

21-11-2005

## Introducing N-Tree-Space: A Classification System for Knowledge Library Applications

E. Rayner  
*University of Wollongong*

I. C. Piper  
*University of Wollongong, [ian@uow.edu.au](mailto:ian@uow.edu.au)*

M. Bunder  
*University of Wollongong, [mbunder@uow.edu.au](mailto:mbunder@uow.edu.au)*

Follow this and additional works at: <https://ro.uow.edu.au/infopapers>



Part of the [Physical Sciences and Mathematics Commons](#)

---

### Recommended Citation

Rayner, E.; Piper, I. C.; and Bunder, M.: Introducing N-Tree-Space: A Classification System for Knowledge Library Applications 2005.  
<https://ro.uow.edu.au/infopapers/498>

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: [research-pubs@uow.edu.au](mailto:research-pubs@uow.edu.au)

---

## **Introducing N-Tree-Space: A Classification System for Knowledge Library Applications**

### **Abstract**

This paper identifies and defines the Knowledge Library Application (KLA) class. KLAs are an important class of information resource/knowledge management applications. It will be clear from the definition that KLAs are highly relevant to both the commercial and academic worlds and existing techniques such as keyword and tree classification, On-Line Analytical Processing (OLAP), and Formal Concept Analysis (FCA) do not provide key KLA features. N-Tree-Space (NTS), a technique detailed in this paper, enables the provision of these features and should facilitate the development of a range of KLAs that offer significant advantages over currently available software tools.

### **Disciplines**

Physical Sciences and Mathematics

### **Publication Details**

This article was originally published as: Rayner, E, Piper, I & Bunder, M, Introducing N-Tree-Space: A Classification System for Knowledge Library Applications, IEEE International Region 10 Conference (TENCON 2005), Melbourne, November 21-24 2005, 1-5. Copyright 2005 IEEE.

# Introducing *N*-Tree-Space: A Classification System for Knowledge Library Applications

Eric Rayner

Faculty of Informatics  
University of Wollongong  
Wollongong NSW 2522  
Australia  
Email: ericr@uow.edu.au

Ian Piper

Faculty of Informatics  
University of Wollongong  
Wollongong NSW 2522  
Australia  
Email: ian@uow.edu.au

Martin Bunder

Faculty of Informatics  
University of Wollongong  
Wollongong NSW 2522  
Australia  
Email: mbunder@uow.edu.au

**Abstract**—This paper identifies and defines the Knowledge Library Application (KLA) class. KLAs are an important class of information resource/knowledge management applications. It will be clear from the definition that KLAs are highly relevant to both the commercial and academic worlds and existing techniques such as keyword and tree classification, On-Line Analytical Processing (OLAP), and Formal Concept Analysis (FCA) do not provide key KLA features. *N*-Tree-Space (NTS), a technique detailed in this paper, enables the provision of these features and should facilitate the development of a range of KLAs that offer significant advantages over currently available software tools.

**Index Terms**—Information retrieval, information technology, knowledge engineering, knowledge representation.

## I. INTRODUCTION

In today's knowledge economy, the technology we use to organize and communicate information is critical. Techniques of information organization and retrieval currently in use are based on paradigms that were developed before the information explosion. These techniques do not adequately meet the needs of sophisticated users. *N*-Tree-Space is a new approach that introduces information organization concepts that provide new possibilities in the way we interact with information. This could have profound implications for the way we communicate, educate, and research.

## II. PRELIMINARY DEFINITIONS

### A. Knowledge Libraries

A Knowledge Library (KL) is a set of elements where each element is a moderately complex and cohesive information structure. Some examples of potential KL sets are: academic papers, company reports, product fact sheets, letters, algorithms, chess games, web sites, and encyclopedia entries. Individual elements in a KL are significantly more complex and meaningful than simple numerical data such as the domain of attributes in relational databases. All elements in a particular KL share a number of common properties, for example academic papers all have: at least one author, a publication date, a title, a subject, a number (zero or more) of cited papers, a number (zero or more) of citations, and a text or content.

