Expert Systems With Applications, Vol. 4, pp. 355–362, 1992
Printed in the USA.

0957-4174/92 $5.00 + .00


# Integrating Legal Support Systems Through Document Models

TREVOR J. M. BENCH-CAPON, FRANS COENEN, AND PAUL E. S. DUNNE

Department of Computer Science, University of Liverpool, Liverpool, Great Britain

**Abstract**—*The need to integrate legal support systems, especially expert systems and retrieval systems, is described. Correspondences between the components of the various classes of system are noted, and some current difficulties with integrating the systems are described. Both the potential for and the benefits accruing from the use of a uniform graph-theoretic model as a basis for the systems are outlined. A detailed description of the use of this approach in the construction of an application in the legal domain is given.*

## 1. INTRODUCTION

THE LAW has proved a popular domain for the development of computerised support systems. Computer systems have been developed to assist legal work in the retrieval of information, ranging from the keyword-in-context searching of LEXIS and WESTLAW, to more advanced methods of conceptual retrieval as in Hafner (1987) and Bing (1987), and, more recently still, hypertext-based systems such as JUSTUS (Wilson, 1988). Support can be provided for document preparation ranging from basic use of word processors, to semiautomated methods, such as those of Sprowl (1979) for will preparation and Morris et al. (1987) for the preparation of Internal Revenue Service documents. Assistance in the application of the law is provided by expert systems such as the Retirement Pensions Forecast and Advice System of the UK Department of Social Security (Springel-Sinclair, 1988) and the Latent Damage Advisor of Capper and Susskind (1988). Additionally, of course, any organisation concerned in legal work will have the usual office systems and database requirements. Work on such systems has tended to progress independently, but these systems should not be conceived of as separate technologies, as there are sufficient grounds of commonality between them to give great potential synergy through integration. We will focus on the need to integrate expert systems with retrieval systems, illustrating our points by reference to a system we are currently developing to assist in the processing of claims for compensation made to British Coal, an application which we see as being fairly typical of those which can currently be addressed.

Requests for reprints should be sent to Trevor J. M. Bench-Capon, Department of Computer Science, University of Liverpool, Liverpool L69 3BX, Great Britain.

With regard to the application of AI techniques to supporting legal work, it has been traditional to follow (McCarty, 1984) and to divide systems into two categories: legal analysis and planning systems, which are the traditional legal expert systems, wherein the user is asked to state the facts of a particular case and receives a decision or advice from the system; and conceptual retrieval systems, in which legislation and previously decided cases are retrieved in a way intended to be somewhat more intelligent than the normal keyword in context Boolean search systems. In a subsequent paper (McCarty, 1988) McCarty stressed that whilst the distinction is useful, and reflects a genuine distinction among research prototypes, there is no need to keep the systems separate. Indeed he says "a hybrid system should be our ultimate goal." The reasons for this are:

> The analysis and planning system would be more useful if it could provide direct access to the case materials which justified its conclusions, and this would be possible if the system were linked to a conceptual retrieval system. The retrieval system would be more powerful if it could follow the patterns of inference suggested in the cases, and this would be possible if the system had access to the rules of the legal analysis system. (p. 20)

Such an integration requires, however, that the various technologies be put on some uniform foundation, so that coherence and harmonisation can be achieved. McCarty's answer is that both systems should therefore be written in the same representation language: his "Language for Legal Discourse" (McCarty, 1989). This work is interesting, but in this paper we will propose a different approach towards integrating these systems—moreover, an approach which we believe can be extended to facilitate the integration of other conventional systems as well. The main thrust of this paper is therefore to outline how a single coherent document model together with appropriate formal transformation rules could be used to provide a foundation for the integration of these different systems.

## 2. THE NEED TO INTEGRATE EXPERT AND RETRIEVAL SYSTEMS

The need to integrate expert systems with retrieval systems can be shown by considering a very small and simplified example. The example is taken from the field of Social Security, in which a number of promising prototypes have been developed. In this field and in others, such as Housing Law administration and the British Coal application described later, decisions are made not by lawyers but by lay employees working in accordance with detailed guidance on how they are to apply the law. As the example shows, this detailed guidance is of great help in producing a useful system.

