ILL: 210424408

FTU TN: 923721

Charge
Maxcost:

Borrower: INL

QA76.76.E95 M69 AR

32103007168663

Moving towards expert systems globally in the 21st century /

Vol: Issue:

Date: 1994 Pages: 1152-1165

Title: Natural Language - An Appropriate Knowledge Representation Scheme for the Administrative Domain Dayal, S., Johnson, P., Mead, D., Liebowitz, J.

Natural Language: An Appropriate Knowledge Representation Scheme for the Administrative Domain

Surend Dayal*, Peter Johnson** and David Mead***

*Legal Analyst
**Director
***Principal Technical Director
SoftLaw Corporation, PO Box 772
Dickson, ACT, Australia 2602

Abstract

In this paper, we argue that a representation scheme based around natural language is particularly appropriate for use by developers in the production of expert systems for the administrative domain. Firstly, traditional knowledge representation schemes are discussed. The weaknesses of these symbolic representation schemes for the representation of administrative knowledge are highlighted, in terms of both theoretical and practical limitations. Secondly, we propose a re-analysis of what constitutes a knowledge representation scheme, and suggest a scheme in which the developer and users work with natural language, while the machine operates on a different level. Finally, we present a natural language representation scheme which we have used to resolve the theoretical and practical difficulties highlighted in the first section.

Introduction: Current Paradigms and Their Limitations

This paper is confined to problems and issues encountered when using expert systems technology in the construction of legal or administrative applications. These types of applications include systems for the automation of business processing, compliance-testing systems, systems based around legislative or other regulatory documents, and systems intended to assist with auditing business requirements. For the purposes of this discussion, we are not concerned with applications for scientific domains such as geology, medicine or pathology. Nor are we dealing with those products which focus on modelling concrete processes, such as inventory control systems, diagnostic evaluation systems or simulation software.

It is certainly true that knowledge based systems can play a role in facilitating better decision making in organisations whose main functions are related to the application of rules, particularly where those rules are laid out in a substantial document defining the domain. Such systems can assist decision makers by:

- reducing the logical complexity of the decisions which the user must make,
- making decisions more consistent, because the user is taken through all of the material relevant to a given decision.

Substantial assistance can also be offered to help users through issues requiring the exercise of discretion, or more fluid concepts which require human judgement for resolution.1

However, there are both practical and theoretical problems currently restraining the development of expert systems for use in the administrative domain. The serious practical problems presently facing expert system developers are the following:

- the problems with trying to massage complex concepts into symbolic form - the modelling problem,
- verification of large knowledge bases,
- maintenance of large knowledge bases,
- provision of adequate user interaction facilities with the knowledge base.

The main reason that these problems still constrain the application of expert systems technology to administrative processes, is that most of the tools and techniques being used in the expert system sphere date back to technology which originated in the late 70s and early 80s. In particular, the knowledge representation schemes used by expert system development tools are inadequate for capturing administrative rules. In our experience, a careful evaluation of most of the applications touted as "successes" for mainstream AI in the administrative field is less than encouraging. Often, close examination will reveal that one or more of the following is true:

- the application is quite trivial in terms of what it actually does,
- there are huge costs associated with developing and maintaining the software,
- it is almost impossible to adequately maintain the software.

This observation is borne out by the fact that despite the many claims to success, expert systems have not been embraced by the corporate or government world. Particularly in the administrative domain, corporate and government agencies have not regarded working systems as numerous enough or successful enough to be a viable option. This lack of credibility has resulted in funding setbacks for the industry.2

In the administrative domain there is typically a body of text such as audit rules, procedures or legislation which defines much of how the business must operate. Alternatively, domain experts can be found to articulate the rules of the business as natural language rules which can then be modelled in a knowledge base. However, capturing this knowledge has been a very difficult process given the traditional representation languages used in expert system development. It is possible to identify five significant characteristics of the administrative rules:

- The knowledge is framed in terms of statements of logic - it is naturally expressed as rules.
- The logical relationships captured are often quite complex, requiring a powerful logic syntax.
- The logic is declarative. Its meaning should be independent of the particular order in which it is articulated.
- Objects are subsidiary to rules in this domain. It is the rules which define the appropriate action in a given situation, and the relationships between the objects.
- The rules are semantically rich. They can contain complex statements and define complex relationships.

In this paper we show how a rich scheme of knowledge representation, based around natural language, can address both these theoretical characteristics of administrative/business knowledge as well as the practical problems of knowledge modelling, verification, maintenance and user interaction.

In the first section, we show that traditional schemes of knowledge representation used in the expert system community fail to adequately cater for either the theoretical nature of administrative knowledge, or the practical problems of expert system construction. In the second section, we show how a reconsideration of what constitutes a knowledge representation scheme has allowed us to provide a representation based around natural language. Such a scheme actually addresses both the theoretical considerations and the practical requirements of a representation scheme for the administrative domain. We have chosen to use English as the basis of the natural language scheme which is discussed in this paper.

Traditional Representation Schemes

Introduction

Here, we look at the three main types of knowledge representation scheme previously used for expert system development in the administrative domain. Firstly, we describe and illustrate each scheme, and highlight their theoretical inadequacies in terms of one or more of the five characteristics of administrative rules. Secondly, the practical difficulties introduced by using any of the three traditional symbolic schemes of representation are analysed.

Theoretical Inadequacies

Object-Attribute-Value Schemes

Schemes based on the old object-attribute-value tuples date back to the days of Prospector and Mycin, and form the basis of the representations used by many modern expert system development environments. Knowledge is represented by objects which have attributes which can assume particular values. In simpler schemes, there may only be attribute-value pairs not tied to any object, or even just states which take on a truth or certainty value.

On a theoretical level, the difficulty with these schemes is that the logic supported by the scheme is very limited: the only verbs supported by the scheme are to be and to have: an object can have-an attribute, the value can be defined. For example, a Porsche has-a colour. The colour is an attribute of the Porsche. The colour is red. The link "is" is a definitional link. The attribute-value or simple state schemes are semantically even weaker. This type of logic is too limiting in the administrative domain, where the rules can be expressed in terms of any verb in the English language.

