Information Technology and Legal Education: Towards 2000

9th & 10th April 1992
Sponsored by
Context Ltd; Linklaters & Paines; Needham & James

Mapping Legal Hypertext Systems

Mark L. Macaulay

Centre for Law, Computers and Technology University of Strathclyde

Keywords: Legal hypertext systems - Mapping hypertext links - Book maps - Question maps - Flowcharts - Hierarchical data model - Network data model - Relational data model solution - Commercial development.

Abstract: A significant problem with hypertext systems is that the tracing of links between information is vexatious. A knowledge of the links that exist is important because the information may require to be updated to take into account changes in the law. Traditional structured methods of mapping have proved to be ineffective at mapping hypertext systems, so this paper looks at computer science data models as an alternative solution. An examination is made of the hierarchical, network and relational data models and conclusions drawn as to their relative merits for the task of mapping legal hypertext systems.

1. Introduction

There are primarily two different types of legal hypertext system: document assembly and text retrieval systems. Document assembly hypertext systems, such as Justus' Clerk (Wilson 1991), employ clauses from different precedent documents, combine them together in a new document and probe for discrepancies between clauses to produce an acceptable first draft. Text retrieval systems, such as Hyperlaw (Painter 1990), take various categories of information relating to a designated domain and interconnect the information so thatit is available as a single, coherent form rather than several disparate segments. A text retrieval hypertext system may take the form of statutes and case law relating to the same knowledge domain being interlinked, or a complex contract being linked with its assorted annexes and the arbitration decisions on its clauses. Irrespective of which type of hypertext system we are considering, the power of hypertext emanates from its ability to link related information.

Pre-supposing that the law evolves, what operations are likely to be performed on a legal hypertext system? Using the illustration of a text retrieval system containing statutes, case law and articles relating to the same knowledge domain, then the subsequent operations may require to be effected at some interval to maintain the currency of the system:

- Deletion - new legislation passed by parliament outdating a previous statute contained within the hypertext system;
- Insertion - a court deciding a case interpreting a section of one of the statutes contained within the hypertext system;
- Modification - an article updated taking into account recent case law developments.
- Performance of these operations to maintain the currency of a legal hypertext system cannot be achieved in an efficient manner unless the system is mapped. It should be noted that mapping does not entail forcing a structure on hypertext but on maintaining a record of what information the system has linked together.

This paper considers the mapping of information within a text retrieval hypertext system analogous to the one outlined above, that is, one which contains statutes, case law and articles relating to the same knowledge domain. The rationale behind this is that the number of discrete categories of information which are interlinked is smaller and more clearly demarcated, making the precepts relating to the mapping of hypertext systems simpler to discuss and demonstrate. However, the principles discussed apply equally to text retrieval hypertext systems and to document assembly hypertext systems alike.

2. Structured Methods of Mapping

Typically, structured methods of mapping have focused on the following: book maps, question maps and flowcharts. These methods, and in particular flowcharts, have proved to be reasonably successful at this task More specificaHy, book maps have proved to be adept at mapping Knowledge Presentation Systems, question maps at mapping Computer Aided Learning Systems and flowcharts at mapping Expert systems. A book map fundamentally replicates the appearance of the contents page of a book. A theme of information is partitioned and re-partitioned under headings and sub-headings, so that this reiteration continues until all the information is represented. The headings after the First partitioning has accurred are extremely general, becoming more specific as the amount of partitioning increases. Each time partitioning occurs, that partition level is indented from the last such level to show that further partitioning has in actuality ensued. Where there are distinct information themes on the same knowledge domain, each theme is portrayed as a discrete book contents page and linked together via undirected lines. For example, see Digram 1.

