Computable Law and Electoral Act: Concepts and Model of an Expert Assistant

C. P. E. Agbachi

Department of Mathematical Sciences, Kogi State University
Anyigba, Kogi State, Nigeria

Abstract — In many a developing country, such as Nigeria, disputes arising from election process can be fraught and common. The lingering nature, controversy all impact adversely on the nation’s body politics. It is evident, the overwhelming number of court cases in the judicial system. Inherent in this situation are long delays and attendant ill effects. With trend in the upward, and no sign of abating, it would be welcome indeed any contributions towards accelerated judgement of court cases in our system. Thus, this paper discusses the concepts of Artificial Intelligence, in application to automated Legal Assistant, as it pertains to the Electoral Act of Nigeria.

Keywords — Jurisprudence, Expert System, Knowledge Engineering, Legal Reasoning, Hybrid-Base.

I. INTRODUCTION

Computable law emanates from the study of Law and Computing, discussions in many forums [1, 2]. It is an ongoing investigation into the process of law and how models can be constructed so that same process can be assigned to a machine. They have been varying degrees of success across organisations over the decades, as well as issues of acceptability. Yet, it remains well received within the context of an Expert System as a legal assistant.

A. Expert System

Expert System is a relatively new and modern concept evolving from main stream application of Artificial Intelligence. By this model, a machine can achieve the Turing Test [3] when the scope is confined to a narrow domain of specificity. Over the decades, there have been many Expert Systems and Assistants, with results as good as human expertise and thus, one reason to adopt use in the field of law.

An Expert System includes key components, Knowledge Base, Inference Engine and Work Base, comprising of facts and databases, Fig 1.

1. User Interface:

In generic terms anyone can be a user of an expert system. But there are cases where the ES is not supposed to supplant the human of which a trial judge is one. So while the system may be as good as a judge, it should operate with human oversight. Thus, the user should be a legal professional.

The user interface would provide the same environment as an election court room. By this model, legal opinions, facts and adjudications etc. are available from the interface.

2. Domain Expert:

The domain expert is vital in constructing an expert system. It is this knowledge that is represented in the machine as source of expertise. This expert therefore must be very knowledgeable in the field of law in matters arising from electoral act and adjudication, in order to achieve congruence between formal and informal reasoning, Fig 2.

The form and models for translating this wisdom into machine readable format is known as knowledge representation (KR). The process is carried out through Knowledge Engineering in order to create a Knowledge Base (KB).

3. Knowledge Base:

Knowledge Base is the one of the modules of an expert system, which consists of information provided by the expert. It is where the knowledge is stored, the output of Knowledge Representation.
Acquired information from expert must be in the form of problem solving rules or procedures, Fig 3. Other familiar methods include, Predicate Logic, Fuzzy Logic, Semantic Networks and Frames.

4. Work Base:

A Work Base consists strictly of working memory that comprises of facts of the matter. However in real situations this embraces extended and associated databases. In this context, apart from facts at the tribunal [4], public references include Electoral Act as available at National Election Commission. It also includes the Nigerian Constitution and landmark cases in the Appeal and Supreme Court, Fig 4.

5. Inference Engine:

The inference engine is a representation of the human mind and comprises of generic control mechanism that applies the axiomatic knowledge in the knowledge base to the task-specific data to arrive at some solution or conclusion. There are quite a number of inference techniques that include:

- Pattern Matching
- Case-Based Reasoning, analogy
- Logical inference:
  - deduction
  - abduction
  - induction
- Rule-Based Inference:
  - forward chaining
  - backward chaining
  - top-down
- Problem reduction (AND-OR graph search)
- Consistency-Based Reasoning
- Graph Transformations, Graph Grammars

II. MODELS

The fields of law are quite numerable and with various areas of specializations. Branches include Constitutional Law, Land Law, Property Law, Medical Jurisprudence etc. For each of these fields, there are different models for optimisation. With the potential application in Electoral Act, the interest would be in a jurisprudential representation.

