Software Architecture of a Learning Apprentice System in Medical Billing

Aftab Ahmed, Umair Abdullah and Mohammad J. Sawar

Abstract—**Machine learning is an emerging field of computer science concerned with the learning of knowledge from exploration of already stored data. However, effective utilization of extracted knowledge is an important issue. Extracted knowledge may be best utilized via feeding to knowledge based system. To this end, the work reported in this paper is based on a novel idea to enhance the productivity of the previously developed systems. This paper presents the proposed architecture of a Learning Apprentice System in Medical Billing system being developed for medical claim processing. A new dimension is added whereby, the process of extracting and utilization of knowledge are implemented in relational database environment for improved performance. It opens enormous application areas as most business data is in relational format managed by some relational database management server. The major components of the proposed system include knowledge base, rule engine, knowledge editor, and data mining module. Knowledge base consists of rules, meta rules and logical variables defined in the form of SQL queries stored in relational tables. Rule engine has been successfully developed and deployed in the form of SQL stored procedures. Knowledge editor and data mining modules are under development. Given architecture depicts over all business process of medical billing along with major components of the system. The proposed architecture effectively integrates all three pertinent components given by data mining (production rule discovery), rule based systems technology and database systems environment.**

Keywords-component; Artificial Intelligence, Intelligent Systems, knowledge engineering.

I. INTRODUCTION

Learning apprentice systems were introduced since 1980 with inception of famous LEAP system by T. M. Mitchell [12] and followed by work of many researchers including [3], [4], [5], [6], and [7] in this area. Originally, the main technique practiced was to integrate learning module with inference engine of an expert system in order to speed up the knowledge acquisition process leading to the improved performance of the system. The same basic concept is still valid and used in all research and development work being carried out today. However, an additional dimension of 'data mining' has been combined in order to take advantage of huge volumes of data accumulated in almost every major organization of the world. For instance, an interesting architecture of classical expert system with and without data mining module has been described by Nada Lavra'c [9]. Nevertheless, many researchers in the domain are working on integration of data mining and Artificial Intelligence (AI) systems (rule based, intelligent, knowledge based, expert etc.). Generally, it has been widely accepted that integration of AI system feed with knowledge from domain experts, data mining and machine learning techniques has potential to offer enhanced learning of knowledge than configuration involving AI systems, domain experts and machine learning (without data mining techniques) as knowledge source. System developed by Ortega [10] is an excellent example of intelligent system which uses data mining techniques.

The work reported in this paper is based on a novel idea to enhance the productivity of the previously developed systems by integration of data mining and AI techniques as mentioned above. Accordingly, a new architecture of Learning Apprentice System is being proposed and implemented in the field of medical claim processing for identifying and fixing (if possible) errors in medical claims. The proposed architecture is based on the development in relational database environment to maintain the medical billing data is in relational format (as generally data is available in this format) and database management system. Subsequently, implementation of data mining module and rule engine/inference engine is in languages like Structure Query Language (SQL). It is evident from our experience that the proposed architecture has enhanced the performance of the respective system. Moreover, the proposed architecture will also eliminate need of transforming data from relational environment into the environment of rule engine/ inference engine.

The major components of the developed system including knowledge base, rule engine, knowledge editor and data mining modules are represented in Figure: 1. Knowledge base consists of rules, Meta rules, and logical variables defined in

Manuscript received March 14, 2009. This work has been supported by Higher Education Commission of Pakistan under '5000 – indigenous PhD Fellowships Scheme'.

Aftab Ahmed is Dean and Director at Foundation University of Engineering and Management Sciences (FUIEMS) Rawalpindi, Pakistan (e-mail: aftab_ff@hotmail.com).

Umair Abdullah is PhD scholar at FUIEMS Rawalpindi, Pakistan (e-mail: umair_pitafi @yahoo.com).

Mohammad J. Sawar is Dean and Director at Faculty of Engineering & IT, Northern University Nowshera, Pakistan. (e-mail: sawar@northern.edu.pk).