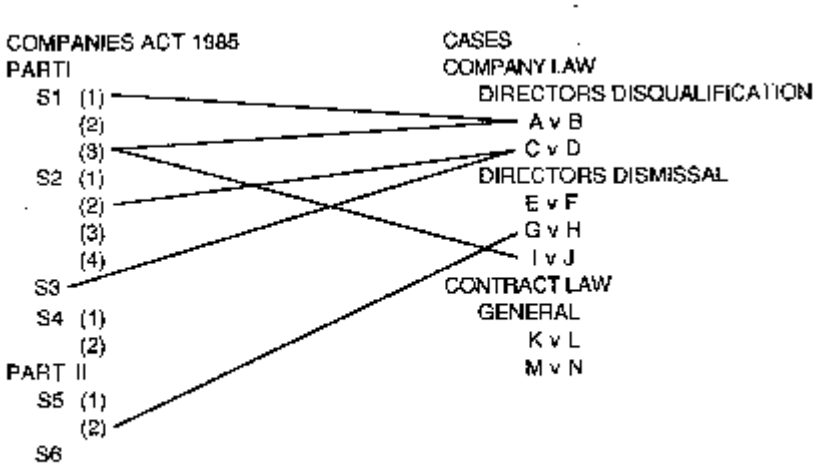

Diagram 1 The Book Map Structured Method

A question map guides the user through a series of boxes, containing questions and answers, until an end result is concluded. The boxes are linked together by directed lines from an answer to a question or a result, so that the answer to each question posed dictates which boxes will be accessed and in what sequence. Each box, excluding the end boxes, contains a question with two or more answers.

When an end box is reached it will contain a result corresponding to the answers the user selected during the question boxes. For Example, see Diagram 2

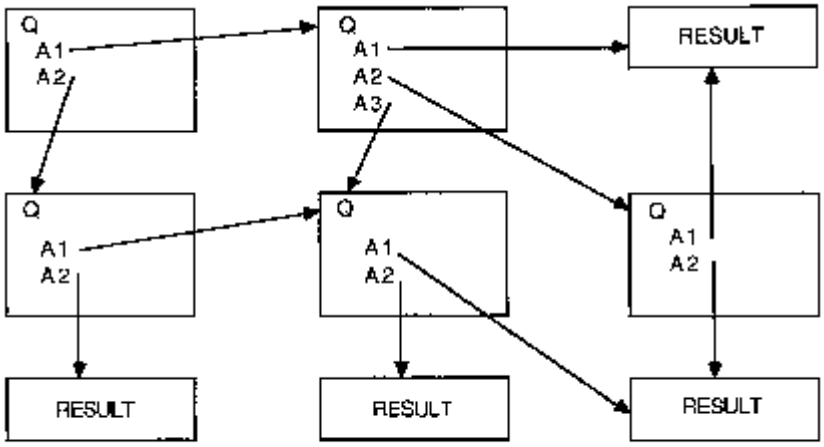

Diagram 2 The Question Map Structured Method

A flowchart is a list of instructions that have to be carried out in a certain order. Essentially, flowcharts commence and cease with start/stop boxes (oval-shaped) and contain intermediary steps consisting of the following categories of boxes:

- process boxes (rectangular-shaped), sometimes better known as instruction boxes;
- decision boxes (rhombus-shaped), sometimes better known as question boxes;
- input/output boxes (parallelogram-shaped), sometimes better known as read/write boxes.

These are only some of the variety of boxes available and the user may encounter online storage, offline storage, auxiliary operation, preparation and visual display boxes as well as others. Traditionally, decision boxes had only two exits, yes and no, but they have been corrupted by computer programmers to allow any number of potential answers so that any number of exits are now feasible from this box. All the boxes are connected by directed flowlines: the direction of the arrows dictates the sequence in which the boxes should be computed, and it is possible to loop round from any one box to another. For example, see Diagram 3