Suppose that a certain benefit paid an additional amount for heating, with the intention that people should not be made to suffer simply because they had the bad luck to live in a house that was hard to heat. The primary legislation probably would simply introduce the existence of such an additional amount for heating, and contain an enabling provision allowing regulations to prescribe the circumstances in which such an amount was payable. This would be to give a degree of flexibility. A regulation might now be made along the following lines:

A person shall be entitled to an additional amount for heating if his dwelling is hard to heat.

If we formalise this legislation we get something like:

R1 X is entitledToHeatingAddition if Y dwelling-of X, and
Y is hardToHeat.

Executing this as an expert system might give rise to the following dialogue:

User: Is Tony entitledToHeatingAddition?
System: Which Y is dwelling-of Tony?
User: 23 Railway Cuttings
System: Yes, if 23 Railway Cuttings hardToHeat.

Whilst the temptation is to say that such a system brings us very little further forward, such a reaction is misplaced. The role of such a system is to direct us to the question we must answer, and away from irrelevant matters, such as the age and state of health of the claimant, and to show us the consequence of the answer we give to the important question. What such a system does not do, and what it cannot be expected to do since the legislation gives no guidance, is to provide help in answering the key question. In such a system, making no use of information outside of what can be found in legislation, the answer must be left to the unaided judgment of the user. Of course the user cannot just answer the question any way he wants. He is charged with making an actual legal decision, one that is hoped will conform with other decisions made on this matter in the past. To this end, the user of such a system typically will have been provided with a manual of guidance that operationalises the decisions that must be made in terms of questions for which answers can be expected. These operational definitions may be based on past cases, the policy of the organisation, or on the collected experience of adjudicators. Such operationalisations enable us to come up with rules such as:

R2 X is hardToHeat if X built-before 1927 and
X is detached-House.
R3 X is Not hardToHeat if X is energyEfficient
R4 X is energyEfficient if X built-by Basset.

Such rules provide, however, only sufficient conditions and are applicable only in certain, tightly defined circumstances. It is quite inconceivable that every possible circumstance that may arise can be covered by a set of operational definitions, although, of course, a decision in a new circumstance may well give rise to a new operationalisation. If there is no guidance, or when the specific guidance runs out, the user has no alternative but to consider the original sources, the legislation, the guidance, and the case reports, and to use personal judgment to interpret the facts of the case on hand in the light of the documents. Here is where the sort of support that can only be provided by an effective e covered by a set of operational definitions, although, of course, a decision in a new circumstance may well give rise to a new operationalisation. If there is no guidance, or when the specific guidance runs out, the user has no alternative but to consider the original sources, the legislation, the guidance, and the case reports, and to use personal judgment to interpret the facts of the case on hand in the light of the documents. Here is where the sort of support that can only be provided by an effective retrieval system is required. Ultimately, the effectiveness of the system relies on the user applying the leaf predicates correctly, and this will require consultation of the original sources in unusual cases—the cases in which the user is in most need of help.

Of course it would be possible to provide access to these sources through separate systems; perhaps a keyword in context system for the legislation, a hypertext system for the operational guidance, and a heavily indexed set of cases for previous decisions. But this would be to ignore two important points. First, the user has no wish to browse through the source material: The user is accessing it to answer a particular question in light of the facts of a particular case, and so the sections of the material that are relevant are determined by these factors. Moreover, these factors are encapsulated in the point of the computation reached by the expert system and the facts the user has already supplied, and thus an integration that would permit this information to be made available to the retrieval system is highly desirable. Second, there are clear relations between the various source documents: The guidance guides the application of a particular piece of legislation in light of certain past cases. These relations need to be made available to the user also.

Thus, only an expert system integrated with a means of retrieving the original source documents can provide optimal support. In the next section we shall look at some of the commonalities between the various systems, which make integration possible, and some of the problems of harmonisation, which make the need for a formal foundation for this integration essential.