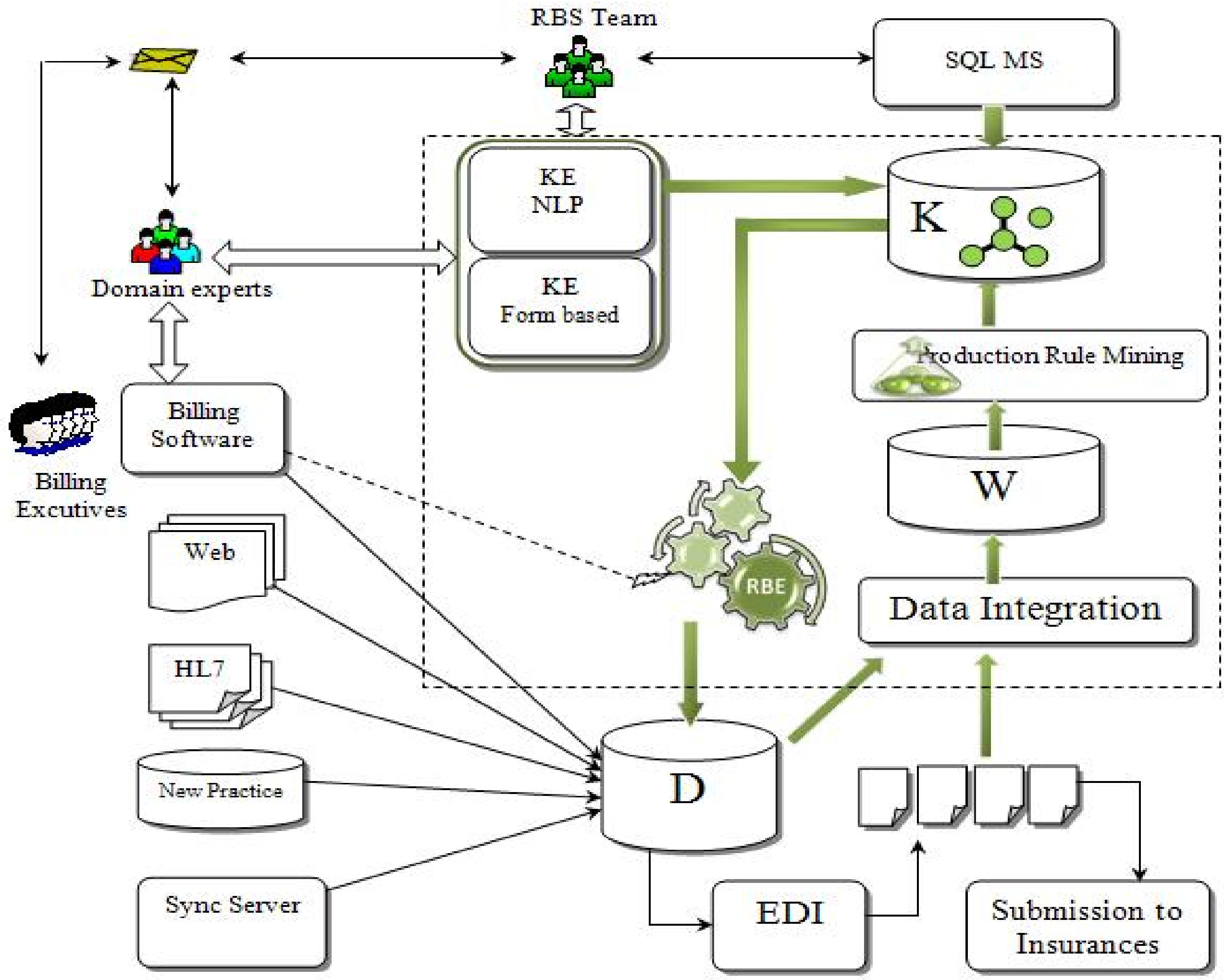

Figure 1. Architechture of Data Mining Driven Learning Apprentice System for Medical Billing.

the form of SQL queries. Rule engine has been successfully developed in the form of SQL stored procedures and deployed in the production environment [14]. However, rule editor and data mining modules are under development. The reported research and development work is being carried out at a USA based medical billing company by the team of domain experts, software engineers and an AI consultant. The team is known as 'RBS team', and the system is known as simply 'RBS' (i.e. Rule Based System).

