META-INFORMATION SYSTEMS: A DESIGN STUDY

Mark S. Madsen, Iain S. Fogg & Clive L.N. Ruggles
Department of Mathematics and Computer Science
Charles Wilson Building, University of Leicester
Leicester LE1 7RH, UK
Fax: +44-533-523604

ABSTRACT

Metadata systems are concerned with the management of data which describes other data, whether datasets, catalogues or entire database management systems. Metadata systems can be used to store richly detailed forms of information, perform seamless wide-ranging searches of information distributed across networks, and to integrate information stored in disparate repositories. Here we describe the design and methods of implementation derived from the experience of the Leicester University Metadata Project (LUMP). Our approach utilises the incorporation of semantic metadata in addition to resource metadata, resulting in a generally more powerful system than existing global directory services. A feature of the LUMP design is flexibility of implementation, with the ability to provide a coherent metadata system functioning above heterogeneous autonomous distributed databases.

INTRODUCTION TO METADATA SYSTEMS

The term *metadata* means, literally, *data about data* (Lillywhite, 1991). In this paper, we use the term to denote both data which describes elements of discourse — *semantic* metadata — and data which describes datasets or DBMSs — *resource* metadata. The reason for considering metadata is that a powerful solution to the management challenge posed by the large quantities of data now held globally is the development of *metadata systems*, whose basic function is to locate data which is relevant to a particular user, given information about the type of data required and the details of the *context* in which it is to be used. Ideally, this context should be defined in terms of the user who requires the data: their background, interests, understandings, and preferences. The main function of a metadata system is therefore to identify data which may satisfy the requirements of the user, and to store information about its location, content, and quality relative to the interests and situation of the user — taking into account such practical considerations as accessibility, cost, and ease of use.

The metadata approach to information systems provides a number of powerful advantages over other approaches:

- Metadata systems provide a level of abstraction between the user and the stored data. This allows the construction of specialised, custom databases as described by Neelameghan (1992) which can present their data in more conventional forms and under normal cataloguing conventions, depending on the manner in which they are accessed.
- Such a level of abstraction can provide the conceptual space for information to be structured into knowledge as described by Hsu *et al* (1991) and Ingwersen (1992). The metadata environment can typically supply this in the form of information about the user which in turn can be used to filter the user's view of the metadata and data stored in the system.
- Metadata can provide an integrating framework for disparate databases so that the differences between these are transparent to the user. This can be the case even when the databases are heterogeneous and widely distributed. This issue is discussed by Madsen, Fogg & Ruggles (1994).

A complete metadata system should be capable of performing all of these functions automatically, so as to provide the user with a transparently personalised view of the information space. The overall problem set with which the system needs to be able to cope is extensive. Generic problems which need to be faced are sheer volume of data managed by heterogeneous DBMSs, and resilience of function against inevitable failures of metadata consistency.

It is important to understand how the ideas presented here relate to existing directory/data dictionary services, such as WWW (World Wide Web), WAIS (Wide Area Information Service), X.500, Whois++ and the Gopher interconnection protocols. The overall functionality of all of these differs in a subtle but crucial manner from general metadata models. While the directory services are intended as general enquiry services which allow connection to local services via some uniform protocols, the metadata model abstracts their functions and additionally provides for the integration of knowledge modelling within the system (Hsu *et al*, 1991).

Existing studies of metadata systems include the use of a metadata approach to accessing protein structure databases from a knowledge-based system (Eccles & Saldanha, 1990). There Prolog is used as a system metalanguage for the construction of database queries directly from the knowledgebase. As the authors point out, this approach amounts to using the metadata system as a rich communication channel. Another study is concerned with the management of decision processes via the integration of three kinds of metadata (Hsu *et al*, 1991). These are *operating knowledge* concerned with individual data or functional subsystems, *control knowledge* such as data transfer rules and equivalence tables for distinct data items relevant to the same logical subject, and *decision knowledge* for global and local processes (the terminology used is that of Hsu *et al*, 1991). Closely related with this last-mentioned study is the collaborative work system KMS, described by Akscyn, McCracken & Yoder (1988). A specific model which incorporates some of the ideas described in this article is set out in the paper of Ishikawa *et al* (1993).

Those aspects of metadata storage which are relevant to the efficiency of its entry and retrieval have been detailed by Ruggles *et al* (1992). A conceptual study of some of the issues arising in the design of metadata systems has recently been carried out by Fogg, Ruggles & Newman (1993).

The remainder of this paper is set out as follows. The next section describes the structural aspects of the design, which are mainly concerned with the metadata model. The section after that describes the functionality of the distributed components of the system, along with the strategies adopted in the design for the everyday management of metadata. A section is then devoted to the characteristics of the metadata query language MQL. In the final section, we bring together the disparate aspects of the system and describe a prototype implementation (the STILE Phase I system) currently being field-tested at our institution and others.

Limitations of space preclude a complete formal description being given, so we restrict ourselves generally to a discussion of the important features of the abstract design. We remark here that the formal system specification has been written in Object-Z (Duke & Duke, 1990, Hall, 1990, Cusack, 1991, and Stepney, Barden & Cooper, 1992) and will be presented elsewhere.

STRUCTURAL DESIGN

Informally, our starting point is that the world of discourse can be described in terms of *concepts*. Davis (1986) identifies three broad categories of concept — *temporal* concepts, *spatial* concepts, and the vast remainder of *pure* concepts (Davis simply calls these all *concepts*, but this terminology is too confusing for the present purpose). Our approach is to encapsulate conceptual information within more general structures which we call *topics*. The immediate advantages of doing so are twofold: first is that higher-order metadata (data about metadata) can easily be accommodated within the system, second is that topics can in principle be extended so as to incorporate other attributes beyond resource or semantic metadata. This approach to data via the creation of a neutral database has been successfully used in the integration of commercial applications — see, for example, Busby & Hutchison (1992).

The implementation of topics requires a representation that is amenable to human users. For this reason, textual, graphical, audio, and visual representations of topics are permitted. Furthermore, it is quite natural for us to identify relationships between topics in the real world. To facilitate sophisticated browsing and querying of our metadatabase, it is also necessary for us to represent the relationships between topics in the system. The function of our metadata system can therefore be seen as providing a repository for topics, and managing relations between them. In essence, our metadatabase system is a management system for a semantic net — see Quillian (1985) and Sowa (1991) — although with some features peculiar to the distributed metadata environment within which it exists. This view of metadata systems is close to that expressed by Hsu *et al* (1991).