A. Jurisprudence

Jurisprudence [5], in the proper sense of the word, is the science of law. It is that science which has as its function, to ascertain the principles on which legal rules are based. By so doing, it is not only able to classify those rules in their proper order, and show the relation in which they stand to one another, but also settle the manner in which new or doubtful cases should be brought under the appropriate rules. Jurisprudence is more a formal than a material science. When a new or doubtful case arises to which two different rules seem, when taken literally, to be equally applicable, it may be, and often is, the function of jurisprudence to consider the ultimate effect which would be produced if each rule were applied to an indefinite number of similar cases, and to choose that rule which, when so applied, will produce the greatest advantage to the community.

Given the above overview, questions arise on suitable foundations for a Legal Expert System in relation to jurisprudence. While there are a variety of views a number of important opinions stand out and may be stated as follows [6, 7]:

a. All expert systems in law necessarily make assumptions about the nature of law and legal reasoning. As such they must embody theoreies of legal knowledge, legal science, the structure of rules, the individuation of laws, legal systems and sub-systems, legal reasoning, and of logic and the law (as well perhaps as elements of
a semantic theory, a sociology, and a psychology of law), theories that must all themselves rest on more basic philosophical foundations. If this is so, it would seem prudent that the general theory of law implicit in expert systems should be explicitly articulated using (where appropriate) the relevant works of seasoned theoreticians of law.

b. Successful legal knowledge engineering presupposes a profound familiarity with the nature of law and legal reasoning. It is therefore scarcely imaginable that such mastery could be gained other than through immersion in jurisprudence.

c. The scientific character of law is justifiable on three grounds: the demand for full justice, that is for solutions that go to the root of controversies; the demand for equal justice, that is a like adjustment of like relations under like conditions; and the demand for exact justice, that is for a justice whose operations, within reasonable limits, may be predicted in advance of action. In other words, the marks of a scientific law are conformity to reason, uniformity, and certainty.

d. Judgment machine naturally follow the concept of scientific character of law and is welcome in the sense of disclose all possible available alternative legal rules. But while a machine could perform some of the functions of a judge, that role for humans would remain because the solution to a legal problem may depend upon extra-rational factors, involving the whole of human experience.

e. From a jurisprudential point of view, the law has been made completely determinable. If a computer can be programmed to make judicial decisions, human discretion will have been completely removed. By this argument, people would live under the rule of law and not under the rule of persons.

There are also equally strong opinions on the other side of the divide. For example, responding:

f. It is true that all legal expert systems necessarily make assumptions about the nature of law and legal reasoning. However, a lawyer must have a model of the law which includes assumptions about the nature of law and legal reasoning, but that model need not rest on basic philosophical foundations. It may be a pragmatic model, developed through experience within the legal system. Many lawyers perform their work with little or no jurisprudential knowledge and there is no evidence to suggest that they are worse at their jobs than lawyers well-versed in jurisprudence.

g. It is normal to acquire technical skills of legal reasoning and legal argumentation which make up the concept of "good lawyer" by immersing oneself in substantive legal subjects. So, Jurisprudence has to do, not with the lawyer’s role as a technician, but with any need he may feel to give a good account of his life’s work to community.

h. A legal expert system need only operate at the same level of abstraction as a lawyer, rather than at the philosophical level of a jurist. The fact that many lawyers have mastered the process of legal reasoning, without having been immersed in jurisprudence, suggests that it may indeed be possible to develop legal expert systems of good quality without jurisprudential insight.

i. The ideal of certainty in law is tolerable only in the context of an empirical world in which forces inducing change are so manifold that the attainment of the goal is never possible. A judgment machine in removing uncertainty in the law raises the possibility of the petrifaction of the law.