## 3. COMMONALITIES AND PROBLEMS

The relationships between hypertext- and knowledge-based systems have been discussed elsewhere (Barlow et al., 1989a, 1990). Broadly, these turn on the fact that the links in the hypertext system are typically construed as arcs of a semantic net; these arcs correspond to the predicates that are found in an expert system. Whilst some of the relations will be found in only one of the systems, others are common. However, the uses made of them within the two classes of system are different; they are employed to facilitate document traversal in the hypertext system, whereas in the expert system they will provide a vocabulary for the construction of rules to license inference. There is, however, considerable scope for the information latent within each of the systems to inform the other system. Turning to conceptual retrieval systems, we see that the proposals of Hafner (1987) combine the use both of semantic nets, which relate to hypertext-like systems, and issue discrimination trees, which have much in common with the rules of an expert system. Similarly, alternative approaches are founded either ultimately on a semantic net or, as in the case of Bing's normative thesaurus (1987), on expert system-like rules. Conventional systems likewise will tend to use keywords that correspond to the predicates found in the expert system.

Given the similarities noted above, it might be thought that harmonisation through common predicates would be a straightforward task. But this is not so, for the different uses to which the relations are put give rise to different attitudes towards their definition in the different systems. Suppose, for example, we wished to integrate an expert system with a hyper document relating to its domain. Whilst it is true that hypertext systems are based on some form of semantic net, very often the relations represented by the arcs in this net are not defined with any great precision, rather relying on the connotations of the labels of the arcs to convey the meaning. Thus, for example, a typical net may use a relation "is-a" to link nodes without making it clear whether this class–superclass relation is intended to be exhaustive, so that any instance of the superclass must be an instance of at least one of the subclasses, or whether it is intended to be exclusive, so that any instance of a superclass must be an instance of, at most, one of the subclasses. Other decisions about the relation need to be made, and these are not necessarily taken consistently across the whole net. In a hypertext system, and probably a conceptual retrieval system also, this may not matter too much, since the user of the system can supply the required intelligence and, so, recognise and reconcile any inconsistencies that may occur. In contrast, the relations of an expert system are precisely defined, because their extension can be computed from the system. Often, too little attention is paid to thinking precisely about the relations, in which case, the computation may conflict with the expectations of the system's builder, but nonetheless the executability of the expert system does mean that the relation has an objective definition via the computed extension. In an expert system the knowledge base that will be debugged until the actual, computed interpretation is not significantly different from that of the system's builder.

If the predicates of the various systems are to be merged into a single system, however, these differing degrees of precision in relations become unacceptable. The expert system will apply its computation-based understanding of relations to the more loosely used relations of the semantic net within the hypertext system, which may or may not coincide with the similar relations of the conceptual retrieval system, with the inevitable result that some uses of these relations will be inappropriate. Therefore, it is essential that at least some of the relations employed in the different systems, namely those on which the systems will interact, should be harmonised through some common model. This means that the use of the relations in the hypertext and conceptual retrieval systems must be disciplined and based on some precise definition of those relations, and this definition must correspond to the computational behaviour of the expert system. The question thus becomes one of how we are to express the precise definitions of these relations and how such expressions are to be made operational when constructing the systems. It is clear that we must make the major change in our approach to the retrieval systems rather than expert systems, because an ambiguous relation cannot be used as if it were unambiguous without unwanted conclusions being drawn (or desired conclusions not being drawn), whereas an unambiguous relation can be catered for in a system that uses that relation in other senses also. The sentence "every student uses a computer" is ambiguous in that every student may use a different computer if they all have micros, or they may all use the same computer where they use a mainframe. This distinction must be made if we formalise the sentence into predicate logic, and different conclusions will follow, according to the choice made. In natural language, however, the ambiguity can be allowed to pass, relying on the reader to understand the intended meaning according to the context and judgment. The first concern, therefore, must be to provide a means for making the relationships used within the hypertext system precise, and in such a way that this conformity with the chosen interpretation can be imposed on the system builder. Our belief is that the requisite precision can be imposed through the use of document models. The next section therefore introduces the ideas underlying such models.

