

Migrating toward a Next-Generation Clinical Decision Support Application: The BJC HealthCare Experience

Yan Huang^b, MA, Laura A. Noiro^a, BS, Kevin M. Heard^b, BS, Richard M. Reichley^b, RPh,
Wm. Claiborne Dunagan^{a, b}, MD, and Thomas C. Bailey^{a, b}, MD

^aWashington University School of Medicine, Department of Internal Medicine, St. Louis, MO

^bBJC Healthcare, Center for HealthCare Quality and Effectiveness, St. Louis, MO

Abstract

The next-generation model outlined in the AMIA Roadmap for National Action on Clinical Decision Support (CDS) is aimed to optimize the effectiveness of CDS interventions, and to achieve widespread adoption. BJC HealthCare re-engineered its existing CDS system in alignment with the AMIA roadmap and plans to use it for guidance on further enhancements. We present our experience and discuss an incremental approach to migrate towards the next generation of CDS applications from the viewpoint of a healthcare institution. Specifically, a CDS rule engine service with a standards-based rule representation format was built to simplify maintenance and deployment. Rules were separated from execution code and made customizable for multi-facility deployment. Those changes resulted in system improvements in the short term while aligning with long-term strategic objectives.

Introduction

Numerous reports cite the effectiveness of clinical decision support (CDS) systems in improving the quality of health care services^{1, 2}. However, CDS is not widely implemented within the United States. This fact is recognized in the *Roadmap for National Action on Clinical Decision Support* released in June 2006 by the American Medical Informatics Association (AMIA), which recommends a series of activities “to ensure that *optimal, usable and effective* clinical decision support is *widely available* to providers, patients, and individuals *where and when they need it* to make health care decisions.”³

The model for a next-generation CDS application outlined in the AMIA Roadmap suggests that key features such as standardized knowledge representation, scalability and outcome-driven focus would facilitate realization of the full potential and widespread adoption of CDS. The model is a summary of a consensus that emerged from recent studies on guideline modeling⁴, knowledge management⁵, and CDS service frameworks⁶. Since 2005 BJC HealthCare (BJC) has been re-engineering its existing CDS systems with the same objectives in

mind.

BJC HealthCare comprises 13 hospitals and multiple community health locations in the greater St. Louis, southern Illinois and mid-Missouri regions. It had 3,508 staffed beds and 145,584 hospital admissions in 2005. BJC HealthCare has used CDS tools to deliver patient-specific assessments and interventions since the early 1990s. In the last decade, its CDS tools evolved into a collection of individual stand-alone agents with functionalities from dose checking, microbiology surveillance, allergy monitoring, automated adverse event detection, drug interaction detection and guideline adherence monitoring. This expansion of functionalities increased system complexity and, consequently, maintenance costs. Ongoing review and maintenance of CDS rules was difficult because rules were embedded in execution code. Additionally the use of CDS interventions varied from facility to facility and rule customization at the facility level was necessary to facilitate deployment.

We started the re-engineering process based upon lessons from past experience and best practices suggested in the literature. A CDS rule engine service with a standards-based rule representation format was built to simplify maintenance of existing functionality and deployment of new ones. Rules were separated from execution code and were made customizable for multi-facility deployment. In this paper we present the successful implementation of the first part of our multi-phase strategy. At this emerging stage of the next-generation CDS application envisioned in the AMIA Roadmap, it is our hope that this work might provide valuable input for a robust, systematic approach for CDS adoption at the national level. We also discuss how a health care network like BJC HealthCare can benefit from a national CDS initiative and the next steps we may take using the AMIA Roadmap as guidance.

Methods

Design overview

To leverage the existing CDS capabilities at BJC, we started the architectural design of a new CDS application based on best implementation practices recommended in the literature, such as the “ten commandments” for effective CDS⁷. The long-term goal of this project is to build a next-generation CDS application as described in the AMIA Roadmap. We intended to accomplish the following design objectives during the first phase:

- Separate knowledge representation from execution code;
- Encode knowledge in both human and machine-interpretable format;
- Replace standalone rule-based applications with a generic CDS rule engine service that supports both real-time and batch assessments.
- Make CDS rules easily customizable for deployment in any facility within BJC HealthCare.

