Improving the Accuracy of Neuro-Symbolic Rules with Case-Based Reasoning

Jim Prentzas[1], Ioannis Hatzilygeroudis[2] and Othon Michail[2]

Abstract. In this paper, we present an improved approach integrating rules, neural networks and cases, compared to a previous one. The main approach integrates neurules and cases. Neurules are a kind of integrated rules that combine a symbolic (production rules) and a connectionist (adaline unit) representation. Each neurule is represented as an adaline unit. The main characteristics of neurules are that they improve the performance of symbolic rules and, in contrast to other hybrid neuro-symbolic approaches, retain the modularity of production rules and their naturalness in a large degree. In the improved approach, various types of indices are assigned to cases according to different roles they play in neurule-based reasoning, instead of one. Thus, an enhanced knowledge representation scheme is derived resulting in accuracy improvement. Experimental results demonstrate its effectiveness.

1 INTRODUCTION

In contrast to rule-based systems that solve problems from scratch, case-based systems use pre-stored situations (i.e., cases) to deal with similar new situations. Case-based reasoning offers some advantages compared to symbolic rules and other knowledge representation formalisms. Cases represent specific knowledge of the domain, are natural and usually easy to obtain [11], [12]. Incremental learning comes natural to case-based reasoning. New cases can be inserted into a knowledge base without making changes to the preexisting knowledge. The more cases are available, the better the domain knowledge is represented. Therefore, the accuracy of a case-based system can be enhanced throughout its operation, as new cases become available. A negative aspect of cases compared to symbolic rules is that they do not provide concise representations of the incorporated knowledge. Also it is not possible to represent heuristic knowledge. Furthermore, the time-performance of the retrieval operations is not always the desirable.

Approaches integrating rule-based and case-based reasoning have given interesting and effective knowledge representation schemes and are becoming more and more popular in various fields [3], [13], [14], [15], [17], [18], [19]. The objective of these efforts is to derive hybrid representations that augment the positive aspects of the integrated formalisms and simultaneously minimize their negative aspects. The complementary advantages and disadvantages of rule-based and case-based reasoning are a good justification for their possible combination. The bulk of the approaches combining rule-based and case-based reasoning follow the coupling models [17]. In these models, the problem-solving (or reasoning) process is decomposed into tasks (or stages) for which different representation formalisms (i.e., rules or cases) are applied.

However, a more interesting approach is one integrating more than two reasoning methods towards the same objective. In [16] and [10], such an approach integrating three reasoning schemes, namely rules, neurocomputing and case-based reasoning in an effective way is introduced. To this end, neurules and cases are combined. Neurules are a type of hybrid rules integrating symbolic rules with neurocomputing in a seamless way. Their main characteristic is that they retain the modularity of production rules and also their naturalness in a large degree. In that approach, on the one hand, cases are used as exceptions to neurules, filling their gaps in representing domain knowledge and, on the other hand, neurules perform indexing of the cases facilitating their retrieval. Finally, it results in accuracy improvement.

In this paper, we enhance the above approach by employing different types of indices for the cases according to different roles they play in neurule-based reasoning. In this way, an improved knowledge representation scheme is derived as various types of neurules' gaps in representing domain knowledge are filled in by indexed cases. Experimental results demonstrate the effectiveness of the presented approach compared to our previous one.

The rest of the paper is organized as follows. Section 2 presents neurules, whereas Section 3 presents methods for constructing the indexing scheme of the case library. Section 4 describes the hybrid inference mechanism. Section 5 presents experimental results regarding accuracy of the inference process. Section 6 discusses related work. Finally, Section 7 concludes.

2 NEURULES

Neurules are a type of hybrid rules integrating symbolic rules with neurocomputing giving pre-eminence to the symbolic component. Neurocomputing is used within the symbolic framework to improve the performance of symbolic rules [7], [10]. In contrast to other hybrid approaches (e.g. [4], [5]), the constructed knowledge base retains the modularity of production rules, since it consists of autonomous units (neurules), and also retains their naturalness in a large degree,

[1] Technological Educational Institute of Lamia, Department of Informatics and Computer Technology, 35100 Lamia, Greece, email: dprentzas@teilam.gr.
[2] University of Patras, Dept of Computer Engineering & Informatics, 26500 Patras, Greece, email: {ihatz, michailo}@ceid.upatras.gr.

since neurules look much like symbolic rules [7], [8]. Also, the inference mechanism is a tightly integrated process, which results in more efficient inferences than those of symbolic rules [7], [10]. Explanations in the form of if-then rules can be produced [9], [10].

