Informatica e diritto, XX annata, Vol. III, 1994, n. 2, pp. 177-194

Exploiting Document Structure in Retrieval from Legal Hypertexts

TREVOR J.M. BENCH-CAPON, PAUL E.S. DUNNE, GEOF STANIFORD

1. INTRODUCTION

There are many different types of documents to be found in the legal domain: legislation, both primary and secondary, commentaries, case reports, guidelines issued to adjudicators, and information supplied to members of the public, to name but a few. For a given task it is often difficult to predict which material will be relevant: general principles, cases relevant to other areas of law, and other legislation may be needed for a balanced understanding of a particular question. A person trying to get what is needed for a particular purpose must extract from this large amount of potentially relevant documentation the specific material needed in the given situation, with reference both to the current level of background knowledge possessed by the reader and the problem to be solved.

Because it is so central to almost every task in the legal domain, the retrieval of documents has received much attention in the application of computers to law. Early applications of computers centred on conventional retrieval systems, such as LEXIS and WESTLAW, which rely on Boolean searching methods. Whilst of great general utility, these techniques are often only of limited value for supporting specific tasks. In particular, problems arise from the fact that one legal concept may be represented by a variety of different words, and that the context may lead to one and the same keyword being used to express a range of different concepts. Relevance must be determined therefore not by the presence of particular words, but by the underlying concepts with which the document deals.

This has led to much investigation being directed towards the «conceptual» retrieval of legal documents. Examples of such techniques can be found in Bing (1987), Hafner (1987), and, most elaborately, in Dick (1991). These approaches have achieved some impressive results in experimental work, but they all require a formidable degree of skilled analysis both of the domain and of individual documents. Together with some other obstacles, this has so far prevented these techniques from being scaled up to realistic practical applications.

Connectionist methods of retrieval hold some promise of reducing the amount of expert analysis needed, by automating the recording of associations between documents. Example of this work may be found, applied to the legal domain, in Belew (1987), and, as a general technique, in Boughanem and Soulé-Dupuy (1992). Again these techniques have had their successes, but the empirical results are not entirely conclusive, and the improvements in recall and relevance have yet to be achieved with consistency.

In recent years hypertext has become very popular. Wilson (1988) describes an early system which presents legal material as a hypertext. Hypertext seems to hold out a great deal of promise for the legal domain in that the grain size of the documents is smaller, avoiding problems whereby the relevance of a passage of a document is lost in a generally irrelevant context, and by the use of semantic links between the document fragments which directly support the conceptual association of these fragments. In a hypertext, materials are primarily located by a strategy of browsing these semantic links. It must, however, be recognised that browsing is a strategy developed from recreational reading, where the reader is not seeking specific information, and so is not directed towards situations where there is a definite task in view, which is normally the case when a legal information retrieval system is being used. Hypertext also has another disadvantage: the smaller grain size means that there is an ever present danger of material being encountered out of context, and so misinterpreted. Linear documents are organised by their authors and this gives them a structure which can be lost when they are transformed into a hypertext. In some domains this may not matter, but it is a serious issue in the domain of law. Legal documents tend to conform to a fairly rigid conventional structure, and this structure makes important contributions to the meaning of the document. As is argued in Routen and Bench-Capon (1991), the need to capture this structural meaning is critical.

Moreover, it must be recognised that structure is important not only with respect to the the source documents: the document the reader is trying to obtain (and a reading of a hypertext can usefully be seen as the retrieval of a customised document) will also have a desired structure, relating to the target reader and the goals of that reader. This can be clearly seen in much of the work done on document preparation systems, such as Sprowl (1979), Morris (1987) and Branting (1993).

The aim of this paper is to suggest ways in which the undeniable benefits of the use of hypertext can be gained while avoiding the problems described earlier. The key idea is to respect and record the structure of the documents involved, so as to direct the traversal of the hypertext to the task in hand. We will therefore be concerned with the description of a document manipulation formalism that addresses the problems of representation and retrieval in a way which exploits the structures that are present in the documents found in the domain and so to counteract the problems exhibited by existing systems. The key elements of the formalism may be summarised as follows:

- Representation of documents as hypertext graph structures that are constructed and updated through use of the graph modification rule paradigm amplified in Bench-Capon and Dunne (1989), so as to capture the physical and conceptual organisation of the documents.
- Exploitation of *linearisation schema*, as introduced in Bench-Capon et al. (1992), as a formal mechanism both for extracting hard-copy documents from the system and as an aid to traversal of the hypertext.
- Definition of an *agent based* system to control searching and traversal of the hypertext, developed using the agent formalism propounded in Staniford et al. (1993), Staniford (1993) and Staniford and Dunne (1994).

