# Legal knowledge based systems
JURIX 92
Information Technology and Law

The Foundation for Legal Knowledge Systems
Editors:
C.A.F.M. Grütters
J.A.P.J. Breuker

H.J. Van den Herik
A.H.J. Schmidt
C.N.J. De Vey Mestdagh

C. Groendijk, Neural Schemata in Automated Judicial Problem Solving, in: C.A.F.M. Grütters, J.A.P.J. Breuker, H.J. Van den Herik, A.H.J. Schmidt, C.N.J. De Vey Mestdagh (eds.), Legal knowledge based systems JURIX 92: Information Technology and Law , The Foundation for Legal Knowledge Systems, Lelystad: Koninklijke Vermande, pp. 147-158, 1994 ISBN 90 5458 031 3.

# NEURAL SCHEMATA IN AUTOMATED JUDICIAL PROBLEM SOLVING

C. GROENDIJK
Computer/Law Institute, Vrije Universiteit, Amsterdam, The Netherlands

*Summary*

*In most contemporary legal knowledge based systems, conclusions are reached by applying rules to case descriptions. A case description usually consists of a limited set of facts. In human judicial problem solving, the application of legal rules is not based on the facts directly, but on a structured interpretation of these raw data. A structured data interpretation serves as a guide through the problem space; it enables the problem solver to ask context sensitive questions and to make plausible default assignments. In this paper, a neural method to create structured data interpretations is advocated and a method to integrate these networks with a rule based system is presented.*

## 1. Introduction

In the last decade, a new form of information technology, neural networks, has gained popularity in the scientific community. Scientists in the field of A.I. and Law have also taken notice of this technique and its potentials for application in legal knowledge based systems. Neural networks have often been suggested as instruments to tackle the problem of so-called vague terms [Philipps, 1989][Opdorp et al., 1991][Rose & Below, 1989]. In this paper, however, neural networks are used to create structured schematic interpretations of case descriptions.

In most contemporary legal knowledge based systems conclusions are reached by applying rules to a case description which consists of user provided facts and consequences of applied rules. One of the basic assumptions of the research presented here is that such a case description is to shallow in a domain where there seems to be no limit on the facts that might be relevant to the analysis of a particular legal problem [Gardner, 1987]. A human problem solver seems to fit a case immediately to a description that captures the structure and expectations inherent in the case and such a knowledge-rich description is necessary to enable efficient problem solving, context sensitive and adequate questioning and the recognition of relevant jurisprudence.

Of course humans always outdo knowledge based systems with regard to the amount of knowledge they possess, but, and maybe more importantly, the knowledge is also better structured. In human problem solving the knowledge seems to be organized in conceptual structures or schemata which, during problem solving, interact smoothly with the knowledge of the current problem and the way the problem is solved.

The idea that knowledge is organized in conceptual structures has appeared to many investigators, especially in the field of cognitive psychology. On the level of more or less implementational ideas, Minsky [1975] has developed the concept of frames and Schank & Abelson [1977] developed the concept of scripts. Although these models capture the basics of the general idea, they are still very inflexible. [Rumelhart et al., 1988] present a neural model in which most of the inflexibility of frames and scripts are circumvented. This paper presents a method to use these neural conceptual structures in a knowledge based system based on first order predicate logic.

Section 2 presents a short introductory to neural networks; furthermore, it contains an overview of how neural networks have been used in the field of A.I. and Law. Section 3 discusses the basics of frames and scripts, or, more generally, schemata and presents a neural version of this idea. The fourth section discusses the integration of these neural structures with a rule based system at two levels: the domain level and the level of control.

## 2. Neural networks

Neural networks are structures which resemble the functioning of the brain on a metaphoric level. Typically, a neural network consists of a large number of processing units, also called neurons or nodes, which are highly interconnected. A processing unit typically performs very simple arithmetic: as a function of the input it receives from its neighbours, it reaches a certain degree of activation; as a function of its activation an output value is computed which is sent to (other) neighbours. Connections between neurons are either excitatory or inhibitory: excitatory connections increase the input to connecting neurons whereas inhibitory connections tend to lower the activation of following neurons. The effectiveness with which a signal is transported through a connection may vary and is expressed as a weight: excitatory connections have positive weights; inhibitory connections have negative weights. It is important to see that although a neuron's own computation is very simple, its final activation is a result of the activation of connecting neurons and the weights of the connections; the intelligence of neural networks emerges as a result of the complex interconnections between its processing units.