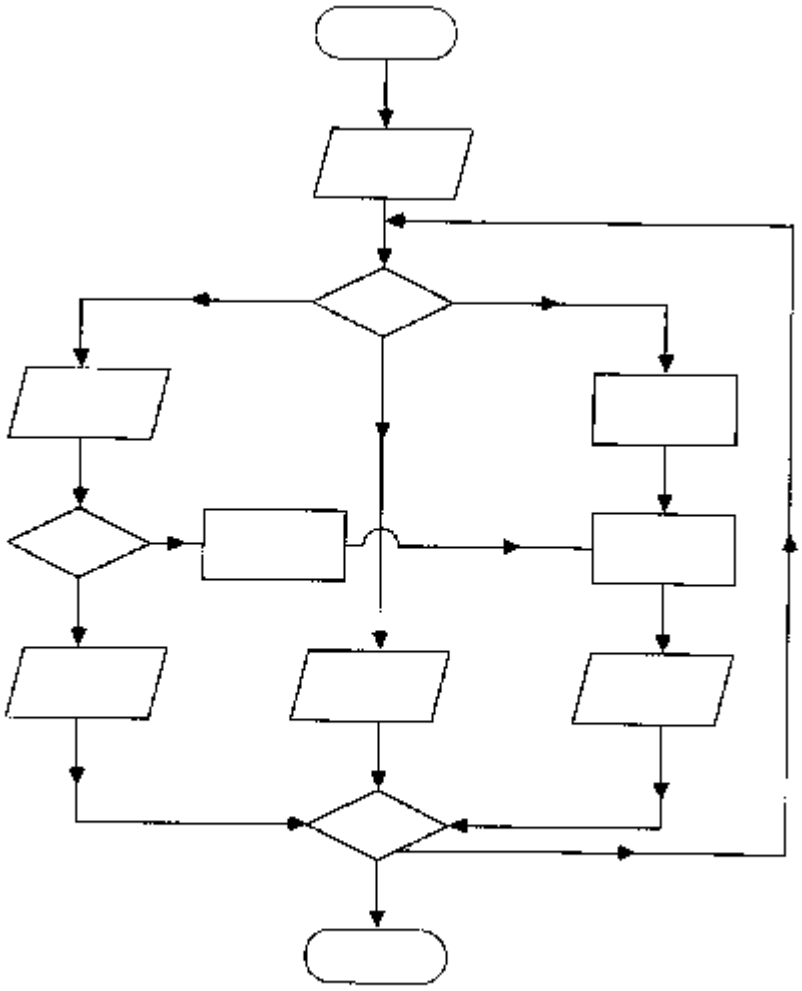

Diagram 3 The Flowchart Structured Method

However, these structured methods have proved to be unable to cope with the nature and magnitude of the cross-reference links contained in hypertext systems. Attempts to map hypertext systems utilising these methods inevitably results in hyper-spaghetti. This term simply refers to the resultant diagrams being an unintelligible morass of links when they are employed to undertake the mapping of hypertext systems.

3. Data models of mapping

Since hypertext is a development from computer science database theory, a solution to the problem of mapping hypertexts may be found by examining database data models. Data models are employed to specify how the separate pieces of information in a database are related. The most valuable level to examine in relation to mapping will be the conceptual schema of the data models because it portrays a representation of the information which is interjacent to how the computer stores it and how the user visualises it, resulting in a model which is not so low level that we have immense difficulty in understanding it yet not so high level that it becomes a morass of links. The conceptual schema in fact consists of an abstract description of the various types of entity that need to be processed in any way by the enterprise in question. There are three data models: the hierarchical model, the network model and the relational model.

3.1 The Hierarchical Data Model

The hierarchical model is historically the oldest of the data models, and the most elementary to conceptualise. It therefore seems logical to commence by examining it first. The hierarchical model exists as a tree structure consisting of entities. Each entity contains records, and each record is composed of attributes. The records are connected to each other via links. For example, statute is an entity containing records composed of a statute name and section number.

There are a number of precepts associated with the hierarchical data model which must be observed:

1. No entity can be connected to more than one object higher up in the hierarchy,although it can be connected to several entities at a lower level. This means that there can only be one enhty at the top of the tree; this entity is termed the root. The hierarchical model is sometimes designated as the 'Parent-Child Model', so that the rule may also be expressed by saying that parents may have many children but each child, except the root which has none, has only one parent.
2. There cannot be any self-referencing links. That is, a record type cannot be both the owner and a member of the same set type.

Mapping in the hierarchical model takes place in three stages. First, tree hierarchies are created. Entities are mapped into record types, which are diagrammatically represented by rectangular boxes.

Then the relationships between these record types are mapped into hierarchies (infra). Secondly, attributes are associated with the record types: the attributes associated with a record type displace the record type name from the box, and it becomes a label for that box instead.

Thirdly, records are associated with the record types, which are then linked to their children within the hierarchy. The linking is shown as a directed arrow from parent to child.

The classes of hierarchy which may exist between entities (A, B, C, D and E) in the hierarchical model are:

a) One-to-one (1:1) relationships

Entity A (root) is the parent of entity B.

b) One-to-many (1:M) relationships

Entity A (root) is the parent of entities B and C. As well as being a child of entity A, entity B is also the parent of entity D.

c) Many-to-one (M:1) and many-to-many (M:N) relationships

Many-to-one and many-to-many relationships may be represented as a hierarchy, but only by instituting replication into the tree. For example, if A is the parent of B and C, and B is the parent of D, but E is also the parent of C and D then this would be represented as 2 separate trees.

We can represent the relationship as a multiple hierarchy database rather than a single hierarchy one, but replication still remains. However, replication may be virtually eradicated by the use of virtual records. A pointer associates a virtual record in one tree with an actual record in another tree. The virtual record replaces a redundant copy of the actual record with a reference to the actual record.

Entities C and D in the second tree are replaced by virtual records which point to the entities C and D in the first tree, thus eliminating the replication of those two entities.