The most important attributes of a topic are the descriptive elements called *nuggets*, which communicate meaningful aspects of the topic to the human user, and the links from that topic to related ones. Links to other topics are represented by the *Id* of the target topic and a weight denoting the importance of that link. (Although this aspect is not treated explicitly here, a requirement is that the system should provide effective learning capabilities which will allow link weights to be adjusted according to the results of usage.) The full structure of a topic is illustrated in Figure 1, and is to be understood as follows:

- Each topic possesses at least one *TopicId.*
- A topic may have any number of nuggets. These may consist of text, or involve graphics, audio, video or other combinations of media. Each nugget is labelled by a *ContextId,* denoting the context in which that nugget is relevant to the topic. The uses of contexts are described in more detail below.

- A topic may have any number of *links* to other topics. A link consists of the *TopicId* of the related topic and a numerical weight expressing its importance. Like nuggets, each link is labelled by a *ContextId.*
- A topic may have a *dictionary* consisting of any number of *TopicIds* representing *equivalent topics*: those which have been locally identified (that is, from the point of view of the user who controls that topic) as being the same as that topic containing the dictionary. Note that these equivalent topics may be controlled by a different user.
- A topic may have any number of *resource entries.* Each resource entry represents an *external resource* such as an online database, ftp server, or file viewer. The resource entries consist of scripts which contain sufficient information to facilitate online access to those external resources directly from the topic where that is possible, otherwise information which will enable the user to locate and access the resource offline.

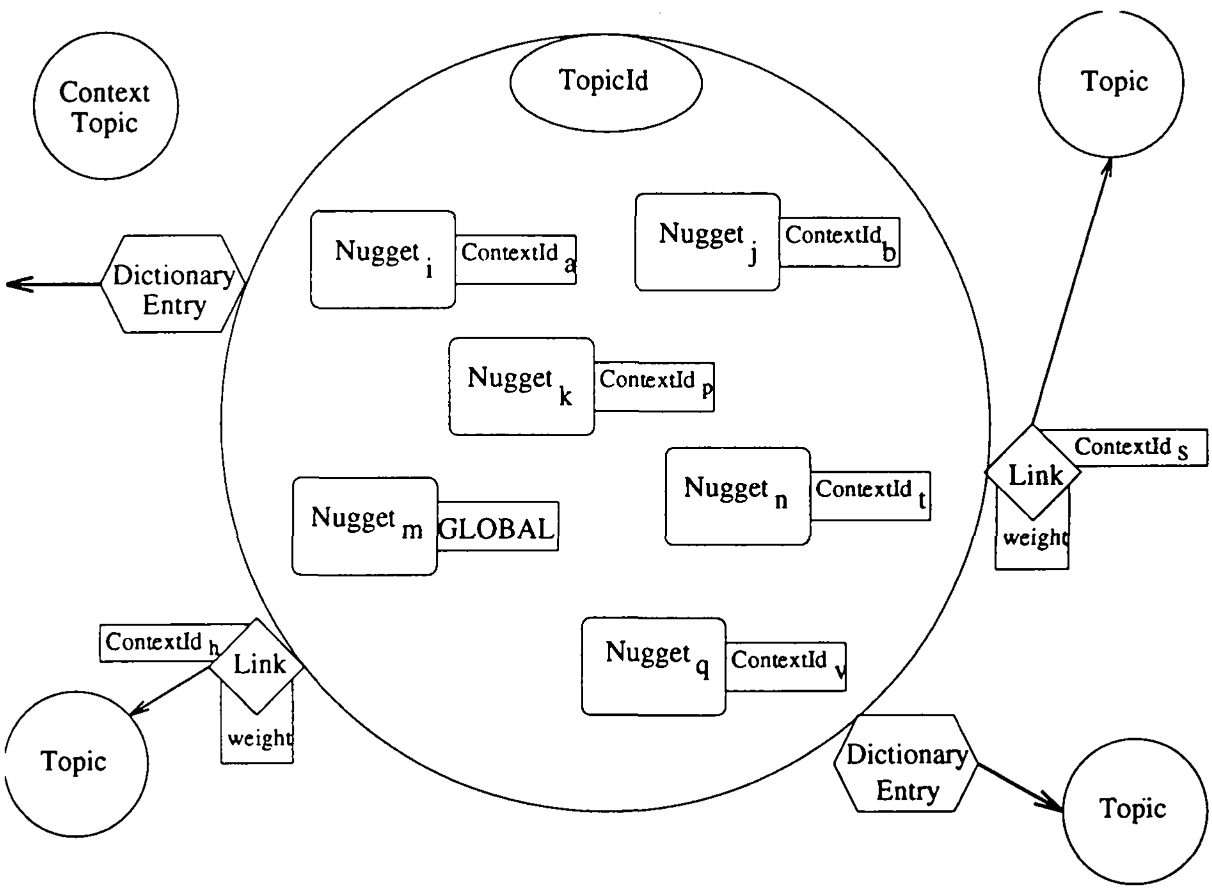

Figure 1: Illustration showing the structure of a general topic and its connections to other topics within the metadata system's topic network.

These ideas are summarised in the entity-relationship diagram of Figure 2. It would also be possible to introduce link types to our model (Croft and Turtle, 1989). There are a number of reasons for not doing so at this stage of development of our metadata management model. For one thing, the benefits of link typing need to be weighed carefully against the extra complexity it implies (Glushko, 1989). In our model, we would need to introduce yet another subclass of topics called Link Type Topics, so as to avoid the danger of infinite regress. The strongest argument against link typing, however, is that even without it, our metadata model is capable of effective representation of all forms of recorded information or knowledge, and is also efficient and simple for the user to comprehend.

Topics with nonempty resource sets are called *resource topics.* They are the interface between the metadata contained in the topic net and the external data world. Topics with no resource entries, on the other hand, contain what is called *semantic metadata,* namely descriptions of what is meant by particular terms or concepts, and which other semantic materials are relevant to specific topics. The reason for allowing topics to have multiple identifiers is so that deletion or merging of topics can proceed by removal of their contents, but without detriment to the referential integrity of the topic network.