j. Science does not and will not offer any law machines that give automatic answers to specific questions put to them, whether as to particular cases or as to ultimate legal issues such as the relative importance of interests that may be in conflict. It may, indeed, provide data from which social or ethical judgments may be made; but the judgments will remain with man.

k. As the law is expressed in natural language, it is subject to considerable semantic indeterminacy. It is therefore possible to use rules deductively to solve clear cases, where there is general agreement that they fall within the scope of a rule. The English system of precedent has, by its use, a body of rules of which a vast number, of both major and minor importance, are as determinate as any statutory rule. They can now only be altered by statute, as the courts themselves often declare in cases where the merits seem to run counter to the requirements of the established precedents.
An appraisal of these positions may leave one ambivalent. However any decision on model has to be situated in context of challenges the Legal Expert System is expected to overcome. This task can be amplified as in Fig 5, of a typical judicial process. It can be seen that same arguments, similar cases, often lead to different court resolutions.

![Non-determinism in Judicial Process](image)

As such disputes arising from election process are fraught and common. The lingering nature, controversy all impact adversely on the nation’s body politics. Inherent in this situation are long delays and attendant ill effects. With upward trend, it is timely any contributions towards accelerated judgement of court cases in our system, which minimizes non-determinism.

With this objective, the arguments in (a) to (e) are supportive and appear to offer solution. Even the opposing contentions in (h) and (k), give grounds in favour of automation. The Electoral Statute is in nature of clear rules with mostly clear cases, and thus amenable to jurisprudential model and software application.

### B. Knowledge Representation

Knowledge Representation (KR) can be described as a concept for representing a problem in the language that a computer can understand. For the electoral statute this means constructing, through knowledge engineering, a knowledge base representation of the opinion of a legal expert.

There are three basic methods in consideration, namely:

- Rule-based
- Case-based and
- Hybrid model

#### 1. Rule-Base:

A rule is a conditional statement that links given conditions to actions or outcomes [8]. By itself, a rule-based system consists of if-then rules, a bunch of facts, and an interpreter controlling the application of these rules, given the facts. These if-then rule statements are used to formulate the conditional statements that comprise the complete knowledge base. A single if-then rule assumes the form 'if x is A then y is B' and the if-part of the rule 'x is A' is called the antecedent or premise, while the then-part of the rule 'y is B' is called the consequent or conclusion.

There are two broad kinds of inference engines used in rule-based systems: forward chaining and backward chaining systems.

In a forward chaining system, the initial facts are processed first, and keep using the rules to draw new conclusions given those facts. As the processing progresses, new subgoals are also set for validation. Forward chaining systems are primarily data-driven, while backward chaining systems are goal-driven.

Consider an example with the following set of if-then rules:

**Rule 1**: If A & C then Y

**Rule 2**: If A & X then Z

**Rule 3**: If B then X

**Rule 4**: If Z then D

Suppose the task is to prove that D is true, given that A and B are true. According to forward chaining, start with Rule 1 and go downward till a rule that fires is found. Rule 3 is the only one that fires in the first iteration. After the first iteration, it can be concluded that A, B, and X are true. The second iteration uses this valuable information. After the second iteration, Rule 2 fires adding Z is true, which in turn helps Rule 4 to fire, proving that D is true.
Forward chaining strategy is especially appropriate in situations where data are expensive to collect, but few in quantity. However, special care is to be taken when these rules are constructed, with the preconditions specifying as precisely as possible when different rules should fire.

In the backward chaining method, processing starts with the desired goal, and then attempts to find evidence for proving the goal.

Resuming with the same example, the task to prove that D is true would be initiated by first finding a rule that proves D. Rule 4 does so, which also provides a subgoal to prove that Z is true. Now Rule 2 comes into play, and as it is already known that A is true, then the new subgoal is to show that X is true. Rule 3 provides the next subgoal of proving that B is true. But that B is true is one of the given assertions. Therefore, it could be concluded that X is true, which implies that Z is true, which in turn also implies that D is true.