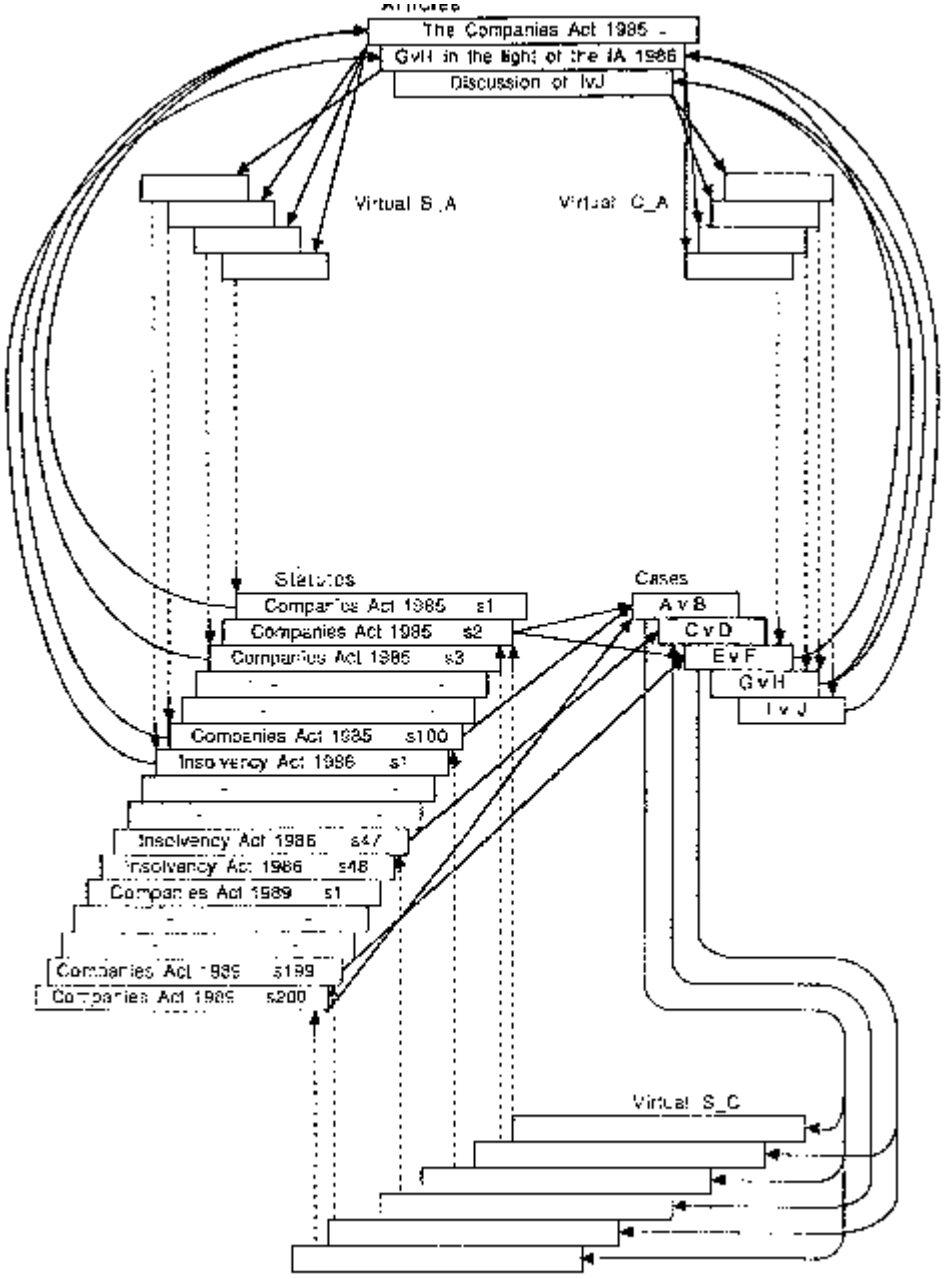

Diagram 4 The Hierarchical Data Model

Statutes, cases and articles have many-to-many-relationships between each other. As a result, virtual records require to be utilised to represent this situation. Had it simply been the case that there was a one to many relationship between statutes and cases and articles, the need for virtual records would be redundant: the relationship would be represented simply as directed arrows from the statute records to the case and article records. It is obvious that the hierarchical model has difficulty in representing many-to-many relationships. It should also be noted that it is not possible to map two records from one entity together, for example, it is impossible to map a reference to s2 of the Companies Act 1985 from s199 of the Companies Act 1989 using this model.

