A Self Evolutionary Rule-base

A. M. Khattak, Z. Pervez, W. A. Khan, S. Y. Lee, Y. K. Lee
Department of Computer Engineering, Kyung Hee University, Korea
{asad.masood, zeeshan, wajahat.alikhan, sylee)@oslab.khu.ac.kr, yklee@khu.ac.kr

ABSTRACT

The rapid growth in domain knowledge is the main reason for the evolution of knowledgebase's maintaining the domain knowledge. Rule-based Decision Support Systems (DSS) are the most effected systems with the growing knowledge. The experts need to continuously update the rule-base for the new knowledge. This manual and periodic updates in rule-base are time consuming and less useful. In this paper we propose a Self Evolutionary Rule-base algorithm for rule-bases of DSS to decrease the burden from experts and also provide updated knowledge on time. To achieve this objective, we develop a generic structure for rules storage that not only provide efficient manipulation of rules but a generic structure for storage of rules regardless of rules nature/format. The detail working of proposed Rule-base system for rules storage and manipulation is provided in this paper. For the proof of concept, we have implemented the Self Evolutionary Rule-base algorithm in Socially Interactive Clinical Decision Support System (SI-CDSS). The focus is on diabetes disease patients and the overall SI-CDSS is deployed in Microsoft Azure environment. In its implementation, Rough Set generated rules are used and the algorithm is executed on Rough Set generated rules.

Keywords: Rules, Rule-base, Decision Support System, Algorithms, Performance, Verification.

INTRODUCTION

Rapid growth of information is the main reason for the increase in complexity of the collected data in a specified field of study. These complexities also introduce various types of uncertainties in the data collected specifically relating to problems in healthcare applications and services [2 and 3]. To extract the useful information from the uncertain data, different researchers from mathematics, computer science, and medical related areas have worked on number of theories [10] that supports in building expert systems. However, these theories do have limitations. These limitations restrict the theories and its complaint systems to limited domains where they can perform better.

Rules are the important paradigm for representing expert knowledge. In rule-based expert systems, the rule-base contains the domain knowledge coded in the form of rules. In healthcare domain, theories that support rule-base models are preferred over those of machine learning based theories [2]. The fact behind it is that medical doctors can interact with systems using rule-based approaches. They can compile new rules while in machine learning based approaches, medical doctors are not aware of the internal working of the systems. In addition, most of the domain specific expert systems are also based on rule-base (knowledgebase) [11]. Rule-based systems store, manipulate, and interpret information in a useful way. Rule-based systems are frequently used for diagnosis, recommendations, and symptoms clustering [8].

The rule-base systems use rules for variety of different purposes such as conflict resolution, decision making, and recommendation. However, the main issue is that these rules are maintained in text files or in XML files and are static in nature; on the other hand, the domain these rules are defined for is not static. So there is a need for a system that can help in rule evolution with the evolution in the field of knowledge. Rough Set based technique is used to generate new rules from the data collected [3]. The new rules need to update the existing set of rules or in other words, making the rule-base to evolve so that the rule-base can accommodate the new or changed rules. Another issue is a common structure for the storage of rules. When a system uses different rule-based approaches (rules in different format) then for rules storage they also need different structures.

In this research, we focus on a generic storage structure that is used for storage of rules (from healthcare domain) of different representational format. On top of the proposed storage structure, we propose a Self Evolutionary Rule-base algorithm for rules with changes reflected from the evolved domain knowledge to accommodate the new and changed knowledge in Rule-base. A detail discussion on storage structure for rules and the implementation of Self Evolutionary Rule-base algorithm is provided in this research. The proposed algorithm for Rule-base is implemented with our underdevelopment Socially Interactive Clinical Decision Support System (SI-CDSS) [2] shown below.

The remaining paper is arranged as follows: Related Work section presents details on Rule-base and Rule-based systems. Section Proposed Scheme provides detail description of proposed storage structure and evolutionary algorithm. Implementation and Results section presents the preliminary implementation details. Finally we conclude our discussion in Conclusions section.