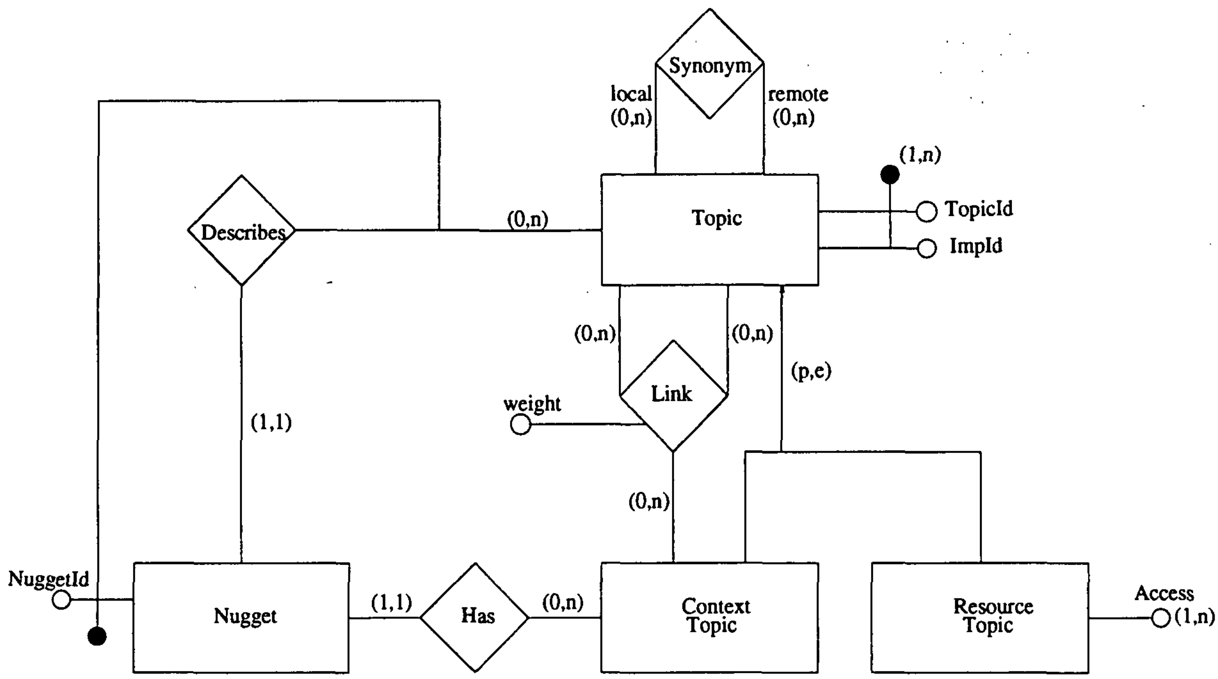

Figure 2: *Entity-Relationship diagram for the topic structure described in the text.* The Access attribute of the Resource Topic corresponds to an external resource.

An advantage of the metadata approach to information is that it allows for the possibility of incorporating knowledge management as one of the functions of metadata systems (Hsu *et al*, 1991). The fact that the topic model is constructed so as to facilitate the interchange of metadata relating to knowledge, rather than simply information, makes it necessary to include some scheme for classification of metadata within the capabilities of the metadatabase (Ranganathan, 1967). This point is important since it is the classification and interrelation of information which transform it into knowledge as discussed by Belkin (1978), Lyotard (1979) and Brookes (1980). Such a scheme will also turn out to ease the problems of deciding how queries can be dealt with.

Crucial to the ability of the system to configure the metadatabase effectively so as to suit the needs of a particular user is the use of the *context* attribute. The purpose of this attribute, as in every form of discourse, is to establish a framework within which meaningful communication can take place. Our design also gains further benefits from contexts. In response to the user specifying which context they are using, the system can simplify their view of the system. Additionally, contexts are used to construct an efficient query method — see below.

In our system, context is realised via the introduction of a label attached to each nugget and link of every topic within the system. This indicates within what circumstances that nugget or link is to be understood as part of the defining structure of the topic within which it is found. Contexts are themselves implemented in terms of topics. The *ContextId* mentioned above can therefore be replaced by a *TopicId*, with the introduction of suitable invariants. In our present design, we insist that there be specific *context topics*. These are not allowed to be resource topics, and they are restricted to have nuggets and links whose context label is that of the *global* context — this is defined as the context in which all elements of all topics are relevant, and implemented via a special invariant identifier supplied as part of the initial system configuration.

A modest example which clarifies some of these ideas is illustrated in Figure 3. This might be represented in the system by the topic with *TopicId* number 57, which in addition to the nugget "Water" may consist of the textual nuggets "Substance used for drinking and swimming", "Substance found in rivers and seas", "Combustion by-product", and the links ("Boats",4), ("Fish",8), ("Whisky",1). Here the *Ids* of linked topics have been replaced by their textual names for clarity, and the number following the name is the weight of that link. The application of the context "Sports", whose *ContextId* is the *TopicId* 92 in the example, may pick out the subtopic with the nuggets "Water" and "Substance used for drinking and swimming", and the link ("Boats",4). Although drinking is not (always) regarded as a sport, the nuggets themselves are not divisible by the actions of the system, and so the entire nugget is extracted by the sports context. It is easy to see how the topic structure allows this example to be extended to incorporate other forms of media such as sound and video.