Frame-based/Object-oriented Schemes

The more sophisticated version of the simple tuples provided in an object-attribute-value scheme are systems based around frames or object models of the world. These provide more complex functions on the objects captured by the representation, and include more sophisticated properties of objects such as inheritance. The developer is given a language in which he/she can attempt to build an object-model of the world.

A typical rule construction in the code would be as follows (taken from NASA's Monkey and Bananas using CLIPS):3

```
(defrule unlock-chest-to-hold-object ""
      (goal-is-to (action holds) (arguments ?obj))
      (chest (name ?chest) (contents ?obj))
      (not (goal-is-to (action unlock) (arguments ?chest)))
=>
      (assert (goal-is-to (action unlock) (arguments ?chest))))
```

Here we run into the cardinal difficulty with any symbolic schemes of representation: lack of semantic richness. Most of the semantics of the code are contained in the rule names, such as "unlock-chest-to-hold-object", or in the actions defined in the scheme.

These representation schemes are further unsuited to the administrative domain because they are designed to capture too much extraneous information about objects and their relationships. An elaborate object model of the world is simply not required in a legal or business setting. In the administrative domain, the objects are subsidiary to the rules which contain them, and these rules can easily be divined either from printed documents or domain experts. In this sphere, creation of an object model is very difficult, time consuming, error prone, inaccurate, subject to interpretation by the knowledge engineer and difficult to verify and maintain. An object based representation may be more suited to situations in which there is a complex set of objects which will interact in complex ways.4

Predicate Logic Schemes

In the administrative domain, probably the best of the traditional approaches to the representation of rules is the use of a logical formalism in which each administrative rule can be viewed as an expression of first order logic, or predicate logic. This allows rules to be written using symbolic languages such as Prolog, in the form of standard horn clauses.5 One of the prime examples of this representation scheme was the formalisation of the British Nationality Act (1981) by the Imperial College Group (ICG) in 1986.6 A logical model of the British Nationality Act was implemented in Prolog. For example, section 1(1) of the Act read as follows:

"1. - (1) A person born in the United Kingdom after commencement shall be a British citizen if at the time of birth his father or mother is
  (a) a British citizen; or
  (b) settled in the United Kingdom."

This logic could then be converted into Prolog syntax. The rule above would become:7

```
acquires_British_citizenship_by_section_1.1_on_date_(X,Y) :-
          born_in_the_UK(X),
          born_on_date(X,Y),
          after_or_on_commencement(Y),
          has_a_parent_who_qualifies_under_1.1_on_date(X,Y).
```

Prolog is closer to the characteristics of business knowledge than the previous two schemes discussed. Being based around predicate logic, it is quite powerful for capturing complex logical relations. It suffers from the fact that the rules are not production rules - their meaning depends on the order in which they are entered, since evaluation of the rules proceeds from top to bottom through the knowledge base. This can be overcome by the use of a shell to define rules, or to implement special reasoning strategies.

Secondly, the definition of complex objects within the rule structure is not very well supported by Prolog. While it is possible to define a large set of logical relations which might tell the developer something about an object (for example that a kiwi is a bird), this is not naturally supported.8 Finally, like the other two schemes of knowledge representation, the major problem with Prolog is that it does not have the semantic richness of a natural language. At best, it could be said that a quasi-natural language can be provided by the use of long predicate names.

Practical Inadequacies

The Key Flaw: Semantic Richness

The key practical deficiencies with all of the traditional schemes used by developers to represent administrative rules stem from the fact that they do not have the richness of natural language, and as a result they are very weak semantically.9 Even in a language like Prolog, which provides a powerful means of inferencing between propositions and can handle complex logical statements, the developer is offered very little assistance in determining the meaning of the concepts into which the logic is imparted.

The use of symbolic languages to implement expert systems in this way can seem attractive, because it allows one language to be both used by the developer, and executed by the machine. However, a very real difficulty with modelling social knowledge arises when the developer tries to relate the model to reality. In the case of symbolic formalisms, this step is purely informal and intuitive. The reader of the code must look at the non-natural representation language and fathom its meaning in a natural language. Often, the only place where the actual meaning of the rules is captured is in the mind of the person who wrote them, and even then this will fade with time.10 Ideally the developer should be able to determine what is being represented directly from the formalism, rather than having to take an intermediate step which only obscures the knowledge contained in the system.11

It is this problem with semantics which inevitably has repercussions in the practical sphere. The other four characteristics of expert system development can almost be fulfilled with a sophisticated symbolic scheme of knowledge representation. For example, using a shell based on augmented Prolog, a scheme could be provided which:

- was rule-based,
- provided for a powerful logic,
- was composed of production rules, and
- incorporated objects within the rule structure.

However, without the rich semantics of natural language, the representation scheme would still fail to be useful for the construction of large systems. In the administrative domain, the developer who uses a semantically weak representation scheme will run into all of the practical difficulties experienced in this sphere to date: the modelling problem, verification, maintenance and the provision of adequate user interaction.

Knowledge Representation: The Modelling Problem

Where a symbolic representation scheme is used, a significant problem in the construction of an expert system is the actual time spent translating natural language rules formulated by the domain expert, or contained in the source documents. In fact it is often the case that the concepts cannot be properly captured at all.

We could take the example of deontic logic to demonstrate this modelling problem. The systems we are discussing are examples of normative systems, in that they incorporate rules which specify a complex web of powers, rights and obligations to regulate human behaviour. The tool of language for expressing such normative concepts has always been that of deontic logic and the deontic operators such as may, ought, must, shall, should, etc. While there has been some discussion on how to go about accounting for the normative nature of administrative rules, as yet very few systems representing such norms have made any use of a deontic logic.12 This is due in no small part to the fact that in a symbolic representation scheme, deontic logic is difficult to handle. The common solution is to try to add the concepts directly into the symbolism, either by the inclusion of new logical operators such as "permittedA" or "oughtA", or in a language like Prolog, by the introduction of deontic operators in the predicate names themselves.

Due to the normative nature of administrative rules, any knowledge representation scheme which purports to account for these rules will usually require some sort of deontic logic.13 The only way to ignore the problem is to confine use of the representation scheme to situations in which the deontic operators can be ignored. Even then, to represent a provision such as "The seller may exercise his rights under section 61 when ..." without articulating the normative nature of the language only serves to obscure the knowledge rather than faithfully represent it.