RELATED WORK

Expert systems are mostly rule-based systems. They use human expert knowledge (human intelligence) to solve real-world problems. The expert knowledge is often represented in the form of rules. Rule-based systems in healthcare domain are preferred over machine learning based systems due to the fact that rules are easily understandable to doctors. The rules in healthcare system represent medical doctors and domain knowledge. In addition, medical doctors feel comfortable while interacting with rule-based systems. The problem with these systems is that the rules are static and needs expert intervention for updates [3]. A medical guidelines based (i.e., rule-based) clinical decision support system proposed in [4] mainly focus on metabolism synthesis. The rule-base is constructed from the predefined logic (i.e., medical logic) and then used in inference engine for decision making. It is composed of four different components namely; data, model, inference, human computer interaction. The rule-base is static and does not consider evolution for the updates in the field of knowledge (domain knowledge). So, all the updates are carried out by experts and doctors manually.

In [7], the authors focused on real-time activities performed by patients and then use these activities and domain knowledge for decision making. The system used description logic rules after match making process to make appropriate decisions based on situation analysis. The rules used are stored in text file and are periodically updated by experts based on user and system needs. In [5], the authors proposed inference mechanism with Electronic Health Record (EHR) for existing hospital information system. The rule engine and the rule-base for containing the clinical guideline are integrated together. With static rules, their main drawback is strict modeling of information according to its input types for different components.

A healthcare service using conventional clinical decision support system and ontology to manage user healthcare data has been proposed in [6]. The main functionality of the system was to generate, deploy, and manage patient information. The information is inferred using rule-base and ontology. The results are then propagated to dependent components. In [9], the authors presented a tool for multi-objective job scheduling problems. An interactive multi objective genetic algorithm for decisions has been proposed. It's decision function is defined as a measure of truth of a linguistically quantified statements (rules). The tool also provides support for what-if analyses.

Research work and actual deployment of rule-based decision system is visible in different areas including healthcare such as~~;~~ in development of clinical decision support systems. These systems are very strict with the structure for rule representation and one rule-base cannot support two or more inference engines with different formats of rules. Another issue is that the rule-base of these systems is static and is not updated with the new knowledge. Our proposal is a generic structure for rules storage regardless of the rules nature and keeps the rule-base dynamic with evolving domain knowledge.

PROPOSED SCHEME

Clinical Decision Support System (CDSS) is widely used now days in every country for better, timely, and low cost healthcare services. Most of these systems are based on the concept of using rules for decision making and because of this these systems are also termed as expert systems. This section provides details on proposed scheme i.e., Self Evolutionary Rule-base for SI-CDSS [2]. SI-CDSS continuously generate new rules based on patient's experience and expert interventions. To cope with the growing knowledge, our proposed scheme keeps the rule-base updated. To support the proposed scheme, a generic structure is also build for rules storage regardless of rules structure/format. Based on the generic structure and evolutionary algorithm, this section is divided into two subsections (see Figure 4 for overall system architecture).

Rules-base Storage Structure

Information systems as well as decision support systems are mainly based on reliable data and facts. Facts (conditions) represent our knowledge about the situation/problem. Rules are used to represent relationships among the Facts. Based on these rules, Inference Engine makes the inference for situation analysis and decision making. The rules are mainly used to get end results based on the given facts. For example, Figure 1 shows the consequent effects of combination of facts (A….X) that builds confidence for a rule in a given situation and results in consequent effects (actions).

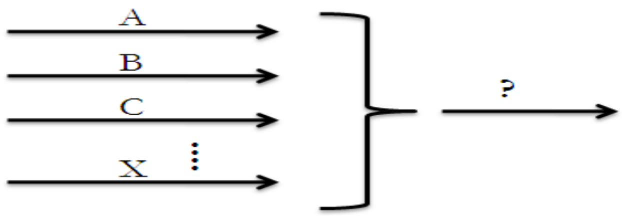

Figure 1. Shows abstract structure of rule. Set of conditions (facts), confidence building, and resultant action.

