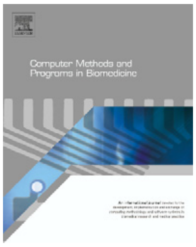
journal homepage: www.intl.elsevierhealth.com/journals/cmpb



Developing guideline-based decision support systems using protégé and jess

Chiehfeng (Cliff) Chen[a,b,c,d], Kung Chen[e], Chien-Yeh Hsu[c,*], Yu-Chuan (Jack) Li[c,f,*]

[a] Graduate Institute of Medical Sciences, College of Medicine, Taipei Medical University, Taiwan
[b] Department of Surgery, Taipei Medical University – Wan Fang Hospital, Taiwan
[c] Graduate Institute of Biomedical Informatics, Taipei Medical University, Taiwan
[d] Department of Public Health, School of Medicine, College of Medicine, Taipei Medical University, Taiwan
[e] Department of Computer Science, National Chengchi University, Taiwan
[f] Department of Dermatology, Taipei Medical University – Wan Fang Hospital, Taiwan

ARTICLE INFO

Article history:
Received 6 April 2010
Received in revised form
20 May 2010
Accepted 23 May 2010

Keywords:
Computerized physician order entry
Clinical decision support system
Clinical practice guideline
Protégé
Jess

ABSTRACT

The Institute of Medicine has identified both computerized physician order entry and electronic prescription as keys to reducing medication errors and improving safety. Many computerized clinical decision support systems can enhance practitioner performance. However, the development of such systems involves a long cycle time that makes it difficult to apply them on a wider scale. This paper presents a suite of guideline modeling and execution tools, built on Protégé, Jess and Java technologies, which are easy to use, and also capable of automatically synthesizing clinical decision support systems for clinical practice guidelines of moderate complexity.

1. Introduction

A clinical practice guideline (CPG) is a systematically developed statement designed by expert clinicians to assist practitioners in deciding about appropriate healthcare choices for a specific clinical condition [1]. Adherence to evidence-based practice guidelines examined in a study averaged approximately 59% nationwide in the US [2]. Guideline adherence by physicians still has room to improve. Most guidelines, such as the Adult Treatment Plan III (ATP III) for hypercholesterolemia [3], are relatively complicated so that the physicians might not adhere to them everytime.

Computerized clinical decision support systems (CDSSs) are information systems designed to improve clinical decision-making [4]. They may make use of a spreadsheet, graphical tools, and artificial intelligence to present data. In general, CDSSs are especially helpful when the clinical data and rules are highly complex and on a large scale [5]. Historically, much of the development of expert systems occurred in the 1970s [6,7]. Literature on the effects of health information technology (HIT) revealed that implementing an electronic health record (EHR) system with a CDSS could increase the delivery of care that would adhere to guidelines and protocols, enhance the capacity of care delivery, reduce rates of medication errors, and decrease utilization of care [8].

[*] *Corresponding author at*: Graduate Institute of Biomedical Informatics, and Cancer Excellency of Clinical Research, Taipei Medical University, 250 Wu-Xin Street, Taipei City 110, Taiwan. Tel.: +886 2 27361661.
E-mail addresses: cyhsu@tmu.edu.tw (C.-Y. Hsu), jack@tmu.edu.tw (Y.-C. Li).

0169-2607/$ – see front matter © 2010 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2010.05.010

To address these issues, a CDSS which integrates the CPG into a computerized physician order entry (CPOE) system that aimed to improve the health care quality and enhance the adherence to CPG has been developed [9]. Although the CPG-embedded CDSS works well, it was developed manually by hard-coding the logic of a CPG into the underlying CDSS. Consequently, it is not easy for physicians to understand the logic working behind the scene. Furthermore, any changes in the logic of the CPG necessitate significant re-working of the CDSS. Therefore, we looked into the feasibility of developing guideline-based CDSS using existing software tools to overcome such weakness.