### B. Knowledge Library Applications

A Knowledge Library Application (KLA) is a software program for the creation, storage, modification, querying, and display of a KL. KLAs provide the following key features.

- 1) KLAs inform users of the content (i.e. the elements) of the library and the relationships that exist between elements in the library.
- 2) KLAs allow users to select a (possibly empty) subset of the library containing elements that satisfy arbitrary requirements specified by the user.

KLAs also provide core functionalities such as allowing users to publish elements to the library and view selected elements. The features above have been identified as *key* because they will be *the* defining characteristics of knowledge library implementations and will determine the library's usability and usefulness.

KLAs are information resource management (IRM) applications. Information resource management has been described as:

the concept that information is a major corporate resource and must be managed using the same basic principles that are used to manage other assets, such as employees, materials, equipment, and financial resources. [1]

Furthermore, if Knowledge Management (KM) is defined as:

...the classification, dissemination, and categorization of information and people throughout an organization [2]

then we can appreciate that KLAs are also knowledge management applications.

Although readers should agree that the KLA class is an important class of applications, to the author's knowledge, no definition of this class and its key features (as given above) has been attempted before.

## III. EXISTING TECHNIQUES

### A. Keyword Classification

The ubiquitous *keyword classification* is a technique that is both flexible, and very easy to implement and use. Library

elements are classified according to one or more keywords. Users simply type in keywords and all elements (stored in a list) that include these keywords are selected.

While keyword classification works well enough on simple domains, the effectiveness of keyword classification in complex technical and scientific domains is linked to the *naming strategies* researchers employ when assigning names to, previously unnamed, entities and qualities. There are two naming strategies available.

First, a researcher may choose to *create new words*. In the context of keyword classification, this strategy has the drawback that it tends to create information ‘pools’ or ‘islands’ of research. In order to discover a research island, one must know the special terminology that is used on that particular island, but if you know the terminology, then you must have already discovered the island. This isolating circularity creates research communities that are unaware of parallel research in related fields.

Second, a researcher may choose to *extend the meaning of existing words*. In the context of keyword classification, this strategy has the drawback in that it creates information ‘swamps’ or ‘infogluts’ [3]. Keyword searches on words that are polysemes (that is, words that have different, but related, meanings) or homonyms (that is, words that have different unrelated meanings) results in an overlarge selection that is only loosely tailored to requirements.

Apart from these significant (and costly) shortcomings, keyword classification fails to provide the first key feature of a KLA in that it does not inform users of the content of the library nor is it effective in showing relationships that exist between elements in the library. This last point is especially telling:

...wit and genius are related processes, for both consist in bringing together ideas from unrelated fields. [4]

### B. Tree Classification

In contrast, the venerable *tree classification* technique defines *parent*, *child*, *ancestor*, and *descendant* relationships between nodes [5]. Users normally navigate from the single *root node* down through the tree towards one of the numerous *leaf nodes* by repeatedly choosing from one of the child nodes, each choice further restricting and defining the set of selected elements.

In general, tree classification is somewhat inflexible in that it only allows a single perspective for both classification and selection. For example if we tree classify organisms according to their evolutionary relationships (as does the ‘tree of life’ [6]) we cannot classify or select animals according to their habitat, or in any other way apart from their evolutionary relationships.

In terms of KLA key features, tree classification may (depending on the implementation) provide users with good information about the contents of the library. Tree classification is also quite good at displaying the relationships that exist between elements (the first key KLA feature) – but only from one point of view. Also, tree classification by definition only

allows for a single path to each node, so it does not allow the user to select elements according to arbitrary specifications (the second key KLA feature).

### C. Formal Concept Analysis (FCA)

FCA, which is succinctly described in [7], provides a way of structuring a collection via *formal concepts* which are represented in a *concept lattice*.

Unfortunately FCA has a number of deficiencies that make it an unsuitable underlying structure for implementing KLAs.