Application domain of this RBS system is medical billing, which is data rich domain [15] with huge body of knowledge. As generally defined medical billing is the process of submitting and processing of claims to insurance companies in order to get reimbursement for services rendered by a healthcare provider (doctor, physician, surgeon etc). A customer can do billing by himself/herself known as in-house billing or can contract a billing company for handling his/her medical bills [15].

Due to the complication of billing process, about 30% of all medical claims are denied at first level. 35% of initially rejected claims are rejected again. Up to 10% of rejected claims are never collected [15]. By fixing preventable medical billing errors, payment collections can be increased, and rejections can be reduced.

Many companies have developed customized software to resolve claim rejection issue generally known as claim 'scrubbing'. Claim scrubbing can be defined as verification and checking of the medical claims for any potential errors and perform legitimate change in information before it is sent to insurance payer.

A well designed and technology supported medical billing procedure does not allow errors to proliferate and will process over 90% claims to be paid after their first submission [15].

The proposed architecture of the overall system developed is presented in Section II. A brief description of components interacting with our learning apprentice system has been already explained. The components of learning apprentice system (represented in dotted line in Figure: 1) have been described in Section III, Section VI, Section V, Section VI and Section VII. Finally, summary and conclusion is presented to summarize the contributions of this work along with future research directions.

II. SYSTEM ARCHITECTURE

The organization of the overall system is known as software architecture [1] developed through integration of many components. Architecture of a data mining driven learning apprentice system for medical billing is given in Figure: 1. Billing software is the main entity of the entire system and most of the users interact with it. Billing executives use it for inserting, updating, and modifying data concerning claims. Domain experts use it for pulling various types of reports, importing data and following up claims. Billing software triggers Rule Based Engine (RBE) to perform billing compliance related checks on a claim saved at that time. Thus RBE applies its production rules on the claim (which is being saved) and tells the user his/her faults found in the claim. RBE also performs scrubbing activity as defined in Section-I as part of the production rules.

Main database contains claim related data, input from billing software, websites, HL7 (i.e. Health Level 7) files, data imported from database of new practices and data sent by sync server (synchronization server). Electronic Data Interchange (EDI) module gets data from the database and converts it into text files in specific format (like 837 format), which is acceptable by the insurance companies. Submission module sends these files to the insurance companies via internet. It is handled under the process outside of our learning apprentice system (represented by dotted line box). Learning apprentice system has four major components (shown in dotted line box of Figure 1). Knowledge Editor (KE) is shown on top left corner consisting of two parts. One part of knowledge editor is conventional graphical user interface form based. Second part of KE is Natural Language Processing (NLP) based. It has facility to dialogue with the user for learning and verification of learned knowledge. Knowledge is in the form of production rules stored in knowledge base represented by cylinder with symbol 'K' (Refer to the top right corner of the system).

Rule Based Engine (RBS) is depicted in the form of gear present at bottom left area of dotted box. Rule engine has been developed in Structured Query Language (SQL) in the form of stored procedures [14]. A triggering mechanism linked to it from billing software has been also shown. Remaining blocks in learning apprentice system belong to data mining module. These include 'data warehouse' (represented by cylinder with symbol 'W'), data integration module, and production rule mining module.

People shown on top of the diagram are development team of this learning apprentice system (known as RBS team). Currently the team is using Microsoft Management Studio (SQL MS) to develop, debug and update knowledge base of the system. RBS team and domain experts communicate with each other through email. Domain experts use knowledge editor to add new knowledge to the knowledge base (in the form of production rules) or verify any new rules found by production rule mining module.

III. KNOWLEDGE BASE

Knowledge base of the system consists of rules, Meta rules and logical variables [14]. Condition part of a rule is implemented in the form of where clause of SQL query. If condition is true i.e. rule query gives some results (one or more records) then action part of the rule is executed which is suffixed after the condition of where clause of rule query. In traditional rule based system [2] logical variables get their value by matching with the working memory elements. In this system whole database is working memory and logical variables get their value(s) by executing a SQL query. Each logical variable has a SQL query associated with it, which when executed returns value(s) of that logical variable.

