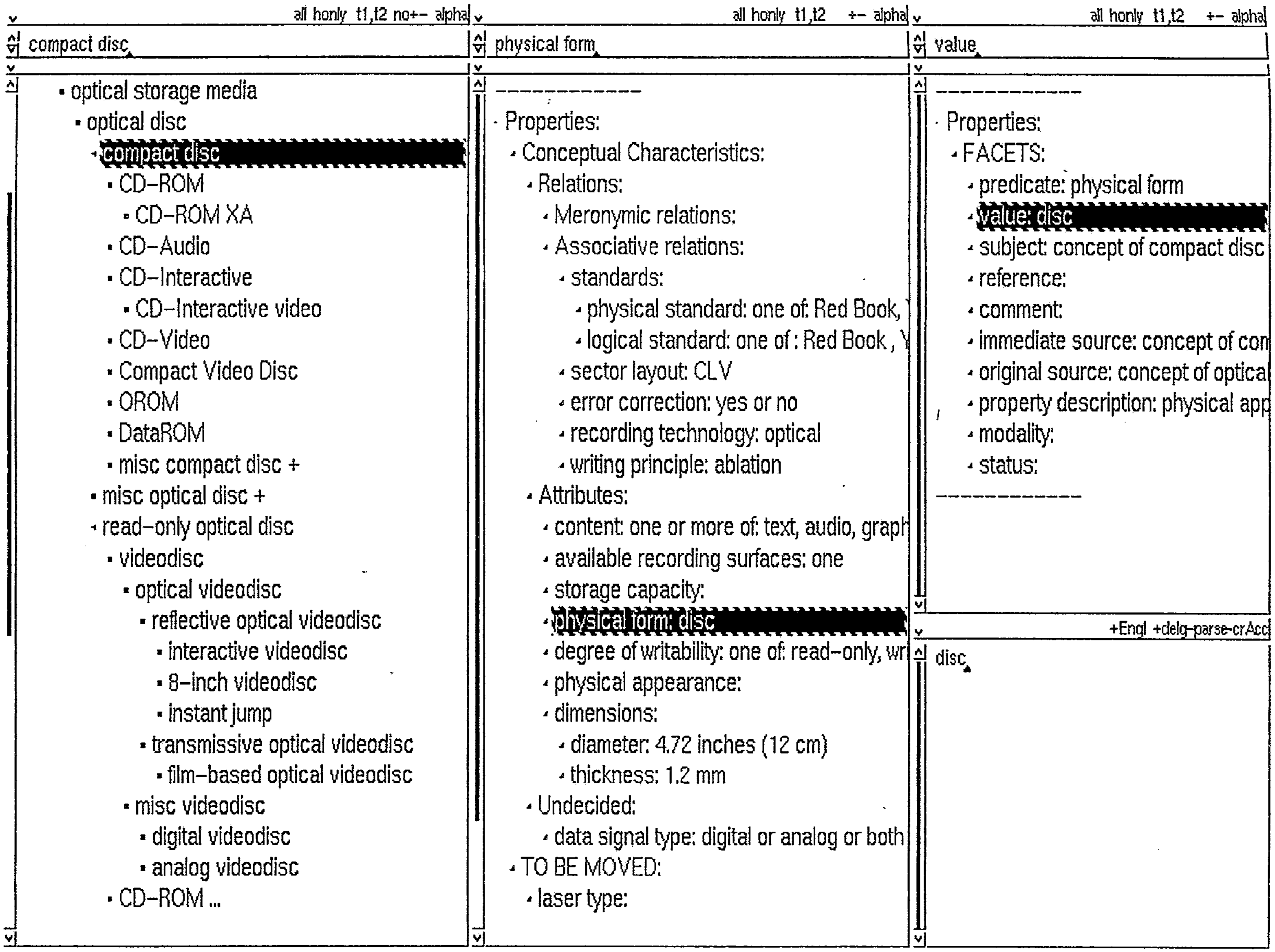

Figure 1. A CODE4 outline browser showing concepts, properties and facets. Hierarchical relations are shown by indentation.