One major failing of FCA in this respect relates to *formal concepts*. Formal concepts represent FCAs method of selecting elements (the second key feature of a KLA). The problem is that the formal concepts that exist may change depending upon which elements are added to the set (or which *objects* are added to the *formal context* in FCA terminology). It is well established that it is undesirable to have this kind of directly causal relationship between the contents, and the structure, of a taxonomy.

For example, in the relational model of data, a *deletion anomaly* occurs when structural information is lost as specific instances are deleted [1].

Indeed, this criticism is relevant to any classification system where the taxonomy is automatically generated. Perhaps this is why discussions of FCA stress that the formal context consists of *fixed* objects and attributes—but, of course, if this is a restriction then FCA can not be used to create a KLA.

Another problem with FCA, and automatic classification systems generally, is that it is often desirable to have taxonomies that conform to established ideas about what classifications should exist. It may be necessary to manipulate ‘automatic’ systems into reflecting these ideas, after which, such systems can no longer be considered to be properly automatic.

Furthermore, for large and complex element sets, as are typical of KLAs, FCA’s representation of formal concepts in a *concept lattice* is not an effective user-interface concept. Generally, there are far too many formal concepts in a typical KLA formal context. Attempting to solve this problem by only displaying formal concepts that select more than some number (perhaps ten) of objects only worsens the *deletion anomaly* problem – with well established concepts seemingly arbitrarily popping out of existence when objects are deleted.

Finally, the usefulness of FCA is necessarily restricted to collections containing objects that can be well described and differentiated from one another by listing attributes. This may not be adequately descriptive for certain types of object, which could be better described using *property-value* combinations. For example, documents in general may have the *property* of *text length* which could have the *value*, for a particular document, of 1000 words. There does not appear to be an elegant way of representing property-value combinations with attributes alone (without introducing property-attributes and value-attributes).

#### D. On-Line Analytical Processing (OLAP)

The term OLAP was introduced by Codd et. al. in 1993 [8]. This paper has been criticized as being

...controversial due to being vendor sponsored rather than mathematically based... The white paper should therefore be regarded as a vendor-published brochure... rather than a serious research paper... [9]

Although various definitions and standards have been proposed [9], [10] there remains no complete formal definition of OLAP and no clear agreement within the industry about precisely what OLAP is. Key features of OLAP have been identified as:

...the multidimensionality of the data, which enables browsing the data by dimensions and by measures (facts). A dimension is a list of values that provides an index to the data... [Other] key features are on-line operations such as roll-up, drill-down, slice-and-dice and rotate. The analyst can use them to select appropriate dimension values and view selected measures in tables and graphs. Other key features of OLAP are built-in and programmable analytical capabilities, and different presentation and reporting options. Common characteristics of KDD [Knowledge Discovery in Databases] and OLAP algorithms are that they operate on large data sets and that their result, which mainly consists of aggregated information from existing databases, is previously unknown [11].

Even though the OLAP concept does not appear to be an obvious basis on which to create KLAs, it has been used to implement a ‘bibliographic database’ [11] with some success. In this case however, no attempt was made to create a taxonomy that would allow users to select a subset of library elements for viewing. Instead OLAP has been used to build a reporting tool that presents and summarizes information relating to document meta-information and interrelationships:

We believe that KDD, and especially the OLAP technology, could help libraries and information centers solve important issues regarding bibliographic databases. OLAP systems can be used for periodic and ad hoc reporting, quality assurance, and data integrity checking [11].

In general, OLAP cannot be directly applied to the problem of providing the key KLA features because it is a tool to enable analysts to efficiently query data warehouses, not a classification tool for organizing knowledge libraries.

#### E. Faceted Classification

Faceted classification was introduced in 1933 by Ranganathan through his ‘colon classification’ scheme, which he began work on in 1924 [12], [13]:

The presence of books with multi-faceted subjects was a fact... Library classification should reckon with them [12] (quoting Ranganathan)

In essence, faceted classification schemes involve identifying the main properties or ‘facets’ of a particular subject and constructing a hierarchical taxonomy for each facet. In colon classification, colons (:) indicate a change from one facet to the next [14]. Other faceted systems may employ other symbols to indicate particular relationships between facets [15].

