

A Clinical Rule Editor in an Electronic Medical Record setting: Development, Design, and Implementation

Rachel Regier, MS; Rupali Gurjar, MS; Roberto A. Rocha, MD, PhD

Knowledge Management, Clinical Informatics Research and Development, Partners
HealthCare Systems, Wellesley, MA

Abstract

Clinical decision support (CDS) implemented as part of an electronic medical record (EMR) has a well-documented history of improving patient safety and quality of care; however, the difficulties of keeping CDS up to date have also been documented. At Partners HealthCare, we initially implemented CDS reminders in our 'homegrown' EMR system as 'hard-coded' rules. The challenges of updating existing rules and implementing new rules in the hard-coded state, however, soon made this model unsustainable. After evaluating our needs and requirements for rule creation and maintenance, we designed and created a browser-based rule editor that would decrease turnaround time for logic changes, allowing us to respond to CDS requests more efficiently. We have been able to maintain the older reminder rules with the rule editor, and have added a number of new reminders. Our work to date has confirmed the strengths of the editor, but has also identified a few limitations.

Introduction

There is often a considerable delay between the confirmation of a clinical research finding in the medical literature and the implementation (or incorporation) of that finding into clinical practice.¹ Over the years, healthcare organizations have looked to "computerized clinical decision support" (CDS) to bridge the gap between research and clinician action. More recently, this has been done using computerized clinical decision rules ('reminders'), which have been demonstrated to improve both clinical outcomes for patients and quality of clinician performance.²

The ability to view these reminders on a computer screen, however, does not inherently overcome some of the challenges of knowledge dissemination. Medical knowledge evolves rapidly and it is difficult for a busy practicing clinician to stay up-to-date with the latest clinical evidence and research findings. Consequently, the initial creation of CDS is often the easy part; with knowledge changing so rapidly, even

the best CDS knowledge bases can become outdated unless careful thought is given to the maintenance process.³ This maintenance process is critical to the long-term success of CDS,⁴ and the maintenance effort can be made more efficient through a careful construction of the knowledge base.³

At Partners Healthcare, CDS has long been a priority. In the inpatient setting, reminders were implemented starting in the late 1990s.⁵ In the ambulatory setting, multiple reminders were implemented in 2000 as part of a study intended to primarily assess clinician compliance with guidelines and, secondarily, to assess the impact of reminders on workflow and quality of care.⁶ These reminders were displayed on the patient-specific summary screen in our homegrown outpatient EMR, and the rules were written as part of the system source code ('hard-coded'). The Knowledge Management (KM) team was excited to have these new reminders, as we believed that this CDS would help to disseminate knowledge and to encourage clinician action.

The fact that these reminders were hard-coded, however, prevented the KM team from acting quickly on new research findings to update the existing rules in a timely manner. All changes to the reminders required a software engineer to modify the system source programs, whether the change was adding an additional rule, or making a minor change to an existing rule, such as adding a problem or medication. Significant time, ranging from weeks to months, was required for any change, no matter how minor, as the software engineer's time had to be requested, the programs carefully reviewed, and each modification tied to a specific system release date. In addition, whenever a question arose around the validity of a reminder on a particular patient's chart, the code had to be reviewed by both a software engineer and a knowledge engineer with programming experience to determine if the problem was due to outdated rules, data integrity issues, or programming errors. Due to these limitations, we were unable to update the reminders on a regular basis, and changes to the rules were made infrequently.

As additional reminders were requested, it became increasingly clear that this model was unsustainable, and that it would ultimately hamper our ability to support existing and new CDS. In addition, we felt strongly that it was less than ideal for knowledge engineers to depend on software engineers for the maintenance of CDS, a conclusion also came to by Geissbuhler and Miller.⁷ These realizations led to us to build an editor in our EMR that would consist of reusable logic pieces and a structured format to ensure a consistent look for each reminder.⁸

Methods

To design an editor for the reminders, the KM team identified several important goals: 1) Greater control over the reminder change process; 2) Reduced turnaround time for modifications; and 3) Rules that a non-programmer could easily understand.

After identifying the project goals, we reviewed the existing 39 reminders to uncover the commonalities among them. At the same time, subject matter experts (SMEs), who were also practicing clinicians, were asked to review the reminders and confirm that the logic was correct and appropriate for the targeted patient populations. Since we anticipated an expansion in the scope of the reminders, SMEs in areas not represented by the existing reminders (e.g. pediatrics, chronic renal disease) were consulted to identify potential new reminders, which were also reviewed for commonalities. Fourteen new pediatric reminders were created, and several other preventive healthcare reminders were proposed.