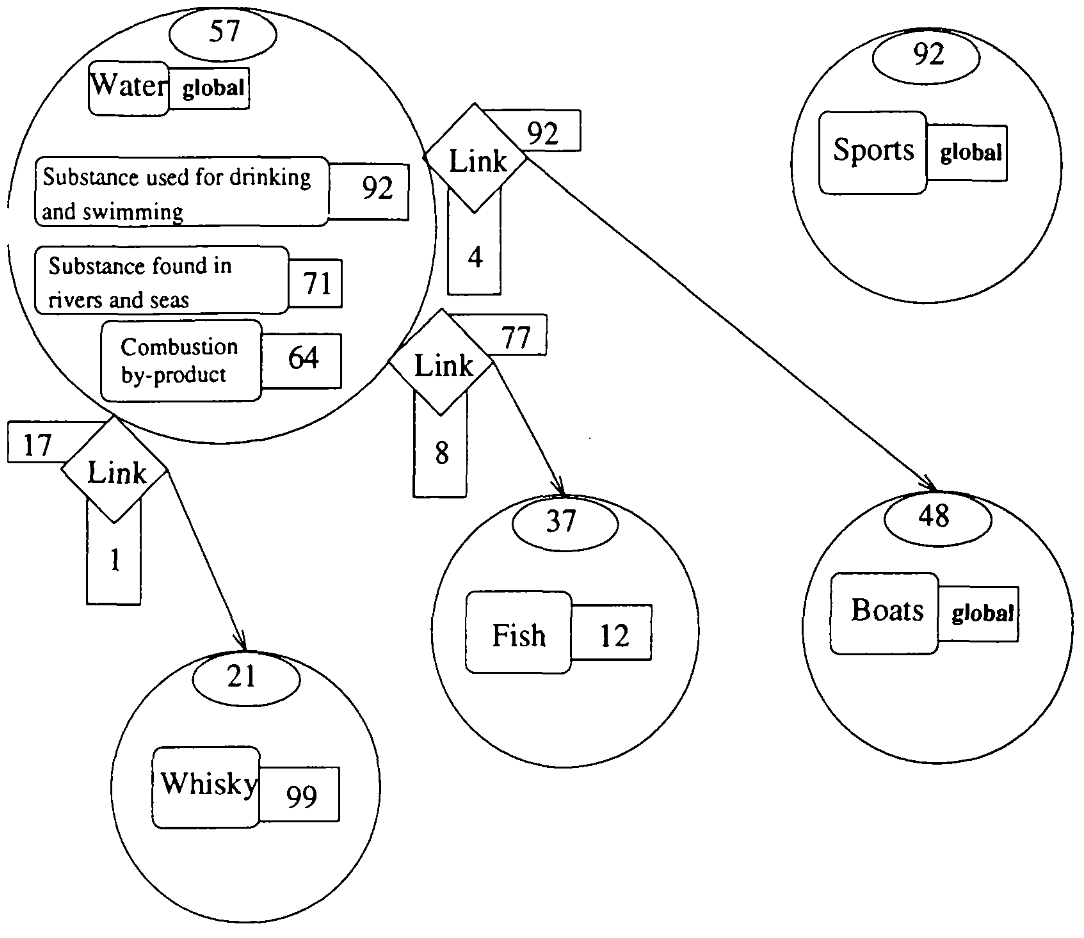

Figure 3: Illustration of the example discussed in the text: the representation of the concept "water" as a topic within the system.

FUNCTIONAL DESIGN

The basic functional unit of the system is the IMP (Information Management Processor) as imagined in Fogg, Ruggles & Newman (1993). The user's logical view is that each user possesses a unique IMP which is responsible for the maintenance of all gathered and stored metadata, and for communication with IMPs belonging to other users for querying their metadata stores or responding to such received queries. As such, the IMP is the user's *representative*, and must be able to evolve over time in such a manner as to reflect the user's own interests and background, so that the system will often be able to resolve ambiguities and uncertainties of input and queries on the basis of the stored user profile held by the IMP. For example, we could use the textual objects "lager" and "beer" to describe a particularly well-known thing in the real world. The problem of course is that "lager" and "beer" are terms used by British people to denote quite different things in the real world, while to an Australian or South African, they are naturally the same. So while we may use textual objects to describe topics, the topic that is being denoted by the text depends on the experience and biases of the user (see Ruggles *et al*, 1992).

The management of the metadata contained within the system, and particularly within any IMP's metadata store, is clearly the central issue in the development of a meta-information system. It is, however, a nontrivial task to anticipate all possible management needs and effective policies. The solution to the management problem consists of three parts: the manner in which the topics describing metadata are communicated between the IMPs, the definition of the structure of a metadata query, and the strategies to be adopted in the everyday management of each IMP's topic network. The approach to communications taken in this paper is that all communications between IMPs are expressed in terms of the system scripting language MQL (see below). All lower level communications and transport protocols are expected to be supplied by the system interface (see Figure 5). The second and third aspects are dealt with below. Our system design requires that each IMP be capable of at least the following suites of operations:

Topic Operations

- creation — accomplished by assigning an unused *TopicId.*. Other elements of the topic are then added by use of the appropriate operations listed below.
- merging — accomplished by creating a new topic possessing all the *TopicIds* of the topics to be merged, and then adding all the elements of the contributing topics to the new (multiply identified) topic.
- deletion — performed by removing all elements except the *TopicId*, leaving the (multiply identified) null topic.

Nugget Operations

- creation — performed by assigning an unused *NuggetId.* Each new nugget has a default context assigned, which is the context active in the user interface at the time of creation.
- modification — performed by reading the new nugget from an external buffer into a nugget with an existing *NuggetId.* Editing is performed by reading the existing nugget contents into the external buffer first.
- context assignment — performed by overwriting the existing *ContextId* of the nugget with a specified *ContextId.*
- retrieval — performed by specifying the *NuggetId* of the desired nugget. This permits the possibility of browsing all nuggets within a topic.
- deletion — performed by removing the *NuggetId* and freeing the storage used by its contents.

Link Operations

- creation — performed by assigning an unused *LinkId.* Each new link has a default context assigned in the same manner as for nuggets, and a default weight of unity.
- modification — performed by overwriting the existing target *TopicId* of the link with a specified *TopicId.*
- context assignment — performed by overwriting the existing *ContextId* of the link with a specified *ContextId.*
- modification of weight — performed by overwriting the existing weight of the link with a specified real value.
- traversal — performed by specifying the *LinkId*, retrieves the entire target topic.
- deletion — performed by removing the *LinkId.*

Dictionary Operations

- creation — performed by assigning a *DictionaryId.*
- modification — performed in a similar manner to the modification of nuggets. The invariant is that the resulting dictionary must consist only of a set of legal *TopicIds.*
- browsing of equivalences — achieved by returning all the remote *TopicIds* of topics which are listed as equivalent with the one containing the dictionary. The system should provide the option of retrieval of the remote topics where possible.
- deletion — performed by removing the *DictionaryId.*

Resource operations

Resource operations like creation, modification, and deletion are similar to those for nuggets and links. Different is

- resource access — given the *ResourceId,* returns the script representing the method of directly accessing the resource. The system may provide for transparent access to the resource via the resource manager — see Figure 5.

METADATA QUERY LANGUAGE

Discovering whether an IMP's metadata store contains any topics relevant to answering a particular query is obviously best accomplished by presenting that IMP with a topic to which the topics stored in that IMP's topic network can be compared. (Of course, the metadata query will also need to specify the context within which the search is to be conducted.) This in turn requires the construction of a quantitative metadata evaluation function. This evaluation of a topic is defined relative to another topic — the similarity between topics is all that the system needs to be able to quantify, since any query requires that a set of topics be evaluated relative to that query, and returned if their evaluation is sufficiently high, according to criteria which are to be included in the structure of the query. Note that the user interface is responsible for packaging all user input into the form of topics.