Verification

Applications in the legal domain (ie. based on statutory documents) have a relatively reliable knowledge foundation: a document which substantially defines the domain. This is also true in the business environment - in a large organisation there are usually policy and procedures manuals to guide lower level staff. Furthermore, where the system is being built for a large organisation, to administer their point of view, domain experts of the organisation are also readily available to offer their guidance. These facts should allow business applications to be readily and reliably verified - in the sense that the rule-based portion of the application accurately captures the rules as articulated by the domain experts.14 However, our experience has been that a knowledge representation scheme in which the developer employs symbolic or abbreviated terms leads to severe difficulties in verifying the correctness of the rulebase.

Symbolic representation of rulebase items appears acceptable in small scale applications. The use of "English-style" phrases within logical predicates or object models certainly assists in the verification of small scale systems. However, the self-explanatory nature of this type of rule language and its apparent reliability can be misleading. A scheme of knowledge representation in which the developer uses a symbolic language, even when extended to a full object-attribute-value or frame-based scheme, suffers from the lack of richness adverted to previously. The individual's overview determines which quasi-natural phrases are used, and what they mean. Effectively, the analyst modelling the law invents a new language every time he/she creates a new knowledge base or changes any part of an existing knowledge base.15

Checking the correctness of a symbolic rulebase becomes a complex process of understanding a piece of computer code and verifying that it correctly maps back to the subject material. This is even more difficult for someone not well versed in the computing sphere. In the administrative area it is typical that those in a position to verify the correctness of the rulebase do not have detailed computer knowledge.16 Hence there is a grave danger that large systems implemented using symbolic logic will stray from the desired point of view.

Maintenance

While a symbolic scheme of knowledge representation merely leads to difficulties in verifying the correctness of large rulebases, maintaining such knowledge bases is actually prohibitive. A rulebase which is not visually coherent to the developer is inherently difficult to maintain. In a domain which habitually relies on symbolic or abbreviated terms, a rulebase which utilises those terms will be visually coherent to an expert. Where symbolic representation is foreign to the domain, any attempt to read the rulebase will require a constant process of translation. It is extremely difficult to update a rulebase which requires translation at every step.

The difficulty is easily represented. Imagine a rulebase of several hundred rules, written in Martian. For the average English language speaker, inserting changes in that rulebase, even with a copy of the subject Martian document translated, would be a painstaking process, fraught with danger. This is precisely the problem which arises when a domain which lacks an internal system of symbols or abbreviation is mapped into a symbolic language. To anyone but the person who invented the quasi-natural language phrases, the system might as well be written in Martian. Small systems can be readily maintained. This ease is misleading when the intended final application is of any significant scale.

User Interaction: Questions and Reports

Symbolic schemes of knowledge representation do not internally generate coherent literal reports unless the knowledge representation accurately reflects conventional symbolic communication in that domain. In the administrative domain, the following report to the end user is unlikely to be acceptable:

```
eligible
because
           under_65
and        male
and        NOT has_disease
and        seek_work
and        CES_rego.
```

A similar report would be unacceptable in various scientific domains (for example, a geographic domain) where graphical output would be required.

The problem with investigation of symbolic rulebases is similar. Asking the user a series of questions like under_65?, male?, has_disease?, seek_work?, and CES_rego? will be inadequate. These are the sort of questions which will be generated directly from a symbolic rulebase, unless there is an intermediate step which associates some canned text with the individual propositions.

In the administrative domain, the problem of limited user interaction is usually approached in one of three ways:

(i) it is ignored, and questions and reports are generated in quasi-natural language;
(ii) the output can be composed of text strings, which are substituted for the literal, abbreviated phrases; or
(iii) a knowledge-based composition of canned text paragraphs is generated (which requires its own small knowledge base and introduces another level of maintenance and potential unreliability).

Method one is clearly not good enough when users are involved in intensive processing. While the actual output generated by methods two and three may be adequate, these methodologies introduce at least three problems:

(i) the system's summary of the method and path used to arrive at the conclusion is removed from the process of arriving at that conclusion and therefore is inherently unreliable - the summary may not disclose questionable inferences or even mistakes in the inferencing process;
(ii) as a practical issue, such a methodology is inelegant and introduces issues of additional maintenance and checking - in a substantial application, this can become a significant problem;
(iii) the addition of the textual explanations requires extra time and effort during the construction of the rulebase.

The Natural Language Solution

Requirements for the Administrative Domain

In the administrative domain, it can be seen that the theoretical inadequacies with traditional representation schemes, in particular their lack of semantic richness, lead into practical difficulties with those schemes. It is clear that a better scheme is required.

Let us assume for a moment that it is possible to use a scheme based on natural language for the representation of knowledge. We instantly provide the developer with all of the semantic richness which the domain requires. The use of natural language as a vehicle for knowledge representation immediately overcomes some of the modelling limitations evident in conventional systems, and would allow the characterisation of all the complex relations capable of expression in the full natural language.

However, this would not be enough. Given a natural language base, all of the other theoretical characteristics of administrative rules must be accounted for:

- The representation scheme must allow rules to be constructed in the natural language.
- The rules must be able to capture complex logical relationships. This will be the case, provided the rules themselves can be expressed in the natural language.
- The natural language rules must be declarative in nature, and have meaning independently of the order in which they are entered. In other words, they must be production rules.
- Within the natural language sentences, there must be the capacity to define objects.

It is only if the knowledge representation scheme adequately accounts for these characteristics of administrative rules that it can hope to solve the practical problems with expert system development such as the modelling problem, verification, maintenance and user interaction.

Traditionally, this sort of knowledge representation scheme has been almost impossible to implement. Past attempts have concentrated on building semantic symbolic models onto which natural language is mapped. We propose a re-examination of the notion of knowledge representation to see whether there is a way of harnessing the power of natural language without solving all of the major problems with semantic modelling of natural language.