Knowledge representation format

Given an example rule that *if a patient is an inpatient and has active warfarin drug interaction and an INR lab result that is greater than 3 then alert a pharmacist*, we can see the knowledge representation format of a rule should have at least the following elements:

- a) patient care context (inpatient)
- b) triggering event (INR lab result)
- c) data requirements (active warfarin drug order and any interacting drug orders)
- d) rule parameter (INR threshold = 3)
- e) action specification (alert a pharmacist)
- f) patient data model (a patient data structure that can be referenced by an expression language)
- g) executable format (the if-then decision logic)

We separated all the above elements from execution code so that they are stored and managed externally. This greatly facilitates configuration management and deployment. Here we introduce the implementation details of the patient data model, action specification, and executable format, which are essential to the communication between the rule service engine and clients.

A structured patient data model is the cornerstone of a computer-interpretable knowledge representation format. It is referenced in patient data queries as well as decision criteria evaluation. Observing the best practice for patient data model design suggested by developers for GLIF⁸, SAGE⁹, and SEBASTIAN⁶, we implemented a Virtual Medical Record (vMR) as a simplified view of the HL7 RIM v2.13. We only

selected those objects and attributes that were relevant to our CDS rule set. Because of its object-oriented design, representing vMR in an XML format is straightforward. XPath¹⁰ is the W3C standard expression language for extracting and evaluating XML data. Hence a patient evaluation such as *whether the last LDL of a patient was greater than 100* could be written in XPath syntax as:

```
/vmr/observation [./codingSystem='LOINC' and
./codeValue= '2089-1' ]/[last()] > 100.
```

This expression demonstrates how vMR helps to overcome the heterogeneity of a local clinical information system (CIS) and vocabularies since it is not bound to any institution-specific database field or object attribute.

The action specification is modeled as a subclass of the Act object in HL7 RIM. The CDSAction object has attributes for rule identifier, action, action reason, scheduling setting, and action target such as drug orders or labs. One assessment from the rule engine will return one or more CDSAction messages.

Business Process Execution Language for Web Services (BPEL)¹¹ is an XML-based language used to define an executable business process with formal specifications for logic, state and data handling. We find that it provides the necessary language support for decision reasoning and data handling required by an executable rule format. Here is an example to assign the last LDL of a patient to a variable in BPEL:

```
<assign name="getPatientLDL">
  <copy>
    <from variable="input" part="vmr"
query="/vmr/observation [./codingSystem='LOINC' and
./codeValue= '2089-1' ]/[last()]/value/quantity"/>
    <to variable="lastLDL"/>
  </copy>
</assign>
```

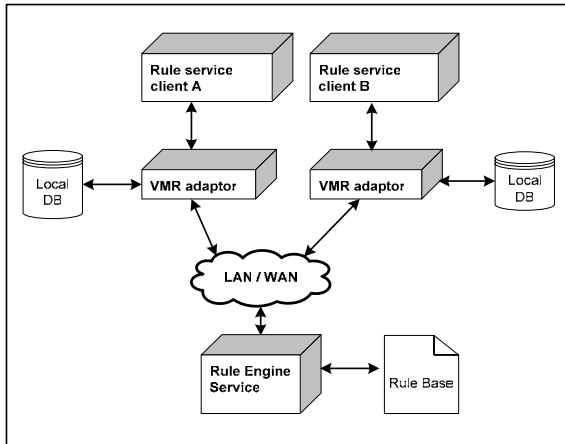
Below is an example of the logic to alert if a patient's last LDL is greater than 100:

```
<switch name="checkLDL">
  <case condition=
    " bpws:getVariableData('lastLDL') > 100" >
    <throw faultName="alert"/>
  </case>
</switch>
```

Service-oriented architecture

Service-oriented architecture (SOA) uses loosely coupled services that can be invoked across networks without knowledge of their underlying implementations. The loose coupling principle could potentially improve software reusability and adaptability to changing business needs. Encouraged by the widespread implementation of SOA in other

Figure 1. System deployment architecture



industries, the idea of adopting SOA in CDS applications is gaining momentum. The SOA principle was applied to our rule engine service. Figure 1 displays the deployment scenario for the optimal use of the rule engine service.

A *rule service client* is a CDS application that calls the rule engine service as a remote subroutine to perform rule-based patient assessments across a network. The client should provide its own implementation of a *vMR adaptor* mapping the local patient data model to the supported vMR format. The service invocation as well as the input and output message formats should also be known to the client. We use the Java Rule Engine API (JSR-94)¹², which is a standard specification endorsed by the Java Community Process, to wrap around the BPEL interface and serve as a client-side façade. Once the service interface is established, the implementation details of either the rule engine or the client become irrelevant behind the façade.