It was apparent that a new governance model needed to be established to ensure that the implemented reminders were representative of Partners' clinical

guidelines and compatible with external guidelines. Checks and balances were built into this model to ensure that only clinically important reminders were implemented, using the criterion that the clinical evidence had to be 'important to remember but easy to forget.'

After ensuring that the existing reminders were up-to-date and after authoring new reminders, the identified commonalities were used to develop a generalized model for reminder activation (Figure 1). This model influenced the design of a rule editor with a graphical user interface on a Cache platform, allowing knowledge engineers to visualize the rules for existing reminders and create the logic for new ones.

The generalized reminder activation model allowed us to break down each reminder into the simplest logic expressions and to model these expressions as 'primitives' that were used repeatedly throughout the existing reminders. When developing the rule editor, we wanted the ability to reuse expressions, so we created a set of 'primitives' that represented all commonly used expressions. One example is the "medication primitive", where the appropriate groups ('subsets') of medications for that reminder can be entered, and specific medications can be included or excluded. This primitive also allows the user to define a medication start date or minimum number of medications to make the primitive true. Another primitive, one of the simplest, is the "age primitive," where an age range can be created for each reminder (Figure 2). A more complicated primitive is used for health maintenance items (e.g. mammography, eye exam, HbA1c). This Health Maintenance primitive provides the ability to look at ranges of results, service dates, and ages at the last service date (Figure 3). Prior to the final design of the rule editor, the

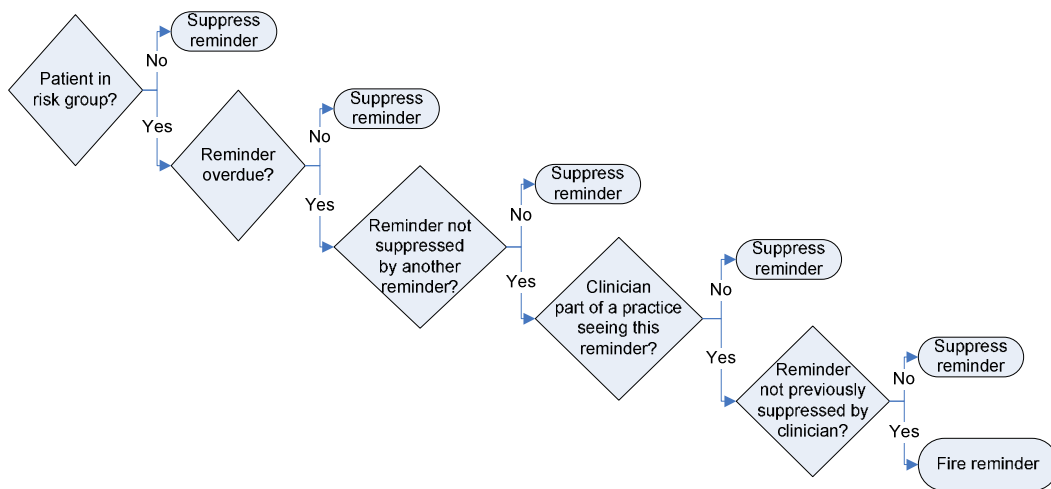


Figure 1. Generalized model for reminder rules.

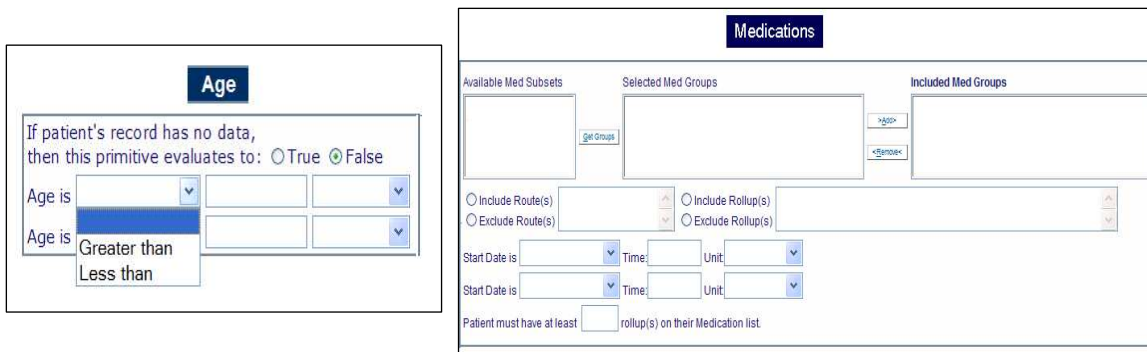


Figure 2. Two examples of rule editor.

rules and primitives were prototyped in a stand-alone database, which allowed us to ensure that the logic pieces being created would be sufficient not only for the existing 39 reminders but also the 14 new ones.