ISSN: 2324-9773

Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

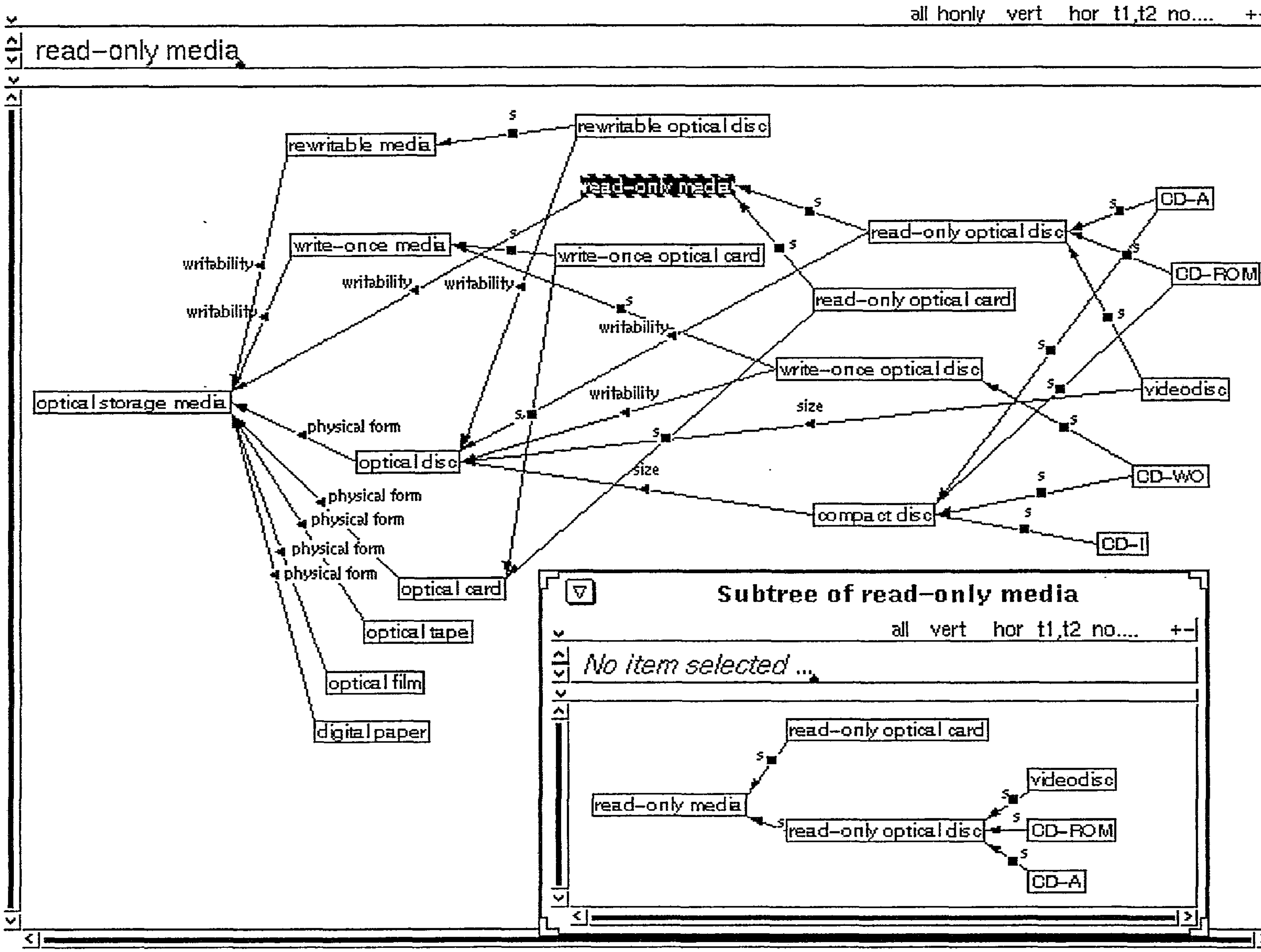

Figure 2. A CODE4 graphical browser with a subgraph.

ISSN: 2324-9773

Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

opening a subgraph, or by hiding or restricting parts of the hierarchy as discussed above. Graphs are generally created showing generic-specific (i.e., *is-a*) relations, and are capable of displaying multiple inheritance (cf. Section 3.6.2). Graphs showing partitive and associative (i.e., *non-hierarchical*) relations can be viewed in separate windows as subgraphs of a main generic- specific graph. The nodes and links in the graphical browsers are generated automatically by CODE4, but the layout can easily be rearranged and new layouts saved.

2.1.3 Property comparison matrix (PCM) browsers. PCM browsers, such as the one shown in Figure 3, facilitate comparisons between two or more concepts. Knowledge is displayed in a series of rows and columns, with concept names appearing along the horizontal axis, and property names along the vertical axis. The intersecting cells show the value of a property for a given concept. The PCM browser can be used to show a comparison of any selected properties for any set of concepts.

optical disc: (writability)	read-only optical disc	rewritable disc	WORM disc
physical standard		(ISO, ANSI)	
logical standard		(ISO, ANSI)	no universally accepted standard
error correction	yes or no	yes or no	yes: Reed-Solomon protocol
diameter	3.5, 4.72 (12 cm), 5.25 (13 cm), 8, 12, or 14 inches	3.5 or 5.25 inches (13 cm)	3.5, 4.72 (12 cm), 5.25 (13 cm), 8, 12, or 14 inches
writing principle	ablation	one of: polarity reversal, phase change, ablation, bubble formation	one of: phase change, ablation, bubble formation
available recording surfac	one of: one, two	one of: one, two	two
storage capacity		128 MB or more	200MB - 10GB
degree of writability	read-only	erasable	write-once

Figure 3. A CODE4 property comparison matrix browser comparing READ-ONLY OPTICAL DISC and its co-ordinate concepts.