2. Constructing Representations According to Document Structure

Hypertext systems are increasingly advocated as the best means of representing and perusing large collections of documents. A hypertext may be viewed as a labelled directed graph in which the nodes of the graph correspond to textual blocks and the edges describe relationships between different sections of text. It has been the convention that the edge labelling is viewed as a semantic net covering the content of the document.

Formally, a hypertext has been defined as

Definition 1: [Bench-Capon et al., 1992] A *hypertext, H*, over the character set (or alphabet) Σ is defined by a quintuple:

$$H \equiv (V; E; \lambda_V; \lambda_E; \chi)$$

where $V = \{1, 2, \dots, n\}$ is a finite set of nodes; $E \subseteq V \times V$ is a finite set of links; $\lambda_V : V \rightarrow \Sigma^*$ is a node labelling function; $\lambda_E : E \rightarrow S^*$ is a link labelling function; and $\chi : V \rightarrow \Sigma^*$ is the mapping describing the textual content of each node in the hypertext. •

While hypertext allows for more flexible traversals of text, it has been observed that this very flexibility presents considerable difficulties even to experienced users of hypertext. Users may become disoriented in a network with an intricate linkage structure and they may be unable to find the precise section of text in which they are interested, and unable to integrate the information they do retrieve into a coherent whole which will meet their information needs. It can be argued that these problems arise for two reasons: many hypertexts are, initially, constructed from conventional linear source texts, which were of course written and intended to be accessed as such, and the processes by which such sources should be 'hyperised' so as to reflect the original organisation imposed by the author are poorly understood. Secondly, since the linkage structure may be modelled on a semantic net, there are the problems of subjective interpretation of relation names that have been well discussed by Brachmann (1975) and others. A further difficulty that has been identified is that often material is ultimately needed in a conventional hard-copy format and so a mechanism is needed by which relevant information can be extracted in a sensible order from the mass of the hypertext. This, the *linearisation problem*, has been the subject of much investigation, see e.g. Simpson and McKnight (1990). Its difficulty is exacerbated by the fact that different classes of user, having different purposes in mind, will wish to see different linearised routes through the same hypertext. Thus what is needed is a linearisation strategy which can be tailored to specific tasks.

A potential solution to these problems has been suggested in Bench-Capon and Dunne (1989). Following the observations of Stotts and Furuta (1988), Koo (1989) and others we noted that different classes of document are associated with different organisational structures – these structures being implicitly understood and observed by authors and readers of documents in a specific class. Any document can also be seen as a graph of text nodes linked by edges.

Thus in Bench-Capon and Dunne (1989) we defined a document graph as

Definition 2: A *document graph* is a directed acyclic graph, $G(V,E)$. The nodes in V denote *objects* in a document and the edges in E depict logical connections between objects. •

Notice that such document graphs are specialisations of general hypertexts since they enforce an acyclicity condition. In the more general framework introduced in Bench-Capon et al. (1993b), different graph-theoretic structures, of increasing richness, are used to model a hierarchy of document forms progressing from linear text through to hypertext. Document graphs appear as an intermediate stage of this hierarchy. Moreover, the types of nodes permitted in a document, the edges that link them, and the grammar which determines which edges may link which nodes are determined by the class of a document.

Formally, the class to which a document belongs is described by its *specification*, thus in Bench-Capon and Dunne (1989) we have

Definition 3: A *document specification* is a pair $DS = (C, Init)$. Here C is a finite set of *constraints* $\{C_i : 1 \leq i \leq k\}$ where each C_i is a computable predicate on document graphs. *Init* is a set of *initial* document graphs. Given a document specification, DS, and a document graph G, G is said to satisfy specification DS if $G \in Init$ or $C_i(G)$ is true for each constraint in C. •

With this definition, by making explicit the structural characteristics that typify a class of documents, we can construct representations consistent with this structure and determine whether modifications proposed subsequently will preserve it. Some example specifications of document classes, e.g. for mathematical papers, were given in Bench-Capon and Dunne (1989).

We are now in a position to give a specification of the structural constraints on particular document classes, and so specify these constraints for the various classes of legal document that we may wish to have available in our hypertext. The resulting graph may be seen as providing an overlay to the hyperised text which will mean that some of the links in the hypertext will reflect these structural links derived from the class of document stored.

Thus the initial hyperisation of a text should be produced in a manner consistent with the implicit structures of the documents being hyperised, present in any class of document but of especial importance in legal documents. Any subsequent changes to the hypertext must only be permitted if they preserve this structure. In order to facilitate this process, Bench-Capon and Dunne (1989) illustrated how simple graph grammars may be used to control modifications to the resulting hypertext.