Analysing the Notion of Knowledge Representation

The issue of what actually constitutes knowledge representation has received scant treatment over the years,17 nevertheless it is assumed that any particular expert system has a knowledge representation scheme. This assumption often blurs the distinction between a number of discrete functions which must be provided by the scheme. A knowledge representation scheme used to build an expert system must provide:

(i) a language through which a computer manipulates the subject knowledge;
(ii) a language through which the developer manipulates the subject knowledge, in order to represent that knowledge in a manner which the machine can manipulate;
(iii) a language through which the machine communicates the subject material to the user (for example, in questions and reports).

The discrete nature of these functions has not always been clearly articulated. The traditional approach used by all three of the symbolic representation schemes discussed earlier is to conflate these functions into a single language. Where the same language is used by both the developer, in writing the knowledge base, and the machine, in manipulating the logic, the conflation makes it appear natural that there is one representation language at the foundation of any expert system.

As shown in the first portion of this paper, the attempt to provide the developer with a language which can be executed at machine level has profound theoretical and practical consequences for expert system development. There are different constraints and theoretical problems which arise from the needs of the developer and the needs of the machine. Conflation can mean that these constraints and problems needlessly bedevil the first language, and vice-versa.

The requirements of each function are different, and the criteria for evaluating the adequacy of the different languages are different. In particular, we suggest that in the administrative domain there are profound benefits in using natural language as the representation language for the second and third functions, and hiding the machine level operations from developers and users. We move from the analysis of diagram 1, in which there is no distinction between the first and second functions of a knowledge representation scheme, to the analysis in diagram 2.

Diagram 1

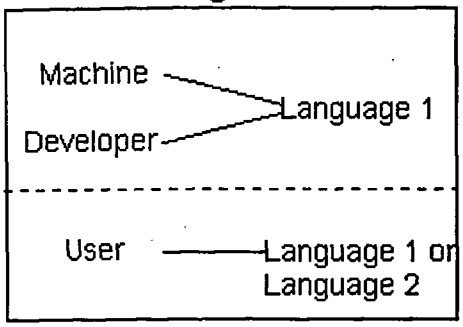
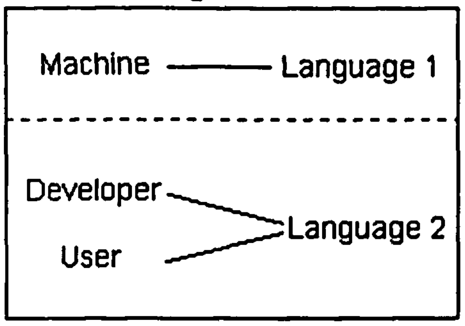


Diagram 2



Of course, these languages must interact seamlessly. If they do interact seamlessly, the theoretical and practical difficulties can be confined in the appropriate language and managed in an appropriate way, without introducing needless complexity.

Theorists have naturally concentrated on the problems of extending the boundaries of machine knowledge. The proper place for pure research in this field is the interface between the knowledge and the machine. However, this purist endeavour should not constrain the capacity of applied researchers to construct practically useful applications based on current capabilities. Within this practical sphere, it is limiting and misleading to conflate the various aspects of knowledge representation into a single scheme.

By separating the schemes of representation in this manner, it is possible with current technology to provide the developer with a language which meets the theoretical characteristics of administrative rules, and at the same time solves all of the practical difficulties we have highlighted in expert system construction.

STATUTE

Background

STATUTE is a knowledge base development environment tailored specifically for the creation and maintenance of applications based on documented rules. *STATUTE* was created over a period of 7 years, working under research grants privately and later at the University of Canberra and the Law Faculty of the Australian National University. *STATUTE* was developed explicitly in order to construct a large expert system application based on Australian Social Security law.

Over the period of the design and creation of *STATUTE*, through several prototypes of expert system shells, we developed a scheme of knowledge representation which we believe to be suited to the domain, a methodology for the creation and maintenance of large knowledge bases, and a set of tools and facilities for building and maintaining appropriate production applications in the legal and business environment.18 We began with symbolic schemes of representation similar to those discussed in the first portion of this paper. However, we ran into all of the practical difficulties with symbolic representation discussed earlier. So we had to re-analyse the theoretical nature of the rules we were trying to model and come up with a representation scheme which captured all of the characteristics of administrative rules

Semantic Richness: Natural Language

STATUTE differs from the type of logical formalism described above in that the language which is used by the developer to represent the logic contained in the legal documents consists of complete English sentences. The "natural language knowledge representation" which is discussed in this paper is not a true Natural Language Interpreter. Our aims are more modest. The significance of the system of knowledge representation is that it provides a natural language in which the developer can capture the administrative rules. This has significant implications for the practical problems confronting developers, which will be discussed in detail below.

STATUTE performs a functional parse (down to phrase level) on the sentences in the rulebase. The sentences can then be restructured by *STATUTE* into any of 6 forms:

(i) a sentence in the positive sense,
(ii) a negation of the sentence,
(iii) a question,
(iv) a statement of possibility that the fact may be true,
(v) a positive imperative statement of the fact,
(vi) a negative imperative statement.

Each sentence used in the knowledge base can thus be manipulated by *STATUTE* to automatically create questions and reports. Since the sentence has been parsed down to phrase level, these can be generated directly from the text of the rulebase (see the appendix for some examples).

STATUTE separates the knowledge representation used by the developer and the user from that operated on by the machine. The internal representation is a form of predicate logic implemented in Prolog and C. However, the parsing overlay ensures that the developers, any independent verifiers and the users of the final system work exclusively with complete English sentences. Hence, the *representation scheme* used by *STATUTE* is really the following:

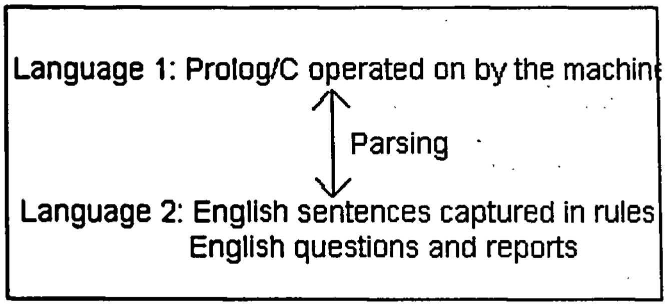

Language 1: Prolog/C operated on by the machin
Parsing
Language 2: English sentences captured in rules
English questions and reports

Diagram 3

A System of Rules

STATUTE uses a rule-based system to represent knowledge. Standard logical operators are provided between the natural language premises - for example disjunction (logical *or*), conjunction (logical *and*) and negation.