2.1 Syntax and Semantics

The form of a neurule is depicted in Fig.1a. Each condition C_i is assigned a number sf_i, called its *significance factor*. Moreover, each rule itself is assigned a number sf_0, called its *bias factor*. Internally, each neurule is considered as an adaline unit (Fig.1b). The *inputs* C_i (i=1,...,n) of the unit are the *conditions* of the rule. The weights of the unit are the significance factors of the neurule and its bias is the bias factor of the neurule. Each input takes a value from the following set of discrete values: [1 (true), 0 (false), 0.5 (unknown)]. This gives the opportunity to easily distinguish between the falsity and the absence of a condition in contrast to symbolic rules. The *output D*, which represents the *conclusion* (decision) of the rule, is calculated via the standard formulas:

$$D = f(\mathbf{a}), \quad \mathbf{a} = sf_0 + \sum_{i=1}^{n} sf_i \ \ C_i$$

$$f(\mathbf{a}) = \begin{cases} 1 & \text{if } \mathbf{a} \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

where **a** is the *activation value* and *f(x)* the *activation function*, a threshold function. Hence, the output can take one of two values ('-1', '1') representing failure and success of the rule respectively.

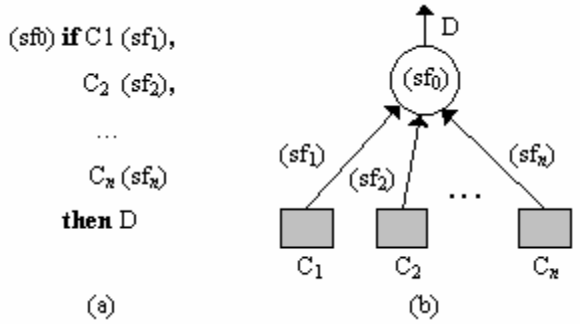

Fig. 1. (a) Form of a neurule (b) a neurule as an adaline unit

The general syntax of a condition C_i and the conclusion D is:

<condition>::= <variable> <l-predicate> <value>
<conclusion>::= <variable> <r-predicate> <value>

where <variable> denotes a *variable*, that is a symbol representing a concept in the domain, e.g. 'sex', 'pain' etc, in a medical domain. <l-predicate> denotes a symbolic or a numeric predicate. The symbolic predicates are {is, isnot} whereas the numeric predicates are {<, >, =}. <r-predicate> can only be a symbolic predicate. <value> denotes a value. It can be a *symbol* or a *number*. The significance factor of a condition represents the significance (weight) of the condition in drawing the conclusion(s). Table 1 (Section 3) presents two example neurules, from a medical diagnosis domain.

Neurules can be constructed either from symbolic rules, thus exploiting existing symbolic rule bases, or from empirical data (i.e., training examples) (see [7] and [8] respectively). An adaline unit is initially assigned to each possible conclusion. Each unit is individually trained via the Least Mean Square (LMS) algorithm. When the training set is inseparable, special techniques are used. In that case, more than one neurule having the same conclusion are produced.

Table 1. Example neurules

NR1: (-23.9) **if** patient-class is human0-20 (10.6), pain is continuous (10.5), fever is high (8.8), fever is medium (8.4), patient-class is human21-35 (6.2), fever is no-fever (2.7), ant-reaction is medium (1.1) **then** disease-type is inflammation	NR2: (-13.4) **if** patient-class is human21-35 (6.9), pain is continuous (3.2), joints-pain is yes (3.1), fever is low (1.5), fever is no-fever (1.5) **then** disease-type is chronic-inflammation

2.2 The Neurule-Based Inference Engine

The neurule-based inference engine performs a task of classification: based on the values of the condition variables and the weighted sums of the conditions, conclusions are reached. It gives pre-eminence to symbolic reasoning, based on a backward chaining strategy [7], [10]. As soon as the initial input data is given and put in the working memory, the output neurules are considered for evaluation. One of them is selected for evaluation. Selection is based on textual order. A neurule fires if the output of the corresponding adaline unit is computed to be '1' after evaluation of its conditions. A neurule is said to be 'blocked' if the output of the corresponding adaline unit is computed to be '-1' after evaluation of its conditions.