ISSN: 2324-9773

Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

2.2 Masks

A mask is a mechanism that allows CODE4 users to select a subset of concepts from a larger set according to specified criteria (cf. Table 1). Masks can be applied to any type of browser.

has a name matching the string:
has a property value matching the string:
has any term matching the string:
has empty property value
has facet = value:
has metaconcept property = value:
has property = value:
inherits all of the properties:
inherits any of the properties whose name matches the string:
inherits any of the properties:
inherits to all of:
is a descendent of (or equal to) one of:
is a system concept (term, statement or metaconcept)
is an instance
is in any of dimension(s):
is in any of inherited dimension(s):
is in the hierarchy of any of:
is included in the set:
is root of or in any of inherited dimensions:
is the property of an inherited dimension
is the property of the dimension

Table 1. Selection criteria predicates for CODE4 masks.

To apply a mask, the user calls up the mask function from the menu and then selects a predicate and specifies the desired value. Existing CODE4 predicates are shown in Table 1, but new predicates can be added. Predicates can also be negated to get the opposite effect. When applied, the mask is either "true" or "false" for each concept. The set of conditions specified in a mask is often very simple (e.g. show only those concepts that have the property 'writability'); however, complex masks combining several conditions can also be created.

Masks can be used to focus the display or reduce the amount of information shown at one time, for example to show only one subtree. Masks can also be used for search and retrieval tasks, for example, as shown in Figure 4, a user could request CODE4 to show only those concepts having the value 'disc' for the property 'physical form'. There are two types of masks: visibility masks and selection masks.

ISSN: 2324-9773

Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

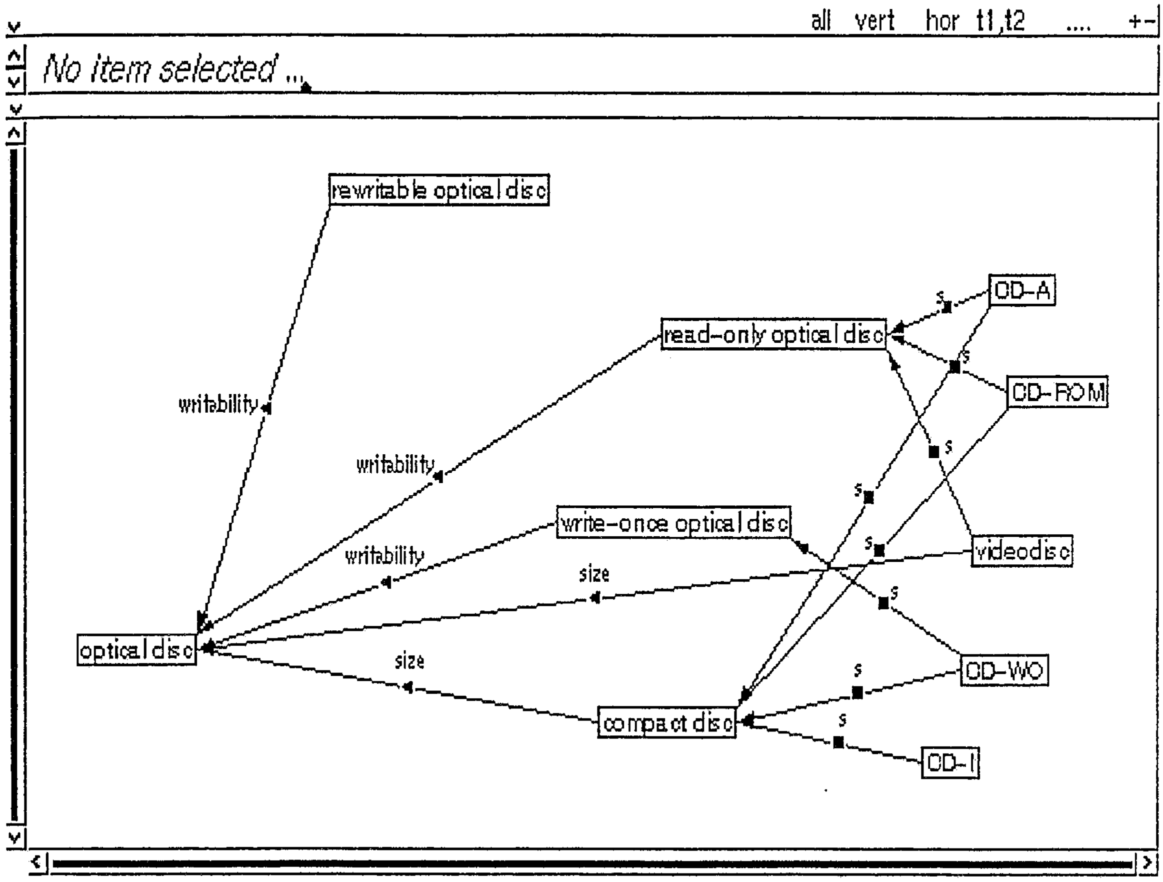

Figure 4. A CODE4 visibility mask applied to a graphical browser. Only those concepts having the value 'disc' for the property 'physical form' are displayed.