IV. RULE ENGINE DESIGN

Design of Rule Engine has been discussed in details in [14]. It has been implemented in structured query language. The purpose of the system is to handle 'claim scrubbing' but implementation approach is based on rule based system technology [2]. The main entities of this rule based system are 'meta rules', 'rules' and 'logical variables', all of them stored in their respective tables[15].

Rule engine applies all the rules on a given claim and finds out errors and faults. Each rule is equivalent to a check with some action part associated to it. Each rule has been implemented as 'where' clause of a SQL query [14] and 'select' portion of the rule query is attached by the rule engine.

For example date of birth of patient could not be greater than its date of service (i.e. obviously a patient cannot be treated before his/her birth). But this could happen due to manual data entry mistake. Therefore, the check of 'date of birth follows date of service' is implemented as *'where '<DOB>' > '<DOS>' '*. Where <DOB> (i.e. date of birth of the patient) and <DOS> (i.e. date of service of the patient claim) are logical variables and will be replaced by their respective values from the given claim and patient. The respective values will be changed when rule is applied on some other claim. Rule engine will get their values by executing the query stored in Logical_variables table. Suppose if date of birth of the patient is '25/01/10' and date of service (mistakenly entered) is '20/01/10'. The rule query after replacing the value of logical variable will become

select @rowcnt=count(*) where'25/01/10' > '20/01/10'

Now 'where' clause is true, @rowcnt SQL variable will get value 1 indicating the error of 'patient date of birth follows date of service'. The claim having these values will be

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

identified by RBS as faulty claim and hence will be blocked from submission.

Now suppose in case of a newly born baby date of birth of is ‘20/01/10’ and date of service is also ‘20/01/10’. Rule query after replacing value of logical variables will become

select @rowcnt = count(*) where ‘20/01/10’ > ‘20/01/10’

In this case of @rowcnt SQL variable will get 0 value (as condition is false in ‘where’ clause) thus indicating that error of ‘patient date of birth follows date of service’ has not occurred.

Note that RBS has the ‘scrubbing’ ability i.e. it can tweak the claim for obvious valid and legal change. Each rule has its own scrubbing action. Most of the critical rules block the faulty claim from submission to insurance payers, while non critical rules just display warning message to the user [15]. In future current rule engine design will be replaced by ‘Enhanced Rule Engine Design’ proposed by Sawar [8].

V. Data Warehouse

A data warehouse is required for mining/extracting new knowledge in the form of database. Although a copy of operational database can be used for extracting new knowledge, specific format of data warehouse will facilitate data mining algorithm for learning of new knowledge). Design of data warehouse should be carefully constructed as construction and maintenance of data warehouse will demand various resources including time and money.

First and the main reason is that when a claim is rejected by an insurance record of the claim, it is modified in operational database to remove the rejection reason. After modification corrected claim is resubmitted to the insurance and the respective data with which claim was rejected is modified in operational database. However, the same data is important for a production rule mining algorithm to build rules to avoid future rejections. Operational database contains either information of unpaid claims or paid claims (obviously with correct and clean data) accepted by the insurance companies. Moreover, the data warehouse contains all the historic records of claims i.e. with faulty data rejected from insurance companies and final clean data for the corresponding accepted claims.

VI. Production Rule Mining

In Artificial Intelligence language a ‘rule’ is an implication of type X → Y. where X and Y are some entity sets, where X is called antecedent and Y is known as consequent. Data mining and machine learning (inductive learning) have many algorithms for mining of rules. But data mining algorithms are for mining ‘association rules’ not ‘production rules’. Many researchers have modified famous apriori algorithm for their specific needs [13]. Concept of association rule with NULL or empty consequent has been introduced in [13]. However, association rules cannot be used as knowledge for claim scrubbing.