The concept of faceted classification has influenced the development of other major classification systems. Universal Decimal Classification [16] can be thought of as a faceted version of Dewey Decimal Classification. BC2 is a faceted version of Bliss’s ‘Bibliographic Classification’ [17]. Ultimately it seems that the faceted library classification revolution was completely swallowed up by the more fundamental and radical revolution of computerization. It is only now, when the dust from the computer revolution is beginning to settle, that the notion of *computerized* facets is beginning to take hold [18], [19], [20].

## IV. INTRODUCING $N$ -TREE-SPACE

### A. Information Space

In this paper we define the term *information space* as follows.

An information space is a theoretical construct that, partly by analogy to physical space, describes a system for organizing, relating, and presenting information.

An information space defines the relationships between the information elements it contains. It also describes how users may navigate the space and select the information elements.

### B. $N$ -Space

We define  $N$ -space as an  $N$ -dimensional space where each *point* can be completely described in terms of  $N$  integer indices along  $N$  orthogonal axes. Each axis is associated with a dimension. A *region* in  $N$ -space is a set of points. Certain regions can be described in terms of ranges along any of the axes. Dimensions in  $N$ -space are *inclusive*—meaning that each point in  $N$ -space has a coordinate on each axis.

Objects can be *attached* to  $N$ -space by associating them with regions.  $N$ -space provides a navigable space where relationships between objects can be discussed in terms of distance and direction.

### C. $N$ -tree-space

$N$ -tree-space (NTS) is an information space that is essentially an  $N$ -space where each axis consists of the set of leaves of a rooted tree (rather than the set of integers). Thus a point in NTS can be completely described in terms of  $N$  leaves from  $N$  different trees.

More general *classification regions* in NTS can be described in terms of nodes (both interior nodes and leaf nodes), rather than leaves. Classification region descriptions may include zero or more nodes from each dimension. KLA elements can be *attached* to an NTS taxonomy by associating them with nodes, and hence classification regions, in NTS.

A *classification* is a set of dimension–node pairs that describe the classification region that a particular element is

attached to. A *classification in dimension  $d$*  of an element is a subset of its classification containing only the dimension–node pairs where  $\text{dimension}=d$ .

The relationship between two or more elements in a particular dimension can be discussed using traditional tree terminology such as *common ancestor*, *parent*, *child*, *sibling* and so on.

#### D. Creating an NTS KLA

Creating an NTS KLA necessarily involves implementing a number of components including:

- an NTS taxonomy (classification system),
- a KL where each element is *attached* to a region in the taxonomy, and
- an interface for browsing the taxonomy and selecting and viewing elements.

By way of example, consider a very simple two dimensional NTS consisting of two dimensions:  $D_1$  and  $D_2$ .  $D_1$  consists of the root node  $\alpha$ , which has child nodes  $\beta$  and  $\epsilon$ , while node  $\beta$  itself has two child nodes  $\gamma$  and  $\delta$ .  $D_2$  simply consists of the root node  $A$ , which has child nodes  $B$ ,  $C$ , and  $D$ .

We attach three elements ( $e_1, e_2$ , and  $e_3$ ) to the taxonomy:  $e_1$  to the nodes  $\beta$  and  $B$  (which we represent by  $\langle \beta, B \rangle$ ),  $e_2$  to  $\langle \gamma, D \rangle$  and  $e_3$  to  $\langle \epsilon, A \rangle$ . Because this is a very simple NTS we can depict it graphically (see Figure 1).

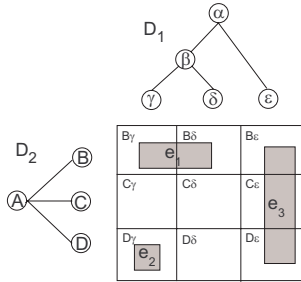


Fig. 1. A simple two dimensional NTS including elements

A user's *position* in an NTS KLA can be described in terms of some number of *selected nodes* from some number of dimensions (zero or more nodes from each dimension).