2.2.1 Visibility masks. This type of mask determines whether the concept will be displayed ("true") or hidden ("false"). The default visibility mask displays all concepts. Concepts remain hidden until the mask is turned off.

2.2.2 Selection masks. This type of mask determines whether a concept will be highlighted/underlined ("true") or not ("false"). The default selection mask highlights/underlines no concept. If highlighting is chosen, the concepts remain highlighted only until another concept is selected. If underlining is chosen, the concepts remain underlined until the underlining feature is toggled off.

ISSN: 2324-9773

Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

3 KNOWLEDGE ORGANIZATION AND CLASSIFICATION FEATURES IN CODE4

With CODE4, users create knowledge bases (KBs), which consist of collections of related concepts, and properties which describe these concepts. CODE4 incorporates many organizational principles and features that are very conducive to managing classification schemes. In this section, we will endeavour to discuss CODE4's knowledge organization and classification features in a general light; however, some examples may reflect specializations made for managing classification schemes in Terminology work as this is the application with which we are most familiar.

3.1 Inheritance Hierarchies and Consistency Checking Mechanisms

In concept networks dominated by generic-specific relations, concepts are arranged in *inheritance* or *is-a hierarchies*. In a CODE4 inheritance hierarchy, the generic concepts are called *superconcepts,* and they are classified into successive levels of specific concepts called *subconcepts,*[1] as seen in Figures 1 and 2. Two subconcepts sharing the same superconcept are called *siblings*. Inheritance is a powerful AI technique that allows for the properties of any given concept to be implicitly true for all subconcepts of this concept, for all subconcepts of these subconcepts, etc. CODE4 also supports inheritance from multiple superconcepts (cf. section 3.6.2). Coupled with CODE4's inheritance mechanisms are mechanisms for detecting inconsistencies. For example, if a user tries to reparent a concept such that its new parent is one of its descendants, a feedback panel will pop up on the screen advising the user of this error.

3.2 Properties

Associated with each concept is a set of properties which describe it. CODE4 does not presuppose any particular theory of concepts, so users can create any number and type of properties. All properties are explicitly recorded in an easy-to-read slot-filler format where the slot is the **property name** and the filler is the ***property value*** (e.g. **physical form:** ***disc***).

In CODE4, properties can also be arranged in a hierarchy, which is distinct from the inheritance hierarchy. There is a single top property, and users typically create several levels of subproperties, as shown in Figure 1. This allows users, for example, to separate Attributes from Relations, and to separate Meronymic Relations from Associative Relations, etc.

As previously mentioned, properties are inherited from superconcepts to subconcepts. As concepts become more specialized, the value of the property may either remain the same, or undergo modification/specialization. New properties may also be added at each level of the classification. In some special cases, a concept may have a property which should not inherit to its subconcepts so CODE4 also allows for there presentation of non-inheriting properties.

1. The identification of a concept as being a superconcept or subconcept is relative; a subconcept at one level of the classification can be a superconcept at another level.

ISSN: 2324-9773

3.3 Facets

Often it is not enough to merely specify the property of a concept, so CODE4 allows users to attach *facets*[1] to properties. One way of looking at a facet is to think of it as a property of a property. For example, all properties could have a facet called 'comment'.

3.4 Linguistic Knowledge

Because terminology is highly important in the representation of knowledge, CODE4 has features for representing linguistic knowledge. A single concept can be represented by multiple terms (*synonymy*), such as "CD-ROM" and "Compact Disc Read-Only Memory". One term is chosen as the preferred term and is used to actually designate the concept, while the other term(s) is(are) recorded as synonyms in the linguistic information window, as shown in Figure 5. CODE4 also