Backward chaining is useful in situations where the quantity of data is potentially very large and where some specific characteristic of the system under consideration is of interest. If there is not much knowledge about what the conclusion might be, or there is some specific hypothesis to test, forward chaining systems may be inefficient. In principle, the same set of rules is applicable for both forward and backward chaining. In the case of backward chaining, since the main concern is with matching the conclusion of a rule against some goal that is to be proved, the 'then' (consequent) part of the rule is usually not expressed as an action to take but merely as a state, which will be true if the antecedent part(s) are true.

Rule base logic follows the so-called rule of the thumb. That is model of human reasoning. The Electoral Statute comes under definitional law and in rule-based environment. It allows one to propose solutions quickly, thus avoiding the long process of decomposition and recomposition involved in assembly process. It is useful in situations where the domain knowledge is not completely available or difficult to obtain. The past cases may help to provide warnings of potential problems that have occurred in the past and to avoid repeating the mistakes. However, it is to be emphasised that blind use of past cases will lead to situations should be avoided and knowledge cum expertise is needed to transform or to adapt the past case to the current problem.

CBR provides many advantages to problem solving in a knowledge-based environment. It allows one to propose solutions quickly, thus avoiding the long process of decomposition and recomposition involved in assembly process. It is useful in situations where the domain knowledge is not completely available or difficult to obtain. The past cases may help to provide warnings of potential problems that have occurred in the past and to avoid repeating the mistakes. However, it is to be emphasised that blind use of past cases will lead to situations should be avoided and knowledge cum expertise is needed to transform or to adapt the past case to the current problem.

With regard to Electoral Law, there are cases that are resolved in the Appeal and Supreme courts that serve as references. In this respect, the principle of stare decisis applicable, which holds that courts should apply the doctrine of precedent.

2. Case-Base:

An expert is one who has vast specialised experience, having witnessed numerous cases in the domain and generalised this experience to apply it to new situations. When confronted with a problem, the expert is reminded of previous similar problems and their respective solutions. It might be that the expert has so many exemplary cases for a given problem that the experience has been distilled into a general rule to be applied. Still, this general rule has its root in actual experience. Hence it can be stated that the design expert derives the knowledge from experience and the basic unit of knowledge is case but not rules. Experts gain knowledge through accumulating new design episodes, remembering their own experiences and the lessons learnt from mistakes. They can reason by analogy and solve the new problems [9].

Reasoning based on the similar past problem-solving experience helps the designer to exploit the useful details for application to a particular similar case. This problem-solving strategy is termed case-based reasoning (CBR) [10]. It is based on the observation that human reasoning processes are founded on specific experience rather than a set of general guidelines. Thus compared to other AI-based reasoning methods, CBR is a process of considering past cases and arriving at decisions on comparison between the current situation and the old cases. The solutions to problems are accomplished from past experience, stored in the form of cases, rather than from rules or first principles. That is, the case-based problem solver works by recalling what has happened in the past in similar situations rather than by projecting what could work in the future.

CBR provides many advantages to problem solving in a knowledge-based environment. It allows one to propose solutions quickly, thus avoiding the long process of decomposition and recomposition involved in assembly process. It is useful in situations where the domain knowledge is not completely available or difficult to obtain. The past cases may help to provide warnings of potential problems that have occurred in the past and to avoid repeating the mistakes. However, it is to be emphasised that blind use of past cases will lead to situations should be avoided and knowledge cum expertise is needed to transform or to adapt the past case to the current problem.

With regard to Electoral Law, there are cases that are resolved in the Appeal and Supreme courts that serve as references. In this respect, the principle of stare decisis applicable, which holds that courts should apply the doctrine of precedent.

A case-based system is typified by four key phases: Retrieve, Reuse, Revise, and Retain.

![Fig.7](http://www.ijcttjournal.org)
Whenever a new input case has to be dealt with, the system performs inference in this cycle Fig 7.