Although the hierarchical model is the simplest data model to conceptualise, a number of problems exist with it in relation to mapping hypertext systems. Navigation in a hierarchy is complicated, and altering the structure once the model has been established is also complicated. For example, parent records are necessary to the existence of all child records, so that deleting the former entails the deletion of the latter. Also, the model does not permit self-referencing links, and more vitally, there is a lack of adequate means to represent entities that have many-to-many associations. Many-to-

many relationships are modelled using virtual records, which is both complex and involves a degree of duplication. Mapping hypertext systems with the hierarchical model is therefore not a particularly viable option.

3.2 The Network Data Model

In the network model, a record represents an entity set or relationship set and is associated with other records via links. A given record occurrence may have any number of immediate dependents, as well as any number of immediate superiors. Associations are represented by the non-mathematical concept of a co-set, which merely means that it intersects records. Thus, the co-set is the network representation for a relationship. There are a number of precepts associated with the network data model which must be observed:

1. There cannot be any self-referencing links. That is, a record type cannot be both the owner and a member of the same set type.
2. An instance of a record type cannot exist in more than one instance of a specific set type at a time. This means that many-to-many associations are not directly irnplementable.
3. Many-to-many associations are represented in a network by interposing an intermediate record type between the two record types having a many-to-many association.

Mapping in the network model (Oxborrow 1989) takes place in three stages. First, Bachrnan diagrams are created. Entity sets are mapped into record types, which are diagrammatically represented by rectangular boxes (supra). The relationships between the entity and relationship sets are then mapped into co-sets (infra). Secondly, attributes are associated with the record types. The attributes associated with a record type displace the record type name horn the box, and it becomes a label for that box instead (supra). If any link boxes are created (infra), they are left undesignated and contain a combination of the primary keys of the entities they are linking.

Thirdly, records are associated with the record types and link boxes, if any exist. Records of record types are thereafter linked by directed lines to either records of other record types or records of link boxes and then re-linked back to themselves.

The classes of co-set which may exist between entities (A, B, C and D) in the network model are:

a) One-to-one (1:1) and One-to-many (1:M) relationships

i) Singular Co-set

A singular co-set has one occurrence only in the database. It can be considered as a single set in its own right.

ii) Network

The record type C is a member in both the A-C and B-C co-sets. That is, ithas two owners.

iii) Multi-member Co-set

The record types B and C are members of the same co-set, which is owned by record type A.

iv) Hierarchy

The co-set A-B represents a single level hierarchy, while the complete structure represents a multilevel hierarchy.

Relationships are mapped into co-sets with the arrows directed towards the 'many' end in a one-to-many relationship, and towards the most appropriate record type to represent the member of the co-set in the case of a one-to-one relationship.

b) Many-to-many (M:N) relationships

The many-to-many relationship uses a link record type to link A and B. The link record type performs a similar function to the relationship relation of the relational model (infra).