2. Materials and methods

2.1. *Overview*

This paper presents a suite of guideline modeling and execution tools which we developed. Our tools were built on top of *Protégé* [10] (version 3.0), *Jess* [11] (version 6.0), and Java technology (version 1.4, Sun Microsystems, Inc., USA). There are two major parts of the system: (1) the modeling part in Protégé and (2) the execution part based on Jess. Protégé is a free, open source ontology editor and knowledge-base framework developed at Stanford University (Palo Alto, CA, USA). It provides a very powerful environment and a plug-in application programming interface (API) to develop customized knowledge-based applications. Furthermore, Protégé also supports the construction of object instances from predefined classes using a drag-and-drop mechanism in a visualized environment. Based on Protégé's mechanisms, we have built a guideline authoring environment that models the logic and rules of a guideline using a decision graph (DG) with a visual representation. A plug-in of Protégé was developed to translate the visual representation of guideline rules to an internal representation in extensible markup language (XML), which in turn was converted to Jess rules for execution. The overall process is illustrated in Fig. 1.

The Jess rules generator (a small Java program written by the authors) converts the XML representation into a format that Jess can understand (Jess rules). Jess is small, light, and one of the fastest rule engines available. Jess is a famous rule engine and scripting environment written in Java by Ernest Friedman Hill at Sandia National Laboratories [11]. Its powerful scripting language gives us access to all of Java's APIs. Jess provides a structured way of testing a given format and ultimately provides an executable answer.

Since rule interpretation is the major part of guideline execution, we designed our guideline execution environment on top of Jess. We treated Jess as a library component in a Java application, and divided the guideline execution environment into two parts: a generic part and a rule-specific part. The generic part is pre-built and contains the main program that drives the guideline execution process, including user interaction, patient data retrieval, and activation of Jess. The guideline rules are embodied in Jess rules after the translation process and finally are interpreted by the Jess engine at the command of the main program. In short, a guideline-based CDSS can be synthesized automatically and executed without extra coding.

2.2. *Guideline modeling and authoring*

We modeled the logic and rules in a guideline using a DG with a visual representation. There are three kinds of nodes in a Protégé's DG: (1) start node, (2) information node, and (3) terminal node. A start node marks the beginning of a decision process; an information node collects and computes patient health-related data as required by the guideline logic; and a terminal node gives patient-specific recommendations according to the guideline.

On the edge between two nodes, we attach a Boolean criterion involving patient information that is derived from the guideline rules. A special criterion named *Otherwise* is used to simplify the specification of Boolean criteria. The truth values of the Boolean criteria attached on the edges determine a path from the start node to one of the terminal nodes. The idea is quite intuitive; we learn about a patient's health status step-by-step along the nodes in the graph; and finally a path leads us to a patient-specific recommendation as determined by the guideline. Fig. 2 shows a simplified version of the rules of the ATP III guideline.

Inside a Boolean criterion or an information node, one needs to refer to various data items concerning the patient encountered and other guideline-specific information. These items are like variables in common programming and are defined using the Protégé ontology editor. In addition, we have also provided a few library functions, such as computing the time interval between the period when the laboratory data were first acquired and the current time. If needed, more functions can be easily added using Protégé's Java API. These variables and function definitions are all sharable and reusable across similar guidelines.

Furthermore, in order to make the process of constructing a DG user-friendly, we made use of the visual support of Protégé which provided a visualized approach to draw the decision graph and to build the Boolean criteria attached on the edges. Fig. 3 displays a screen shot of the guideline authoring environment we developed on top of Protégé. The leftmost column shows the ontology definitions of the data items (Medical Term) and various entities used to construct a DG. The panel in the middle is the visual mechanism for specifying the DG.

2.3. *Guideline representation and translation*

As stated earlier, rules have to be extracted from a DG and translated into Jess format for execution. A closer look at the structure of a DG reveals that it is an embodiment of a finite state machine (FSM). The nodes of a DG correspond to the states of the associated FSM and the edges specify the transition between two states, while the Boolean criterion attached on an edge decides whether a state transition should be triggered. Hence, what the translator needs to do is to represent the state transition diagram of an FSM in Jess rules. Fig. 4 shows an example of Jess rules generated by the translator.