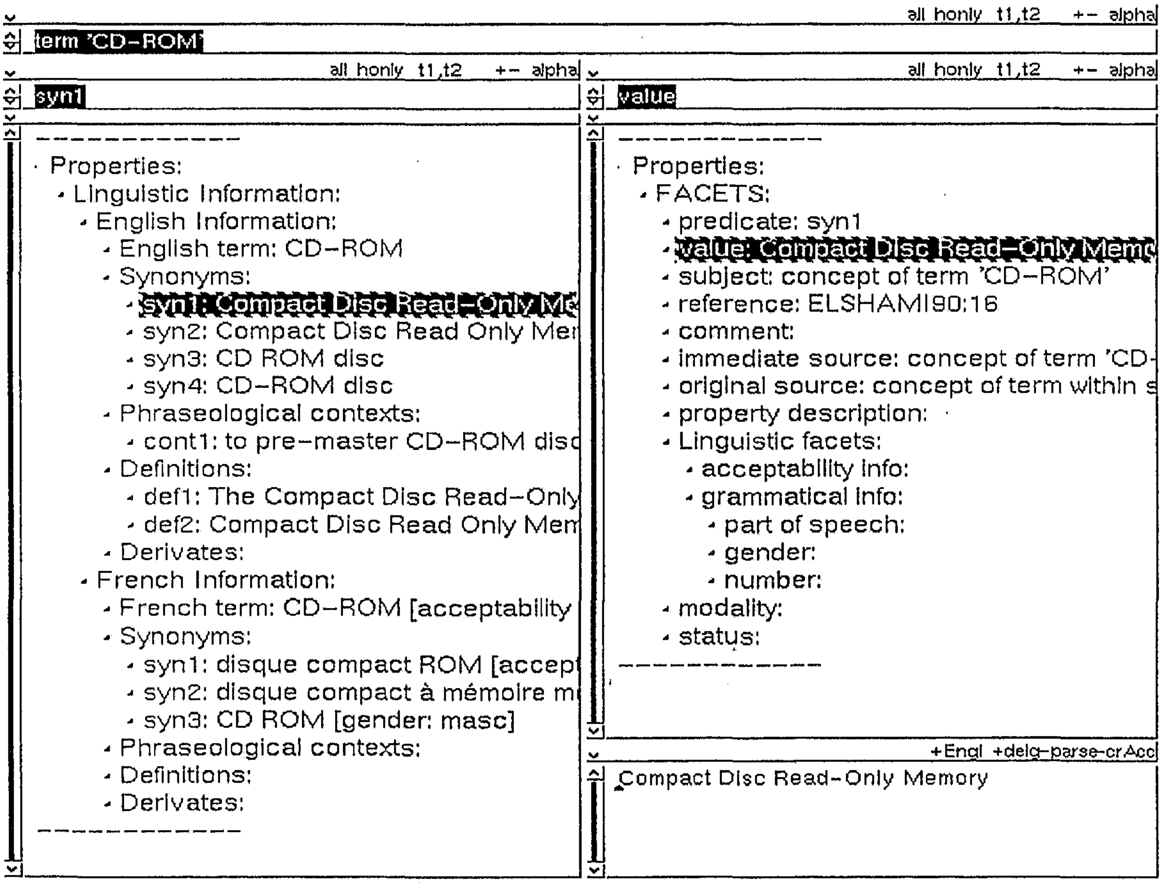

Figure 5. A CODE4 linguistic information window.

1. The use of the term *facet*, which is standard in Artificial Intelligence, should not be confused with the idea of facet as it is used in Information Science (we use the term *dimension* to represent this meaning, cf. section 3.6).

ISSN: 2324-9773

Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

allows a single term to represent several different concepts (*polysemy*) if necessary, for example "Mercury" the mythological god, "Mercury" the planet, "Mercury" the telecommunications firm, etc. This is possible because every time a term is given to a concept CODE4 creates a new instance which has a property called 'meanings'. If the term already exists, an addition is made to its 'meanings' property since it is now the name of more than one concept. Terms can have such linguistic properties as part-of-speech, number, gender, derivates, collocations, contexts, definitions, foreign language equivalents, etc.

3.5 Metaknowledge

Users may wish to represent knowledge *about* the knowledge itself. CODE4 provides facilities for maintaining administrative knowledge, such as reference sources for the knowledge, who entered the knowledge, date of entry, etc. This information is stored as *metaconcept properties*, and some of it is generated automatically by CODE4 (e.g. 'last time changed').

3.6 Support for Multidimensionality

CODE4 has many features which support a particular phenomenon of classification which we refer to as *multidimensionality* [Bowker1992]. Multidimensionality occurs when a concept can be classified in more than one way, based on different properties that it may have. For example, as shown in Figure 2, the concept OPTICAL STORAGE MEDIA can be classified according to the property 'writability' into the subconcepts READ-ONLY MEDIA, WRITE-ONCE MEDIA and REWRITABLE MEDIA. However, it can *also* be classified according to the property 'physical form' into the subconcepts OPTICAL DISC, OPTICAL TAPE, OPTICAL FILM, OPTICAL CARD and DIGITAL PAPER. Each way of classifying a concept is referred to as a *dimension.* A concept which has been given more than one dimension is said to be *multidimensional.* CODE4 has many features which facilitate the creation and manipulation of multidimensional concept networks.

3.6.1 Metaconcept properties: 'has dimensions', 'dimension' and 'inherited dimension.' CODE4 has three metaconcept properties (cf. Section 3.5) that are important for multidimensional classification: 'has dimensions', 'dimension', and 'inherited dimension'.

Has dimensions. A concept that can be classified in more than one way based on the different properties that it can have will have information filled in under the property 'has dimensions'. For example, the concept OPTICAL STORAGE MEDIA has the value 'writability' and 'physical form'.

Dimension. A top level concept in any dimension (i.e., a concept which is the immediate subconcept of a concept which 'has dimensions') will have information filled in under the property 'dimension'. For example, the concept READ-ONLY MEDIA has the value 'writability', while the concept OPTICAL DISC has the value 'physical form'.