In most systems these rules are stored in text files that are hard to maintain. In case of systems using two or more inference engines such as; SI-CDSS [2], the systems will maintain two different rule-bases as the rules format is different (see Figure 2) and cannot be stored in a rule-base fix for one particular format. As it is obvious that rules have two main components i.e., condition and action that makes it easy to model it. With variety of different rules representational format, using different logical, mathematical, and relational operators makes it a bit tough to model them in a single storage structure. To handle this, we develop a schema and in that we model rule as an entity separate from its condition and action parts. The rule entity contains the information about its associated conditions and

if BMI<=3, Lethargic=1, Fats=1, then Diabetes type=1 if Exercise>=2, Fats=0, then Additional Medication=0
Patient(p) $\sqcap$ hasSymptoms(cool skin) $\sqcap$ hasSymptoms(dam skin) $\sqcap$ hasBP(160/120) $\rightarrow$generateRecomendation(drink juice) $\sqcap$ generateRecomendation(take rest)
UrineProblem,Weight,WeightLoss,Age,Fatigue,Pain$\rightarrow$Class Yes,73,*,48,No,* $\rightarrow$ 5
Figure 2. Shows the example of procedural (if-then) rules, description logic rules, and rules used by Rough Set.

actions information by distributing its key as a foreign key. On rule entity, it also contains information about the rule format and the type of disease it is focusing on. All the conditions that can be part of a rule are modeled separately along with the operators. This also solves another problem for storing and handling unknown number of conditions of a rule while in conventional systems, the numbers of conditions are fixed. The same way, the actions can be many for a given rule so we also model it in a separate entity that makes a many-to-one relationship with the rules entity. For more detailed view of the storage structure please see Figure 3. The class entity is for extra recommendations from medical doctors while the user entity is for tracking user and user access privileges.

Self Evolutionary Rule-base

Rule-base is one of the main components of SI-CDSS that store and manipulate decision rules based on user's request. Inference Engine needs to take a decision and for decision making rule-base is contacted and corresponding rules are extracted for decision making. These rules are mainly composed based on experts (doctors) knowledge [1], however, in SI-CDSS, patient's experience is also incorporated that further help in recommendations and decision making. As updates in the domain knowledge and patients experience sharing is very frequent, and is the main reason for frequent rules generation by Rough Set based inference engine. At the same time these rules need to be updated in the Rule-base to provide up to date services and recommendations.

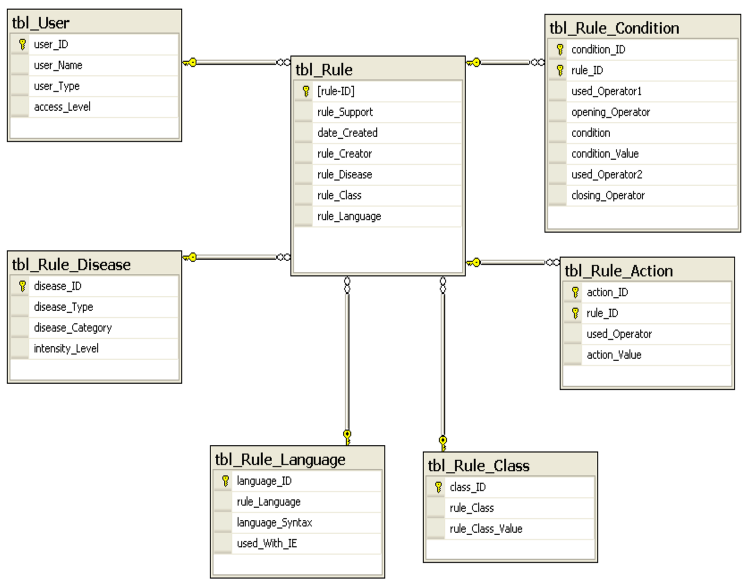

Figure 3, Shows the generic storage structure for rules storage in rule-base.