A condition evaluates to 'true' ('1'), if it matches a fact in the working memory, that is there is a fact with the same variable, predicate and value. A condition evaluates to 'unknown', if there is a fact with the same variable, predicate and 'unknown' as its value. A condition cannot be evaluated if there is no fact in the working memory with the same variable. In this case, either a question is made to the user to provide data for the variable, in case of an input variable, or an intermediate neurule with a conclusion containing the variable is examined, in case of an intermediate variable. A condition with an input variable evaluates to 'false' ('0'), if there is a fact in the working memory with the same variable, predicate and different value. A condition with an intermediate variable evaluates to 'false' if additionally to the latter there is no unevaluated intermediate neurule that has a conclusion with the same variable. Inference stops either when one or more output neurules are fired (success) or there is no further action (failure).

During inference, a conclusion is rejected (or not drawn) when none of the neurules containing it fires. This happens when: (i) all neurules containing the conclusion have been examined and are blocked or/and (ii) a neurule containing an

alternative conclusion for the specific variable fires instead. For instance, if all neurules containing the conclusion 'disease-type is inflammation' have been examined and are blocked, then this conclusion is rejected (or not drawn). If a neurule containing e.g. the alternative conclusion 'disease-type is primary-malignant' fires, then conclusion 'disease-type is inflammation' is rejected (or not drawn), no matter whether all neurules containing as conclusion 'disease-type is inflammation' have been examined (and are blocked) or not.

3 INDEXING

Indexing concerns the organization of the available cases so that combined neurule-based and case-based reasoning can be performed. Indexed cases fill in gaps in the domain knowledge representation by neurules and during inference may assist in reaching the right conclusion. To be more specific, cases may enhance neurule-based reasoning to avoid reasoning errors by handling the following situations:

(a) Examining whether a neurule misfires. If sufficient conditions of the neurule are satisfied so that it can fire, it should be examined whether the neurule misfires for the specific facts, thus producing an incorrect conclusion.

(b) Examining whether a specific conclusion was erroneously rejected (or not drawn).

In the approach in [10], the neurules contained in the neurule base were used to index cases representing their exceptions. A case constitutes an exception to a neurule if its attribute values satisfy sufficient conditions of the neurule (so that it can fire) but the neurule's conclusion contradicts the corresponding attribute value of the case. In this approach, various types of indices are assigned to cases. More specifically, indices are assigned to cases according to different roles they play in neurule-based reasoning and assist in filling in different types of gaps in the knowledge representation by neurules. Assigning different types of indices to cases can produce an effective approach combining symbolic rule-based with case-based reasoning [1].

In this new approach, a case may be indexed by neurules and by neurule base conclusions as well. In particular, a case may be indexed as:

(a) *False positive (FP)*, by a neurule whose conclusion is contradicting. Such cases, as in our previous approach, represent exceptions to neurules and may assist in avoiding neurule misfirings.

(b) *True positive (TP)*, by a neurule whose conclusion is endorsing. The attribute values of such a case satisfy sufficient conditions of the neurule (so that it can fire) and the neurule's conclusion agrees with the corresponding attribute value of the case. Such cases may assist in endorsing correct neurule firings.

(c) *False negative (FN)*, by a conclusion erroneously rejected (or not drawn) by neurules. Such cases may assist in reaching conclusions that ought to have been drawn by neurules (and were not drawn). If neurules with alternative conclusions containing this variable were fired instead, it may also assist in avoiding neurule misfirings. 'False negative' indices are associated with conclusions and not with specific neurules because there may be more than one neurule with the same conclusion in the neurule base.