Inherited dimension. Any subconcepts of the top level concept ina dimension (or any of their subconcepts) will have information filled in under the property 'inherited dimension'. For example, the concept COMPACT DISC has the inherited value 'physical form ' from its superconcept OPTICAL DISC.

3.6.2 Multiple inheritance. CODE4 allows a subconcept to inherit properties from two or more superconcepts, even when the superconcepts belong to different dimensions. For example,

ISSN: 2324-9773

Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

as shown in Figure 2, the concept CD-ROM has two superconcepts: READ-ONLY OPTICAL DISC in the dimension based on the property 'writability' and COMPACT DISC in the dimension based on the property 'physical form'.

3.6.3 Simultaneous super-subconcept/sibling relationships. It is also possible for two concepts to be simultaneously involved in both a super-subconcept *and* a sibling relationship. For example, the concept OPTICAL DISC can be classified into VIDEODISC and COMPACT DISC according to the property 'size', and also into READ-ONLY OPTICAL DISC, WRITE-ONCE OPTICAL DISC and REWRITABLE OPTICAL DISC based on the property 'writability'. In this classification, the concepts VIDEODISC and READ-ONLY OPTICALDISC are siblings; that is, they share the same parent (i.e.,OPTICAL DISC). However, VIDEODISC is also necessarily a subconcept of READ-ONLY OPTICAL DISC since all videodiscs have the value 'read-only' for the property 'writability'. This means that the concepts READ-ONLY OPTICAL DISC and VIDEODISC also have a super-subconcept relationship. The simultaneous super-subconcept/ sibling relationship between these concepts is depicted in Figure 6.

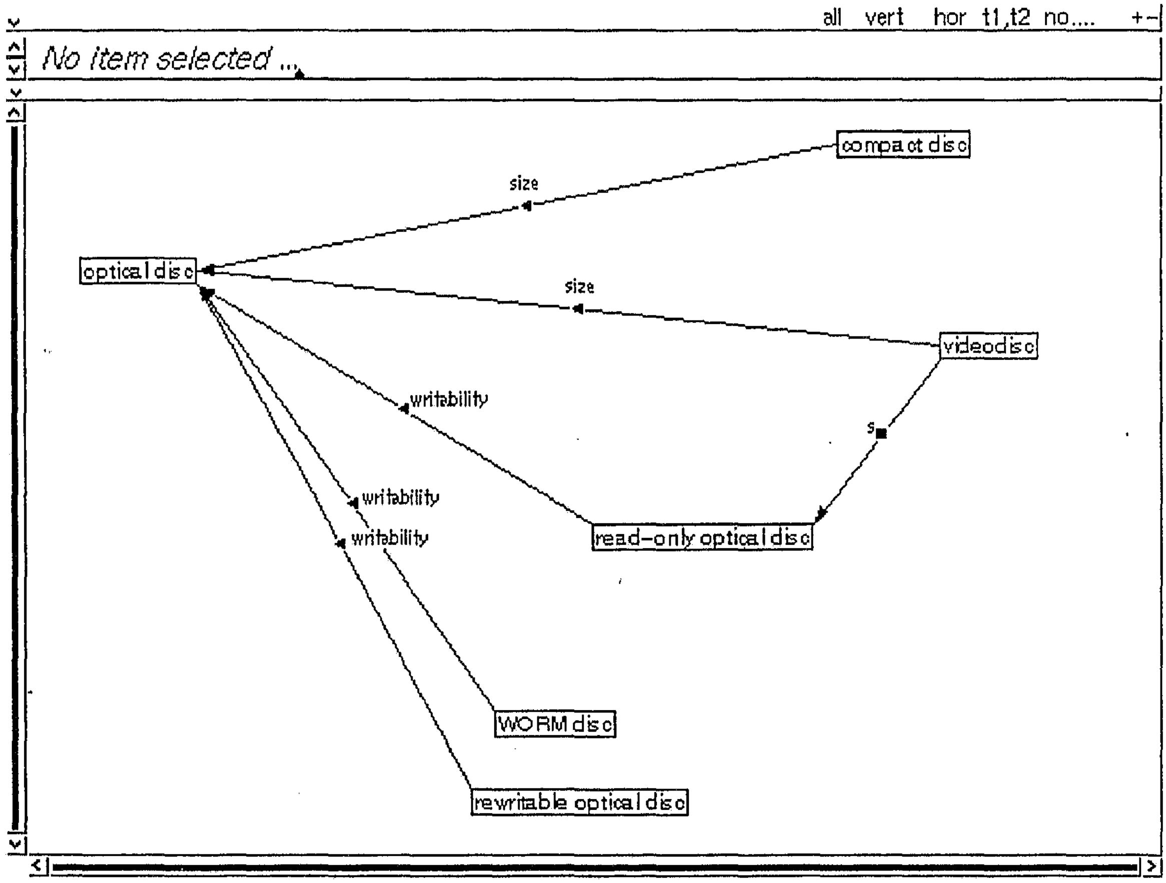

Figure 6. A depiction of the simultaneous super-subconcept/sibling relationship between VIDEODISC and READ-ONLY OPTICAL DISC.