A major task is the design and translation of an expression language for specifying the Boolean criteria derived from

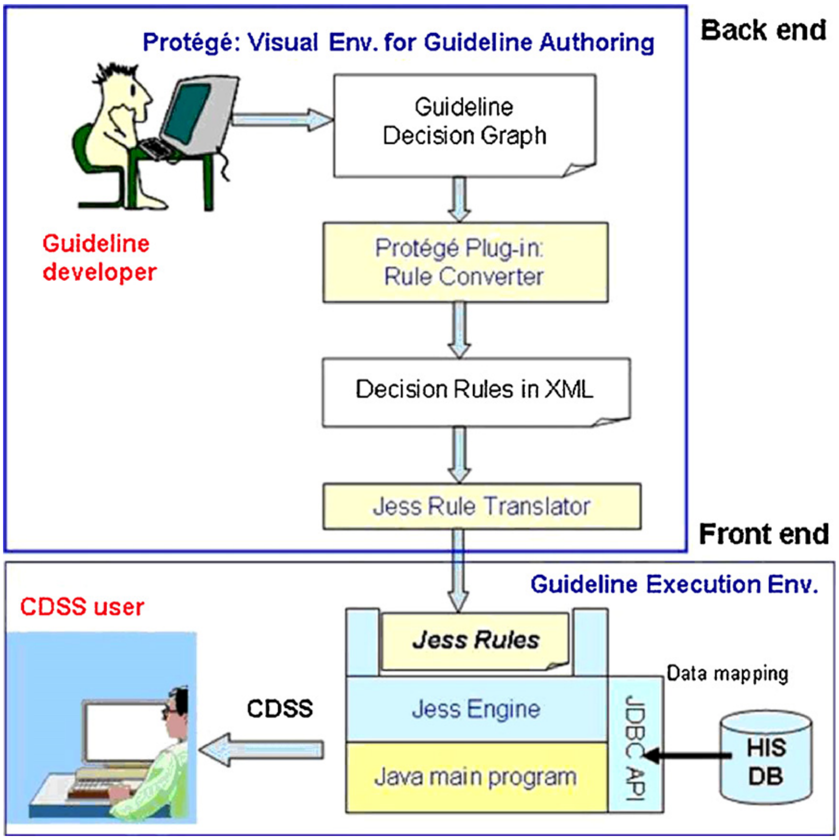


Fig. 1 – Overall process of our approach.

guideline decision rules. We defined two syntactic categories for our expression language. Basic criteria support common arithmetic operators and relational operators between the values of two medical terms, including constants, while combo criteria build complex Boolean expressions using the logical connectives of *not*, *and*, and *or*. For example, the ATP III guideline [3] recommends "if total cholesterol (TC) is ≥200 mg/dL and the quotient of TC over high density lipoprotein (HDL) is >5 mg/dL or HDL is <40 mg/dL, a follow-up lipoprotein profile is needed for appropriate management based on low density lipoprotein (LDL)", which could be translated to a combo criteria composed of three basic criteria as follows:

$\mathrm{TC} \geq 200$ and $\mathrm{TC/HDL} > 5$ or $\mathrm{HDL} < 40$

Furthermore, many criteria of laboratory data require more complex operations involving time-related calculations. To address such needs, we have provided a few library functions to implement those operations that can be invoked in the Boolean criteria. For example, the ATP III guideline [3] recommends "If LDL remains ≥130 mg/dL after 3 months of therapeutic lifestyle changes, consideration can be given to starting an LDL lowering drug to achieve the LDL goal". The following criterion demands that the LDL value of the patient must be obtained more than 3 months ago:

$\mathrm{LDL} \geq 130$ and time_interval(TC, current_time(), "M") > 3

2.4. *Guideline execution environment*

The main function of the guideline execution environment is to interpret the Jess rules translated from the guideline as represented in the DG with respect to a specific patient's health data. Hence, an important task is to get the patient's health data from the hospital information system (HIS) and use them during the rule interpretation process. A general solution for this data integration task needs to consider various complex data mapping issues that are beyond the scope of our study. Here we take a simpler approach to solve the problem. In particular, we use a configuration file that specifies the name mapping between the physical data items stored in the HIS database and the logical entities defined in our medical term ontology [12], and a generic data retrieval library component built for loading a patient data given his or her ID number. When the synthesized CDSS is brought up, the data retrieval component is invoked to retrieve the patient's data. The data mapping component then acquires the needed data from the result set and initializes the medical terms declared in the guideline using the configuration file.

Finally, the task of interpreting the guideline rule with respect to the patient's health data is dispatched to the Jess engine. After finishing the interpretation, the Jess engine then reports the result back to the main program, which in turn displays the recommended action to the clinician.

We attempted to establish a system composed of 5 modules, including Swing graphical user interface (GUI), Jess rules,

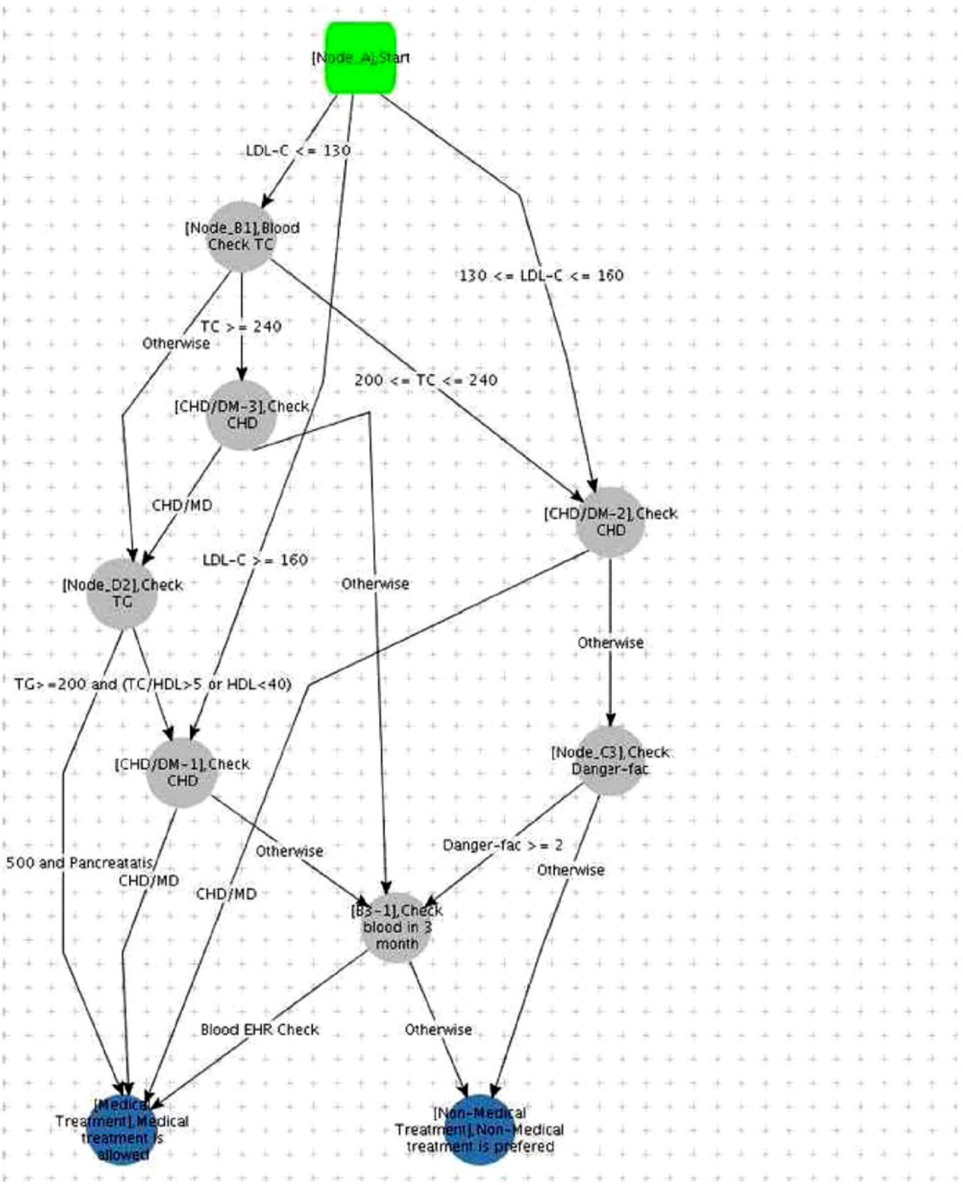


Fig. 2 – A simplified decision graph for ATP III.

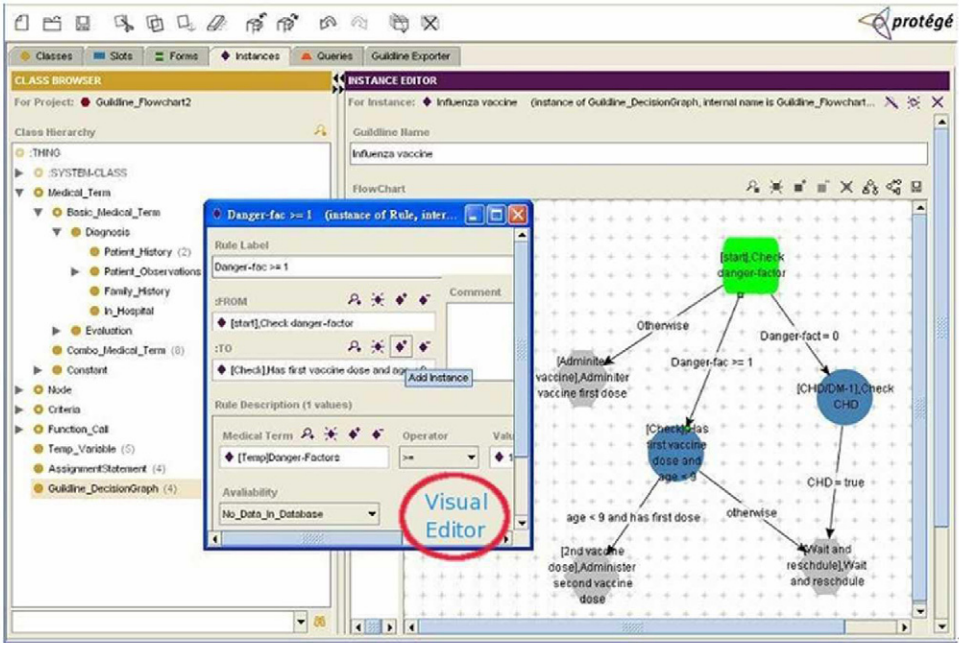
data access or Java data base connection (JDBC), Jess engine, and main program.

The components of the CDSS are defined as follows:

(1) The interface or Swing GUI is the component that puts the picture on the screen of CPOE in a format that can be interfaced by the guideline users (e.g. physicians).
(2) The main program is the manager that controls the flow of execution.
(3) Jess engine absorbs the rules and makes a decision; it is the smart part of the system that gives the executable answer. The Jess engine is a recent development in our CDSS, which we instituted not because it is an efficient device, but rather as an attempt to standardize the methodology.
(4) Data access: JDBC allows the CDSS to retrieve patient data and provides a construct for the program to talk to the data base, i.e., the librarian function.
(5) Jess rules: the actual rules in a format that Jess can understand.

In a summary of methods, except for the guideline authoring application, our tools for synthesizing guideline-based CDSS are all written in Java and integrated with Protégé and Jess. The guideline authoring application is a customized Protégé application with a pre-defined ontology and other related entities that provide a visualized environment for developing guideline representations using DGs. In Protégé, this application is a collection of classes, slots, and instances defined in a project file.