## 4. DOCUMENT MODELS AS A FOUNDATION FOR HARMONISATION

Any meaningful document has a structure, an understanding of which is essential if a reader is to interpret the text correctly. Over the years, conventions for different classes of documents have developed; thus a textbook will, at the highest level, typically comprise a preface, followed by a table of contents, followed by a series of chapters, followed by an index. Chapters themselves have structure, being broken down into sections, paragraphs and sentences. Legal documents in particular exhibit a high degree of conventional structure. A law report will comprise a header note summarising the point of law, a statement of the facts of the case, and the decision. Statutes also follow strict structural conventions. Normally the reader will be expected to read these documents in a particular order; thus, later chapters of a book may presuppose material in earlier chapters. Understanding of any document is enhanced if the reader is aware of the structural conventions, but this means in turn that the author must be aware of, and observe, these conventions also.

Today many documents are prepared on computer systems, although intended for publication in hard copy. The electronic document on which the author is working thus may have a different set of structural conventions from the intended finished product. It is important therefore that there be correspondence between the two; which requires that the electronic document have a formal structure which can be used as a guide for the output paper document. Only if these correspondences exist can the author ensure that he or she is observing the required conventions in a natural and effortless manner.

One structure that has attracted much attention as an underlying document model is directed acyclic graphs. Examples of such models may be found in Delisle, Schwartz, & Neptune (1986), Kimura & Shaw (1984), Koo (1989), and Meyerowitz (1986), all of which employ arbitrarily directed acyclic graph structures; and in Bertino, Rabitti, & Gibbs (1988), Christodoulakis, Theodoridou, Ho, Papa, & Pathria (1986), Standard ECMA-101 (1985), and Thomas et al. (1985), which are restricted to tree structures. This model is appealing since it permits logical connections between sections to be represented simply and directly. Koo (1989) extends the earlier models of Delisle et al. (1986), Kimura & Shaw (1984); and Meyerowitz (1986), by introducing (amongst other ideas) the concept of graph modification rules. These are employed to control modifications to a graph in order that it should reflect changes (either in structure or interpretation) made to the underlying document: Thus rules may encapsulate how to modify the graph in the event of sections being added or deleted, and rules may indicate how new logical links in the document structure are to be reflected in the graph form. More formally, a graph modification rule is a production rule in a graph grammar. A rule consists of three parts and acts on a given graph as follows: The rule consists of a predicate, $P$; a left-hand graph $G_l$; and a right-hand graph $G_r$. For a given input graph $G$ if the predicate $P$ holds for $G$ and $G$ contains $G_l$ as a sub-graph then $G_l$ may be replaced by the graph $G_r$ in $G$. Koo (1989) illustrates how a simple set of rules can be applied to create and modify tables of information, and it is observed that the correctness of the given rules may be verified formally by a simple inductive proof.

The properties of directed acyclic graphs offer several advantages over more general representation methods, particularly when their semantic capabilities are enhanced by the provision of a formal modification regime. These are discussed in Bench-Capon & Forder (1989).

The above formal model for electronic documents can, we contend, be applied to conventional documents, hyperdocuments, the semantics nets underpinning hyperdocuments and conceptual retrieval systems, and the rule bases of expert and other automated document creation systems. Essentially, the formal model consists of a set of constraints on the allowed form of graphs, together with a set of traversal and modification rules. The formal representation of documents in this way has a number of advantages independent of integration: For example, legal texts provide one example of documents which are sometimes the work of several individuals and which possess a highly structured form. A formal model would allow exploration of issues such as the automated production of a hyperdocument from legal documents, formal specification of classes of legal document (e.g., contracts, legislation, wills, etc.), and the automated translation of such specifications into rewriting and production systems, managerial strategies for cooperative production of legal hyperdocuments, and the serialisation of such hyperdocuments, which will subsume current navigation problems. In addition, as has been argued in Barlow & Dunne (1989), a graph-theoretic approach allows the importation of formal models of concurrent behaviour as a vehicle with which to describe cooperative authorship activities.