The mechanism employed is termed a *graph modification system*, and is defined as

Definition 4: A graph modification system (or GMS) is a finite set

$$S = \{R_1, R_2, \ldots, R_m\}$$

of *graph modification rules*. Each rule is a triple $< P, G_p, G_R >$ where P is a predicate on document graphs and G_p, G_r are document graphs. A rule operates as follows on a document graph G: if $P(G)$ is satisfied then any instance G_l in G may be replaced by G_r to yield a new document graph F. •

It should be noted that this structural specification is not primarily concerned with the physical layout of text, as is the case for existing systems such as SGML or ODA (1985), but with the *conceptual* organisation of the text.

By observing these conventions, it is possible to ensure that the process of hyperisation does not lose the structure of the source documents: so that the original context of a particular text node remains available, and the meaning implicit in the structuring of the original document is respected.

3. Retrieving Information in a Structured Form

The above discussion concerns the creation and maintenance of hypertext forms of document in order to preserve the structural coherence expected from highly organised documents. Such an approach, however, can also be used when recovering information from a hypertext representation, be it as an orthodox hard-copy document or in the process of navigating the hypertext to recover particular data. Thus, the linearisation and navigation problems associated with hypertext can also be addressed by exploiting the underlying structure of texts.

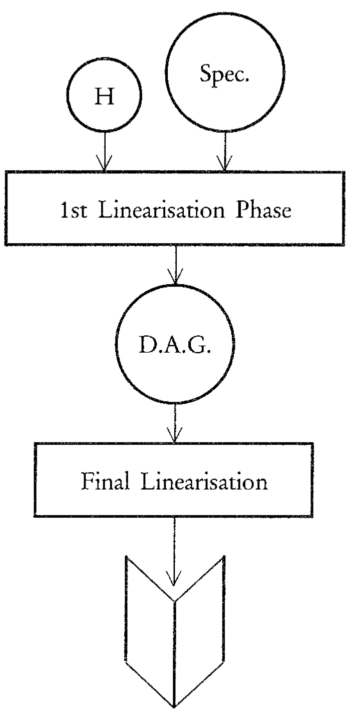
A problem faced in the navigation and linearisation of hypertext forms is that different user classes have widely differing requirements. Users will typically be performing well defined tasks and have specific information needs, so that the document that they are trying to extract from the hypertext itself has a structure capable of specification. These needs, and the diversity of forms they may take, suggest that in order to facilitate the extraction of relevant sections of the hypertext network, mechanisms to aid traversal are needed: such traversal guides must be customised to recognise the structures and information with which different user classes are concerned. We have argued above that classes of document exhibit particular structural properties and thus, in dealing with the traversal of hypertext to relate to specific user classes, it seems natural to use the graph-theoretic form of the document structure expected to be retrieved as a guide to navigating the hypertext.

In Bench-Capon et al. (1992,1993) a formalism respecting such an approach is described. The idea underlying this formalism is to extract from the complete hypertext form a document graph, in the sense of

Definition 2, that conforms to a specification (in the sense of Definition 3) of a class of documents of interest. The document graph output can be seen as an 'intermediate' form between the hypertext and the specific information required. The use of such an intermediate form has several advantages: since its structure conforms to the structural specification of the class of documents of interest it will be easier for the user concerned with the class to navigate and extract information from; secondly, since the graph form is directed and acyclic, the intermediate form contains a finite collection of well defined source-to-sink paths: the textual information obtained by traversing the nodes in these will correspond to a suitable user-specific linearisation of a portion of the hypertext. Of course, in order to tailor intermediate forms produced to specific user needs, it is necessary to describe the class of documents of interest to the traversal system. This we do by means of a *linearisation schema* and *target graph specification.*

An overview of the retrieval process is given in Fig. 1, where a specification of the target document is used in conjunction with the hypertext H to extract a document graph representing the set of retrievable docu-

FIG. 1. *Specification Led Linearisation of Hypertext*



ments which fall within the specified class. This document graph can then be used in the final linearisation phase to retrieve the specific information required.