A user's position determines the set of *selected elements* in that an element in the library is selected if each node from that element's classification is either currently selected by the user, or is a *descendant* of a node that is selected by the user.

In our example (see Figure 2), position  $B\beta$  selects element  $e_1$  and position  $A\gamma$  selects element  $e_2$  (but not element  $e_1$ ).

#### V. DISCUSSION AND CURRENT AND FUTURE RESEARCH

NTS appears to be a near ideal scheme for KLAs. Appropriately implemented NTS KLAs [21] allow users to get a rapid overview of the contents of the library and the relationships that exist between library elements. Furthermore, by

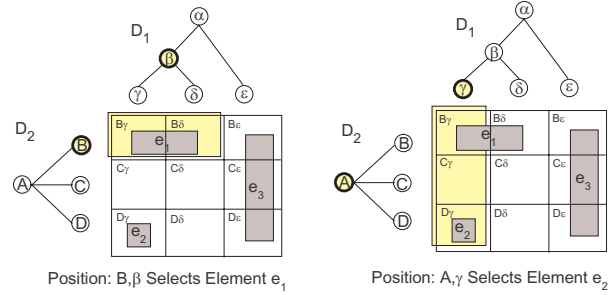


Fig. 2. A simple two dimensional NTS with library positions indicating element selection

allowing the definition of a number of different classification hierarchies, and the selection and classification of elements by position in an arbitrary number of hierarchies, NTS enables users to select sets of elements that are precisely tailored to their requirements.

#### A. Contrasting NTS and Tree Classification

A tree taxonomy with  $n_1$  nodes allows only  $n_1$  different possible classifications. NTS is significantly more descriptive. For example, take our  $n_1$  tree classification nodes and assume they can be distributed evenly over  $N$  NTS dimensions. If we require that each element in the NTS taxonomy is classified *exactly once* on each dimension—which is significantly more restrictive than our actual requirements—then we have a possible:

$$c = \left(\frac{n_1}{N}\right)^N$$

different NTS classifications. For realistic values of  $n_1$  and  $N$  (eg.  $n_1 \gg N$ ) it is clear that NTS has much greater descriptive power than tree classification. For example if  $n_1 = 100$  and  $N = 4$  then  $c = 390625$ .

Another way to compare NTS and tree classification is to ask the following question.

If we have a tree taxonomy with  $n_1$  nodes, how many nodes ( $n_2$ ), evenly distributed over  $N$  dimensions, do we need to construct an equivalently descriptive NTS taxonomy?

In this case, assuming the same restrictions as above, the equation is:

$$n_2 = N \sqrt[N]{n_1}$$

For instance if  $n_1 = 100$  and  $N = 4$  then  $n_2 = 12.65$  (2d.p.). This analysis shows that an NTS taxonomy with 4 dimensions and no more than 4 nodes in each dimension is at least as descriptive as a conventional tree taxonomy with 100 nodes.

#### B. Contrasting NTS and OLAP

NTS shares with OLAP the notion of a multidimensional view consisting of hierarchies on multiple dimensions. However, whereas OLAP has a strong connection with relational databases and may, in practice, even be thought of as a relational database interface, NTS is a data model in itself and is (otherwise) completely unrelated to relational databases.

Also, the application area for *NTS* is completely different to OLAP. In particular, OLAP terminology such as *aggregation level, drill-down, roll-up, and slice-dice* has no *NTS* context.

*NTS* could be regarded as a relative of OLAP. In the future *NTS* may help produce more intuitive and formal definitions—and so a deeper understanding—of OLAP.

### C. Contrasting *NTS* and faceted classification

Just like *NTS*, faceted classification is hierarchical and multi-dimensional. Despite these similarities, there are a number of important differences.

First, unlike faceted classification, *NTS* is an ‘information space’ and is defined with reference to *N*-Space. This provides *NTS* with a more solid mathematical/theoretical foundation than faceted classification.

Second, according to Ranganathan, the chief purpose of faceted classification is to:

...arrange books in a helpful sequence, or... to mechanise the arrangement of books in a helpful sequence.