To achieve better healthcare services, recommendations, and decision makings; more sophisticated and exhaustive list of rules that can serve better are required. With the passage of time, the advancement in expert knowledge and user experience may introduce new rules as well as changes in the existing rules. These all need to be accommodated appropriately in the Rule-base. Change in rules based on patient's experience is very sensitive. For this reason, the generated rules will only be stored if they are verified by the experts and allowed to be stored in the repository. We use the notion of social as the system is more interactive and also use patient's experience for recommendations and decision makings. These rules are used by inference engine for decision making or analysis, so its main interaction point is the inference engine. Updates in rules and Rule-base will all happen through inference engine. The complete process of rules parsing, addition, and updates is given in the system overall architecture shown in Figure 4.

Wrapper component is responsible for selecting the appropriate rules from the rule-base regarding its format and the calling inference engine. Parser handles all the rules retrieval and submission requests. Rule Engineering and Verification is responsible for providing facility to experts for creating new rules and also verification of existing rules in the Rule DB of Rule-base. Rule Updates component is actively listing for changes in existing rules. Once change is found then it update the corresponding rule and propagate it to Rule DB. It is important to note that not all the rules are updated based on simple change in rules but in fact, a rule is updated if there is a change in the consequent (action) of the rule. The update can be a change of existing consequent or addition of another consequent and in our proposed structure we have provision for any number of additional consequent. The detail working of proposed Self-Evolutionary Rule-base system is given in Algorithm 1.

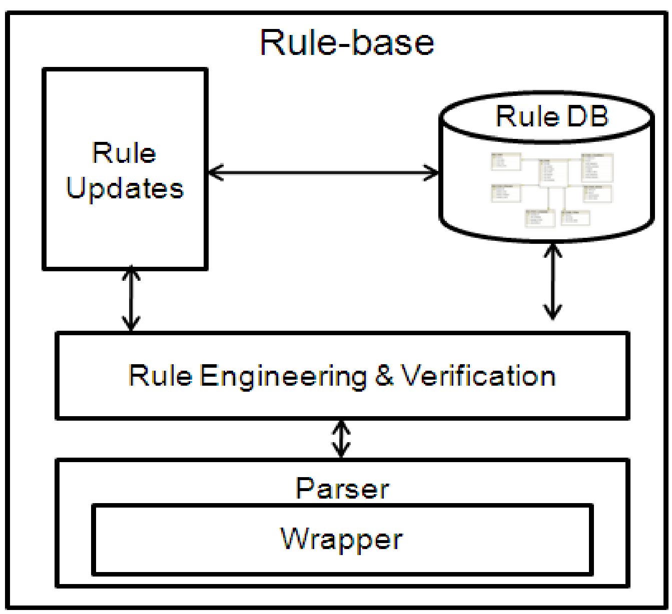

Figure 4,Overall architecture diagram of proposed Self Evolutionary Rule-base.

Algorithm EvolutionaryRule-base ( ): Self Evolutionary algorithm to evolve Rule-base for accommodating the updated knowledge.

Input: Rules form Rule DB and newly generated rules from inference engine.

Input: User entered rules for new knowledge.

Output: Set of new and updated rules storage.

```
/* Check for type of inference engine and then activate appropriate wrapper.*/
    Wrapper.initiate(IE.RulesType)
/* Fetch the rules generated by the inference engine or expert entered rules.*/
    Rules ← IE.Rules
    Rules + ← EXPERT.Rules
/* Fetch rules from RuleDB to be updated.*/
    Rulesdb ← RuleDB.Rules
/* Check if rules from inference engine are new then add in RuleDB.*/
    If NOT(Rules= ∃Rulesdb) then
    Rulesdb ← {x | <RulesΔ, x >New}
    Endif
    /* Check for rules updates, update rule, and store in RuleDB*/
    If (Rules= ∃RulesDB) ⊓ (∃Δ ⊓ Δ.Rules.Change) then
    Rulesdb.Rule ← {x | <RulesΔ, x > Change}
    Endif
    /* Update the original RuleDB for the new and updated rules.*/
    Execute.update(RulesDB,Rulesdb)
    End
```

Algorithm-1, A Self Evolutionary Rule-base algorithm for dynamic updates in Rule DB for evolving domain knowledge.