Retrieval of appropriate cases is a most important task. The recall of past cases is done based on the similarities between the current case and the past cases. It is possible that many cases may be available for an current problem situation. However a case, in general, is a contextualised piece of knowledge representing an experience and represents knowledge at an operational level. For example, it contains specific knowledge that was applied or the particular strategies that were applied for solving a problem. Thus, there are two parts of a case:

- The knowledge it contains and
- The context in which it can be used

It is this second part which is important for selecting or retrieving a case for a given context. One of the widely adopted techniques is to use indices and similarity metrics for selecting cases.

In the reuse phase, a solution for the new case is created based on the retrieved most relevant case(s).

The revise phase validates the correctness of the proposed solution, perhaps with the intervention of the user. Finally, the retain phase decides whether the knowledge learned from solution of the new case is important enough to be incorporated into the system. Quite often the solution contained in the retrieved case(s) is adapted to meet the requirements of the new case.

Usual adaptation methods are substitution, transformation and derivational replay [11, 12]. For the adaptation task, domain knowledge, usually in the form of rules, is employed. Incorporation of knowledge during the operation of a case-based system enhances its reasoning capabilities. This is a major advantage, since the knowledge base of intelligent systems employing other representations remains rather static during operation.

3. Hybrid Base:

The point has been made for rule based system as being ideal for representing knowledge involving electoral statute. However, in any set of rules there are bound to be exceptions and in such a situation the system would be found wanting.

A case based system embodies the knowledge and experience of an expert who can reason by analogy. Yet when the case is entirely new, without any precedent, or when case library is limited to a few records, the best recourse would be to available statute. Against this background, the best option lies in an integrated system of rule and case based legal expert system, Fig 8.

The form of this integration varies with applications. However, as described in [13], there are modes of operation in Rule-Dominant, Balanced Mode and Case-Dominant models.

In Rule Dominant model, rule-based component prevails in the inference process, such that the case-based component plays a complementary role. That is to say the priority is for the rules to fire to conclusion but in the event they are unable to do so, case-based components are invoked to deal with the situation. Examples include IKBALS II [14].

At Balanced Mode, components are equal in supportive mode. Examples include CABARET [15], dealing with legal reasoning. The architecture consists of two co-reasoners, the rule-based component and the case-based component of equal status. There is also a controller, which observes the operation of the whole system and each co-reasoner separately. It then decides how they will proceed in the reasoning process as a whole and individually, assigning tasks to each co-reasoner.

Case Dominant category consists of models in which the case-based component plays a more important role and the rule-based component is less significant. This situation can arise when the case library contains limited number of cases such that rules are invoked when the cases cannot produce a solution. In this paradigm, the rules play a supportive role to case-based reasoning.

III DESIGN

The design follows hybrid base architecture. This includes interface, and control mechanisms, Fig. 8. A client and server, internet based configuration is visualized for parallel use and access from any court location. The starting point though is a model of legal reasoning.

Fig. 8

ISSN: 2231-2803  http://www.ijcttjournal.org  Page 96
A. Model of Legal Reasoning

An Expert System advice should mirror the forms and interface that are generally used by lawyers. Hence, in this respect, the system being considered will be so designed as to operate at the same level of abstraction as would a lawyer.

A lawyer examines the facts of the case in question, the instant case, and determines which area of law, and which statutes (if any) apply. These statutes are applied to the facts of the instant case. The meaning of a concept in a statute may be open-textured, and may determine the result of the application of that statute to the instant case. A lawyer argues about the meaning of an open-textured concept by reference to the facts of the instant case and those of previously decided cases.

The results ascribe a meaning to an open-textured concept which, when the statute is applied, leads to a desired result in the instant case. No two cases can be completely identical, given the plethora of facts associated with any given case. Some of these differences may be insignificant, and much of a lawyer’s reasoning by analogy concerns the legal significance of these differences. Thus, a lawyer argues with cases in the following fashion:

- If the result of a previously decided case is desirable, she/he argues that there are no legally significant differences between the previous case and the instant case, so the previous case should be followed.