Here is an example use case to monitor the care of diabetes patients using the rule engine service. An automated process is monitoring new patient

admissions. Once a patient is identified as diabetic from previous medical records, the monitoring process will invoke the rule engine for diabetes guideline compliance checking. The rule engine will pass back data requirements for each assessment request. The monitoring process will gather the required patient-related data, including lab, drug orders and/or diagnosis codes, and generate a vMR through the vMR adaptor. The vMR is then sent to the rule engine and assessed against the encoded diabetes guideline. The assessment result is one or more actions if the guideline is not satisfied. Each implementation of the rule engine client must determine how to handle the returned actions.

Configuration management for multi-facility deployment

We knew from our past experience that the deployment of CDS functionalities in a facility would fail if it could not meet the unique needs of the facility. As a major benefit of separating knowledge from execution, the configuration management for multi-facility deployment was made flexible enough to overcome the variance in patient population, services, and staff workload among facilities. A rule can be easily activated or inactivated for individual facilities. A rule can be configured to run in either real-time or batch mode to fit into different workflows. Rule parameters such as critical lab thresholds or drug duration times can be customized at the facility level to support different rule sensitivities. The priority of rules can also be adjusted to reflect its clinical impact in a given patient care context.

Results

The CDS rule engine service was developed over a 2-year period and took approximately 2,200 hours of work. It was initially deployed with 25 rules in April 2006. By March 2007, the number of rules has increased to 87. Table 1 shows the count of rules by

Table 1. CDS rule count by category

Category	Sample rules	Count
Dosage/Contraindication *	Cefazolin dosage checking for bariatric patients Propoxyphene/acetaminophen contraindication	16
Dose Duration	Maximum duration of ketorolac therapy exceeded Antibiotic duration exceeds 24 hours post-op	3
Adverse Drug Event	Digoxin drug level > 2 Elevated INR results with warfarin drug interactions	44
Clinical Action Alert	No INR test for patient with warfarin drug interaction Precautions for patient on inhaled ribavirin	21
Guideline Adherence	Diabetes, Heart Failure, Acute Myocardial Infarction	3

* This is an add-on to the existing dosage checking program using the commercial Multum knowledge base.

category. Alerts/reminders are sent to clinicians in 8 of 13 hospitals within the BJC HealthCare system including a home health facility. The rule engine service handles more than 30,000 assessment requests per day. Each patient assessment will take 200~700 milliseconds depending on the complexity of the rule.

The rule engine service client is an event-driven program running against a centralized clinical data repository. It replaced three standalone applications for complex dose checking, adverse event detection, and guideline compliance checking.

Discussion / Conclusion

Local perspective of the CDS national roadmap

CDS interventions have been used as an effective tool to improve patient safety and the quality of care at BJC HealthCare. However, we were striving to optimize system performance and maximize utilization of the CDS system at all facilities while the latest CDS innovations were waiting to be applied. Our experience mirrored the national CDS experience addressed in the AMIA roadmap. From the local perspective of a health care organization, the benefit of the AMIA roadmap can be interpreted in

two ways. First of all, for our internal development the AMIA roadmap confirmed our observations for ways to improve the existing system. Moreover, it provided a comprehensive conceptual framework that helped us to form a long-term strategy. Secondly, we expect the national initiative to have a positive impact on the external environment. For instance, it may accelerate standardization processes in the medical informatics field. To avoid “reinventing the wheel” and reduce development costs, a standardized patient data model, knowledge representation, and CDS service interface are of the greatest value based on our experience.

An incremental approach

Migrating towards the next-generation CDS application envisioned in the AMIA roadmap is a long journey that needs strategic planning. The AMIA Roadmap identified three pillars for realizing the promise of CDS and six corresponding strategic objectives. We believe that a practical approach is to start with system changes that are based on existing resources and have a visible impact in the short term while aligning with long-term objectives. Table 2 shows our attempted multi-phase plan. The accomplishment of the first phase is presented in this paper and we are currently working on the second