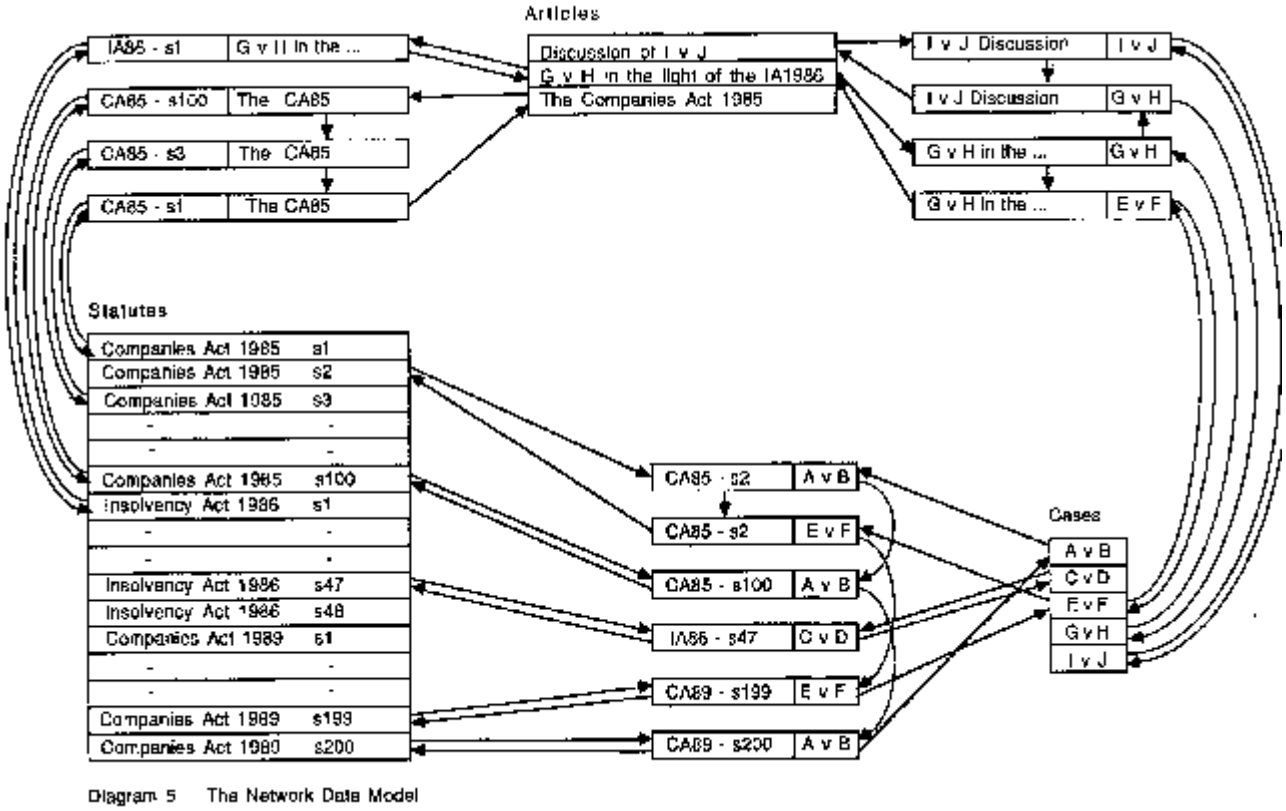

Diagram 5 The Network Data Model

Diagram 5 shows an example of the network model. As a result of the many-to-many relationships between the record types in the diagram, intermediary link boxes require to be created and linked to the record types. Whilst the network model copes with many-to-many relationships rather better than the hierarchical model, the number of directed lines going to and from the link boxes make the links difficult to trace. Again, there is no provision for mapping two records from one entity together.

The network model overcomes the problem of having to duplicate information by introducing the concept of co-sets. Thus, a network permits many-to-many relationships to be represented without the data duplication necessary in the hierarchical implementation. However, navigation in a network is complicated, and many-to-many relations can only be represented by an indirect set. More crucially, the model does not permit self-referencing links. The solution to mapping hypertext systems does not therefore lie with the network model either.

3.3 The Relational Data Model

The concept of the relational database model was introduced by E.F. Codd in the late 1960s (Codd 1970). It is founded on the mathematical theory of relations and sets. A relation represents an entity set, and the attributes of a relation are characteristics associated with the entity set which it represents. A relation is designated by its name and associated attributes. For example,

STATUTE (Name, Section_Kurn)

where 'STATUTE' is a relation which is described by the associated attributes of its name and section number. Record occurrences in a relation are definitively termed tuples. The underlined attribute, 'Name', is the primary key of the relation: the primary key is the attribute (or attributes) which uniquely identifies a tuple in a relation.

There are a number of precepts associated with the relational data model, which must be observed:

1. No two tuples in a relation should be identical. This facilitates each tuple in a relation being uniquely identified by means of one or more attributes in the relation.
2. No primary key and no element of a multi-attribute primary key may have a null value.
3. The value of attributes in one relation which are primary keys of some other associated relation must either be null or correspond to an existing primary key value in an associated relation.

Mapping in the relational model (Oxborrow 1989) takes place in four stages. First, relations are created for the entity sets: one relation is created for each entity set, which contains the attributes associated with that entity set. The entity identifier, where it exists, becomes the primary key of the relation. Secondly, the relationships between the entities must be mapped. There are two predominant techniques of snapping relationships:

a) The Relationship Relation Approach

One relation is created for each relationship, with the attributes consisting of the primary keys of the related entities; and

b) The Foreign Key Approach

For each set of related entities, the primary key(s) are placed in one of the entities of the foreign keys, where such action does not conflict with the semantics of the conceptual model.

The Relationship Relation Approach (infra) will be used in this paper because the Foreign Key Approach tends to be problematic (Chen 1976), the principal reason for this being that it is incapable of representing many-to-many relationships. Thirdly, the relations require to be completed: identifiers (primary keys) should be provided for relations which do not have them. For example,

CASE (Name) would become CASE (Case#, Name).

The # symbol simply represents a number given to each case. Fourthly, records are associated with the relations.