The indexing process may take as input the following types of knowledge:

(a) Available neurules and non-indexed cases.

(b) Available symbolic rules and indexed cases. This type of knowledge concerns an available formalism of symbolic rules and indexed exception cases as the one presented in [6].

The availability of data determines which type of knowledge is provided as input to the indexing module. If an available formalism of symbolic rules and indexed cases is presented as input, the symbolic rules are converted to neurules using the 'rules to neurules' module. The produced neurules are associated with the exception cases of the corresponding symbolic rules [10]. Exception cases are indexed as 'false positives' by neurules. Furthermore, for each case 'true positive' and 'false negative' indices may be acquired using the same process as in type (a).

When available neurules and non-indexed cases are given as input to the indexing process, cases must be associated with neurules and neurule base conclusions. For each case, this information can be easily acquired as following:

Until all intermediate and output attribute values of the case have been considered:

1. Perform neurule-based reasoning for the neurules based on the attribute values of the case.
2. If a neurule fires, check whether the value of its conclusion variable matches the corresponding attribute value of the case. If it does (doesn't), associate the case as a 'true positive' ('false positive') with this neurule.
3. Check all intermediate and final conclusions. Associate the case as a 'false negative' with each rejected (or not drawn) conclusion that ought to have been drawn based on the attribute values of the case.

To illustrate how the indexing process works, we present the following example. Suppose that we have a neurule base containing the two neurules in Table 1 and the example cases shown in Table 2 (only the most important attributes of the cases are shown). The cases however, also possess other attributes (not shown in Table 2).

'disease-type' is the output attribute that corresponds to the neurules' conclusion variable. Table 3 shows the types of indices associated with each case in Table 2 at the end of the indexing process.

To acquire indexing information, the input values corresponding to the attribute values of the cases are presented to the example neurules. Recall that when a neurule condition evaluates to 'true' it gets the value '1', whereas when it is false gets '0'.

For example, given the input case C2, the final weighted sum of neurule NR1 is: -23.9 + 10.6 + 10.5 + 8.8 = 6>0. Note that the first three conditions of NR1 evaluate to 'true' whereas the remaining four (i.e., 'fever is medium', 'fever is no-fever', 'patient-class is human21-35' and 'ant-reaction is medium') to 'false' (not contributing to the weighted sum).

Table 2. Example cases

Case ID	*patient-class*	*pain*	*fever*	*ant-reaction*	*joints-pain*	*disease-type*
C1	human21-35	continuous	low	none	yes	chronic-inflammation
C2	human0-20	continuous	high	none	no	inflammation
C3	human0-20	night	high	none	no	inflammation
C4	human0-20	continuous	medium	none	no	inflammation
C5	human21-35	continuous	no-fever	medium	yes	chronic-inflammation
C6	human0-20	continuous	low	none	no	chronic-inflammation

The fact that the final weighted sum is positive means that sufficient conditions of NR1 are satisfied so that it can fire. Furthermore, the corresponding output attribute value of the case matches the conclusion of NR1 and therefore C2 is associated as 'true positive' with NR1.

Table 3. Indices assigned to the example cases in Table 2

Case ID	*Type of index*	*Indexed by*
C1	'True positive'	Neurule NR2
C2	'True positive'	Neurule NR1
C3	'False negative'	Conclusion 'disease-type is inflammation'
C4	'True positive'	Neurule NR1
C5	'False positive'	Neurule NR1
C5	'True positive'	Neurule NR2
C6	'False negative'	Conclusion 'disease-type is chronic-inflammation'

Similarly, when the input values corresponding to the attribute values of cases C1 and C4 are given as input to the neurule base, sufficient conditions of neurules NR2 and NR1 respectively are satisfied so that they can fire and the corresponding output attribute case values match their conclusions. Furthermore, when the input values corresponding to the attribute values of case C5 are given as input to the neurule base, sufficient conditions of both neurules NR1 and NR2 are satisfied so that they can fire. However, the corresponding output attribute case values match the conclusion of NR2 and contradict the conclusion of NR1. In addition, conclusion 'disease-type is inflammation' cannot be drawn when the input values corresponding to the attribute values of case C3 are given as input because the only neurule with the corresponding conclusion (i.e., NR1) is blocked. A similar situation happens for case C6.