When we move to integrating the various systems, the need for a formal underpinning model ceases to be merely desirable and becomes essential. Although the traversal and modification rules will differ for different classes of document, they will ensure that the relations represented by the arcs in all the various documents will be precisely defined. This being so, where a relation is represented in different documents, identification can be made with confidence, and only the correct consequences will be drawn from this identification. The advantages of this are clear: a single representation may be used as the core of the various systems and communication between them in operation will be facilitated. This would enable, for example: (a) the point of the computation reached by an expert system to determine the entry point to a retrieval system; (b) the output from a conceptual retrieval system to be used in the explanation of the output from an expert system; or (c) the product of document creation system to be offered to an expert system for validation.

## 5. KNOWLEDGE BASES AS DOCUMENTS

The above is critically dependent on our ability to represent all the various components of the system to be integrated within the single formalism of our document model. This is least obviously the case when we consider the rule-based components. In this section, therefore, we will show the sense in which rule bases can, and arguably should, be seen in terms of the document model.

Knowledge bases, like documents, have structure: Depending on the domain of application, the knowledge represented may be classified into a number of categories and be of several different types, for example, conceptual, qualitative, and so on. In addition to the actual facts stored in the knowledge base, various rules must be encoded that will attempt to mimic the reasoning processes employed by an expert practitioner in the relevant domain. Again, such rules may be divided into several different classes depending on the style of reasoning they embody and by the degree of confidence that can be placed on the veracity of those conclusions which are inferred by their application to particular sets of data. This underlying core of factual knowledge and this battery of inferential techniques are obviously related: Tight logical links connect facts in the former with rules in the latter. However, there will, in general, also exist logical connections among diverse collections of information in the factual database and similar links among subsets of the various reasoning techniques. An awareness of all such logical interdependencies is crucial at each stage from knowledge elicitation to verification of the final system if the knowledge engineer is to succeed in the aim of constructing a usable (let alone useful) knowledge-based system for the specific target application.

Just as the writing of a document involves the organisation of ideas by an author into a form that can be communicated to readers, this being often accomplished using some computer representation as an intermediary, so too the design and implementation of a knowledge base involves the systematic distillation of a core of expert knowledge into a form an adept but nonspecialist user can apply; and here again a computer system is used as a vehicle for building and storing the elicited knowledge. Thus, given these parallels between the preparation of complex documents and the codification of specialist knowledge, one can contend that those management approaches that endeavour to aid authors in communicating ideas by providing a computer representation supporting the underlying form of a particular class of document will also provide a valuable foundation for the synthesis of knowledge bases.

The analogy between document- and knowledge-base management extends beyond the creation and design phase. An electronic document is ultimately produced in a serial form corresponding to a sensible order in which some reader may traverse it. The process of reading and understanding a document is an interpretive one. Similarly, in application, a knowledge base is traversed by the inference engine applied to it during specific execution instantiations. The traversal is controlled by some execution strategy, and just as different readership imposed orderings of a document may result in different interpretations of the text, so different traversals of the knowledge base during execution will yield different results on each occasion, for example, if varying conflict resolution strategies are chosen. Underpinning the interpretation of a document, there are implicit constraints on how the text should be read so that a sensible understanding of its content can be garnered; similar constraints guide the reasoning processes carried out on a knowledge base to ensure both termination and the validity of conclusions arrived at.

The purpose of expressing a knowledge base in terms of the formal document model (as with any document) is to be aware of these differing execution strategies (readings of a document) in order to impose appropriate constraints on which strategies are permissible and which are not. That a formal management system is necessary arises from the fact that even a single designer will find it difficult to organise a complex database; this holds true even more for teams of designers who need to conform to a single homogeneous view and thereby require organisational support tools in order to ensure systemic correctness.

## 6. EXPLOITING THE STRUCTURE OF LEGISLATION: THE MAKE PROJECT

We will now illustrate some of the above points by reference to a fairly typical legal application, designed to support the processing of claims for compensation for work-related injuries made by employees of British Coal (BC). This application is being developed as part of the Maintenance Assistance for Knowledge Engineers (MAKE) project, a collaborative project between the University of Liverpool, ICL, and British Coal, investigating the issues connected with maintenance of regulation-based KBS. An essential part of the project was the development of a genuine KBS, addressing a realistically sized application, which could be used as a test-bed for maintenance ideas.