The specification form has to be amenable to manipulation by the linearising system whilst being relatively straightforward to generate.The basic component of such a document graph class description is called a *target graph specification*. Recall that with the nodes and links of a hypertext we have associated labelling relationships λ_V and λ_E respectively. If S is an arbitrary set and λ_S a labelling relationship for S then we denote by *Names* (S) the set

$$Names\,(S) \;=_{def}\; \{\alpha \in \Sigma^* : \angle x \in S \text{ such that } \lambda_S(x) = \alpha\}$$

Definition 5: An *extended regular expression* over Σ with connection set Λ is any expression built as follows:

1. $\forall\, \sigma \in \Sigma$, σ is an extended regular expression.
2. If S and T are extended regular expressions, then so are:
   2.1. $S \oplus T$ (alternative)
   2.2. $S \rightarrow T$ (connection)
   2.3. $S \xrightarrow{\lambda} T \;\forall\, \lambda \in \Lambda$ (labelled connection)
   2.4. (S) (bracketing)
   2.5. S^* (repetition)
3. All that are extended regular expressions arise by reason of (1) and (2) alone. •

Definition 6: A target graph specification is any extended regular expression over *Names* (λ_V) with connection set *Names* (λ_E). •

The formalism described in Definitions 5 and 6 has the property that there is an easily computable mapping from target graph specifications onto document graphs. Let *ERE* (Σ, Λ) denote the totality of all extended regular expressions over Σ with connection Λ set and (Σ, Λ)-dag denote the set of all directed acyclic graphs whose nodes are labelled with elements of Σ and whose links may be labelled with elements of Λ. Notice that any graph in (Σ, Λ)-dag may be viewed as a document graph. With these we can define a mapping:

$$\mu : ERE\,(\Sigma, \Lambda) \rightarrow \textit{Subsets of}\,(\Sigma, \Lambda)\text{-dag}$$

by using the rules:

1. If $\sigma \in \Sigma$, then $\mu(\sigma)$ is a single node labelled σ.
2. If $S, T \in ERE\,(\Sigma, \Lambda)$ then:
   2.1. $\mu(S \oplus T)$ is formed by taking a graph, S' from $\mu(S)$ and one, T'

from $\mu(T)$ and forming the graph consisting of a copy of S' and a copy of T' with no links connecting the two.

2.2. $\mu(S \rightarrow T)$ is formed in a similar way to $\mu(S \oplus T)$ except that unlabelled links are added from each sink node of the S' to each source node of T'.

2.3. $\mu(S \stackrel{\lambda}{\rightarrow} T)$ is formed in the same way as $\mu(S \rightarrow T)$ but with the added links being labelled λ.

2.4. $\mu((S))$ is identical to $\mu(S^*)$.

2.5. $\mu(S^*)$ is either the empty graph or is recursively given by $\mu(S \rightarrow S^*)$.

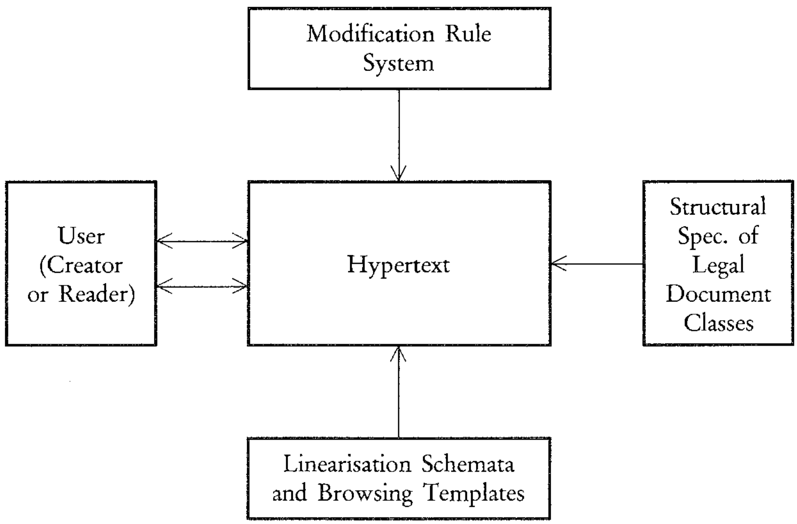
Target graph specifications tied to the document graph specification of a particular class of documents can be used to inform the traversal and retrieval of information from a complex hypertext representation. The specification effectively describes node and link labellings that are relevant to a class of structured documents and how these labellings are permitted to interact. The degree of control provided simplifies the process of filtering the required text from the hypertext form. Of course, the algebraic formalism employed in constructing a specification renders it unreasonable as a mechanism to be employed by users directly. In practice such specifications would be computed and stored as part of the traversal mechanism and hence their actions would be transparent to users trying to retrieve information in a specific form.

While the use of target graph specifications allows arbitrary document graph classes to be captured, in practice there are some drawbacks to its use. In particular, since only structures that are explicitly represented within a hypertext can be extracted, it is not easy to recover document forms that are implicitly present, e.g. different orderings of bibliographic details. In Bench-Capon et al. (1993) an extension of the specification structure is described in which the concept of a *linearisation schema* is developed. Such a schema provides for the representation and retrieval of document forms which may not be explicitly linked into the hypertext representation, so as to allow for the retrieval of documents of classes other than those used to construct the hypertext. Linearisation schemata are defined in terms of various operators defined over target graph specifications and thus these schemata are what should be used to control the traversal process.