ISSN: 2324-9773

Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

3.6.4 Outline browser display. In an outline browser, as shown in Figure 1, CODE4 distinguishes between the bullets preceding different types of concepts. This allows users to see, at a glance, which concepts belong to a dimension (i.e., those preceded by a triangular-shaped bullet(▼)) and which do not. (i.e., those preceded by a square-shaped bullet (■)). If the user clicks on the bullet of a concept belonging to a dimension, the name of the property underlying the dimension will appear in the editing area at the top of the window.

3.6.5 Graphical browser display. Similarly, in the graphical browser, as shown in Figure 2, the links joining non-dimensional subconcepts to their superconcepts are square in shape (■), while those joining top-level concept sin a dimension to their superconcepts are triangular in shape(▼), and have the name of the property underlying the dimension written beside them.

3.6.6 PCM browser displays. In the PCM browser, users can specify whether they wish to compare *all concepts at a certain level of the hierarchy*, regardless of whether they belong to the same dimension (referred to as *siblings*), or *only those concepts at the same level of the hierarchy within the same dimension* (referred to as *co-ordinate concepts*). When comparing co-ordinate concepts, the PCM lists the various dimensions to which a concept belongs, and the user simply has to click on the particular dimension for which he or she wishes to see the co-ordinates. If the user clicks on another dimension, the co-ordinates that are displayed change accordingly. (Note that users can also compare arbitrarily selected concepts.)

3.6.7 Mask predicates. There are five mask predicates (cf. section 2.2) designed to facilitate the manipulation of multidimensional concept systems.The first three predicates can be applied to concepts, and can be used with either visibility or selection masks (cf. sections 2.2.1 and 2.2.2).

> ***Is in any of dimension(s).*** If this predicate is applied, the user will see/highlight/ underline only those concepts which are at the top level of the specified dimension.
> ***Is in any of inherited dimension(s).*** If this predicate is applied, the user will see/ highlight/underline the top-level concepts in a specified dimension as well as all their subconcepts which belong to that dimension.
> ***Is root of or in any of inherited dimension(s).*** If this predicate is applied, the user will see/highlight/underline the concept which is at the root of a dimension (i.e., that concept which *has* several dimensions, e.g. OPTICAL STORAGE MEDIA) as well as all the subconcepts which belong to the specified dimension.

The remaining two predicates can only be used with selection masks (cf. section 2.2.2), and are applied in the property pane of an outline browser. In order for them to work, the name of the property underlying a dimension must also be specified (exactly) as a conceptual property in the property pane. This feature is dynamic; if the users selects a different concept belonging to a different dimension, the corresponding property will be underlined. The underlining can be toggled on and off.

> ***Is the property of the dimension.*** If this predicate is applied, CODE4 will underline the property which underlies the dimension to which the currently selected top-level dimension concept belongs.

ISSN: 2324-9773

Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

Is the property of an inherited dimension. If this predicate is applied, CODE4 will underline the property which underlies the dimension to which the currently selected concept belongs. If the concept belongs to more than one dimension, the properties which underlie each of the dimensions to which it belongs will be underlined.

3.6.8 Update metaconcept prop set. Another feature that was not designed specifically to facilitate the handling of multidimensionality, but which can be used for this purpose is the