The application itself closely conforms to the kind of system discussed in section 2 of this paper. The processing is carried out by lay employees rather than lawyers. Currently, these people need to consult a variety of documents: the legislation that governs the claims, some fairly detailed guidance that partially operationalises the law and policy of their employer in terms of questions they can be expected to answer, and reports of important decisions.

The system was developed using KANT (Knowledge Analysis Tool) and KBB (Knowledge Base Builder). When merged together these form a single environment for analysing and building KBS, a feature of which is the use of the structure of the source documents to structure the developing knowledge base. This is of especial help in the legal domain where the knowledge is derived from source documents—documents that, as has been noted above, have a high degree of structure that can be used to provide the framework within which to organise the knowledge base.

Essentially, KANT is a hypertext-like knowledge analysis tool originally built to assist in the development of a KBS to provide decision support for Department of Social Security (DSS) Adjudication Officers in the assessment of claims for benefits in local DSS offices (Storrs & Burton, 1989). This system was called the Local Office Demonstrator (LOD) System and was one of three applications built as part of the Alvey DHSS Large Demonstrator Project aimed at demonstrating the viability of KBS decision support in large, legislation based organisations. Whilst KANT does not at present enforce a formal document model to regulate user behaviour, such a model is implicit in the way the system is used. An extension to KANT would provide facilities to enforce the model.

KANT is designed to support the construction of KBS in domains, such as legal domains, in which the source knowledge comprises a significant amount of textual material by assisting the knowledge engineer in the analysis of these source documents. Typically, knowledge analysis using KANT consists of the following stages:

1. The source documents are translated into a form suitable for use in KANT, a process known as Kantification. The source documents available for construction of the MAKE Project system were: (a) The Mines & Quarries Act (1954); (b) the H & SE Mines (safety of Exit) Regulations (1988); (c) the Claims inspectors Manual (1990); and (d) a number of significant judgments.
2. One or more Kantified source documents can be viewed through individual windows and relevant Sections identified. These are then copied into a single Source "Structure" on which further analysis can be implemented. This structure can be thought of as a precis of the source material, which maintains the original document structure.
3. The structure created in (2) can then be analysed. A typical approach is to identify Entity, Attribute, Value triples (EAVs) and store them in another storage structure, ensuring at all times that the structure of the source documents is respected.
4. The EAV structure then provides a sound basis from which to build suitable Class Hierarchies and Rule Bases in the KANT intermediate representation, which in turn will reflect the source material.

Thus, as a result of knowledge analysis using KANT, a class hierarchy and Rule Base are produced in an intermediate representation. The class hierarchy in this representation consists of a top level class with subclasses describing different object types. Each class has a unique class name and a number of slots describing attributes of that type of object, and the possible values that this sort of object may have for that attribute. A feature of the class hierarchy is that subclasses inherit attributes of superclasses so as to ensure that a subclass is a strict specialisation of its superclass. In KANT inherited attributes cannot be cancelled, as is the case with some other development environments, for example KEE.

The rule base in the KANT intermediate representation then consists of a set of rules, each of which corresponds to an identifiable section of the source text, and each of which reflects the logical structure of that text. More detail of the intermediate representation can be found in Bench-Capon & Forder (1991).

The KBB, a tool that enables the intermediate representation to be transformed into a more efficiently computable form, then takes the Class Hierarchy and rule based in the intermediate representation resulting from the use of KANT and compiles them into rules and objects in the target representation language. The result is a knowledge base represented in Conjunctive Normal Form, the clauses of which remain tied to rules of the intermediate representation, and through them to the source texts, so that the structure of the original documents is carried right through to the executable form of the knowledge base.

### 6.1. Implications for Integration

At this point we have the source texts in a machine-readable form and the structural links developed in these texts through the analysis process. We also have a rule base that exhibits a structure corresponding to these texts and is composed of rules and predicates that are derived from, match, and are linked to these structured documents. Now when the user is forced to exercise his or her own judgment, the point of the execution will determine which predicate and which rule is under consideration, and through these links place the user in the appropriate place in the appropriate source document. Moreover, by following the links between the source documents, the user can retrieve any other related relevant information and then return effortlessly to the expert system when the decision has been made. Because the source texts and knowledge base share the same structure, therefore, integration of the two is seamless, and readily understood by the user.