Two other major tools are the XML rule converter and the Jess rule translator. The former is a plug-in embedded into Protégé which can be activated inside the guideline authoring application. Once invoked, the rule converter first does a basic check of the DG under development and then converts it into an augmented representation of the corresponding FSM



For Project: Guildline_Flowchart2

For Instance: Influenza vaccine (instance of Guildline_DecisionGraph, internal name is Guildline_Flowchart...

Guildline Name: Influenza vaccine

Rule Label: Danger-fac >= 1

:FROM: [start],Check danger-factor

:TO: [Check],Has first vaccine dose and ag...

Medical Term: [Temp]Danger-Factors

Operator: >=

Availability: No_Data_In_Database

Fig. 3 – A screen shot of our guideline authoring environment in Protégé (version 3.0) Basic_Criteria and Combo_Criteria.

```
(defrule start2node_2
       ?f1 <- (patient (CHD ?CHD) (MD ?MD) (LDL-C ?LDL-C) (state ?state) (done ?done))
       (test (= 0 (str-compare ?done no)))
       (test (= 0 (str-compare ?state start)))
       (test (> ?LDL-C 160))
     =>
       (modify ?f1 (state node_2)) )

(defrule node_22treatment_1,medical-heal
       ?f1 <- (patient (CHD ?CHD) (MD ?MD) (LDL-C ?LDL-C) (state ?state) (done ?done))
       (test (= 0 (str-compare ?done no)))
       (test (= 0 (str-compare ?state node_2)))
       (or (test (= 0 (str-compare ?CHD true))) (test (= 0 (str-compare ?MD true))) )
     =>
       (modify ?f1 (state treatment_1,medical-heal))
       (modify ?f1 (done yes)) )
```

Fig. 4 – Example of Jess rules generated by the translator.

in XML. In addition to the state transition rules, this representation includes a list of data items that are referenced in the DG and need to be initialized with values from the HIS database.

The Jess rule translator takes the XML rules and translates them into Jess rules that implements a forward reasoning scenario according to the FSM embodied in the XML rules. Two Jess rules generated by the translator are shown in Fig. 4. Each rule specifies a state transition between two states that will be made if certain conditions are satisfied

3. Results

We have a CDSS integrated with the ATP III guideline [9]. Therefore, we took historical data of the previous system to evaluate the effectiveness and correctness of the synthesized system. Among the 2514 patient records we tested, 76 produced inconsistent results were detected. After examining these 76 patient records in detail, we found that 70 resulted from using a different version of the criterion on patient's triglycerol (TG) value; our version requires a TG > 1000 mg/dL, while that of the CDSS is a TG > 500 mg/dL. The other six inconsistent results should in fact be attributed to undetected bugs of the CDSS.

4. Discussion

CDSSs have special requirements because they involve patient care, including the need for precision in dosing medication. A 1999 Institute of Medicine (IOM) report estimated that about 80,000 people are hospitalized due to medical errors, and of these, approximately 7000 die annually in the United States [13]. Of these errors, up to 70% are preventable. Similar reports in other countries showed that medication errors indeed have a significant impact on mortality, morbidity, and cost of care [14,15]. Among the 2514 patient records we tested, 76 produced inconsistent results were reported by our system. The result implies that our system can analyze guideline adherence of a patient's HER post hoc.

Our approach to developing guideline-based CDSS takes advantage of the knowledge-base framework of Protégé. Protégé has been used to write the knowledge base (KB) of ATHENA which has been developed as part of a project evaluating the implementation of CPGs for hypertension [12]. In ATHENA, the decision-support component, which makes use of the technology developed in the EON project [16], has two components: a KB written in Protégé that models the guidelines, and a guideline interpreter that applies patient data to create patient specific treatment recommendations consistent with the knowledge in the KB. However, in comparison with the guideline interpreter in ATHENA, Jess is a freeware with high accessibility. Thus, we used Jess as rule engine to interpret the guideline rules. We provide a user-friendly environment to develop easy-to-comprehend visual charts that can model the logic and rules of many guidelines. Moreover, such a visual formalism for representing guidelines is a good medium for both clinicians and software developers to communicate their understanding and opinions. In addition, bugs of the CDSS that are discovered can be corrected at the early stage of guideline implementation.