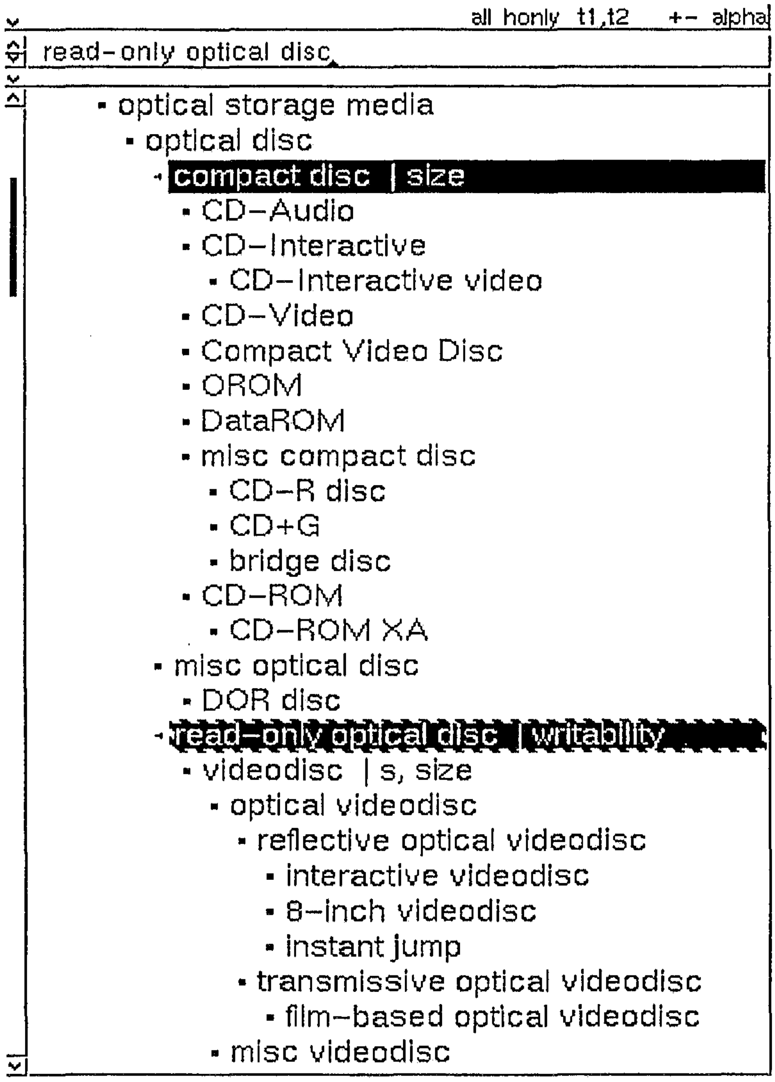

Figure 7. A CODE4 outline browser displaying the metaconcept prop set values for the property 'dimension'.

ISSN: 2324-9773

Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

format>>update metaconcept prop set menu selection. If the user selects this command from the menu, and then clicks on the 'dimensions' property in the metaconcept properties subwindow, every concept in the outline browser is displayed in its usual place in the indented list, but it is followed by a vertical bar (|) and its value(s) for the metaconcept property 'dimension'. This could also be done for the metaconcept properties 'has dimensions' or 'inherited dimensions' if desired. These values remain displayed until the feature is turned off by the user. Figure 7 shows the metaconcept prop set values for the property 'dimensions'.

3.6.9 Facets pertinent to multidimensionality. As discussed in section 3.3, CODE4 allows users to create facets, which can be thought of as properties of a property. Users can create facets that could be relevant to multidimensional classification. For instance, one could create a facet called 'principal user(s)' attached to the property underlying a particular dimension. The value for this facet could state the group for whom a particular dimension has great importance. For example, users may be more inclined to classify OPTICAL STORAGEMEDIA according to the property 'writability', while manufacturers may be more likely to classify OPTICAL STORAGEMEDIA according to the property 'physical form'.

Another possible facet that could be relevant to multidimensionality could be 'frequency'. Again, this facet could be attached to properties underlying dimensions and it could indicate how common or rare a particular dimension happens to be in the literature.

3.6.10 Colour display. For those users fortunate enough to be using colour monitors, CODE4 can be programmed to display the bullets and links in the outline and graphical browsers in different colours. Ordinary concepts could have bullets and links in one colour, while concepts belonging to dimensions could have their bullets and links displayed in other colours.

4 CONCLUDING REMARKS

In this paper, we have endeavoured to show that the potentially complex task of managing classification schemes can be facilitated with the help of knowledge management systems such as CODE4. CODE4 helps users organize knowledge effectively by making use of such powerful features as inheritance, by allowing users to structure knowledge hierarchically and associatively, and by allowing users to consider more than one dimension of a classification. CODE4's display features permit the knowledge to be viewed and manipulated in textual or graphical form, and masks allow users to control the type and amount of information they wish to view at any one time.

Anyone interested in exploring the optical storage technology knowledge base in a running CODE4 system (with a partial interface) can access it via the World Wide Web standard at Uniform Resource Locator (URL):

http://www.csi.uottawa.ca/kaml/CODE4.html

ISSN: 2324-9773

Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

ACKNOWLEDGEMENTS

The research reported here has been supported by the Social Sciences and Humanities Research Council of Canada (SSHRC), the Natural Sciences and Engineering Research Council of Canada(NSERC), and the Committee of Vice Chancellors and Principals (CVCP) of the UK. Several members of the Artificial Intelligence Laboratory of the University of Ottawa, Canada, have contributed to this research in various capacities: K. Eck, K. Iisaka, J.Kavanagh, I. Meyer, and D. Skuce. J. McNaught of the Centre for Computational Linguistics, UMIST, UK, offered comments on an earlier draft of this paper.

REFERENCES

Bowker, Lynne. 1992. *Guidelines for Handling Multidimensionality in a Terminological Knowledge Base*. MA Thesis, University of Ottawa, Canada.

Lethbridge, Timothy C. and Eck, Karen E. 1993. *CODE4.1A/B/CReference Manual.* Ottawa: Artificial Intelligence Laboratory.

Skuce, Douglas and Lethbridge, Timothy C. 1994 (submitted)."CODE4: A Unified Approach to Managing Conceptual Knowledge."*Knowledge Acquisition.*

ISSN: 2324-9773

Bowker, L. (1994). CODE4: Applications for Managing Classification Schemes. 5th ASIS SIG/CR Classification Research Workshop, 17-32. doi:10.7152/acro.v5i1.13775

ISSN: 2324-9773