A further advantage of the use of a shared structure concerns the transparency of the reasoning employed by the system. Early expert systems were often criticised for a lack of topic coherence. Because the knowledge base was structured in accordance with considerations pertaining to the knowledge base rather than to the task or to the user, questions often would be asked of the user in an order the user found difficult to relate to or explain. The MAKE knowledge base, in contrast, has a structure corresponding to the familiar and accessible documents, and, therefore, the motives for asking the questions can be explained by reference to this relatively well understood model, without need to consider an independent knowledge base structure. This is further exploited in the MAKE system by the use of structured dialogues as previously used in the LOD system.

### 6.2. Relations to Isomorphism

It has been cogently argued in Bench-Capon (1989) and Routen (1989) that certain software engineering considerations require that a knowledge base grounded in a written legal source must be in some sense isomorphic with these sources. Whilst there is an intuitive understanding of what such isomorphism entails, precise statement is often elusive. If, however, we have both the source text and the knowledge base modelled in the formal terms sketched above, such isomorphism can be stated with precision. The method of using the structure of the source documents to determine the structure of the knowledge base can therefore be seen as a way of achieving this isomorphism.

## 7. CONCLUSIONS

We have noted the existence of a range of currently separate systems for the support of legal work, and noted that many of them are grounded in common ideas, most notably, the existence of structured texts, sets of predicates describing the relationships between entities in the domain, and rules composed using these predicates that describe their logical relations, and manipulation in the performance of tasks. We have argued that these commonalities could form the basis for the integration of such systems, so that their development can be harmonised, components can be shared across systems, and benefits can be derived from communication between the systems in operation. Such integration is possible only given a suitable formal basis that can underpin all such systems and serve as the common foundation for the integrated system. We have argued that modelling such systems in terms of electronic documents, described as directed acyclic graphs with associated traversal and transformation rules, can provide the required common formal basis. Finally, we have illustrated the benefits of such an approach through reference to a system we are currently producing: the harmonisation achievable in the design of information retrieval and expert systems through the common representation of a single coherent model; the support that such a model provides for regulating the activity of several authors engaged in cooperatively producing texts; and, as has been discussed in detail, how such an approach would facilitate the construction and maintenance of expert system knowledge bases.

This paper, of course, does not go into the specific aspects of models for particular classes of document; we believe, however, that it makes a strong case for their development, and the construction of such models is the subject of ongoing work.

## REFERENCES

Barlow, J., Beer, M.D., Bench-Capon, T.J.M., Diaper, D., Dunne, P.E.S., & Rada, R. (1989). Expertext: Hypertext-Expert System theory, synergy and potential applications. In N. Shadbolt (Ed.), *Research and development in expert systems VI*, pp. 116–127, Proc. of Expert Systems 89, Cambridge Univ. Press.

Barlow, J., & Dunne, P.E.S. (1989). *Cooperative writing schemes based on formal models of concurrency* (Report CS/CSCW/7/1989). Dept. of Computer Science, Univ. of Liverpool.

Bench-Capon, T.J.M. (1989). Deep models, normative reasoning and legal Expert Systems. *Proceedings of the 2nd International Conference on AI and Law* (pp. 37–45). Vancouver: ACM Press.

Bench-Capon, T.J.M., & Dunne, P.E.S. (1989). Some computational properties of a model for electronic documents, Electronic Publishing—Origination, Dissemination and Design (pp. 231–256). New York: Wiley.

Bench-Capon, T.J.M., & Forder, J.M. (in press). Knowledge representation for legal applications. In T.J.M. Bench-Capon (Ed), *Knowledge-based systems for legal applications* (pp. 245–264). New York: Academic Press.

Bertino, E., Rabitti, F., & Gibbs, S. (1988). Query processing in a multimedia document system. *ACM Trans. on Office Information Systems*, **6**(1), 1–41.