Complex Logical Relations

Since the premises consist of complete English sentences, the logic of any verb can be represented in the rulebase. For example, we could model the following sentence (see the appendix for more detail): *the claimant's child went to the shop,* and we could base our logic around the verb "to go". For example, we could say *the claimant's child did not go to the shop,* or the system could generate the query *Did the claimant's child go to the shop?.*

Production Rules

Inferencing based on the rules is forward chaining - once a fact is known to the system, all inferences which flow from that fact will be made. Thus, the rules have declarative status. The meaning is independent of the order of the rules in the rulebase.

Objects

In *STATUTE,* an object is any variable item such as a phrase which can take on a range of characteristics. However, the capacity of the language based knowledge representation to deal with variables is subtle and natural. It is best understood by reference to language, rather than by reference to more limited symbolic systems of knowledge representation.

In the sentence

The claimant's child went to the shop

the phrases *the claimant, the claimant's child* or *the shop* can be made into variables, in much the same way that they are in the English language. *The claimant* and *the claimant's child* can be instantiated as particular persons (say, "Frank" and "Bob") and *the shop* can be instantiated as a particular shop (say, "the butcher"). Thereafter, any use of either of those terms in a rule will limit the application of that rule to the particular instantiation: knowledge can be context sensitive. In addition, certain characteristics can be attributed to any object in the system, such as weight, height, etc.

Since they represent individual phrases, variables which can occur inside sentences are known as phrase variables. Internally, *STATUTE* represents the sentences by a data structure which notes any phrase variables in the sentence. In this way, the knowledge representation scheme goes beyond mere propositional logic, without losing the richness of natural language representation. Multiple instantiations of variable items may be tested using the same rulebase.

Results

Introduction

We have found that using a knowledge representation scheme which accords with all of the theoretical characteristics of administrative rules has had profound consequences for the delivery and maintenance of large administrative applications.

Using the *STATUTE* suite of tools, we have built production systems for the Australian Department of Social Security (DSS), the Australian Department of Veterans' Affairs (DVA) and the New Zealand Accident Compensation Corporation (ACC). Each of these systems is substantial, using at least 3000 rules involving several thousand decision points. Currently, these systems are designed to process from 40,000 claims per year up to 1.2 million claims per year.

In order to show some concrete results of using the scheme for the automated generation of natural language questions and reports, an appendix has been included which shows some source documents, the rules modelling those documents, and questions and reports generated directly from the natural language representation of those rules. The rest of this section discusses the implications for modelling, verification, maintenance and user interaction.

Knowledge Representation: The Modelling Problem

In order to accurately model complex concepts which must be expressed in full English, the developer requires a system which automatically allows the knowledge engineer to express those concepts in English, the user to be queried in full English sentences and the reports to be expressed in full English. Once the developer and the user have to work with anything less than the full English with which *people* deal, then knowledge modelling becomes a significant bottleneck. The separation of the developer/user level from the machine level provides developers with enough power of expression to represent any concept which can be expressed in English.

To take the example described in the first section, it is often necessary to model deontic logic in the administrative domain. An English language representation incorporates all of the modal verbs such as *ought, may, must, could, should,* etc. making them available to the developer. This means that the full range of deontic concepts may be expressed without difficulty, exactly as they appear in the source documents. This is significant in terms of accurately representing natural language texts. If deontic concepts cannot be represented faithfully by the developer then many sentences in the source material would have to be "interpreted" by the developer.19 As can be seen by the problems experienced in verification and maintenance of symbolic knowledge bases, it is desirable to eliminate such developer interpretation as far as possible. Keeping the representation of the deontic operators at the level of the user and the developer removes any need for the machine to incorporate deontic logic in its internal inferencing processes.

Verification

In order to achieve a verifiable rulebase in the business sphere, we have identified four essential elements for a shell:

- a knowledge representation faithful to the domain;
- a style of rule organisation faithful to the domain;
- a strict separation of the modelling of the source documents from the modelling of any interpretative knowledge;
- a strict separation of the rulebase from any procedural aspect of the application.

The last three points are discussed in detail elsewhere.20. Our primary methodology for rulebase verification is essentially visual: domain experts verify the correctness of the rulebase directly against the source documents (in addition to the standard unit and integration testing which accompanies any large application21). Such a methodology requires very high visual correspondence between the rulebase and the source material, not only in organisational format but also in the literal terms of the rule conclusions and premises themselves. Therefore, the methodology is dependent on the rules being represented in a non-opaque, coherent way, which is only possible where a natural language is the modelling language. The verification process is no longer a complex decoding and "unmapping" process.22 In practice, people who have no substantial experience with computers or knowledge based systems, but who are experts in the particular domain, can reliably verify the knowledge base.

In each of our applications, since the rulebase structure is so transparent the representation language used by the developer can be examined by the client's domain experts (who may have little computer knowledge) to verify the correctness of the rulebase. Any discrepancies with the client's policy or point of view are picked up and changed. Exhaustive testing is not possible in large rulebases due to combinatorial explosion, especially when responses other than true or false are allowed for. Case testing is unreliable, and is not well tolerated by clients used to normal system testing. Exhaustive visual verification of a large rulebase is therefore extremely valuable.

Maintenance

We have identified several criteria by which the ease and reliability of maintenance can be measured in this domain. While we do not believe that there can be any "fail-safe" maintenance scheme - all modelling of knowledge is ultimately the preserve of the human system developers - we believe that a maintenance methodology and maintenance tools which are suited to the domain will enhance the reliability of the system.