IMPLEMENTATION AND RESULTS

As mentioned above, the Rule-base system is a subcomponent of overall SI-CDSS [2], which has been developed and deployed on Microsoft Azure environment for its prototype demonstration. In SI-CDSS deployment, Rule-base is also implemented and deployed for Rough Set based inference engine with focus on diabetes disease. In this prototype demonstration, the Rule DB was developed and maintained in Microsoft SQL Azure. Figure 5-a shows the user interface for patients to share their experience of using the medicine. This information is then used by Rough Set for rules generation and after mining the experience, inference engine generates recommendations for the patients as shown in Figure 5-b. The rules used by Rough Set are also in different format than the other rules see Figure 2. All the attributes used in conditions and actions are listed as columns, are of fix number. The values for the conditions are separated using comma that represent the AND operation. The rules contain different values for different combinations of symptoms. The numbers of symptoms are 33 in every single rule. With system learning new rules, we test our system for its claims using proposed structure with database manipulation against text files using string manipulations. We introduce a bunch of random 20 rules (i.e., new rules, existing rules, and some updated rules) and test both techniques performance. All these testing and deployment experiments are carried out on local machine with 3 GB memory and 2.67 GHz of quad core processor.

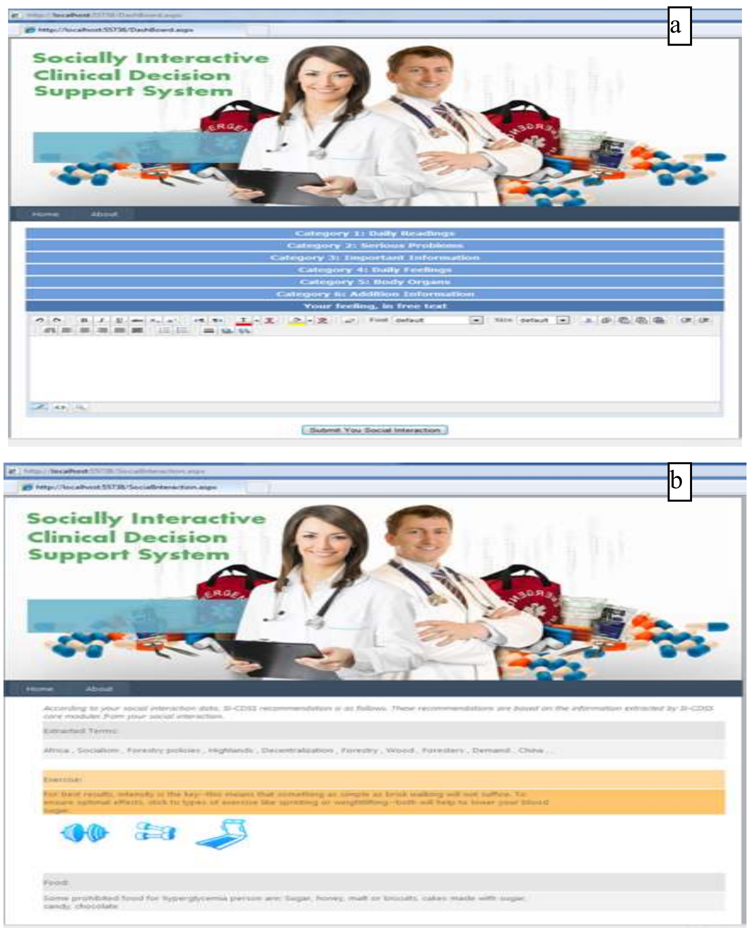

Figure 5. (a) shows the interface for patient's experience entry while (b) shows the recommendations generated by SI-CDSS based on the rules from Rule-base.