- If the result of a previously decided case is undesirable, she/he argues that there is some legally significant difference between the previous case and the instant case upon which the previous case should be distinguished.

B. Interface

Inference embodies models of legal reasoning, with user communications through an interface. This is designed to recreate the environment of a tribunal court room. Furthermore, it should in the forms that are familiar in legal practice.

In a more expanded mode, the interface includes a Blackboard [16]. The blackboard is a shared repository of problems, goals, partial solutions, suggestions and contributed information. The blackboard can be viewed as a dynamic library of requests and contributions that have been recently provided through the cooperation mechanism between the rule base knowledge and the case base knowledge.

C. Control

In some hybrid systems, users have an option to choose rule or case based approach in arriving at a resolution. The design in this application adopts a rule-dominant approach. It is preferable though to automate the process. For this reason a control mechanism is required.

The control mechanism can examine and assess all the options available in rule and case base and assign scores for each as described in [17]. It is then able to compute best rule and best case results and make a selection by comparison.

The control mechanism can also reorganize the knowledge bases and use them in the most effective and coherent method. Facility exists when the need arises, in adaptation, for passing the appropriate part of knowledge from one knowledge base to another and converting from one representation form into another. Control and cooperation mechanisms make use of the work data base, the knowledge bases, and the blackboard to accomplish task.

Work base comprising of working memory, includes facts of the case, Electoral Act, Nigerian Constitution, and landmark election cases, Fig. 4. At the onset, the working memory usually is empty but during the execution of the system it will accumulate and comprise of assertions generated from the processes of cooperation between the knowledge bases.

IV. IMPLEMENTATION

The development of an expert system usually follows a path of construction in the environment of a shell. These shells comprise of all the components but without a knowledge base. This domain information has to be provided in the course of development. The advantage of shells is in the fact that inference engine and interface are already available to use, freeing and assisting the engineer in a speedy exercise.

There are many commercial shells such as CLIPS1, EXSYS2 etc. However for a prototype, Visual Prolog [18, 19] which is freely available is satisfactory. This is the shell in supervised work, Development of a Legal Expert System for Handling Electoral Cases [20]. The salient features of this work are a template to build on and can be described in the data design, program flow and interface.
A. Data Design

The data flow diagram of the system is illustrated in Fig. 9. The DFDs model the relationship between various components and show how input data is transformed to output results through a sequence of functional transformations.

The program flow chart is illustrated above, Fig. 10. It may be added that in a web based client and server configuration, as envisaged, the records and databases would be centrally located within the secure precincts of the Supreme Court.

B. Program Flow

The launch of the system and explanation menu lists six electoral offences in the electoral act. And when any in the list is selected, a new window page containing predefined rule for that offence appears.

Hence a case of impersonation brings up the page in Fig 11, containing the rules guiding against impersonation.

The next course of action is to fill the form and check the boxes against the rules of electoral act. Once satisfied, the Simulate Case button is activated to generate a Conclusion as in Fig 12. OK button saves the judgement to records.
V. CONCLUSION

An analysis and evaluation of the performance of the system prototype shows that the conclusions are in accord with human legal reasoning. Thus, the law is computable and both jurisprudential approach and case procedure are perfectly representable.

However the human oversight is important and the role of any legal system is to enhance the process of law. With further development in progress, the goal of complete coverage and representation of the electoral law is in sight. It is then expected that lawyers and judges would avail of this package. The result will be very welcome, leading to consistency, uniformity, determinism and prompt in judicial decisions.

REFERENCES
[1] Rinke Hoekstra, “Knowledge Representation in the Legal Domain”, Leibniz Center for Law, Faculty of Law, University of Amsterdam.