The classes of relationship which can exist between two entities (A and B) in the relational model using the Relationship Relation approach are:

a) Many-to-many relationships (M:N)

These relationships are represented in the relational approach by constructing a new relation with no attributes apart from the primary key which is formed from the primary keys of the entities which are related. For example,

A-B (A#,B#).

b) One-to-many relationships (1:M)

i) Mandatory in the 'many' direction

An entity of type B can only exist if it is related to an entity of type A. The primary key of A can be placed in B as an additional, extrinsic key, attribute. For example,

A-B (B#, ..., A#).

ii) Optional in the 'many' direction

An entity of type B can exist without being related to an entity of type A, so this type of relationship will be represented by a relationship relation with B# as the primary key. An occurrence of A-B will only exist for those B entities which are related to an A entity. For example,

A-B (A#, B#).

c) One-to-one relationships (1:1)

i) Mandatory

Either the primary key of A may be placed as an extrinsic key in B, or vice-versa. For example,

A-B (8#, ..., A#).

ii) Optional

A relationship relation is created containing the primary keys of A and B, and either one of these may form the primary key of the new relation. For example,

A-B (A#, B#).

iii) Contingent

The primary key of the entity in the optional direction is placed in the relation representing the entity in the mandatory direction. For example,

A-B (B# ....., A#).

The resulting relational model can intelligibly be depicted in tabular form. Its tabular format allows simple, high level navigation by users, so that insertion, deletion and modification can be executed expediently with the resulting tables. The veritable power of the relational model can then be perceived in its proficiency at exercising queries over entire, and conjoined tables.

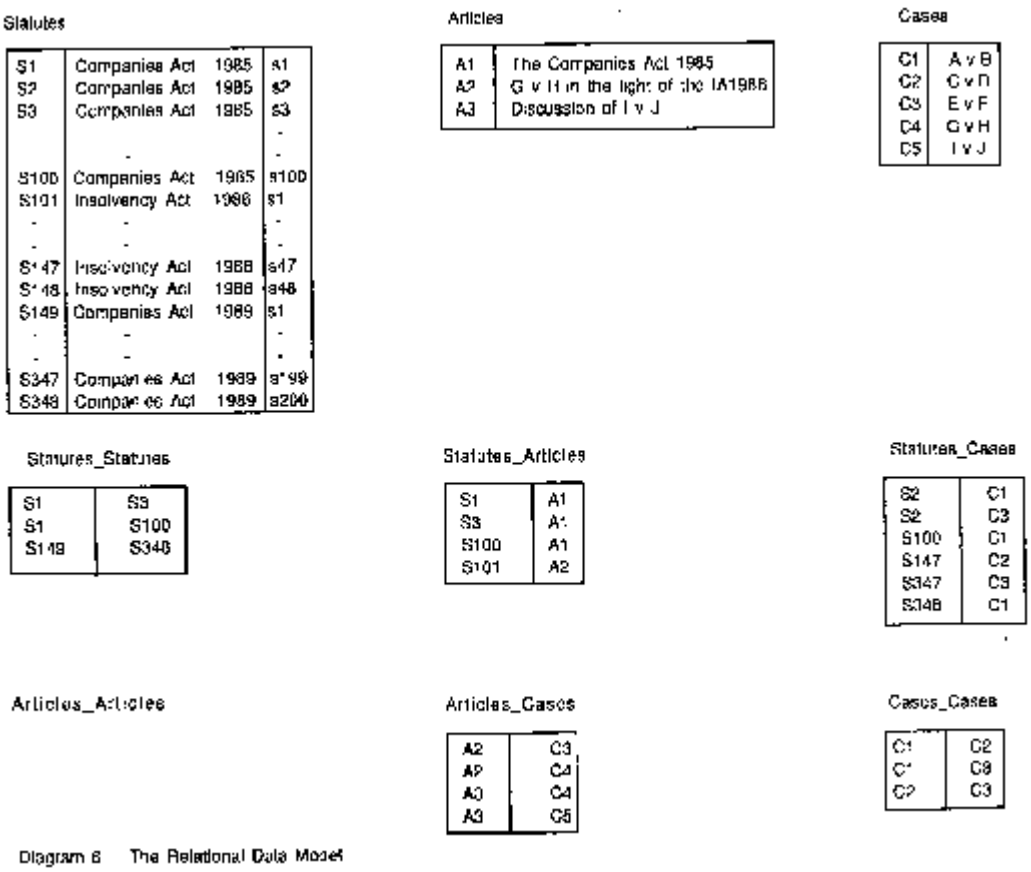

Diagram 6 The Relational Data Model

Digram 6 shows a relational model. In this model, many-to-many relationships are clearly and simply presented as conjoined tables. It is also possible to map two records from one entity together using this method.