Neural networks have often been suggested as a method in the field of A.I. and Law. Philipps [1989], for instance, describes a method to use a backpropagation network to establish the applicability of a judicial predicates. Fernhout [1989] used a two-layered network to find the most likely applicable legal rules of a indictment on the basis of its text. Opdorp & Walker [1990] propose to use neural networks in PROLEXS after they realized that their original method used could be reinterpreted as a perceptron. In [Opdorp et al., 1991] a method is described to use neural nets as a tool to capture expert knowledge on vague terms. Rose & Below [1989] use a combination of symbolic and sub-symbolic (neural) methods to tackle the ambiguities and vagueness occurring in legal language.

## 3. Structured data interpretation

### 3.1. Frames and scripts

One of the basic ideas in cognitive psychology to explain human cognitive functioning is that memory is organized in large and highly structured modules containing generic information about entities of the world we live in. Many have argued that structures like these are the basic building blocks of cognition [Rumelhart, 1980]. During problem solving, or whatever task a human is performing, appropriate memory modules are activated to offer an interpretation of the situation the problem solver encounters; the activated knowledge structure provides an active context which enables the problem solver to use general as well as problem specific knowledge.

A very well-known attempt to model this general idea into a method which is usable by computers is presented in Minsky's so-called "frame paper" [Minsky, 1975]. In this theory, knowledge is organized in a hierarchy of frames, a frame being a collection of slots that describe a stereotyped object, act or event. The intent of Minsky's ideas are best described via one of his examples: an anecdote describing a person's experiences as he opens a door and enters a room. The act of opening the door activates a "room-frame" which tunes the person's perception in order to help him interpret what is behind the door. The person would, for instance, have little difficulty recognizing the living room as such, whereas, if what is seen behind the door consists of a seashore, it would take some time to recognize this scene, and it would leave the person quite disoriented. If the "room-frame" is confirmed by the scenery, the frame not only fits the scenery, but also provides the observer with knowledge inherent in the situation.

Schank & Abelson [1977] developed similar ideas. Their basic knowledge structure is the so-called script: a frame-like structure that is specialized toward describing a stereotypic sequence of actions. In analogy with the functioning of frames, the activation of scripts helps to interpret stories. For instance the sentence sequence: - *John went to the restaurant; - He asked the waitress for a hamburger; - He paid the tip and left,* is very easy to understand because the sequence triggers a knowledge structure (the restaurant script) which describes a stereotypic sequence of events whereas the sequence: - *John went to a park; - He asked the midget for a mouse; - He picked up the box and left,* is very difficult to comprehend although comparable in syntactic structure.

Both in script and frame theory, the knowledge structures are somehow a compromise between rigidness and flexibility. On the one hand, the structure provides stereotypic information. On the other hand, the structure should be usable in more then one situation so the mechanism must allow for some flexibility. A frame or script-like structure in it self provides general abstract knowledge; the task or the problem to be solved provides specific information and a flexible mechanism allows the two together to create an active context (a model) in which the problem is solved. In frames, the flexibility is achieved by allowing for some slots various fillers and providing a default value in the absence of information; other slots have fixed values.

## 3.2. A neural model

Rumelhart, Smolensky, McClelland and Hinton [Rumelhart et al., 1988], present a neural version of the concept of schemata (frames and scripts). The inspiration to do so is derived from the fact that one of their main goals is to offer an alternative framework to describe human thought and this framework should also include the key concepts developed in cognitive psychology. In this model a schema is a neural network in which nodes or groups of nodes represent slots. As is the case with Minsky's frames, one has to distinguish between passive and active schemata: A passive schema contains general and abstract knowledge; an active schema or a so-called instantiated schema is a structure which contains a mixture of problem specific and general knowledge. This distinction is even more prominent in neural schemata: a neural schema gets instantiated by running the network.

In what follows in this section, first the passive, i.e. general and abstract, knowledge representation of neural schemata is discussed. Running a schema is presented thereafter.