Admittedly, our knowledge representation model for CPGs is not as general as some other models, such as GLIF III, which is a language for structured representation of guidelines in order to facilitate sharing of clinical guidelines [17]. As to the guideline execution environment, we could have used other methods, such as direct interpretation, to develop our environment. We built it on top of Jess in order to test the feasibility of our approach quickly since the rule translation task is easier than direct interpretation. However, we encountered some difficulties in basing our execution environment on Jess. Specifically, we found that Jess cannot handle user interaction friendly, as it is designed to be a batch-oriented inference engine. If a clinician needs to interact with the patient to collect more data during the guideline steps, we will have to tweak Jess to accomplish such tasks. On the other hand, if we code the execution environment directly as an interpreter of FSMs, we may overcome such difficulties with less effort.

There are certain additional advantages of the CDSS, especially regarding standardization. In other words, as guidelines change, the CDSS code does not change. The only thing that has to be regenerated is the XML and the rules which are then reloaded into the CDSS enabling Jess to execute a new answer.

In addition, if CPG can be written using Protégé, then other hospitals will not have to re-invent the wheel, so to speak. Some systems for viewing and editing the knowledge model were reported [18,19]. The beauty of Protégé is that both humans and computers can read it. Our Jess rule translator can read the rules exported from Protégé and convert them into Jess rules. In addition, since both Protégé and Jess are freeware, readily accessible on the internet, they are commonly available for use.

In the future, we plan to extend our tools in the following directions. First, we will enhance the user interaction capability of our guideline execution framework by replacing the Jess engine with a FSM interpreter developed using the Model-View-Controller pattern [20]. Second, since many guideline rules require certain kinds of temporal analyses on the laboratory data of a patient, we will develop some high-level modeling mechanisms to express common temporal patterns in our guideline model. Finally, we will consider integrating the guideline-based systems with EHR. A potential solution to these problems is the development of a common model for EHR. Recently, we saw significant advances made toward such a model: the proposal of Taiwan Electronic Medical Record Template (TMT) [21]. We will investigate the feasibility of developing a generic data retrieval component based TMT so that the synthesized CDSS can have greater portability in terms of EHR data integration.

5. Conclusions

We have presented a suite of tools on top of Protégé and Jess for developing guideline-based CDSS. Our tools provide a visualized environment for modeling guidelines and a translator for converting the logic and rules of a guideline into Jess rules that are executable in a Java-Jess environment. Once a guideline is properly modeled in the Protégé-based development environment, an executable CDSS in Java is automatically synthesized from the logic and rules of the specified guideline.

REFERENCES

[1] M.J. Field, K.N. Lohr, Clinical Practice Guidelines: Directions for a New Program, National Academy Press, Washington, DC, 1990.

[2] S.L. Thier, K.S. Yu-Isenberg, B.F. Leas, C.R. Cantrell, S. DeBussey, N.I. Goldfarb, D.B. Nash, In chronic disease, nationwide data show poor adherence by patients to medication and by physicians to guidelines, Manage. Care 17 (2008) 48–52.

[3] Third Report of the National Cholesterol Education Program (NCEP) Expert Panel on Detection, Evaluation, and Treatment of High Blood Cholesterol in Adults (Adult Treatment Panel III) final report, Circulation 106 (25) (2002) 3143–3421.

[4] A.X. Garg, N.K. Adhikari, H. McDonald, M.P. Rosas-Arellano, P.J. Devereaux, J. Beyene, J. Sam, R.B. Haynes, Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review, J. Am. Med. Assoc. 293 (10) (2005) 1223–1238.