4 THE HYBRID INFERENCE MECHANISM

The inference mechanism combines neurule-based with case-based reasoning. The combined inference process mainly focuses on the neurules. The indexed cases are considered when: (a) sufficient conditions of a neurule are fulfilled so that it can fire, (b) all output or intermediate neurules with a specific conclusion variable are blocked and thus no final or intermediate conclusion containing this variable is drawn.

In case (a), firing of the neurule is suspended and case-based reasoning is performed for cases indexed as 'false positives' and 'true positives' by the neurule and cases indexed as 'false negatives' by alternative conclusions containing the neurule's conclusion variable. Cases indexed as 'true positives' by the neurule endorse its firing whereas the other two sets of cases considered (i.e., 'false positives' and 'false negatives') prevent its firing. The results produced by case-based reasoning are evaluated in order to assess whether the neurule will fire or whether an alternative conclusion proposed by the retrieved case will be considered valid instead.

In case (b), the case-based module will focus on cases indexed as 'false negatives' by conclusions containing the specific (intermediate or output) variable.

The basic steps of the inference process are the following:

1. Perform neurule-based reasoning for the neurules.
2. If sufficient conditions of a neurule are fulfilled so that it can fire, then
   - 2.1. Perform case-based reasoning for the 'false positive' and 'true positive' cases indexed by the neurule and the 'false negative' cases associated with alternative conclusions containing the neurule's conclusion variable.
   - 2.2. If none case is retrieved or the best matching case is indexed as 'true positive', the neurule fires and its conclusion is inserted into the working memory.
   - 2.3. If the best matching case is indexed as 'false positive' or 'false negative', insert the conclusion supported by the case into the working memory and mark the neurule as 'blocked'.
3. If all intermediate neurules with a specific conclusion variable are blocked, then
   - 3.1. Examine all cases indexed as 'false negatives' by the corresponding intermediate conclusions, retrieve the best matching one and insert the conclusion supported by the retrieved case into the working memory.
4. If all output neurules with a specific conclusion variable are blocked, then
   - 4.1. Examine all cases indexed as 'false negatives' by the corresponding final conclusions, retrieve the best matching one and insert the conclusion supported by the retrieved case into the working memory.

The similarity measure between two cases c_k and c_l is calculated via a distance metric [1]. The best-matching case to the problem at hand is the one having the maximum similarity

with (minimum distance from) the input case. If multiple stored cases have a similarity equal to the maximum one, a simple heuristic is used.

Let present now two simple inference examples concerning the combined neurule base (Table 1) and the indexed example cases (Tables 2 and 3). Suppose that during inference sufficient conditions of neurule NR1 are satisfied so that it can fire. Firing of NR1 is suspended and the case-based reasoning process focuses on the cases contained in the union of the following sets of indexed cases:

- the set of cases indexed as 'true positives' by NR1: {C2, C4},
- the set of cases indexed as 'false positives' by NR1: {C5} and
- the set of cases indexed as 'false negatives' by alternative conclusions containing variable 'disease-type' (i.e., 'disease-type is chronic inflammation'): {C6}.

So, in this example the case-based reasoning process focuses on the following set of indexed cases: {C2, C4} ∪ {C5} ∪ {C6} = {C2, C4, C5, C6}.

Suppose now that during inference both output neurules in the example neurule base are blocked. The case-based reasoning process will focus on the cases contained in the union set of the following sets of indexed cases:

- the set of cases indexed as 'false negatives' by conclusion 'disease-type is inflammation': {C3}.
- the set of cases indexed as 'false negatives' by conclusion 'disease-type is chronic-inflammation': {C6}.

Therefore, in this example the case-based reasoning process focuses on the following set of indexed cases: {C3} ∪ {C6} = {C3, C6}.