### 3.2.1. Knowledge representation

Fig. 1 presents a "person-schema" as a small example of a neural schema. A person-schema is a knowledge structure that is able to provide a structured interpretation of what qualities might be attributed to a person. Each node represents such a quality and might be considered a slot. However, it is sometimes better to regard a set of nodes as a slot; e.g. the nodes representing "is_male" and "is_female" represent the "gender slot" together.

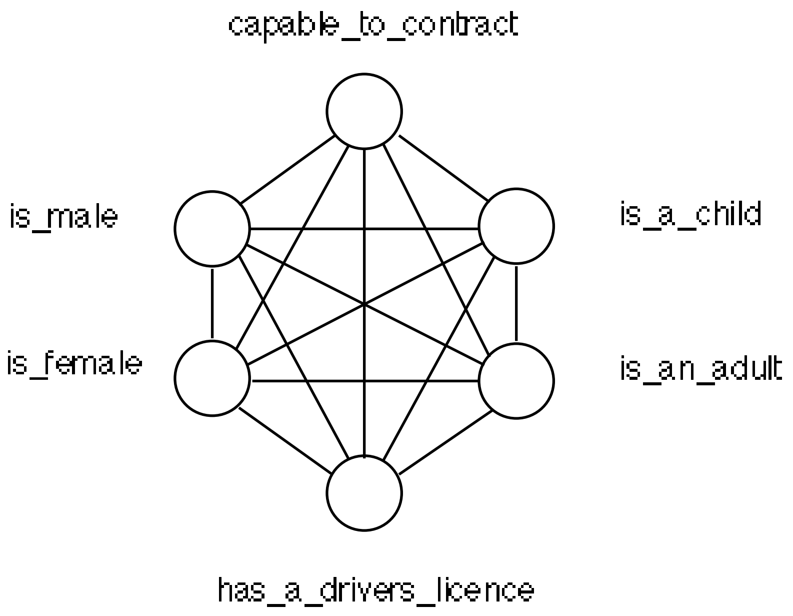

Figure 1: Neural schema of type person

It is obvious that the qualities presented in the neural schema in real life relate to each other. For instance the qualities "is_an_adult" and "capable_to_contract" often go together because most adults are capable to contract whereas "has_a_drivers_licence" and "is_a_child" usually do not go together. These kinds of relations are represented in the weights of the connections between the nodes: a number between -1 and 1; positive weights denote that the connected qualities tend to occur on or off together; a negative weight indicates that qualities tend to exclude each other. The absolute value of the weight represents the strength of the relation.

Each node has a bias attached to it which indicates whether the quality is usually available or not: a positive bias means that the quality is more often available; negative biases indicate the reverse. The influence of the biases will be explained when discussing the instantiation of neural schemata.

It is interesting to note how scientists from various disciplines might perceive this knowledge representation. Of course from the connectionist point of view, the representation consists of a set of nodes and a set of excitatory and inhibitory weights; for instance, the connection between "is_male" and "is_female" is strongly inhibitory because these two characteristics cannot be both applicable for one person. A statistician might argue that this is actually a statistical model in which the connections represent correlations between characteristics. A lawyer might perceive the network as a model of argumentation in which the nodes represent hypothesis; excitatory weights represent arguments in favour of the hypothesis, inhibitory weights represent counterarguments and the absolute weights of the connections represent the strength of the argument. From the A.I. point of view, the model is a representation of the problem of multiple constraint satisfaction. To the psychologist, the network represents a frame-like structure in which nodes or groups of nodes represent slots.

Knowledge for neural schemata (i.e. weights and biases) can be captured via a set of examples. For the "person-schema", for instance, such a set could consists of list of ticked-off characteristics of several real or imaginative persons. The statistical correlations computed on the basis of such a set may provide the weights; the probability that a characteristics is ticked off minus the probability that it is not, may provide the bias for each node.