[5] S. Eslami, A. Abu-Hanna, N.F. de Keizer, Evaluation of outpatient computerized physician medication order entry systems: a systematic review, J. Am. Med. Inform. Assoc. 14 (4) (2007) 400–406.

[6] J.M. Teich, P.R. Merchia, J.L. Schiz, G.J. Kuperman, C.D. Spurr, D.W. Bates, Effects of computerized physician order entry on prescribing practices, Arch. Intern. Med. 160 (18) (2000) 2741–2747.

[7] J. Lazarou, B.H. Pomeranz, P.N. Corey, Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies, J. Am. Med. Assoc. 279 (15) (1998) 1200–1205.

[8] B. Chaudhry, J. Wang, S. Wu, M. Maglione, W. Mojica, E. Roth, S.C. Morton, P.G. Shekelle, Systematic review: impact of health information technology on quality, efficiency and costs of medical care, Ann. Int. Med. 144 (10) (2006) 742–752.

[9] C. Chen, K. Chen, C.Y. Hsu, W.T. Chiu, Y.C. Li, A guideline-based decision support for pharmacological treatment can improve the quality of hyperlipidemia management, Comput. Methods Prog. Biomed. 97 (3) (2010) 280–285.

[10] N.F. Noy, R.W. Fergerson, M.A. Musen, The knowledge model of Protege-2000: combining interoperability and flexibility, in: 2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France, 2000.

[11] Jess, the Rule Engine for the Java Platform. Available at: http://herzberg.ca.sandia.gov/jess/.

[12] M.K. Goldstein, B.B. Hoffman, R.W. Coleman, M.A. Musen, S.W. Tu, A. Advani, R. Shankar, M. O'Connor, Implementing clinical practice guidelines while taking account of changing evidence: ATHENA DSS, an easily modifiable decision-support system for managing hypertension in primary care, Proc. AMIA Symp. (2000) 300–304.

[13] L.T. Kohn, J.M. Corrigan, M.S. Donaldson, To Err is Human: Building a Safer Health System, National Academy Press, Washington, DC, 1999.

[14] C. Vincent, G. Neale, M. Woloshynowych, Adverse events in British hospitals: preliminary retrospective record review, BMJ 322 (7285) (2001) 517–519.

[15] R.M. Wilson, W.B. Runciman, R.W. Gibberd, B.T. Harrison, L. Newby, J.D. Hamilton, The quality in Australian health care study, Med. J. Aust. 163 (9) (1995) 458–471.

[16] M.A. Musen, S.W. Tu, A.K. Das, Y. Shahar, EON: a component-based approach to automation of protocol-directed therapy, J. Am. Med. Inform. Assoc. 3 (6) (1996) 367–388.

[17] M. Peleg, A.A. Boxwala, O. Ogunyemi, Q. Zeng, S. Tu, R. Lacson, E. Bernstam, N. Ash, P. Mork, L. Ohno-Machado, E.H. Shortliffe, R.A. Greenes, GLIF3: the evolution of a guideline representation format, Proc. AMIA Symp. (2000) 645–659.

[18] C.E. Kahn Jr., An Internet-based ontology editor for medical appropriateness criteria, Comput. Methods Prog. Biomed. 56 (1) (1998) 31–36.

[19] C. Bratsas, V. Koutkias, E. Kaimakamis, P.D. Bamidis, G.I. Pangalos, N. Maglaveras, KnowBaSICS-M: an ontology-based system for semantic management of medical problems and computerised algorithmic solution, Comput. Methods Prog. Biomed. 88 (1) (2007) 39–51.

[20] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns: Elements of Reusable Object-oriented Software, Addison-Wesley Longman Publishing Co., Boston, USA, 1995.

[21] W.S. Jian, C.Y. Hsu, T.H. Hao, H.C. Wen, M.H. Hsu, Y.L. Lee, Y.C. Li, P. Chang, Building a portable data and information interoperability infrastructure-framework for a standard Taiwan Electronic Medical Record Template, Comput. Methods Prog. Biomed. 88 (2007) 102–111.