The advantages of the relational model stem from the fact that relations can be conveniently entered as tables. It can therefore represent many-to-many relationships graphically as combinations of vertical and horizontal subsets of tables. Details of the underlying implementation of a relational database can be hidden behind its tables, so that data can be manipulated and viewed using a variety of strategies, while the logical independence of its organising structure is retained. The relational model therefore succeeds as an efficient method for mapping hypertext systems.

4. Conclusion

Traditional methods of mapping structures have been demonstrated to be inept at coping with the nature and magnitude of hypertext links and result in hyper-spaghetti. Hypertext is a development from computer science database theory, so that data models appeared plausible candidates for furnishing a solution to mapping hypertexts. The hierarchical and network data models were also unable to contend with the nature and magnitude of hypertext links, essentially because of their inability to manage entities which had many-to-many relationships efficiently and their inability to allow self-referencing links. A solution to these obstacles was found with the relational data model. The success of this model with regard to mapping hypertext systems is primarily because of its proficiency in mapping many-to-many relationships between entities and its capability to create self-referencing links.

On reflection, it should not have been particularly surprising that the relational model is capable of mapping hypertext systems. This is because there are a number of reasonably prominent similitudes between hypertext systems and relational database systems, the foremost ones of which are:

- they are both employed to organise and manipulate vast amounts of similar information;
- use may be interactive rather than pre-specihed, that is, the user determines the sequence to be observed during operation;
- users stipulate what information to locate, not how or where to search for it;
- proxies for linking, retrieving and displaying information may be furnished as default options for users
- there are levels of abstraction, that is, it is possible to peruse the basic information at a top level or the same information in more detail at a lower level of abstraction;
- information may be combined dynamically with link arrangements.

It can therefore be seen that the relational model exhibits the cardinal characteristics of a hypertext system

What is it therefore that makes a hypertext system diHerent from a relational database system? The foremost divergence is that hypertext permits the user to construct a loose network of connections between various topics, going beyond conventional databases to multiply vastly the manner in which a user can access the information. However, there are a number of less salient, yet significant disparities between hypertext and relational databases:

- databases singularly calculate results and generate reports whereas hypertext systems may do this, or more probably, allow browsing and exploration of the information;
- databases consider text and numbers exclusively whereas hypertext integrates text with miscellaneous media, such as sound, graphics and video;
- databases select information using query languages and relational operators whereas hypertext selects information using natural language and icon links;
- databases place priority on preserving logical form whereas hypertext places priority on objects.

It can therefore be seen that hypertext manages information as objects in natural language form whereas databases manage information as files, that is to say, the information structure in databases is imposed but in hypertext is content-oriented.

The purpose of snapping hypertext systems is not to force a structure on them; this would be contrary to the nature of hypertext. The notion behind hypertext is that the author of the text establishes a number of alternatives for readers to explore, and each individual reader determines for him or herself which alternative to follow at the time of reading the text. Mapping a hypertext system is therefore designed only to maintain a trace of all the links instituted to enable easy insertion, deletion and modification of the links so that a legal hypertext system can readily be kept current. This is indisputably of manifest concern to lawyers in relation to hypertext systems, and the relational data mode) permits this operation to be accomplished proficiently.

Whilst the relational model presents a viable option for mapping legal hypertext systems, it can be tedious creating and maintaining the tables required. The ultimate solution is undoubtedly to be found in the commercial development of a hypertext shell which creates and maintains the relational tables automatically, so that modification of legal hypertext shells becomes an inherently efficient and error free task.

References

Chen P., The Entity-Relationship Model - toward a united view of data, ACM Transactions on Database Systems, vol. 1, no. 1, March 1976.

Codd E., A Relational Model of Data for Large Shared Data Banks, Communications of the ACM,

vol.13, no.6, June 1970.

Oxborrow E., Databases and Database Systems (2nd edition), Chartwell-Bratt Ltd., 1989.

Please note that the concepts involved in the network and relational models are to a large extent culled from this book.

Painter D., Hyperlaw, LTC & BILETA Newsletter, vol.2, no.3, Spring 1990.

Wilson E., A Hypertext Interface for Automated Document Handling, Law Technology Journal, vol.1, no.1, October 1991.

Author

Mark Macaulay
Centre for Law, Computers and Technology
University of Strathclyde
173 Cathedral Street
Glasgow G4 0RQ