### 3.2.2. Running a neural schema

The nodes, weights and biases represent general and abstract knowledge. In the chosen example the schema represents characteristics persons might have and common-sense knowledge about dependencies and relations between these characteristics. Running a neural schema makes the schema specific for a problem solving situation. At run time, each node has an activation in the range from zero to one. A fully activated node represents the fact that the particular person in the problem solving situation possesses the characteristic the node represents. As is the case in most neural network mechanisms, the optimal global state of the neural schema is reached via local optimization. A run consists of a sequence of asynchronous updates of randomly chosen nodes. An update of a node is a function of its previous activation and the net input on the node as expressed in the following formula:

$$(1) \qquad a_j(t+1) = a_j(t) + \begin{cases} net_j(1 - a_j(t)) \text{ if } net_j \\ net_j a_j(t) \text{ otherwise} \end{cases}$$

The activation at time t + 1, of node j, is a function of the activation at time t and the net input on the node. The net input is the sum of all excitatory and inhibitory influences. If the net input is excitatory, i.e. greater than zero, the node's activation is pushed in the direction of its maximum in proportion to its previous distance to the maximum; if the net input is inhibitory, the activation is pushed proportionally in the direction of zero. The net input formula

$$(2) \qquad net_j = \sum_i w_{ij} a_i + bias_j + input_j$$

represents the three factors that influence an individual neuron. The first term represents influences of the other nodes, i.e., the knowledge how characteristics relate: some of the other characteristics try to excite the characteristic being updated while others inhibit the node. The first term is simply the sum of these influences. The second term represents the biases of all nodes. For each node, the bias tends to push the activation up or down with the result that if no other information is available, the node is pushed to its default value. The two terms together represent the general abstract knowledge in the network. The third term incorporates problem specific knowledge in the neural schema. Each singular value represents evidence from outside the network that the node should be on. For instance, if it is known that the person the neural schema is run for in the problem solving situation is an adult, the node representing this characteristic should be activated.

Neighbours, biases and input from outside the network put different demands on the node's activation. For each node, it can be calculated how well its value fits in with these demands. Formula 1 ensures that for each node, this so-called "goodness-of-fit" is incremented with each update. For instance, if a neighbour is activated and the connection with this neighbour is of a excitatory nature, the goodness-of-fit of the node (with regard to this neighbour) is maximized if the node is pushed to its maximum; if the connection is inhibitory the goodness is maximized if the node is not activated.

[Hopfield, 1982] has proven that networks with symmetric weights and asynchronous updates always move to a state with a higher global goodness-of-fit (the sum of the individual goodnesses). The reason that the network needs to operate in small steps is that global optimization does not necessarily imply that each node has reached its individual optimum: because of the interdependencies between nodes, the local optimum of one node may obstruct other nodes in reaching their optimal activation.

In terms of knowledge representation, the neural schema, upon reaching the maximum goodness-of-fit, has found an interpretation which best fits the general knowledge the schema possesses given the current fact situation . For instance, the schema might decide that "capable_to_contract" should be activated given the knowledge: "is_an_adult" and "has_a_drivers_licence". It is important to note that the structured interpretation is not some sort of an average of the input (cf. back propagation); it is a distinct interpretation: the network has chosen one of its stereotypes as the most appropriate for the current fact situation.

### 3.3. Benefits of using neural schemata

With neural schemata, one does not have to decide whether a slot or a node has a fixed value or is variable. This is a real advantage because fixed slots should be regarded definitional and defining concepts such as represented with schemata is often problematic. Moreover, characteristics can also be included in the schema if their relation to the conceptual entity which the schema represents is very unclear or weak. However, the most prominent advantage is, that the concept of a default changes in this representation. Normally, a default value is a fixed value which occurs if no information is available on the subject the default stands for. In neural schemata, the value of unknown slots is determined on the basis of all the information that is available. Related to this issue, the instantiated version of neural schemata may occur in quite different configurations although still a member of the same conceptual entity.

## 4. Integration of neural schemata with rules

Although it is imaginable that a future generation of legal expert systems uses neural networks only, contemporary systems rely heavily on the use of inference rules. In this section a method is presented to integrate neural schemata in a rule based system in which the representation of statements is based on first order predicate logic. Automated judicial reasoning is here regarded as an activity in which a system tries to develop a chain of inferences connecting an initial description of a case to a goal. The case description consist of a set of instantiation predicates, i.e. facts; inferences and answers from the user are added to this set of facts. Applying a neural schema means that the schema tries to find the schema that fits best to what is already known to the problem solver; in other words, it tries to find a structured interpretation of the set of facts.