Another requirement for the editor was a testing environment, as we wanted to avoid any erroneous messages and subsequent clinician confusion when reminders were released into the production environment. This was also an important requirement because the process of entering the reminders could not be automated; a knowledge engineer had to author each piece of logic and each string of medication identifiers and problem identifiers. We met this requirement by creating the rule editor in the testing and production environments of our EMR system, and by creating a way to move the reminders from testing to production once they had been validated.

real time (Figure 4). This rule editor was released in April 2007, and at the time 39 existing reminders were implemented, focused on adult preventive care. An additional 14 new reminders were also added at that time, focusing on pediatric preventive care.

Several members of the KM team have access to the rule editor, and each uses it approximately five times a week, typically to address clinician concerns about reminders displaying on patient charts. At this time, access is restricted, and SMEs and other users are not given access, as any accidental change can affect the reminders in real-time. Batches of new rules are added twice a year, and two to three changes are made on a monthly basis. These changes are predominantly minor, with the most frequent changes involving modifications to medications or problems.

The stand-alone database initially used to prototype the original reminders has continued to prove useful, as it serves as a tool for prototyping additional rules and knowledge. It also provides the ability to generate reports, which can be customized to contain any of the reminder logic and change/review documentation available in the database. These reports, which are easily readable by a non-engineer, are often used to facilitate SME panel review of logic or are used by analysts supporting clinics to identify which reminders a practice should see. These reports of the reminders are also published to an online portal whenever the logic is updated. This portal is available to any Partners user or analyst, and is categorized by topic and searchable by keyword.

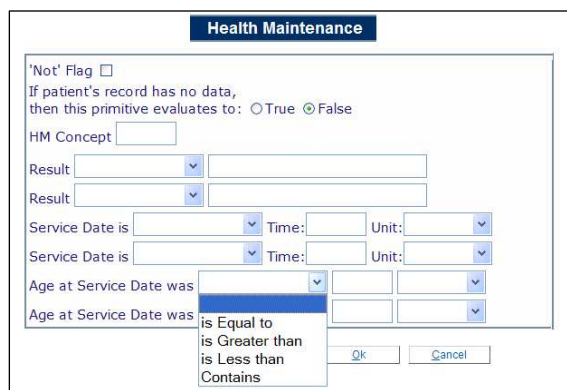


Figure 3. Example of a rule editor primitive: Health Maintenance.

Results

After these careful assessments of our existing and potential needs, and the creation of a prototype, we developed a browser-based rule editor that allowed knowledge engineers to visualize the reminder logic in production and make modifications or additions in

Since the release of the rule editor in April 2007, an additional 60 rules have been added. Nine additional primitives have been created, as new rules have required new content that was not initially anticipated. New functionalities have been added, including the ability to assign new reminders to different sections of the LMR and the ability to sort and search existing reminders.

Figure 4. The browser-based rule editor summary screen for an individual reminder.

The length of the update process has also been dramatically shortened. While in the past a small change, such as the addition of a new medication, could take an average of at least three weeks and would depend on a software engineer, the KM team can now make small changes in one hour, on average. Changes to the content requiring more effort are prioritized against existing work, but are still completed in a matter of days rather than weeks. A new reminder not requiring new functionality can be implemented in a few weeks, including the time needed for establishing the testing and user communication plans.

Discussion

Since the implementation of the rule editor, the KM team has gained the ability to efficiently author and maintain reminders. The team has also found it increasingly important to have a process to keep the reminders up to date. A governance process relying on SMEs has been created in parallel to the rule editor creation and has facilitated content creation and review.

Several advantages to the rule editor have been identified. We have decreased, although not eliminated, the risk for human error by making the reminders visible rather than hard-coded and by limiting the number of people involved in rule construction and modification. Previously, a software engineer, a knowledge engineer, and sometimes a business analyst were involved in rule modification, but now only a trained knowledge engineer is involved with the appropriate testing and validation methods. In addition, we observed a dramatic improvement in the amount of time required to create a new reminder, from several months to several weeks, and have seen a decrease in the time required

to make a minor change, from weeks to days, or even hours for the simplest of changes.