Given the representation of legal texts as graphs reflecting their structure outlined above, navigational templates for browsing and linearisation of the legal hypertext so as to retrieve the information in a task directed manner can be built.

To summarise this section we believe that many of the common problems of hypertext representations may be overcome by rigorously controlling the manner in which such representations are constructed, developed and accessed. The use of document graphs and graph grammar based modification rule systems appears to provide such a control structure. The schematic below, Fig. 2, depicts the main components of the legal document storage system described above:

FIG. 2. *Components of Structure-based and Retrieval System*



4. Agents for Controlling Search with KBS Support

The mechanisms described above support retrieval by ensuring that any material that is retrieved will be presented in a sensible fashion. These mechanisms may, however, be enhanced considerably by the provision of further support to the users of the system. What is required additionally is that the material so presented will be the material which the user will find most useful. Whilst the form of the retrieved material is a necessary constraint, it is, of itself, insufficient: the user does not simply want a legitimate linearisation, but some particular linearisation that will fit the current problem. In other words some «intelligent» support is required to facilitate the traversal.

The relationship between hypertext and knowledge based systems has been observed in Barlow et al. (1989, 1990) and further expanded in Bench-Capon and Dunne (1990). Given that what is required is the judgemental choice between structurally acceptable linearisations, use of selection heuristics encapsulated using KBS techniques, seems an obvious development. In order to guide the navigation and linerisation of texts, therefore, we propose to use knowledge based techniques, so as to incorporate a variety of traversal heuristics. In order to provide an intermediary between the document base and the knowledge base we propose employing a system of communicating agents, each with a formally prescribed set of actions and responses.

The concept of intelligent cooperative information systems is currently evolving from several, until very recently, disjoint technologies including database systems, artificial intelligence, programming languages, software/knowledge engineering, distributed computing, and office information systems, to mention just a few, e.g. Papazogolou and Zeleznikow (1992), Huhns et al. (1993). A number of authors have observed that communicating multi-agent systems offer a rich and flexible design paradigm for the development of such intelligent distributed systems, e.g. Bond and Gasser (1988), Werner (1992).

Genesereth and Nilson (1987) described an elegant, logic based architecture, for computational agents but it lacked the features required to provide agents with the ability to communicate with one another. Building upon Genesereth and Nilson, Staniford et al. (1993) define an extensive formalism which deals with communicating computational agents as a control mechanism for co-operative writing strategies. An agent in the sense used by those authors may, informally, be viewed both as reactive and deliberative, i.e. as an entity which acts in response to external stimuli but whose actions taken as a result are dependent on both the stimulus received and current *internal state* of the agent. Some of the potential offered by agent mechanisms can be seen in their use to control dialogue between human users, and apply heuristic methods to structure a report of that dialogue is described in Staniford *et al.* (1993). Staniford (1993) additionally gives a Prolog implementation of the required agents. Of particular interest, in the context the retrieval of legal information, is the development of agents that use deontic logic, described in von Wright (1951, 1964, 1965, 1971), in conjunction with resolution, described in Robinson (1965), as the reasoning mechanism for the deliberative functionality required within their architecture, described in Staniford (1993,1994). Such agents are highly suitable for applying heuristic methods for the extraction of sets of possible solutions from searches of structured document bases.

As an implemented example, described in detail in Staniford (1993), consider the *rapporteur agent*: whose responsibility it is to synthesise what may be a rambling dialectical discussion, between two participants, into a coherent document setting out the thrust of their debate. The general dialogue graph model that is realised by the rapporteur agent, during the course of the dialectical discussion, may be viewed as a specific instance of the sort of hypertext structure defined in Definition 1. A discussion report is a simple linear example of the more general document models described earlier: see Definition 2. We thus have two graphs, one – a directed cyclic graph – representing the realised dialogue space, and one – a directed acyclic graph – the model of a report of a dialectic discussion; both graphs containing single sources and sinks. The main task facing the *rapporteur agent* is to transform the former into the latter. Rapporteur has no notion of the semantics of a dialectical argument; it provides a way of enforcing a general syntactic structure to a dialogue represented by a graph. This structure is sufficiently flexible to allow the participants to conduct their dialogue using deduction, induction or indeed abduction as the mode of reasoning in their arguments. In its present form the rapporteur agent, while being modelled on Definition 7 (see below) has the requisite knowledge for linearisation encoded in an implicit form; in future work it is intended to develop a knowledge base (see Definition 8 below) of linearisation structures in an explicit way in order to enhance the overall flexibility and robustness of the system.