Clearly, there are many possibilities for the construction of metadata evaluation functions: our design envisions that there will be a predefined set of such functions available to each IMP, so that the user can choose which is to be used in a particular query by choosing one from a menu. We remark here that, in addition to the well-known techniques of word-counting and string-matching, a rich variety of alternative strategies for matching topics to metadata queries is provided by the nonparametric statistical methods known collectively as cluster analysis — for example, see Anderberg (1973), Duran & Odell (1974), Everitt (1974), Van Ryzin (1977) and references contained therein.

Because of the way in which contexts are used to classify stored metadata, a query must consist (at least conceptually) of two parts, the first designed to establish any appropriate contexts, the second to find appropriate response topics to the query. At the same time, a crucial concern in the design of our metadatabase system is performance. In particular, we will need effective and fast indexing to be applied to topics in order to maximise the throughput of the query processor. To achieve this the system uses a two-level indexing strategy: first the contexts are indexed and then the topics within those contexts. With all these components in place, we can describe some aspects of MQL (Metadata Query Language).

Schematically, a metadata query Q consists of two subqueries, a context query Q_c and a topic query Q_t. The main difference between the two queries is their interpretation by the IMP: the first is applied to that IMP's context network, the second to the general topic network. Finally, the result of the query is (at least conceptually) subjected to some form of projection.

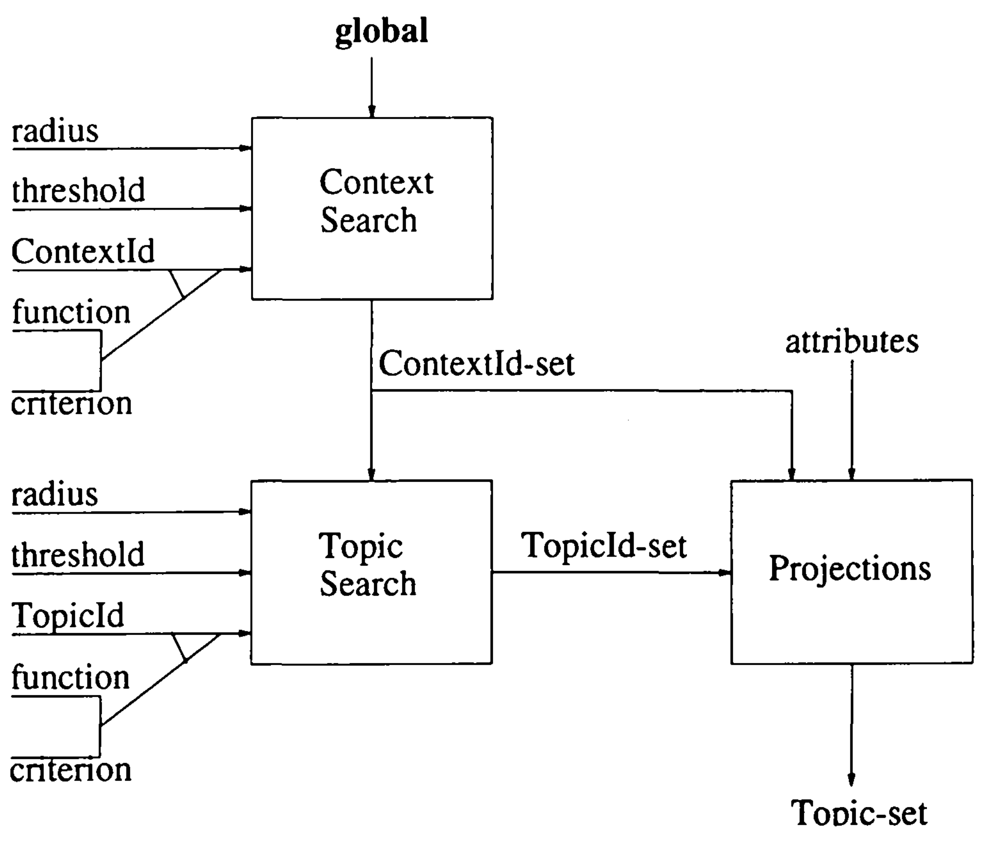

Figure 4: The dynamic structure of an MQL query.

The overall structure of a metadata query is shown diagramatically in Figure 4. The interpretation of this figure is that the context query is of the syntactic form

SELECT(criterion, function, radius, threshold, ***global****)*

where

- *criterion* is the context topic against which the searched contexts are to be compared;
- *function* is the code for the evaluation function to be used by the IMP in comparing the definitions of its contexts with *criterion*;
- *radius* specifies the radius of the context network that is to be returned as the result of each successful match. The radius of the subnet centred on a given context *C* is to be understood as a number *R* such that every context in the subnet can be reached by starting at *C* and following no more than *R* links.
- *threshold* is a number specifying the minimum cumulative weight of links to be followed in constructing the context subnet. Links with a cumulative weight below the threshold are ignored.
- ***global*** is the *ContextId* of the global context.

In its turn, the topic query is of the same form:

SELECT(criterion, function, radius, threshold, contextIdSet)

where the parameters are to be interpreted slightly differently:

- *criterion* is the topic against which the searched topics are to be compared;
- *function* is the code for the evaluation function to be used by the IMP in comparing the definitions of its topics with *criterion*;
- *radius* specifies the radius of the topic subnetwork that is to be returned as the result of each successful match.
- *threshold* is a number specifying the minimum cumulative weight of links to be followed in constructing the topic subnet. Links with a cumulative weight below the threshold are ignored.
- *contextIdSet* is the set of *Ids* of matched contexts returned from the context search, and is used to restrict the search space for topics matching the topic query. Topics with no components bearing a *ContextId* contained in *contextIdSet* will not be examined in the course of the search.

The final projection operation is of the form

PROJECT(contextIdSet, topicIdSet, attributeList)

where *contextIdSet* and *topicIdSet* result from the previous *SELECT* operations, and *attributeList* is a set specifying which attributes of the topic or topics are required.
In order to allow for straightforward retrieval in the case that the relevant *ContextId* or *TopicId* is known to the IMP initiating the query, the alternative forms

SELECT(ContextId, radius, threshold, ***global****)*

and

SELECT(TopicId, radius, threshold, contextIdSet)

respectively are used. A query of a remote IMP is of course constructed in exactly the same way, but is passed to the remote IMP for processing instead of the local one.