Table 2. An incremental approach toward next-generation CDS application

Pillar		
Objective	Best Knowledge available when needed	
	<i>A. Represent clinical knowledge and CDS intervention in standardized format</i>	<i>B. Collect, organize, and distribute clinical knowledge and CDS interventions</i>
Phase 1	Separate knowledge from execution code. Encode knowledge in both human and machine-interpretable representation format.	Build an externalized rule base in which each rule can be individually reviewed and distributed.
Phase 2	Replace local coding system with standard coding system.	Form an enterprise-wide knowledge governance committee
Phase 3	Knowledge is created, validated and maintained directly by knowledge expert.	Establish the enterprise-wide process for authoring, validating, and distributing CDS knowledge
Pillar		
Objective	High adoption and effective use	
	<i>C. Address policy/legal/financial barriers and create additional support and enablers</i>	<i>D. Improve clinical adoption and usage of CDS interventions</i>
Phase 1	Consolidate multiple standalone applications into a generic CDS rule engine service.	Build the CDS rule engine as a service component
Phase 2	Set up security policies and HIPAA compliance auditing.	Integrate with other internal information system components such as terminology server and clinical physician order entry system.
Phase 3		Integrate with external knowledge services if available.
Pillar		
Objective	Continuous improvement of knowledge and CDS intervention	
	<i>E. Assess and refine the national experience with CDS</i>	<i>F. Advance care-guiding knowledge</i>
Phase 1	Collect and store metrics of the effectiveness of the CDS intervention	
Phase 2	Generate and distribute summary reports on the effectiveness of the CDS intervention.	Route user feedback into knowledge refinement.
Phase 3	Adopt best practice collected through the AMIA roadmap initiative.	Set up the enterprise-wide process for authoring, validating, and distributing CDS knowledge

phase.

The CDS rule engine service was chosen as the first installment of a CDS service framework because it consolidates functionalities of existing tools and facilitates rapid deployment of new ones. It also lays the groundwork for knowledge management by separating knowledge representation from the execution code. New services such as a terminology server, user authentication or an ontology server can be added to the system in a plug-and-play way when they are needed. While not all of the desired standards are available, we based our local implementation on *de facto* standards and ready-to-use technologies such as HL7 RIM, BPEL, and XPATH whenever possible and appropriate. Furthermore the architecture is designed to minimize future efforts for adopting standards when they become available. Placeholders for the patient data model, expression language, knowledge representation and execution engine were created so that one implementation can be replaced by another with minimal impact to the system. For example, the BPEL framework supports multiple expression languages. When a production strength parser for the HL7 endorsed object-oriented Guideline Expression Language (GELLO) is released, it can be utilized by this system through changes to the BPEL configuration.

In the second phase, we plan to use the intervention responses we have collected to find a systematic way for routing user feedback into knowledge and system refinement. It closes the feedback loop for CDS rule performance and promotes continuous improvements through corrective actions. In addition, we will pursue proactive user participation in the knowledge authoring and maintenance process, which will lead to the use of authoring tools and a way for all to inspect the knowledge represented in BPEL.

Supported in part by NHLBI Grant R01 HL70790-01A1.

REFERENCES

1. Bates DW, Gawande AA. Improving Safety with information technology. *N Engl J Med.* Jun 19 2003;348(25):2526-2534.

2. Chaudry B, Wang J, Wu S, et al. Systematic review: impact of health information technology on quality, efficiency, and costs of medical care. *Ann Int Med.* 2006 May 16; 144(10):742-752.

3. Osheroff JA, Teich JM, Middleton B, Steen EB, Wright A, Detmer DE. A roadmap for national action on clinical decision support. Bethesda, MD: American Medical Informatics Association; 2006. Available at <http://www.amia.org/inside/initiatives/cds/>. Accessed 1/20/2007.
4. Peleg M, Tu S, Bury J, Ciccarese P, Fox J, et al. Comparing computer-interpretable guideline models: a case-study approach. *J Am Med Inform Assoc.* 2003; 10(1):52-68
5. Greenes RA, Sordo M, Zaccagnini D, Meyer M, Kuperman GJ. Design of a standards-based external rules engine for decision support in a variety of application contexts: report of a feasibility study at Partners HealthCare System. *Medinfo.* 2004;2004:611-615.
6. Kawamoto K, Lobach DF. Design, implementation, use, and preliminary evaluation of SEBASTIAN, a standards-based web service for clinical decision support. *Proc AMIA Symp.* 2005;380-384.
7. 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.* Nov-Dec 2003;10(6):523-530.
8. Johnson PD, Tu WS, Musen MA, Purves I. A virtual medical record for guideline-based decision support. *Proc AMIA Symp.* 2001;294-298.
9. Ram P, Berg D, Tu S, Mansfield G, Ye Q, Abarbanel R. Executing clinical practice guidelines using the SAGE execution engine. *Medinfo.* 2004;2004:251-255.
10. XPath, <http://www.w3.org/TR/xpath>.
11. Business Process Execution Language for Web Services, version 1.1. Available at: <ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf>.
12. Java Community Process. JSR 94: Java™ Rule Engine API. Available at: <http://jcp.org/about/Java/communityprocess/final/jsr094/index.html>.