Cooperation of rules and neural schemata requires integration at two different levels which I will refer to as the domain level and the control level. The method of integration at the domain level enables rules to use the knowledge produced by neural schemata and vice versa. The method of integration at the control level regulates when neural schemata are invoked and when the rule-based system is invoked.

### 4.1. Integration at the domain level

A schema is a set of characteristics capable of describing one object, for instance a room or a person. One such a characteristic can also be regarded as a predicate, more precisely, since it is used to describe one object, a unary predicate. The integration of knowledge at the domain level is basically achieved by interpreting a schema as a set of unary predicates in which the variables can only be instantiated to the same object.

Some word about the meaning of the term "instantiated" is appropriate here because the term is used for schemata and predicates in a different way. In predicate logic, "instantiated" means that the variables belonging to a predicate are not free, e.g. refer to constants. With schemata, "instantiated" means that the structure is made specific for a particular object; for neural schemata, this implies the network has been run.

Just as with rules, neural schemata may run forward or backward. If the neural schema is run backward, it starts out with a query and its application to the set of facts is carried out to establish whether the query is true; in other words, the schema may succeed or fail. For instance, the schema used as an example in the previous section might be triggered to run backward with the query capable_to_contract(john) which might be paraphrased as: Is John capable to contract? In a forward run, a neural schema starts out with an object; the neural schema acquires knowledge about this object based on its internal knowledge and the knowledge of the set of facts and adds this knowledge to the set of facts. In analogy with the previous example, the assignment to the schema might be paraphrased as: "Tel me all you know about John".

A backward run of a neural schema can be described in a seven-step procedure:

1. Instantiate the neural schema
2. Check if query is satisfiable immediately
3. Activate appropriate nodes
4. Fail if not enough knowledge is available
5. Run
6. Verify consistency
7. Succeed if the node representing the query is activated; otherwise fail

As stated above, the characteristics represented in a neural schema should be regarded a set of unary predicates. Of course a neural schema is only run backwards if the predicate of the query is also a characteristic of the schema. The first step in the procedure consists of binding the variables of these predicates/characteristics to the same object as the variable in the query. The procedure fails if the variable is free, because a neural schema tries to fit general knowledge to a specific situation and not to general statements. Binding all characteristics of the schema to an object makes the schema specific for this object. At the second step, the neural schema hooks in with the set of facts. If a fact can be found equal to the query, the schema succeeds (without running) and other steps are omitted. If this is not the case, the system proceeds with the third step which consists of checking for each instantiated characteristic whether it is available in the set of facts. If this is the case, the node representing the characteristic is activated and clamped, i.e. its value will not be changeable during running the schema. This means that the system gives absolute dominance to the outside component of formula 2. Step four checks whether any relevant knowledge was found in the set of facts. If nothing is known about the object in the query, the procedure fails and other steps are omitted. Step five consists of running the network as described in section 3.2.2. This is of course the moment the system creates its interpretation of the current situation. The sixth step checks whether the model the neural schema is forced in, is consistent enough with the general knowledge the schema has; if not the procedure fails. To this end, it is calculated how well the final configuration fits in with its internal knowledge only (neighbours and biases). The last step simply checks the node representing the same predicate as the query. If the node is activated, i.e. above a threshold, the procedure succeeds; otherwise it fails.

Applying a neural schema forwardly is quite similar to the procedure described above. There are three differences. Firstly, the procedure starts with an object only, instead of a query. Secondly, the second step is omitted. Thirdly, in the seventh step, the neural schema adds all the facts it has established and which were not available before running, to the set of facts.

## 4.2. Integration at the control level

This section discusses the issue when to apply neural schemata. In order to follow the discussion a distinction must be made between three types of knowledge units: rules, neural schemata and questions. A question is a knowledge unit which asks the user to either confirm or deny a instantiated predicate, or to instantiate a uninstantiated predicate.

The primary goal of the control schema presented here is to produce efficient problem solving. Questions are considered the most expensive steps; they are avoided as much as possible. The system tries to reach a situation as soon as possible in which a likely path is found towards the goal and invoking questions functions more as a confirmation of the path than as new information gathering.