SYSTEM ARCHITECTURE

The architecture of the system is shown in Figure 5. It consists of the following distinct components:

Stored Metadata

The metadata may be stored using any storage engine (such as a relational, object-oriented, or flat-file database), with the provision that a translation schema must be provided to translate between the physical and logical (namely, topics and links) data models.

Local Gateway

The local gateway performs the translation between the physical and logical data models.

Query Processor

The query processor accepts queries from the user via a graphical user interface, and from remote systems via a system interface, performs appropriate optimisations, and passes the query to the local gateway for subsequent processing. The query processor is also responsible for managing remote queries.

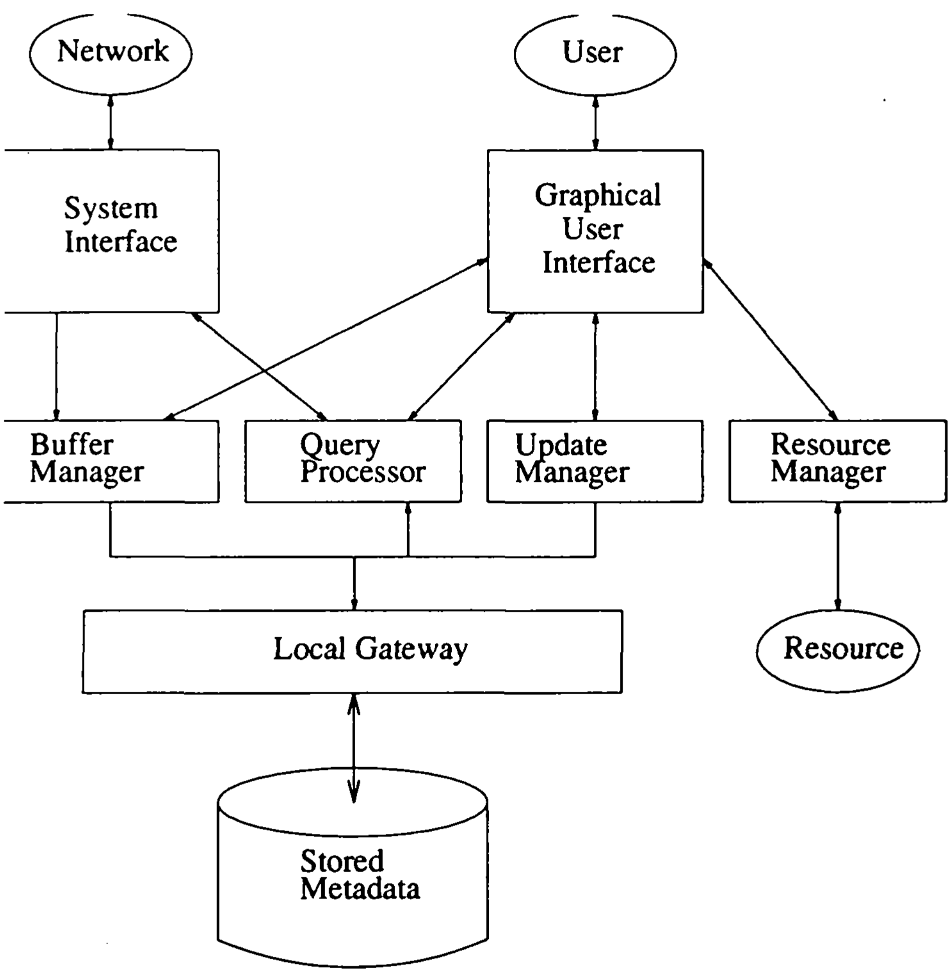

Figure 5: The architecture of the system at the IMP level.

Update Processor

Updates can be initiated only by the local user via the graphical user interface. The update processor ensures that certain constraints (for example, contexts may be linked only to other contexts) are not violated.

Resource Manager

The resource manager is responsible for accessing on-line resources that have been located using the query facilities. The resource manager is compatible with a number of "standard" remote access facilities, such as WWW, WAIS, and Gopher.

Buffer Manager

Metadata accumulated as the result of remote queries is temporarily buffered. The user is responsible for integrating this buffered material with the local metadatabase. Since the buffer is finite, the user can also specify the behaviour of the buffer manager when overflow occurs (for example, discard oldest items first, or discard newest items first). The user may also activate processes to integrate incoming metadata automatically.

System Interface

The system interface deals with the interaction between IMPs, exchange of queries and metadata, detecting IMPs that are out-of-service, interfacing to low-level communication software, and so on.

Graphical User Interface

The graphical user interface must provide a user friendly, configurable, windowing environment for the control and display of metadata, and for the display of data through interaction with the resource manager.

The LUMP design described above has been used in the construction of the STILE (Students' and Teachers' Integrated Learning Environment) system being developed at our institution and at Loughborough University. The aims and agenda of this project are fully described by Ruggles (1994). The STILE Phase I implementation is not yet fully distributed, and does not use contexts to full advantage. It does however use the topic model described above, and implements these in terms of the relational model (based on Figure 2) using a bespoke database management system. Figures 6 and 7 show screendumps from STILE Phase I illustrating both the topic search process and direct resource access.

It is expected that experience with the working STILE system, which is due to be updated to Phase II by late 1994, will be beneficial to the specification of the full final design of which this paper is the outline.

DISCUSSION

Metadata systems provide an exciting and enabling new perspective on the functions and abilities of information systems. This paper shows how a relatively simple, even naive, model of the way that metadata is structured and interelated can be developed to provide a framework for the construction of extremely powerful global, distributed, heterogenous, integrating information systems. Such systems have the virtues of:

Simplicity

The underlying defining structure used for the stored metadata is simple in concept and design. Unlike the metadata form used by many other proposed systems, there is no long catalogue of attributes to be learned before the user can interpret the results of retrieving information or metadata. Likewise, the learning barrier which users must pass in order to be able to collect and store their own knowledge is smaller than that encountered in previous designs — for such an example, see Hsu *et al* (1991).