5 EXPERIMENTAL RESULTS

In this section, we present experimental results using datasets acquired from [2]. Note that there are no intermediate conclusions in these datasets. The experimental results involve evaluation of the presented approach combining neurule-based and case-based reasoning and comparison with our previous approach [10]. 75% and 25% of each dataset were used as training and testing sets respectively. Each initial training set was used to create a combined neurule base and indexed case library. For this purpose, each initial training set was randomly split into two disjoint subsets, one used to create neurules and one used to create an indexed case library. More specifically, 2/3 of each initial training set was used to create neurules by employing the 'patterns to neurules' module [8] whereas the remaining 1/3 of each initial training set constituted non-indexed cases. Both types of knowledge (i.e., neurules and non-indexed cases) were given as input to the indexing construction module presented in this paper producing a combined neurule base and an indexed case library which will be referred to as NBRCBR. Neurules and non-indexed cases were also used to produce a combined neurule base and an indexed case library according to [10] which will be referred to as NBRCBR_PREV.

Inferences were run for both NBRCBR and NBRCBR_PREV using the testing sets. Inferences from NBRCBR_Prev were performed using the inference mechanism combining neurule-based and CBR as described in [10]. Inferences from NBRCBR were performed according to the inference mechanism described in this paper. No test case was stored in the case libraries.

Table 4 presents such experimental results regarding inferences from NBRCBR and NBRCBR_PREV. It presents results regarding classification accuracy of the integrated approaches and the percentage of test cases resulting in neurule-based reasoning errors that were successfully handled by case-based reasoning. Column '% FPs handled' refers to the percentage of test cases resulting in neurule misfirings (i.e., 'false positives') that were successfully handled by case-based reasoning. Column '% FNs handled' refers to the percentage of test cases resulting in having all output neurules blocked (i.e., 'false negatives') that were successfully handled by case-based reasoning. 'False negative' test cases are handled in NBRCBR_PREV by retrieving the best-matching case from the whole library of indexed cases.

Table 4. Experimental results

Dataset	NBRCBR			NBRCBR_PREV		
	Classification Accuracy	**% FPs Handled**	**% FNs Handled**	**Classification Accuracy**	**% FPs Handled**	**% FNs Handled**
Car (1728 patterns)	96.04%	52.81%	64.07%	92.49%	15.51%	20.36%
Nursery (12960 patterns)	98.92%	58.68%	52.94%	97.68%	6.60%	18.82%

As can be seen from the table, the presented approach results in improved classification accuracy. Furthermore, in inferences from NBRCBR the percentages of both 'false positive' and 'false negative' test cases successfully handled are greater than the corresponding percentages in inferences from NBRCBR_PREV. Results also show that there is still room for improvement.

We also tested a nearest neighbor approach working alone in these two datasets (75% of the dataset used as case library and 25% of the dataset used as testing set). We used the similarity measure presented in Section 5. The approach classified the input case to the conclusion supported by the best-matching case retrieved from the case library. Classification accuracy for car and nursery dataset is 90.45% and 96.67% respectively. So, both integrated approaches perform better. This is due to the fact that the indexing schemes assist in focusing on specific parts of the case library.

7 CONCLUSIONS

In this paper, we present an approach integrating neurule-based and case-based reasoning that improves a previous hybrid approach [10]. Neurules are a type of hybrid rules integrating symbolic rules with neurocomputing. In contrast to other neuro-symbolic approaches, neurules retain the naturalness and modularity of symbolic rules. Integration of neurules and cases is done in order to improve the accuracy of the inference mechanism. Cases are indexed according to the roles they can play during neurule-based inference. More specifically, they are associated as 'true positives' and 'false positives' with neurules and as 'false negatives' with neurule base conclusions.

The presented approach integrates three types of knowledge representation schemes: symbolic rules, neural networks and case-based reasoning. Most hybrid intelligent systems implemented in the past usually integrate two intelligent technologies e.g. neural networks and expert systems, neural and fuzzy logic, genetic algorithms and neural networks, etc. A new development that should receive interest in the future is the integration of more than two intelligent technologies, facilitating the solution of complex problems and exploiting multiple types of data sources.