It is also to help mechanise the correct replacing of books returned after use. Again, it is to help fix the most helpful place for a newly added book among those already in a library [14] (p16).

*NTS* has the broader, more contemporary, purpose of providing the key feature of a KLA.

Third (and finally), faceted classification was developed as a manual, rather than computerized, system of classification/organization. Manual systems (in terms of notation for example) have completely different functional requirements to computerized systems.

### D. *NTS* Research

*NTS* has already been the subject of research that resulted in the development of an efficient *NTS* data structure and a working *NTS* algorithm library application [21]. Further research to broaden and strengthen the theoretical foundation for *NTS*, relate *NTS* to existing technologies, and to explore its applications and potential is currently underway.

We believe that *NTS* can be used for educational purposes to advantage—allowing students to rapidly understand the connections between different topics. *NTS* can also offer significant benefits to researchers—enabling more precise automatic searches than is currently possible. *NTS* can also improve communication—by allowing precise definitions and providing a natural, non linear, structure to present information. *NTS* will also assist people working on converging technologies to recognize common research areas.

## REFERENCES

- [1] F. R. McFadden and J. A. Hoffer, *Modern Database Management*, 4th ed. Sydney: Benjamin/Cummings, 1994.
- [2] D. Taft, “Stopping knowledge overflow—knowledge management tools still in developmental phase,” *Computer Reseller News*, p. 14, 2000.
- [3] E. Svenonius, “The epistemological foundations of knowledge representations,” *Library Trends*, vol. 52, no. 2, pp. 571–587, 2004.
- [4] B. Palmer, *The socio-historical background to library classification*. Kent: The library association, 1971, pp. 7–15.
- [5] K. Berman and J. Paul, *Fundamentals of sequential and parallel algorithms*. Boston: PWS, 1997.
- [6] D. Maddison, “The tree of life web project,” 2001. [Online]. Available: <http://tolweb.org/tree/>
- [7] U. Krohn, N. J. Davies, and R. Weeks, “Concept lattices for knowledge management,” *BT Technol J*, vol. 17, no. 4, pp. 108–116, 1999.
- [8] E. F. Codd, S. B. Codd, and C. T. Salley, “Beyond decision support,” *Computerworld*, vol. 27, no. 30, pp. 87–89, 1993.
- [9] N. Pendse, “What is olap?, the olap report,” May 2004. [Online]. Available: <http://www.olapreport.com/fasmi.htm>
- [10] P. Vassiliadis and T. Sellis, “A survey of logical models for olap databases,” *SIGMOND Record*, vol. 28, no. 4, pp. 64–68, 1999.
- [11] E. Hudomalj and G. Vidmar, “Olap and bibliographic databases,” *Scientometrics*, vol. 58, no. 3, pp. 609–622, 2003.
- [12] D. Taulbee, “Classification in information storage and retrieval,” no. 20. ACM 1965 20th national conference, 1965, pp. 119–137.
- [13] H. Bliss, *The organization of knowledge in libraries and the subject-approach to books*, 3rd ed. New York: H.W. Wilson company, 1939.
- [14] S. Ranganathan, *Elements of library classification*. Bombay: Asia publishing house, 1962.
- [15] B. Vickery, *Faceted Classification*. London: Unwin brothers, 1960.
- [16] U. decimal classification consortium (UDC), Hague, 2004. [Online]. Available: <http://www.udcc.org/>
- [17] B. classification association, 2001. [Online]. Available: <http://www.sid.cam.ac.uk/bca/bchist.htm>
- [18] T. knowledge management connection, “Faceted classification of information,” 2004. [Online]. Available: <http://www.kmconnection.com/DOC100100.htm>
- [19] T. Wilson, “Facetmap—your home for faceted classification,” 2003. [Online]. Available: <http://facetmap.com/>
- [20] M. Hearst, “Flameco search home,” 2004. [Online]. Available: <http://bailando.sims.berkeley.edu/flamenco-interface.html>
- [21] E. Rayner, “Developing a method of algorithm classification for computer algorithm library applications,” Master’s thesis, SITACS, 2003.