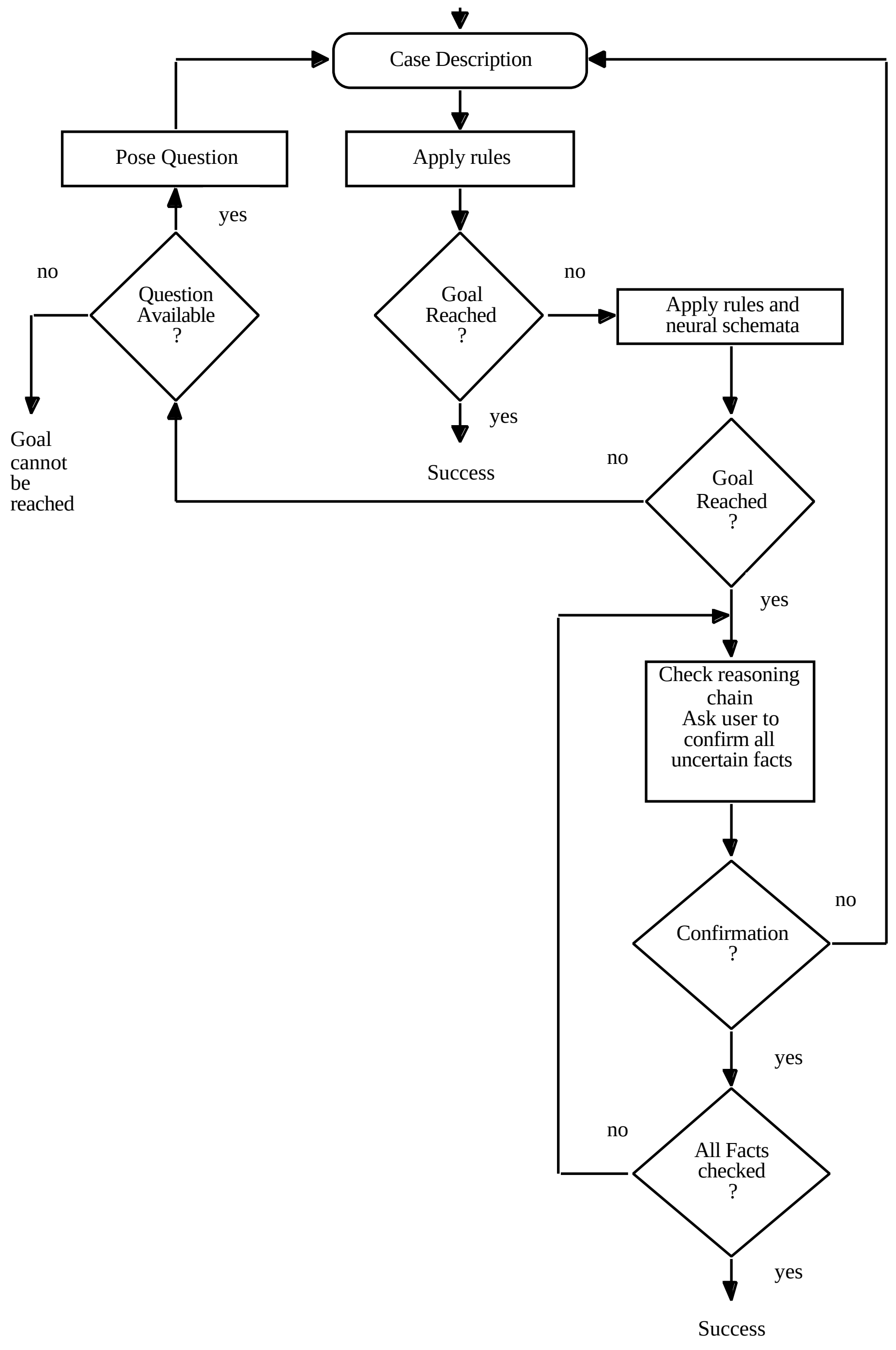

Figure 2: Control strategy to integrate neural schemata in a rule based system

It might be clear from the above that efficient control is considered more an issue of how the system behaves as seen from the outside, then how it traverses the problem space internally. The stress is put on the outside component because if the fact that a systems needs information from outside is ignored, control is not really the problem: the problem space can be search exhaustively because the factual information a legal knowledge based system works with is usually very small (e.g. compared to the bulk of data in a system which tries to interpret radar screens over time).