Flexibility

Because of the simple structural requirements placed upon the stored metadata, our design is capable of incorporating not only new information, metadata, and resources with ease, but can also accomodate existing collections of metadata without difficulty (Eccles & Saldanha, 1990).

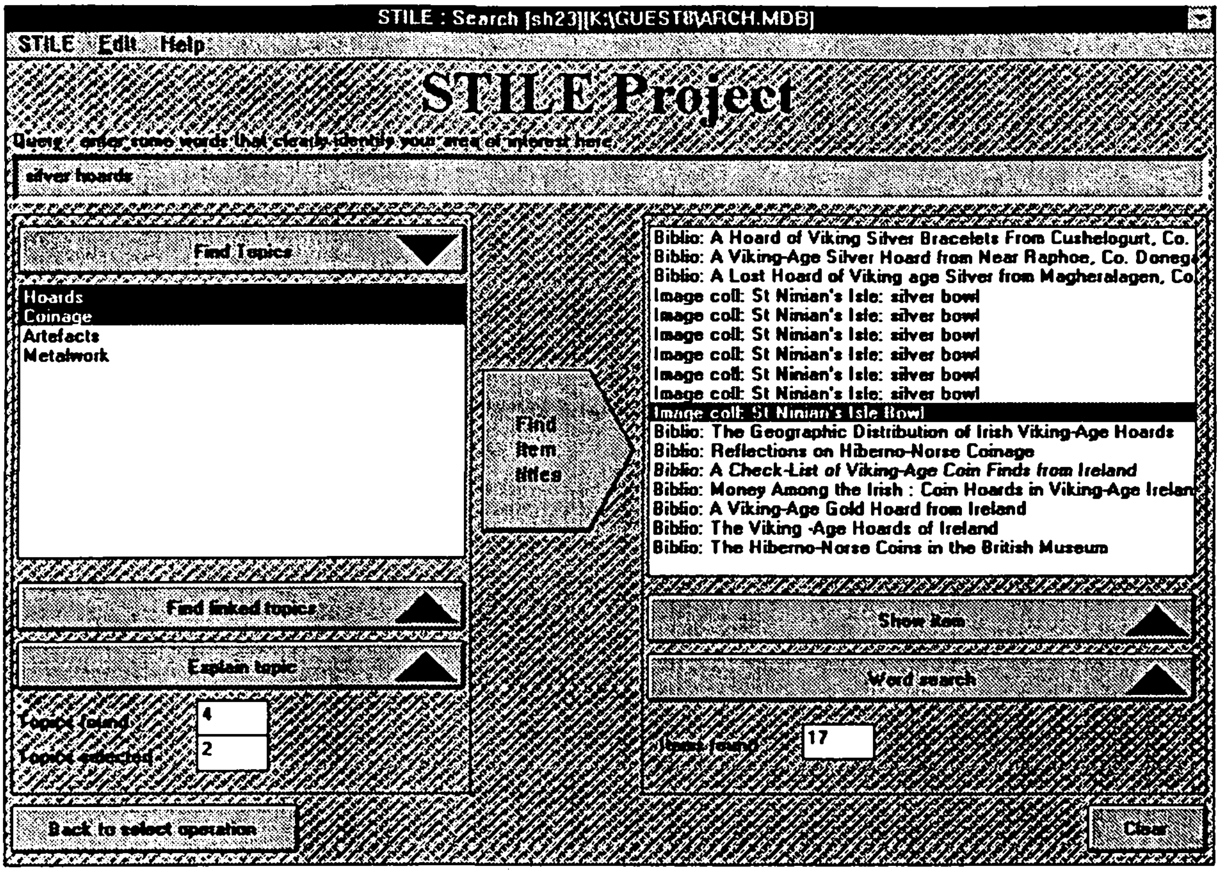

Figure 6: Screen dump from the STILE v1.0 system showing the entry of a search specification.

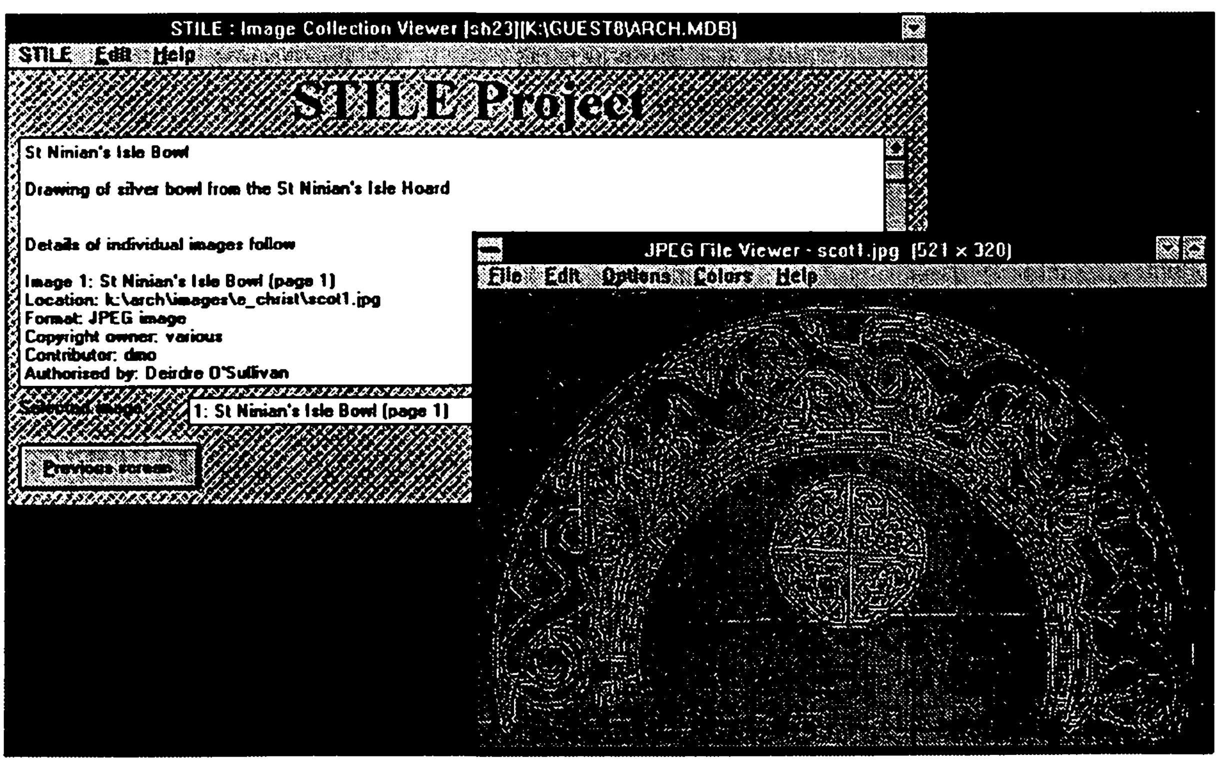

Figure 7: Screen dump from the STILE v1.0 system showing the use of a resource directly from the user's IMP interface.