The practical application of logic based communicating agents to the legal document retrieval domain involves defining a multi-agent structure as a means of controlling search and traversal of document bases. The environment in which this structure will operate thus comprises three main elements:

- A document base;
- A knowledge base;
- A *search-and-traverse engine*.

The document base consists of legal texts which are represented as *specification constrained hypertexts*, i.e. in which the node and linkage structures are required to follow the conceptual organisation of the pertinent document classes: see Definition 1.

The knowledge base encodes specialist information and reasoning strategies, together with information as to the user and the task. It is designed as a logic based deliberative agent with certain constraints: see Definition 8 below. Note that the knowledge base is used to provide information for computational agents directly and its use is transparent to human users of the system.

The search-and-traverse engine is defined as a network of communicating agents: see Definition 7 below. This engine fulfills two roles: it provides the interface between the document base, knowledge base, and user for handling user generated queries; and it assists the user in traversing the document base by retrieving the relevant source hypertext and guiding the user through it in accordance with the appropriate linearisation schema.

Definition 7: A *deliberate agent* in a legal document retrieval system is a 12-tuple of the form

$$< D, H, T, A, R, Q, W, \mathit{see}, \mathit{do}, \mathit{hear}, \mathit{database}, \mathit{action} > \bullet$$

Where the set D is an arbitrary set of predicate calculus databases which represent the agent's internal states. H is a set of hypertexts (Definition 1.) which represent the external states of the environment in which the agent operates. T is a set of partitions of H in the form of a set of *document graphs* (Definition 2.). A is a set of actions, actions that the agent may decide to perform. R is a set of input words that represents the incoming communications from other agents. Q is a set of partitions of R, e.g. requests from a user would fall into a different partition than a set of strategies received from the knowledge base. W is a set of output words and we note that W is not necessarily equal to R.

The function *see*: $H \rightarrow T$ maps each state in H into the partition to which it belongs. The function *hear*: $R \rightarrow Q$ maps each word in R into the partition in which it belongs. These are sensory functions and are used to characterise the way in which an agent perceives stimuli external to itself. We use *see* to enable an agent to determine the local state of its environment and *hear* to enable the agent to receive communications from other agents.

The function *do*: $A \times H \rightarrow H \times W$ is an executory function which maps each action and state into the state that results from the execution of the given action in the given state together with the resulting output; providing the agent with the ability to change the local state of its environment and to communicate a message to another agent.

Unlike *see* and *hear* the *do* function encapsulates acting and communicating in one function because although there will be occasions when we wish to communicate without changing a state, the converse is not the case. We do not allow an act which changes the state of the environment to take place without there also being a corresponding act of communication. We wish to indicate that these two operations are closely bound together in order that we may simplify the coordination of the knowledge – between autonomous agents – that environmental state changes have taken place. The ability to communicate both with and without state changes occurring in the environment is crucial to the notion of networks of cooperating agents.

The function *database*: $D \times N \times T \times Q \rightarrow D$ maps the set of databases and both types of observation into the new set of internal databases; thus allowing the internal state of an agent to change. Finally, *action*: $D \times N \times T \times Q \rightarrow A$ is a function which maps each database, cycle number, external state partition and input partition into the action that the agent is to perform whenever it finds itself with a particular combination of internal databases, inputs and external states; note that, in practice, the cycle number is stored in an element of the set D and is updated by the *database* function but has been shown separately here in order to make its presence apparent.

The key idea in defining agents in this class is the use of an automated inference method like *resolution* combined with a logical method of reasoning such as *deontic logic* to implement the agent's action function. An agent of this sort is *deliberate* in that it deliberates on every cycle about which external action to perform.

In order to provide a useful knowledge base in the context of a multi-agent system we use the architectural specification given in Definition 7 as a general basis architecture and design a specialisation of that basis architecture for the knowledge base component of the system. This methodology provides a central resource that allows system users to operate in parallel from geographically disparate locations.

Definition 8: A *knowledge base* in a legal document retrieval system is a nine-tuple of the form

$$< D, A, R, Q, W, do, hear, database, action > \bullet$$

where the set D is set of predicate calculus databases that encapsulate the general notions presented in Definitions one through six with the addition of internal essentials such as the current cycle number. A is a set of actions which in the case of a knowledge base agent amounts to choosing a suitable output for communication to the interrogating agent, R, Q and W are as described above.