The criteria which we have identified as relevant are:

- The visual transparency of the rulebase, which allows the system to be reliably checked both before and after alteration by the domain expert. As detailed above, rules modelled in a natural language are non-opaque.
- The degree to which the rulebase incorporates flags to assist in the updating process, and the suitability of those flags to the domain. STATUTE rulebases contain very detailed reference points to the subject legislation (see the examples in the appendix).
- The degree to which maintenance of different components of an application can be separated - rulebase maintenance, procedural maintenance and maintenance of source documents. The rigorous separation of these components ensures that one component can be altered without endangering the integrity of the other components. We regard this as essential in any large scale application.23
- The methodology used in rulebase construction, which also must be suited to rulebase alteration. The basis of the methodology supported by STATUTE is the rigorous and explicit separation of interpretative and heuristic rules from the core domain rules.

This methodology has allowed maintenance of large complex applications by a small team. In terms of maintaining the knowledge base, a single person has duties which include alterations to the rulebase in accordance with organisational and policy changes.

User Interaction: Questions and Reports

In the use of conventional shells for legal and business expert system development, links to external interviewing and reporting text are made necessary because the abbreviated knowledge representation is unnatural and foreign to the domain. A scheme of knowledge representation which is transparent to the users of the application allows both the questions asked of the user and reports to be generated directly from the rulebase. Not only is this more compact, it is more reliable. Without automatic queries and reports, another level of difficulty is added to the task of verifying and maintaining the knowledge base.

In a large-scale application, such tasks become significant. For example, in one of the large systems we built the user interaction with the expert system had to undergo independent verification. A large part of this interaction could be verified directly from the knowledge base. With a symbolic representation scheme, the only alternative would have been a process which involved checking a vast body of independently stored text, that might bear no correlation to the source material.

Limitations with the Natural Language Approach

The main limitation currently faced by the system stems from its strength - English is used to represent the rules. This means that the scheme may not be appropriate for legislation or other rules which are not written in English, because all of the translation processes described above in respect of a symbolic representation scheme will apply. However, the parsing component of the representation scheme is a discrete part, and the scheme could be extended to include any language with a recognisable grammar.

Secondly, the process of parsing is not fully automated. The developer must assist the machine by pointing out various components of the sentences used, such as phrases. This introduces some additional training overhead when the developer first uses the system, as opposed to development environments in which they can very quickly write simple rules. Future work is planned to automate more of the process, for example by including built-in dictionaries.

Conclusions

Traditional knowledge representation schemes tend to fall down as a practical tool for developers in the business domain. The concepts embodied in the natural language rules used in the domain have too much syntactic rigour and involve too many of the rich aspects of natural language to be easily and coherently mapped into a symbolic representation. In many cases, the difficulties posed by a symbolic representation will only be apparent when using the formalism to develop a large scale application.

A methodology based around rules written in a natural language squarely tackles the issues of knowledge modelling, system verification and maintenance, and the provision of adequate query and report facilities. The limitations of the scheme have no bearing on these issues, and are merely problems of efficiency and portability. While a natural language representation scheme will not guarantee perfection, it goes a long way towards helping the developer achieve coherent and correct knowledge bases modelling abstract human concepts.

References

1 For a more detailed description of the role of knowledge based systems in public administration, see Johnson and Mead "Legislative Knowledge Base Systems for Public Administration - Some Practical Issues" Proceedings of the Third International Conference on Artificial Intelligence and Law, 1991, pp.108 - 117, and Dayal, Harmer, Johnson and Mead "Beyond Knowledge Representation: Commercial Uses for Legal Knowledge Bases" forthcoming in Proceedings of the Fourth International Conference on Artificial Intelligence and Law, 1993.

2 See Feigenbaum (1993) "Tiger in a Cage: The Applications of Knowledge-Based Systems", Keynote address in the Proceedings of the Eleventh National Conference on Artificial Intelligence, Abstracted on p.852.

3 Taken from Haley, "NASA's Monkey and Bananas using CLIPS", The Haley Enterprise, Inc. 413 Orchard Street, Sewickley, PA 15143.

4 See the discussion in Jackson, Introduction to Expert Systems (1990) Addison Wesley, pp.206 - 212 for greater depth.

5 For example see Sergot, Cory, Hammond, Kowalski, Kriwaczek and Sadri, 'Formalisation of the British Nationality Act' (1986) 2. Yearbook of Law, Computers and Technology, 40, at p.40, Svensson, "Tessec: an Expert System for Social Security Legislation", in Krach, De Vey Mestdagh and Svensson (eds) Legal knowledge based systems: An overview of criteria for validation and practical use", Koninklijke Vermande BV, Lelystad (1990), pp.87 - 92, p.88, Sherman "A Prolog Model of the Income Tax Act of Canada" (1987) Proceedings of the First International Conference on Artificial Intelligence and Law, pp.127 - 136.

6 See Sergot, Cory, Hammond, Kowalski, Kriwaczek and Sadri, 'Formalisation of the British Nationality Act' (1986) 2 Yearbook of Law, Computers and Technology, 40.

7 For a detailed discussion of the process of conversion from reality to object code see Stamper, "A Logic of Social Norms for the Semantics of Business Information" IFIP WG 2.6 Working Conference on Database Semantics, Hasselt, Belgium, January 7 - 11, 1985 at p.5 and Moles and Dayal, "There is more to life than logic" (1992) 3 Journal of Law and Information Science 188, at pp. 205 - 207.

8 See for example the conversion of frames in Bratko, Prolog: Programming for Artificial Intelligence (1990) Addison-Wesley, pp.348 - 359.

9 Stamper, "The Logic of Meaning and the Meaning of Logic in the Context of a Lawyer's Work" 87, Proc. Council of Europe Conf. on computers in legal Education, Rome 1985, pp. 1 - 19, at p.4.

10 Stamper, "The role of semantics in legal expert systems and legal expert reasoning" 115 Ratio Juris, Vol.4 No.2, pp.219 - 244, at p.227.

11 This is especially true in the case of a large system - Stamper, "The Processing of Business Semantics: Necessity and Tools", in Meersman and Sernadas (eds) Data and Knowledge, North Holland, 1988, pp.1 - 20 at p.7.

12 For a comprehensive survey see Sergot "The Representation of Law in Computer Programs: A survey and Comparison" in Bench Capon (Ed.) Knowledge Based Systems and Legal Applications (1991), Academic Press.