Figure 2 presents the control strategy to integrate neural schemata in a rule based system. The system starts out with a goal and a set of facts describing the case to solve (this set may be empty). The system first tries to reach the goal via application of the rules. If this fails, more information is needed, and neural schemata and rules are applied. If the goal is not reached the system has no choice then to ask the user for new information. The system will start again with a case description to which the information of the user is added. Because new facts might invalidate previous conclusions of neural schemata and rules, the system abandons all facts which are not provided by the user. Although much more elaborate schemes are thinkable, this simple solution is chosen because, firstly, without asking questions, applying all knowledge again is not a real problem, and, secondly, starting again is more natural because neural schemata may come up with a completely different interpretation of the current situation if new information is added.

If after the application of neural schemata the goal is reached, a situation arises in which a chain of rule based inferences does indeed lead from the case description to the goal, but its validity is questionable because the application of these rules might be based on facts which, directly or indirectly, were produced by neural schemata. Although these facts are uncertain, they are plausible because they are based on a structured interpretations of what is already known about the case. The next step of the system is to verify the reasoning chain by asking the user to acknowledge all uncertain facts the chain of inferences is based on, one by one. If one of these facts is denied, the systems interpretation has been wrong and the system starts again with user provided knowledge only (including the answers obtained during the failing verification stage).

Finding a plausible path as soon as possible is of course the situation the architecture is aiming at because this creates a situation in which all questions relevant to the path are likely to be confirmed by the user and thus continue to keep their relevance to the solution. The systems behaviour should be comparable with the kind of problem solving strategy [Crombag et al., 1977] that is, lawyers seem to jump to a solution and then work on the details. The questions are posed within the context of this solution and all questions related to a particular object would seem to stem from a consistent view the problem solver has about the object.

This section closes with some remarks about the transparency of the framework. It is often claimed that one of the disadvantages of the use of neural networks is that they cannot provide an explanation of the result of their problem solving [Kowalczyk, 1992][Berg, 1992]). In general this is a issue of debate. However, in the approach presented here, (legal) conclusions are always based on rules and antecedent of rules may function as an explanation or justification. The knowledge of the neural schemata, although on the domain level, functions as a catalyst: the final chain of reasoning contains knowledge verified by the user and rule based knowledge only.

## 5. Conclusions

One of the aims of the research presented here is to develop mechanisms which enables an automated judicial problem solver to base its behaviour on knowledge-rich descriptions of cases to be solved. To this end, neural networks and rules cooperate: neural networks provide a context in the form of structured but uncertain interpretations of the cases to be solved and rules are used to create a chains of defensible inferences. The mean reason to use neural schemata is their flexibility. Adding a neural schema to the knowledge base, of course, increases the amount of abstract passive knowledge of the system, but, more importantly, a neural schema adds knowledge which is very often applicable to the problems the system tries to solve. In principle, a neural schema is applicable as soon as the type of object it describes also plays a role in the problem to solve.

The research presented discloses numerous opportunities for further research such as creating a learning procedure to allow neural schemata to learn from the performance of the system. However, two research efforts should be mentioned here in particular, because these efforts aim at a model which is more in congruence with the nature of judicial problem solving then conventional knowledge based systems. Firstly, the extension of current neural schemata. The neural schemata presented here, relate to single objects (i.e. the nodes represent unary predicates). In fact, in the current implementation objects are typed and it is the type of the object which determines to which neural schema the object belongs. However, in judicial problem solving schema-like knowledge is often also available with regard to relations between two or more objects. Creating neural schemata for these relations opens up quite some new possibilities (and problems) because an interpretation of a relation also puts constraints on the interpretations of the neural schemata of the objects the relation connects. In fact, in doing so, all the facts of a case to be solved could be included in a more global interpretation of the facts instead of a interpretation towards single conceptual entities. Secondly, research efforts will aim at the inclusion of case law in the system. In principle, a case with all its details, can be represented as a large set of different predicates connecting all kinds of objects; that is, the same sort of model that emerges from the neural schemata as described above. The knowledge of the schemata could be biassed towards important casus, with the result that the system tends to interpret fact situations as these important cases and drives the systems behaviour towards verification that indeed the interpreted case is applicable.