The function *hear*: $R \rightarrow Q$ is as defined above but the function *do* simplifies to *do*: $A \rightarrow W$ and maps each action to the resulting output communication. The function *database*: $D \times N \times Q \rightarrow D$ maps the set of databases and the received incoming communication onto the new set of internal databases. Finally, the function *action*: $D \times N \times Q \rightarrow A$ maps each database, cycle number, and input partition into the action that the agent is to perform whenever it finds itself with a particular combination of internal databases and inputs.

Key elements of cooperation in the multi-agent retrieval network are the identification, realisation, and evaluation of agent controlled search and traversal strategies. Such strategies will incorporate heuristic traversal methods involving, for example, the normative thesaurus approach of Bing (1987), domain analysis of the sort found in Dick (1991), and the argument heuristics described in Rissland et al. (1993).

In summary we see the agent formalism developed in Staniford et al. (1993) and refined in Staniford (1993, 1994) as providing the necessary unifying framework with which to control search and traversal in the proposed system. For the legal domain such an interface offers the possibility of greatly enhanced retrieval capabilities, with which the user will be supported in the construction of a document directed at the current task, with the traversal of the hypertext constrained by both form and content.

5. Conclusions

In this paper we have outlined how the current capabilities of document storage and retrieval systems may be enhanced for the classes of documents that possess a highly organised structure: typically a legal document base will be composed of such documents. It has been argued that in building computer-based systems for the purpose of storing and retrieving texts of this kind, it is essential that the underlying structural and conceptual organisation of the document form be taken into account. The paradigms provided by viewing document forms as labelled directed graphs from a particular class allow such structure to be specified and maintained so that the creation, perusal, and retrieval of texts can be accomplished in a manner that is more in line with the needs of the user.

Finally we have argued that the control of the activity of search-and-retrieval for specific information can be enhanced by enlisting the aid of a network of communicating agents to provide a protocol for interactions between the document base and a knowledge base to direct the document traversal. A key advantage of the agent architecture is that it permits the deployment of a plurality of traversal strategies that are available for the provision of «intelligent» assistance to users across a distributed network.

ACKNOWLEDGEMENT

This is a revised and extended version of a paper presented at the conference «Towards a Global Expert System in Law», Florence, December 1993.

REFERENCES

Ashley, K., *Modelling Legal Argument*, Bradford Books, MIT Press, 1990.

Barlow, J., Beer, M.D., Bench-Capon, T.J.M., Diaper, D., Dunne, P.E.S., Rada, R., *Expertext: Hypertext-Expert System Theory, Synergy and Potential Applications*, In: Research and Development in Expert Systems VI (Ed.: N. Shadbolt), Proc. of Expert Systems '89, Cambridge Univ. Press, 1989, 116-127.

Barlow, J., Dunne, P.E.S., Rada, R., *EXPERTEXT: from Semantic Nets to Logic PetriNets*, Expert Systems with Applications, 1, 1990, 51-62.

Belew, R.K., *A Connectionist Approach to Conceptual Information Retrieval*, Proc. of 1st International Conference on AI and Law, Boston, 1987: ACM Press, 116-126.

Bench-Capon, T.J.M., Dunne, P.E.S., *Some Computational Properties of a Model for Electronic Documents*, Electronic Publishing - Origination, Dissemination and Design, 2, 1989, 231-256.

Bench-Capon, T.J.M., Dunne, P.E.S., *An Approach to the Integration of Legal Support Systems*, Proc. Dexa90, Vienna, Springer-Verlag, 1990, 105-111.

Bench-Capon, T.J.M., Dunne, P.E.S., Staniford, G., *Linearising Hypertext through Target Graph Specifications*, Proc. Dexa92, Valencia, Springer-Verlag, 1992, 173-178.

Bench-Capon, T.J.M., Dunne, P.E.S., Staniford, G., *Linearisation Schemata for Hypertext*, Proc. of DEXA 93, Springer Lecture Notes in Computer Science, 720, Heidlelberg, 1993, pp. 697-708.

Bench-Capon, T.J.M., Dunne, P.E.S., Staniford, G., *A Formal Language Basis for Studying Properties of Computer Representations of Documents*, (in preparation), 1993(b).

Bing, J., *Designing Text Retrieval Systems for Conceptual Searching*, Proc. of 1st International Conference on AI and Law, Boston, 1987, ACM Press, 43-51.

Bond, A.H., Gasser, L. (Eds.), *Readings in Distributed Artificial Intelligence*, Morgan and Kaufman Inc., 1988.

Boughanem, M., Soulé-Dupuy, C., *A Connectionist Model for Information Retrieval*, Proc. Dexa92, Valencia, Springer-Verlag, 1992, 260-265.