Bing, J. (1987). Designing text retrieval systems for conceptual searching. *Proceedings of 1st International Conference on AI and Law* (pp. 43–51). Boston: ACM Press.

Capper, P., & Susskind, R. (1988). *Latent damage law—The Expert System*. London: Butterworths.

Christodoulakis, S., Theodoridou, M., Ho, F., Papa, M., & Pathria, A. (1986). Multimedia document presentation, information extraction, and document formation in minos: A model and a system. *ACM Trans. on Office Information Systems,* **4**(4), 345–383.

Conklin, J. (1987). Hypertext: An introduction and survey. *Computer,* **20**(9), 17–41.

Delisle, N., & Schwartz, M. (1986). Neptune: A hypertext system for CAD applications. *Proc. ACM Intnl. Conf. on Management of Data* (pp. 132–143).

Forder, J., & Taylor, A. (1991). *The local office system KBS for legal applications,* T.J.M. Bench-Capon, Ed., (pp. 139–164). New York: Academic Press.

Hafner, C.D. (1987). Conceptual organisation of case law knowledge bases. *Proceedings of 1st International Conference on AI and Law* (pp. 35–41). Boston: ACM Press.

Kimura, G.D., & Shaw, A.C. (1984). The structure of abstract document objects. *Proc. ACM Conf. on Office Information Systems* (pp. 161–169).

Koo, R. (1989). A model for electronic documents. *ACM SIGOIS Bulletin,* **10**(1), 23–33.

McCarty, L.T. (1983). Intelligent legal information systems: Problems and prospects. *Rutgers Computers and Technology Law Journal,* **9**(2), 265–294. (Also in Data Processing and the Law (C. Campbell, Ed). London: Sweet and Maxwell, 1984, pp. 125–151.)

McCarty, L.T. (1988). Intelligent legal systems: An update. In H. Fiedler, F. Haft, & R. Traunmuller (Eds.), Expert Systems in law (pp. 15–26). Tuebingen: Attempto Verlag.

McCarty, L.T. (1989). A language for legal discourse I. Basic features. *Proc. Second International Conference on Artificial Intelligence and Law* (pp. 180–189). Vancouver: ACM Press.

Meyrowitz, N. (1986). Intermedia: The architecture and construction of an object-oriented hypermedia system and applications framework. *Proc. Object-Oriented Prog. Syst., Languages and Applications,* 186–201.

Morris, G., Taylor, K., & Harwood, M. (1987). Handling significant deviations from boilerplate text. *Proceedings of 1st International Conference on AI and Law* (pp. 145–154). Boston: ACM Press.

Rada, R., Dunne, P.E.S., & Barlow, J. (1990). EXPERTEXT: From semantic nets to logic Petri nets. *Expert Systems With Applications,* **1**, 51–62.

Routen, T. (1989). Hierarchically organised formalisations. *Proceedings of the 2nd International Conference on AI and Law* (pp. 242–250). Vancouver: ACM Press.

Routen, T.W., & Bench-Capon, T.J.M. (in press). Hierarchical formalisations. *International Journal of Man-Machine Studies.*

Sprowl, J.A. (1979). Automating the legal reasoning process: A computer that uses regulation and statutes to draft legal documents. *American Bar Association Journal,* **1**, 1–81.

Springel-Sinclair, S. (1988). The DHSS Retirement Pension Forecast and Advice System: An update. In P. Duffin, (Ed.), *KBS in government* 1988 (pp. 89–106). Blenheim Online.

Standard ECMA-101. (1985). *Office document architecture.* Euro. Compu. Manu. Assoc.

Storrs, G.E., & Burton, C.P. (1989). KANT, a knowledge analysis tool. *ICL Technical Journal,* **6**(3), 572–581.

Thomas, R.H., Forsdick, H.C., Crowley, T.R., Robertson, G.G., Schaaf, R.W., Tomlinson, R.S., & Travers, V.M. (1985). Diamond: A multimedia message system built upon a distributed architecture. *Computer,* **18**(12), 65–78.

Wilson, E. (1988). Converting legal texts into expert systems. *Proc. National Conf. on Law, Computers and Artificial Intelligence.* Univ. of Exeter.