Extensibility

Although our simple design is formally complete, it contains a framework within which new types of resource and metadata can easily be incorporated into the system. For example, if it ever becomes possible to create and access online olfactory databases, then these can be straighforwardly incorporated as new resources within the topic network comprising our metadatabase design.

Integration

The metadata approach provides the key to the integration of the functions provided by disparate heterogeneous and autonomous information systems. These systems can then be individually optimised for their own specific design function, and at the same time remain widely accessible through their metadata interface with the rest of the information world (Litwin & Rossopoulos, 1990).

By integrating existing information systems in a flexible manner, allowing access to and use of existing catalogues and utilities, and providing a framework within which users can store and retrieve knowledge, metadata systems generate the potential for rapid and transparent access to the information stored globally both now and in the future.

REFERENCES

Akscyn, R.M., McCracken, D.L. & Yoder, E.A. (1988) "KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations", **Communications of the ACM**, Vol 31, pp 820-835.

Anderberg, M.R. (1973) **Cluster Analysis for Applications**, New York: Academic Press.

Belkin, N.J. (1978) "Information Concepts for Information Science", **Journal of Documentation**, Vol 34, pp 55-85.

Brookes, B.C. (1980) "The Foundations of Information Science. Part I. Philosophical Aspects", **Journal of Information Science**, Vol 2, pp 125-133.

Busby, J.S. & Hutchison, D. (1992) "The Practical Integration of Manufacturing Applications", **Software Practice and Experience**, Vol 22, pp 183-207.

Croft, W.B. & Turtle, H. (1989) "A Retrieval Model for Incorporating Hypertext Links", in **Hypertext '89 Proceedings**, pp 213-224, New York: The Association for Computing Machinery.

Cusack, E. (1991) "Inheritance in Object Oriented Z", **Lecture Notes In Computer Science**, Vol 512, pp 167-179.

Davis, E. (1986) **Representing and Acquiring Geographic Knowledge**, New York: Pitman.

Duke, D. & Duke, R. (1990) "Towards a Semantics for Object-Z", **Lecture Notes in Computer Science**, Vol 428, pp 244-261.

Duran, B.S. & Odell, P.L. (1974) **Cluster Analysis: A Survey**, Berlin: Springer-Verlag.

Eccles, J.R. & Saldanha, J.W. (1990) "Metadata-Based Generation and Management of Knowledgebases from Molecular Biological Databases", **Computer Methods and Programs in Biomedicine**, Vol 32, pp 115-123.

Everitt, B. (1974) **Cluster Analysis**, London: Heinemann.

Fogg, I.S., Ruggles, C.L.N. & Newman, I.A. (1993) "A Distributed Metadata System", Research Report 34, Midlands Regional Research Laboratory, University of Leicester and Loughborough University.

Glushko, R.J. (1989) "Design Issues for Multi-Document Hypertexts", in **Hypertext '89 Proceedings**, pp 51-60, New York: The Association for Computing Machinery.

Hall, A. (1990) "Using Z as a Specification Calculus for Object-Oriented Systems", **Lecture Notes In Computer Science**, Vol 428, pp 290-318.

Hsu, C., Bouziane, M., Rattner, L. & Yee, L. (1991) "Information Resources Management in Heterogeneous, Distributed Environments: A Metadatabase Approach", **IEEE Transactions on Software Engineering**, Vol 17, pp 604-625.

Ingwersen, P. (1992) "Information and Information Science in Context", **Libri,** Vol 42, pp 99-135.

Ishikawa, H. *et al* (1993) "The Model, Language, and Implementation of an Object-Oriented Multimedia Knowledge Base Management System", **ACM Transactions on Database Systems**, Vol 18, pp 1-50.

Lillywhite, J. (1991) "Identifying Available Spatial Metadata: The Problem", in D.J. Medyckyj-Scott, I.A. Newman, C.L.N. Ruggles, and D.R.F. Walker (eds), **Metadata in the Geosciences**, pp 3-12: Group D Publications.

Litwin, W., Mark, L. & Rossopoulos, N. (1990) "Interoperability of Multiple Autonomous Databases", **ACM Computing Surveys**, Vol 22, pp 267-293.

Lyotard, J.-F. (1979) **The Postmodern Condition: A Report on Knowledge**, Manchester University Press.

Madsen, M.S., Fogg, I.S. & Ruggles, C.L.N. (1994) "Metadata Systems: Integrative Information Technologies", **Libri**, Vol 43, in press.

Neelameghan, A. (1992) "Application of Ranganathan's General Theory of Knowledge Classification in Designing Specialized Databases", **Libri**, Vol 42, pp 202-226.

Quillian, M.R. (1985) "Word Concepts: a Theory and Simulation of Some Basic Semantic Capabilities", in R.J. Brachman and H.J. Levesque, (eds), **Readings in Knowledge Representation**. Los Altos, CA: Morgan Kaufman.

Ranganathan, S.R. (1967) **Prolegomena to Library Classification**, Asia Publishing House.

Ruggles, C.L.N. (1994) "An Integrated Environment for Open Learning: The STILE Approach", in **Proceedings of the Eleventh International Conference on Technology and Education**, London.

Ruggles, C.L.N., Newman, I.A., Medyckyj-Scott, D. & Walker, D.R.F. (1992) "Structuring Spatially Related Meta-information for Effective Entry and Retrieval", **The Computer Journal**, Vol 35, pp A475-A486.

Sowa, J.F. (ed) (1991) **Principles of Semantic Networks: Explorations in the Representation of Knowledge**, San Mateo, CA: Morgan Kaufman Publishers, Inc.

Stepney, S., Barden, R., & Cooper, D. (1992) "A Survey of Object Orientation in Z", **Software Engineering Journal**, Vol 7, pp 150-160.

Van Ryzin, J. (ed) (1977) **Classification and Clustering**, New York: Academic Press.