13 Susskind, Expert Systems in Law: A Jurisprudential Inquiry. Oxford University Press, 1987. There has been some debate on the issue - see Bench-Capon, "Deep Models, Normative Reasoning and Legal Expert Systems", in (1989) Proceedings of the Second International Conference on Artificial Intelligence and Law, pp.37 -45.

14 See Dahl, "ESDS: Materials Technology Knowledge Bases, Supporting Design of Boeing Jetliners" (1993) Proceedings of Innovative Applications in Artificial Intelligence, pp.26 - 33, at p.32.

15 See Stamper, "The role of semantics in legal expert systems and legal expert reasoning" 115 Ratio Juris, Vol.4 No.2, pp.219 - 244, at p.227.

16 See for example the large commercial application discussed in Dayal, Harmer, Johnson and Mead "Beyond Knowledge Representation: Commercial Uses for Legal Knowledge Bases" (1993) Proceedings of the Fourth International Conference on Artificial Intelligence and Law, pp.167 - 174.

17 See Davis, Shrobe and Szolovits "What Is a Knowledge Representation?" 1993 14 AI Magazine 1, at pp.17 - 33.

18 The methodology is described in a separate paper by Johnson and Mead - "Legislative Knowledge Base Systems for Public Administration - Some Practical Issues" (1991) Proceedings of the Third International Conference on Artificial Intelligence and Law, pp.108 - 117.

19 An approach to the representation of deontic concepts based on some conceptual analysis by the developer is illustrated in Sergot and Jones, "Deontic Logic in the Representation of Law: Towards a Methodology" (1992) 1 Artificial Intelligence and Law, pp.45 - 64, at pp.59 -60.

20 See "Legislative Knowledge Base Systems for Public Administration - Some Practical Issues" Proceedings of the Third International Conference on Artificial Intelligence and Law, 1991, pp.108 - 117.

21 See Touchton and Rausch, "Putting Expert Systems to the Test: Verification and Validation" (July/August 1993) PC AI, pp.29 - 31.

22 See Stamper, The processing of business semantics: necessity and tools" Proc. IFIP TC2 WG2.6 Working Conference "Knowledge and Data", Albufeira, Portugal, November 1986, at p.7.

23 For a more detailed discussion of the makeup of a STATUTE knowledge base, see Dayal, Harmer, Johnson and Mead "Beyond Knowledge Representation: Commercial Uses for Legal Knowledge Bases" (1993) Proceedings of the Fourth International Conference on Artificial Intelligence and Law, pp.167 - 174.

Appendix: Example of STATUTE Knowledge Representation

The following example demonstrates a small portion (two rules) from a STATUTE based application.

The Legislation

Section 5(1) of the New Zealand Accident Rehabilitation and Compensation Insurance Act (1992) reads as follows:

> 5. Definition of "medical misadventure"---(1) For the purposes of this Act,---
>
> "Medical error" means the failure of a registered health professional to observe a standard of care and skill reasonably to be expected in the circumstances. It is not medical error solely because desired results are not achieved or because subsequent events show that different decisions might have produced better results:
>
> "Medical misadventure" means personal injury resulting from medical error or medical mishap:
>
> "Medical mishap" means an adverse consequence of treatment by a registered health professional, properly given, if---
> (a) The likelihood of the adverse consequence of the treatment occurring is rare; and
> (b) The adverse consequence of the treatment is severe.

The Rulebase

The following is source code from two of the rules which model section 5(1). There are higher and lower level rules surrounding these rules.

Legislative Rule 1 from module s5

The injured person's injury resulted from a medical error or a medical mishap as defined in section 5 if:

option 1:
    The injured person's injury resulted from a medical error

option 2:
    The injured person's injury resulted from a medical mishap

If all the options of this rule are disproved, then it will be concluded that the injured person's injury did not result from a medical error or a medical mishap as defined in section 5

Legislative Rule 2 from module s5

The injured person's injury resulted from a medical error if:

> The injured person's injury resulted from treatment by a registered health professional; and
> The injured person's injury resulted from the failure of a registered health professional to observe a standard of care and skill reasonably to be expected in the circumstances

If this is not the case, then it will be concluded that the injured person's injury did not result from a medical error

Legislative Rule 3 from module s5

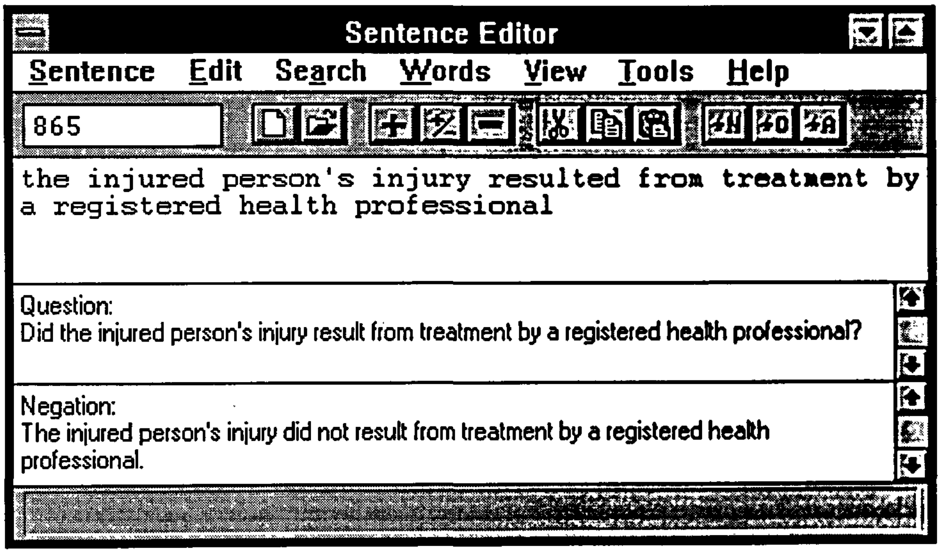
The injured person's injury resulted from a medical mishap if:

> The injured person's injury resulted from an adverse consequence of treatment by a registered health professional, properly given; and
> The likelihood of the adverse consequence of the treatment occurring is rare; and
> The adverse consequence of the treatment is severe; and
> Section 5(3) does not preclude the conclusion that the injured person's injury can be regarded as a medical mishap