References

[1] G. Agre, 'KBS Maintenance as Learning Two-Tiered Domain Representation', In M.M. Veloso, A. Aamodt, (Eds.): Case-Based Reasoning Research and Development, First International Conference, ICCBR-95, Proceedings, Lecture Notes in Computer Science, Vol. 1010, Springer-Verlag, 108-120, 1995.

[2] A. Asuncion, D.J. Newman, 'UCI Repository of Machine Learning Databases' [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA, University of California, School of Information and Computer Science (2007).

[3] N. Cercone, A. An, C. Chan, 'Rule-Induction and Case-Based Reasoning: Hybrid Architectures Appear Advantageous', *IEEE Transactions on Knowledge and Data Engineering*, **11**, 164-174, (1999).

[4] S. I. Gallant, *Neural Network Learning and Expert Systems*, MIT Press, 1993.

[5] A.Z. Ghalwash, 'A Recency Inference Engine for Connectionist Knowledge Bases', *Applied Intelligence*, **9**, 201-215, (1998).

[6] A.R. Golding, P.S. Rosenbloom, 'Improving accuracy by combining rule-based and case-based reasoning', *Artificial Intelligence*, **87**, 215-254, (1996).

[7] I. Hatzilygeroudis, J. Prentzas, 'Neurules: Improving the Performance of Symbolic Rules', *International Journal on AI Tools*, **9**, 113-130, (2000).

[8] I. Hatzilygeroudis, J. Prentzas, 'Constructing Modular Hybrid Rule Bases for Expert Systems', *International Journal on AI Tools*, **10**, 87-105, (2001).

[9] I. Hatzilygeroudis, J. Prentzas, 'An Efficient Hybrid Rule-Based Inference Engine with Explanation Capability', Proceedings of the 14th International FLAIRS Conference, AAAI Press, 227-231, (2001).

[10] I. Hatzilygeroudis, J. Prentzas, 'Integrating (Rules, Neural Networks) and Cases for Knowledge Representation and Reasoning in Expert Systems', *Expert Systems with Applications*, **27**, 63-75, (2004).

[11] J. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[12] D.B. Leake (ed.), *Case-Based Reasoning: Experiences, Lessons & Future Directions*, AAAI Press/MIT Press, 1996.

[13] M.R. Lee, 'An Exception Handling of Rule-Based Reasoning Using Case-Based Reasoning', *Journal of Intelligent and Robotic Systems*, **35**, 327-338, (2002).

[14] C.R. Marling, M. Sqalli, E. Rissland, H. Munoz-Avila, D. Aha, 'Case-Based Reasoning Integrations', *AI Magazine*, **23**, 69-86, (2002).

[15] S. Montani, R. Bellazzi, 'Supporting Decisions in Medical Applications: the Knowledge Management Perspective', *International Journal of Medical Informatics*, **68**, 79-90, (2002).

[16] J. Prentzas, I. Hatzilygeroudis, 'Integrating Hybrid Rule-Based with Case-Based Reasoning', In S. Craw and A. Preece (Eds), Advances in Case-Based Reasoning, Proceedings of the European Conference on Case-Based Reasoning, ECCBR-2002, Lecture Notes in Artificial Intelligence, Vol. 2416, Springer-Verlag, 336-349, 2002.

[17] J. Prentzas, I. Hatzilygeroudis, 'Categorizing Approaches Combining Rule-Based and Case-Based Reasoning', *Expert Systems*, **24**, 97-122, (2007).

[18] E.L. Rissland, D.B. Skalak, 'CABARET: Rule Interpretation in a Hybrid Architecture', *International Journal of Man-Machine Studies*, **34**, 839-887, (1991).

[19] D. Rossille, J.-F. Laurent, A. Burgun, 'Modeling a Decision Support System for Oncology using Rule-Based and Case-Based Reasoning Methodologies', *International Journal of Medical Informatics*, **74**, 299-306, (2005).

[20] H. Vafaie, C. Cecere, 'CORMS AI: Decision Support System for Monitoring US Maritime Environment', Proceedings of the 17th Innovative Applications of Artificial Intelligence Conference (IAAI), AAAI Press, 1499-1507, (2005).