The primary goal of the editor was to decrease reliance on software engineers to maintain the reminders, and we have been successful, at least to some extent. We are no longer dependent on software engineers to maintain individual reminders, although new reminders occasionally require new functionality, which results in additional programming. The initial version of the rule editor simply did not anticipate all of the decision support that would be requested, as new guidelines and research findings have been published and old ones have changed. New insurance pay-for-performance measures have also obligated us to create additional reminders and CDS features. Requests have also been made to display the reminders in various locations of the application, which has required further functionality work effort.

A limitation of our initiative is that we have found a continued dependence on the stand-alone database, as we failed to build a reporting capability into the rule editor. We rely on the database for queries and reports, metadata tagging, and workflow management. We also use this database to track the dates for decisions made and content changed, an important role of any knowledge base maintenance tool.⁷ As a result, knowledge must still be maintained in both the rule editor and the stand-alone database, but knowledge engineers are able to carry out these extra steps as part of the systematic reminder management process.

Another limitation of the rule editor is that it cannot handle complex reminders easily. These reminders must instead be broken down into simpler rules. In some cases, multiple simple rules in the editor are

needed to compose one complex reminder. Based on the available literature and efforts of other groups, rule complexity is a common challenge for rule editors.⁹

With the development of the rule editor, we learned some valuable lessons. First, as mentioned earlier, we have recognized that our dependence on software engineers will never be completely eliminated, because new, unique content is requested on an ongoing basis. Along with this, in some cases, we are still tied to the development cycle as new functionality cannot be added off-cycle, which restricts the release of reminders dependent on the new functionality. Finally, given that much of the knowledge maintenance process has not yet been automated, our team has found a detailed user guide to be of tremendous value, as both a training tool and a reference document.

Our rule editor has come a long way, but remains a work in progress. While some of our design may not be generalizable to other systems, it is our hope that others may benefit from the lessons we have learned. In the future, we would like to add reporting and versioning capabilities to the rule editor itself. This functionality would not be strictly for clinicians' benefit, but would allow us to move away from any dependence on the stand-alone database, and would allow us to more easily identify any issues or inconsistencies within the existing reminders. We also plan to continue adding new functionality as the need arises, and will add new reminders as long as there is a clinical need.

Conclusion

The reminder rule editor has been a useful tool, decreasing our dependence on software engineers and also decreasing the time required to make changes to existing reminders or to add new reminders. Since the rule editor was released in April 2007, we have been able to expand the functionality, along with adding 60 new reminders. All of the previously hard-coded reminders are now in the rule editor and can be updated as guidelines and clinical best-practices change. Additional reminders continue to be requested, and given that some of these will require new functionality from the editor, we expect the functionality of the rule editor to continue to evolve well into the future.

Acknowledgements

We would like to thank the following people for their contributions to this paper: Donald Bugbee, Eric Poon, MD, Steven Morgan, MD, Susan A. Smith, and Gianna Zuccotti, MD.

References

- ¹ Jenders RA, Dasgupta B. Challenges in implementing a knowledge editor for the Arden Syntax: knowledge base maintenance and standardization of database linkages. *Proc AMIA Symp* 2002;355-9.
- ² Shiffman RN, Liaw Y, Brandt CA et al. Computer-based guideline implementation systems: a systematic review of functionality and effectiveness. *JAMIA* 1999;6:104-14.
- ³ Guise DA, Guise NB, Miller RA. Evaluation of long-term maintenance of a large medical knowledge base. *J Am Med Inform Assoc.* 1995 Sep-Oct;2(5):297-306.
- ⁴ Bates DW, Kuperman GJ, Wang S, et al. Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality. *J Am Med Inform Assoc.* 2003 Nov-Dec;10(6):523-30.
- ⁵ Kuperman GJ, Teich JM, Gandhi TK, Bates DW. Patient safety and computerized medication ordering at Brigham and Women's Hospital. *Jt Comm J Qual Improv.* 2001 Oct;27(10):509-21.
- ⁶ Gandhi TK, Sequist TD, Poon EG, et al. Primary care clinician attitudes towards electronic clinical reminders and clinical practice guidelines. *AMIA Annu Symp Proc.* 2003:848.
- ⁷ Geissbuhler A, Miller RA. Distributing knowledge maintenance for clinical decision-support systems: the "knowledge library" model. *Proc AMIA Symp.* 1999:770-4.
- ⁸ Gurjar R, Li Q, Bugbee D, et al. Design and implementation of a clinical rule editor for chronic disease reminders in an electronic medical record. *AMIA Annu Symp Proc.* 2006:936.
- ⁹ Kuperman GJ, Teich JM, Bates DW, McLatchey J, Hoff TG. Representing hospital events as complex conditionals. *Proc Annu Symp Comput Appl Med Care.* 1995:137-41.