Both research efforts aim at a model in which rule based and case based reasoning is a result of a global interpretation of the known facts of the case to be solved. A solution to a legal problem is often not simply the application of rules to the facts or the matching of the current fact situation with known cases but much more a judicial construction which fits a particular interpretation of the facts.

## 6. References

This article may only be cited as: Groendijk C., Neural Schemata in Automated Judicial Problem Solving. In: Grütters, C.A.F.M., J.A.P.J. Breuker, H.J. van den Herik, A.H.J. Schmidt and C.N.J. de Vey Mestdagh (eds), *Legal Knowledge Based Systems: Information Technology & Law, JURIX'92*, Koninklijke Vermande, Lelystad, NL, 1992.

[Berg, 1992] Berg, P.H. van den, Uitleg in juridische kennissystemen. In: *Toogdagbundel*, NVvIR, 1992.

[Crombag et al., 1977] Crombag, H.F.M., J.L. de Wijkerslooth and J. Cohen, *Een theorie over rechterlijke beslissingen*, Tjeenk Willink, Groningen, 1977.

[Fernhout, 1989] Fernhout, F., Using a parallel distributed processing model as part of a legal expert system. In: Martino, A.A. (ed), *Pre-proceedings of the Third International Conference on Logica, Informatica, Diritto*, Florence, 1989.

[Gardner, 1987] Gardner A., *An Artificial Intelligence Approach to Legal Reasoning*, Bradford Books/The MIT Press, Cambridge, Ma., 1987.

[Hopfield, 1982] Hopfield, J.J., Neural networks and physical systems with emergent collective computational abilities. In: *Proceedings of the National Academy of Science*, 1982, no. 79, USA, pp. 2554-2558.

[Kowalczyk, 1992] Kowalczyk, W., Neural networks in Knowledge-based systems. In: *Proceedings of the second symposium on neural networks*, Nijmegen, NL, 1992.

[Minsky, 1975] Minsky, M., A framework for representing knowledge. In: Winston, P.H. (ed.), *The psychology of computer vision*, McGraw-Hill, New York, 1975, pp. 211-277.

[Opdorp & Walker, 1990] Opdorp, G.J van, R.F. Walker, A neural network approach to open texture. In: Kaspersen, H.W.K and A. Oskamp (eds), *Amongst friends in Computers and Law Kluwer*, Deventer, 1990.

[Opdorp et al., 1991] Opdorp, G.J. van, R.F. Walker, J.A. Schrickx, C. Groendijk and P.H. van den Berg, Networks at work, a connectionist approach to non-deductive legal reasoning. In: *Proceedings of the Third International Conference on AI and Law, ICAIL'91*, Oxford U.K., 1991.

[Philipps, 1989] Philipps L., Are Legal Decisions based on the Application of Rules or Prototypes Recognition? Legal Science on the way to Neural Networks. In: *Pre-Proceedings of Third International Conference on Logica, Informatica, Diritto*, Martino, A.A. (ed.), Florence, 1989.

[Rose & Below, 1989] Rose, D.E. and R.K. Below, Legal Information Retrieval: A Hybrid Approach. In: *Proceedings of the Second International Conference on AI and Law, ICAIL'89*, Vancouver, Canada, 1989.

[Rumelhart, 1980] Rumelhart, D.E., Schemata: The building blocks of cognition. In: Spiro, R., B. Bruce and W. Brewer (eds), *Theoretical issues in reading comprehension*, Erlbaum, Hillsdale, NJ, 1980, pp. 33-58.

[Rumelhart & McClelland, 1988] Rumelhart, D.E., J.L. McClelland and the PDP Research Group, *Parallel Distributed Processing*, (8th pr.), MIT Press, Cambridge, Massachusetts, 1988.

[Schank & Abelson, 1977] Schank, R.C. and R.P. Abelson, *Scripts, Plans, Goals, and Understanding*, Erlbaum, Hillsdale, NJ, 1977.