With the novelty of introducing Self Evolutionary Rule-base, in addition, our proposed structure for rule storage also proved better performance than the traditional text file based rules storage. The Self Evolutionary Rule-base not only helps in timely updates in the Rule-base but the rules generated also become more compact. The Self Evolutionary Rule-base process is completely automatic; however, for generated rules verification, the system also have provision for experts (doctors) to verify rules and eliminate rules that are not compliant with standard knowledge of the domain.

CONCLUSION

Rule-based systems are mostly used in healthcare domain for developing CDSS. With the passage of time, the rules containing expert and domain knowledge needs to be updated to accommodate new discoveries. To achieve this, a Self Evolutionary Rule-base system is proposed with a generic structure to store different types of rules. The system is implemented as subcomponent in currently under development system SI-CDSS in Microsoft Azure environment for diabetes patients. The overall working and performance of proposed scheme proves better in comparison against the text based rules management systems performance. In future, our focus is to work for full capacity of the proposed Rule-base i.e., to work with several different inference engines with diverse nature of rules in the Rule-base at a time and still achieve its claimed effectiveness.

ACKNOWLEDGMENTS

This work was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agent)" (NIPA-2011-(C1090-1121-0003).

REFERENCES

1. E. S. Berner, D. Ed, "Clinical Decision Support Systems: State of the Art", Department of Health Services Administration, University of Alabama at Birmingham, AHRQ Publication No. 09-0069-EF, June 2009.
2. I. Fatima, M.Fahim, D. Guan, Y. K. Lee and S. Y. Lee, "Socially Interactive CDSS for u-Life Care", The 5th ACM International Conference on Ubiquitous Information Management and Communication (ACM ICUIMC 2011), Seoul, Korea, February 21-23, 2011.
3. F. Feng, X.Y. Liu, V. Leoreanu-Fotea and Y.B. Jun, Soft sets and soft rough sets, Inform. Sci. (2010) 10.1016/j.ins.2010.11.004.
4. D. Jegelevicius, A. Krisciukaitis, A. Lukosevicius, V. Marozas, A. Paunksnis, V. Barzdziukas, M. Patasius, D. Buteikiene, A. Vainoras, L. Gargasas L. "Network Based Clinical Decision Support System". 9th International Conference on Information Technology and Applications in Biomedicine (ITAB), Larnaca, Cyprus, 2009.
5. J. A. Kim, I. Cho, Y. Kim. "Clinical Decision Support System Architecture in Korea", International Conference on Convergence and Hybrid Information Technology, Washington DC, USA, 2008.

6. E. J. Ko, H. J. Lee, and J. W. Lee. “Ontology and CDSS based Intelligent Health Data Management in HealthCare Server”, world Academy of Science, Engineering and Technology, 2007.
7. A. M. Khattak, Z. Pervez, K. K. Ho, S. Y. Lee, and Y. K. Lee. “Intelligent Manipulation of Human Activities using Cloud Computing for u-Life Care”. The 10th Annual International Symposium on Applications and the Internet(SAINT 2010), Seoul, Korea, 2010.
8. V. Mauno, S. Crina, "Medical Expert Systems", Current Bioinformatics, 3, pp 56-65(10), January 2008, doi = doi:10.2174/157489308783329869
9. D. Petrovic, A. Duenas, S. Petrovic. “Decision support tool for multi-objective job shop scheduling problems with linguistically quantified decision functions” *Decision Support Systems*, 43(4), 1527-1538, 2007.
10. A. Radzikowska and E.E. Kerre, A comparative study of fuzzy rough sets, Fuzzy Sets and Systems 126(2), pp. 137–155, 2002.
11. S. Tsumoto, Automated induction of medical expert system rules from clinical databases based on rough set theory, Information Sciences 112 , pp. 67–84, 1998.