Although, the main theme of [11] shows that the algorithm is meant for production rule discovery and it generates classification rules. Therefore, it is compared with AQ and ID3 algorithms, which are famous classification rule generation algorithms. This algorithm [11] has been further tested by other researchers and found to be better than AQ and ID3. Classification rules can be used for classifying claims as faulty but classification rule mining algorithm cannot serve the purpose of claim scrubbing. Consequent of a classification rule is name of a *‘class’*, while a production rule has some *‘action’* as consequent of a rule. Finding a name of class to which given instance belongs on the basis is training data is relatively simple job as compared to finding ‘actions’. Action can be any sort of modification in the database. However, the finding production rules were needed to merge data mining techniques with machine learning methods.

VII. Knowledge Editor

Currently ‘rule editor’ has been built as given in [14]. It allows user to add and update rules. But we shall upgrade it to ‘knowledge editor’ where domain expert can verify mined rules, test and debug all rules. Rule of ‘date of birth follows date of service’ has been shown in Figure: 2. Rule query is generated by the rule editor when user presses ‘Generate Query’ button.

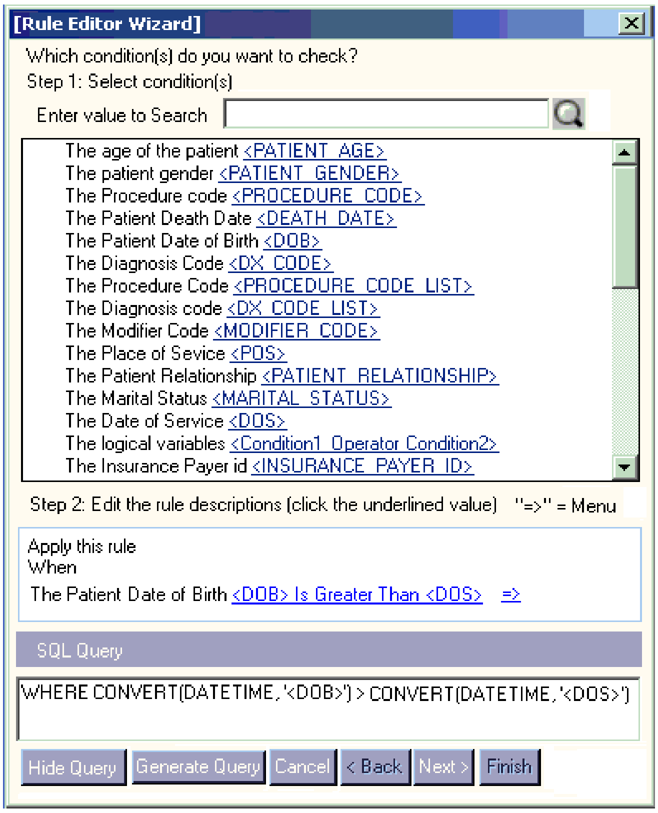

Figure 2. Architechture of Data Mining Driven Learning Apprentice System for Medical Billing.

Knowledge Editor (KE) has two parts; one is form based rule editor shown in Figure: 2. While other will be natural language based, which is currently under research. In rule editor first panel is a list atomic information segments. User can use these information segments to construct rules. Second fragment is the main area, where domain user will construct complex knowledge oriented medical billing compliance rules by joining the information segments selected from upper list, with the help of logical operators 'and', 'or', 'not'. User can use parenthesis to set the precedence of conditions [14].

After completing the logical construct of production rule user can press 'Generate Query' button to view the resultant rule query generated by the rule editor on the basis of rule created by the user. User can edit this query shown in third panel provided he/she has knowledge of SQL programming. At the end the newly created production rule in the form of SQL query is stored in 'rules table' present in production database. Rule engine uses these rules for identifying faults in claims before their submission to medical insurance companies.

VIII. Conclusion

Research and development work on each module of the system shown in this paper is in progress. Rule engine and rule editors have been completed. Rule engine has been developed using SQL in the form of stored procedures. Architecture presented here can be mapped to any knowledge rich domain for applying large number of knowledge oriented checks and performing actions on the basis of results of those checks.