Brachmann, R.J., *On the Epistemological Status of Semantic Networks*, In: Associative Networks: Representation and use of knowledge by computers (Ed.: N.V. Findler); Academic Press, 1975, 3-50.

Branting, L.K., *An Issue-oriented Approach to Judicial Document Assembly*, Proc.

of 4st International Conference on AI and Law, Amsterdam, 1993: ACM Press, 228-235.

Dick, J., *Representation of Legal Text for Conceptual Retrieval*, Proc. of 3rd International Conference on AI and Law, Oxford, 1991, ACM Press, 244-253.

Genesereth, M., Nilson, N.J., *Logical Foundations of Artificial Intelligence*, Morgan Kaufman Inc., 1987, Ch. 13, 307-327.

Hafner, C.D., *Conceptual Organisation of Case Law Knowledge Bases*, Proc. of 1st International Conference on AI and Law, Boston, 1987, ACM Press, 35-41.

Huhns, M., Papazogolou, M. P., Schlageter, G., (Eds.), *Proc. of ICICIS'93*, Rotterdam, IEEE Computer Society Press, Los Alamitos, California, 1993.

Koo, R., *A Model for Electronic Documents*, ACM SIGOIS Bulletin, 10 (1), 1989, 23-33.

Morris, G., Taylor, K., and Harwood, M., *Handling Significant Deviations from Boilerplate Text*, Proc. of 1st International Conference on AI and Law, Boston, 1987, ACM Press, 145-154.

Papazogolou, M. P., Zeleznikow, J., *The Next Generation of Information Systems: From Data to Knowledge*, Springer Verlag, 1992.

Rissland, E.L., Skalak, D.B., and Friedman, M.T., *BankXX, A Program to Generate Argument Through Case Based Search*, Proc. of 4st International Conference on AI and Law, Amsterdam, 1993: ACM Press, 117-124.

Robinson, J.A., A Machine-Oriented Logic based on the Resolution Principle, Journal of the ACM, 1965, 12, 23-41.

Routen, T., Bench-Capon, T.J.M., *Hierarchical Formalisations*, Intnl. Jnl. of Man-Machine Studies, 35, 1991, 69-93.

Simpson, A. and McKnight, C., *Navigation in Hypertext: Structural Cues and Mental Maps*, in R. McAleese & C. Green (eds), Hypertext: State of the Art, Oxford, Intellect, 1990.

Sprowl, J.A., *Automating the Legal Reasoning Process: A Computer that Uses Regulation and Statutes to Draft Legal Documents*, American Bar Assoc. Journal, 1979, 1-81.

Standard ECMA-101, *Office Document Architecture*, Euro. Compu. Manu. Assoc., 1985.

Staniford G., *Multi-Agent Systems in Support of the Cooperative Authorship of Structured Documents*, PhD. Dissertation, University of Liverpool, 1993.

Staniford G., *Multi-agent System Design: Using Human Societal Metaphors and Normative Logic*, (to be presented at ECAI'94, Amsterdam, August 1994).

Staniford, G., Bench-Capon, T.J.M., and Dunne, P.E.S., *Cooperative Dialogues with the Support of Autonomous Agents*, 1993, Proc. ICICIS 93, Rotterdam, May 1993, pp. 144-151.

Staniford, G., and Dunne, P.E.S., *Autonomous Agents in the Support of Cooperative Authorship*, CSCW and Artificial Intelligence (Ed.: E. Edmonds, J.H. Connolly), Springer-Verlag, London, (in press, 1994).

Stotts, P.D., Furuta, R., *Adding Browsing Semantics to the Hypertext Model*, Proc. ACM Conf. on Document Processing Systems, 1988, 43-50.

Werner, E., *The Design of Multi-Agent Systems*, Proceedings of 3rd European Workshop on Modelling Autonomous Agents in a Multi-Agent World., (Ed.) E. Werner, & Y. Demazeau, North Holland, Amsterdam, 1992, 3-28.

Wilson, E., *Converting Legal Texts into Expert Systems*, Proc. National Conf. on Law, Computers and Artificial Intelligence, Univ. of Exeter, 1988.

Wright, G.H. von, *Deontic Logic*, Mind, 1951, 60,1-15.

Wright, G.H. von, *A New System of Deontic Logic*, Danish Yearbook of Philosophy, 1964, 1, 173-182.

Wright, G.H. von, *A Correction to a New System of Deontic Logic*, Danish Yearbook of Philosophy, 1965, 2, 103-107.

Wright, G.H. von, *On the Logic of Norms and Actions*, *New Studies in Deontic Logic*, (Ed.) R. Hilpenen, D. Reidel Publishing Company, Dordrecht, Holland., 1971, 3-35.