If this is not the case, then it will be concluded that the injured person's injury did not result from a medical mishap

The Sentence Editor

This component of the *STATUTE* development environment allows the developer to construct the English sentences used in rules.



the injured person's injury resulted from treatment by a registered health professional

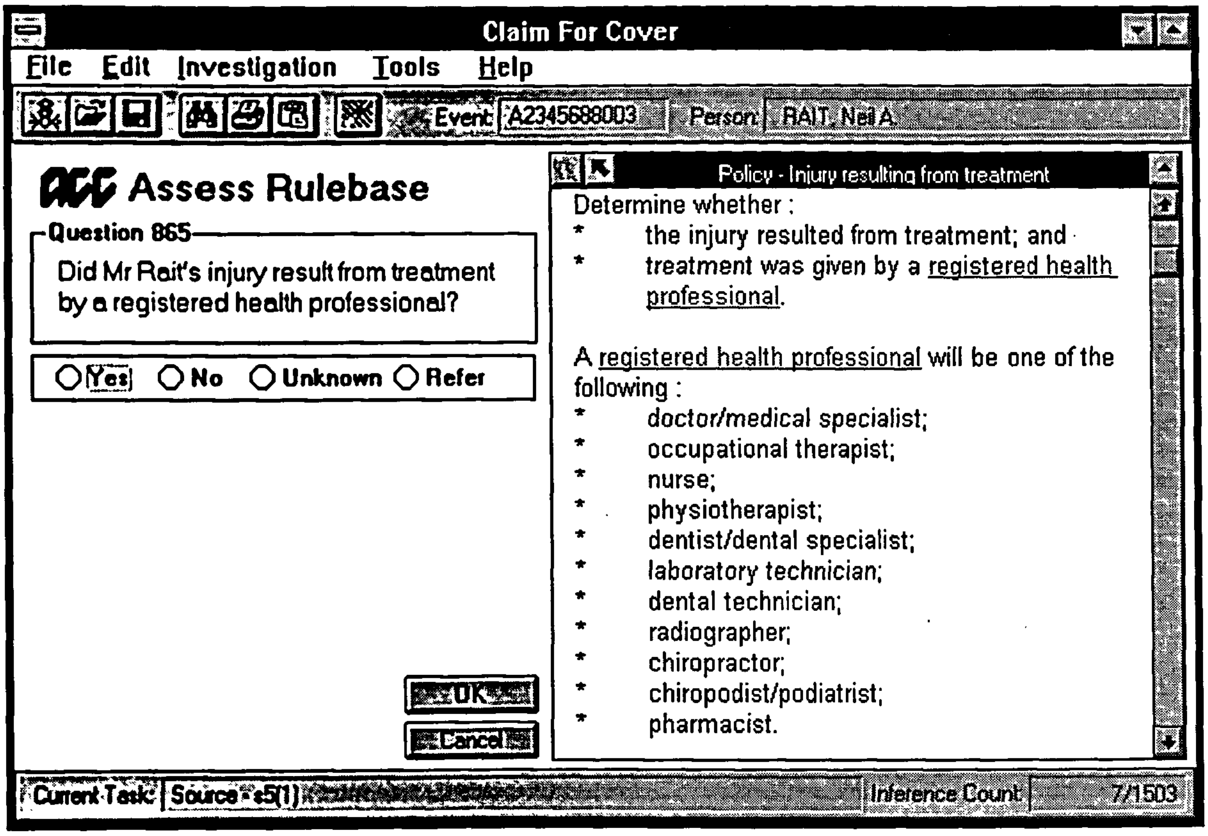
Question:
Did the injured person's injury result from treatment by a registered health professional?

Negation:
The injured person's injury did not result from treatment by a registered health professional.

Diagram 4

A Rulebase Question

An example of a question which can be automatically generated directly from the rulebase. The investigation concerned an instantiation of "the injured person" to "Mr Rait".


Claim For Cover
File Edit Investigation Tools Help
Event A2345688003 Person RAIT, Neil A
ACC Assess Rulebase
Yes No Unknown Refer
OK
Cancel
Policy - Injury resulting from treatment
Current Task Source s5(1)
Inference Count 7/1503

Question 865

Did Mr Rait's injury result from treatment by a registered health professional?

Determine whether :

* the injury resulted from treatment; and
* treatment was given by a registered health professional.

A registered health professional will be one of the following :

* doctor/medical specialist;
* occupational therapist;
* nurse;
* physiotherapist;
* dentist/dental specialist;
* laboratory technician;
* dental technician;
* radiographer;
* chiropractor;
* chiropodist/podiatrist;
* pharmacist.

Diagram 5

A Rulebase Report

One type of report which can be automatically generated directly from the rulebase:

Mr Rait's injury resulted from a medical error or a medical mishap as defined in section 5.
Proof:
Mr Rait's injury resulted from a medical error.
Proof:
Mr Rait's injury resulted from treatment by a registered health professional.
Proof:
Mr Rait's injury was caused by treatment.
The treatment was by or at the direction of a registered health professional.
Mr Rait's injury resulted from the failure of a registered health professional to observe a standard of care and skill reasonably to be expected in the circumstances.

COGNIZANT COMMUNICATION OFFICES:

U.S.A. 3 HARTSDALE ROAD, ELMSFORD, NEW YORK, 10523-3701

AUSTRALIA P.O. BOX 352 CAMMERAY, NSW, 2062

JAPAN c/o OBK T's BLDG 3F, 1-38-11 MATSUBARA, SETAGAYA-KU TOKYO

Library of Congress Cataloging-in-Publication Data

Moving towards expert systems globally in the 21st century / editor, Jay Liebowitz.
p. cm.
Includes bibliographical references and index.
ISBN 1-882345-00-2 : $330.00
1. Expert systems (Computer science) I. Liebowitz, Jay.
QA76.76.E95W69 1994
602'.85 ' 633--dc20 94-4649
CIP

Printed in the United States of America

Printing: 1 2 3 4 5 6 7 8 9 10 Year: 1 2 3 4 5 6 7 8 9 0