AI systems (expert systems, rule based systems, knowledge based system, learning apprentice systems etc.) were originally developed with an inference engine and manual feeding of knowledge by domain experts. It was then followed by the work in the area of machine learning. Subsequently, the data mining researchers are proposing of the utilization of mined knowledge by some AI system. A new dimension of 'development of AI concepts in SQL' has been proposed for its enhanced utilization in business and industry.

Acknowledgment

The authors would like to thank Higher Education Commission of Pakistan and the respective USA based medical billing company for providing excellent research environment (The name of the company is not been disclosed due to the 'Cooperate Identity Disclosure Policy' of the company).

References

[1] D. Garlan, M. Shaw ,"An Introduction to Software Architecture," Advances in Software Engineering and Knowledge Engineering, Volume I, edited by V.Ambriola and G.Tortora, World Scientific Publishing Company, New Jersey, 1993.

[2] F. H. Roth, 'Rule-based systems," Communications of the ACM, vol. 28, pp. 921-932, September 1985.

[3] L. Dent, J. Boticario, J. McDermott, T. Mitchell and D. Zabowski, "A Personal Learning Apprentice," Proceedings of the Tenth National Conference on Artificial Intelligence , AAAI Press, San Jose, CA, 1992, pp. 96–103.

[4] Leonard A. Hermens , Jeffrey C. Shlimmer, A Machine-Learning Apprentice for the Completion of Repetitive Forms, IEEE Expert: Intelligent Systems and Their Applications, v.9 n.1, p.28-33, February 1994 [doi>10.1109/64.295135]

[5] M. J. Sawar, "A learning apprentice system for medical decision support systems", PhD Thesis, School of Computer Studies, University of Leeds, 1993.

[6] M. J. Sawar and R. C. Thomas *"Learning Apprentice System for Turbine Modeling"* in Proceedings of IEA/AIE-90, Charleston, U.S.A. July 1990.

[7] M. J. Sawar, T. G. Brennan, A. J. Cole and J. Stewart *"An Expert System for PostOperative Care (POEMS)"*, Proceedings of MEDINFO-92, Geneva, Switzerland, Sep. 1992.

[8] M. J. Sawar, U. Abdullah, A. Ahmed, *"Enhanced Design of a Rule Based System using Structured Query Language"* submitted to The 2010 International Conference of Computational Intelligence and Intelligent Systems, at World Congress of Engineering, London, U.K., 30 June - 2 July 2010. unpublished

[9] N. Lavra'c, *"Data Mining and Decision Support: A note on the issues of their integration and their relation to Expert Systems"* PKDD'01 workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning: Positions, Developments and Future Directions, p 1-8. 2001.

[10] P. A. Ortega, C. J. Figueroa, G. A. Ruz, *"A medical claim fraud/abuse detection system based on data mining: A case study in Chile"* , DMIN'06, The 2006 International Conference on Data Mining, Las Vegas, Nevada, USA, June 26-29, 2006

[11] M. R. Tolun, and S.M. Abu-Soud. *"ILA: An Inductive Learning Algorithm for Production Rule Discovery,"* Expert Systems with Applications. 14: 361-370. 1998

[12] T. M. Mitchell, Sridhar Mahadevan, and Louis I. Steinberg. "LEAP : A Learning Apprentice for VLSI Design," in Proceedings of Ninth IJCAI. pp. 573-580. 1985.

[13] U. Abdullah, J. Ahmed, and A. Ahmed, "Analysis of effectiveness of apriori algorithm in medical billing data mining" Proceedings of 4th IEEE International Conference on Emerging Technologies, Rawalpindi, Pakistan. 2008.

[14] U. Abdullah, M. J. Sawar, A. Ahmed, "Design of a rule based system using Structured Query Language", 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC09), Chengdu, China. 2009. P 223-228.

[15] U. Abdullah, M. J. Sawar, A. Ahmed, "*Comparative Study of Medical Claim Scrubber And A Rule Based System*" in Proceedings of IEEE 2009 International Conference on Information Engineering and Computer Science (ICIECS 2009), Wuhan, China. 2009.