A highly scalable, interoperable clinical decision support service

Howard S Goldberg,[1,2] Marilyn D Paterno,[1,2] Beatriz H Rocha,[1,2] Molly Schaeffer,[3] Adam Wright,[1,2] Jessica L Erickson,[1,3] Blackford Middleton[4]

[1]Division of General Internal Medicine and Primary Care, Brigham and Women's Hospital, Boston, Massachusetts, USA
[2]Harvard Medical School, Boston, Massachusetts, USA
[3]Information Systems, Partners HealthCare, Boston, Massachusetts, USA
[4]Department of Biomedical Informatics and the Informatics Center, Vanderbilt University Medical Center, Nashville, Tennessee, USA

Correspondence to
Dr Howard S Goldberg,
Partners HealthCare System,
93 Worcester Street, Wellesley,
MA 02481, USA;
hgoldberg@partners.org

Received 6 May 2013
Revised 12 June 2013
Accepted 14 June 2013
Published Online First
4 July 2013

To cite: Goldberg HS, Paterno MD, Rocha BH, *et al. J Am Med Inform Assoc* 2014;**21**:e55–e62.

ABSTRACT

Objective To create a clinical decision support (CDS) system that is shareable across healthcare delivery systems and settings over large geographic regions.

Materials and methods The enterprise clinical rules service (ECRS) realizes nine design principles through a series of enterprise java beans and leverages off-the-shelf rules management systems in order to provide consistent, maintainable, and scalable decision support in a variety of settings.

Results The ECRS is deployed at Partners HealthCare System (PHS) and is in use for a series of trials by members of the CDS consortium, including internally developed systems at PHS, the Regenstrief Institute, and vendor-based systems deployed at locations in Oregon and New Jersey. Performance measures indicate that the ECRS provides sub-second response time when measured apart from services required to retrieve data and assemble the continuity of care document used as input.

Discussion We consider related work, design decisions, comparisons with emerging national standards, and discuss uses and limitations of the ECRS.

Conclusions ECRS design, implementation, and use in CDS consortium trials indicate that it provides the flexibility and modularity needed for broad use and performs adequately. Future work will investigate additional CDS patterns, alternative methods of data passing, and further optimizations in ECRS performance.

INTRODUCTION

Clinical decision support (CDS) technology has been demonstrated to improve the quality and safety of patient care, and is believed to be an integral component in improving patient health outcomes.[1–3] Notably, CDS has shown to encourage adherence to guidelines for prevention and treatment, as well as reduce medication errors.[4–7] Meaningful use legislation has provided further incentive to the healthcare community to incorporate mechanisms into their institutional electronic health records to standardize CDS activities.[8 9]

Due to the high costs of local CDS implementation and associated knowledge maintenance, service-oriented architecture models have begun to emerge as a viable approach to sharing computer-based decision support.[10–12] In response to the need for scalable and shareable CDS, investigators from Brigham and Women's Hospital (BWH), Harvard Medical School, and Partners HealthCare System (PHS) formed the clinical decision support consortium (CDSC), whose goal is 'to assess, define, demonstrate, and evaluate best practices for knowledge management and CDS in healthcare information technology at scale'.[13 14] CDSC investigators have been researching the feasibility and practicality of a service-oriented approach to CDS, and are demonstrating its capabilities to provide recommendations to CDSC members and industry stakeholders.

OBJECTIVE

The goal of the enterprise clinical rules service (ECRS) (table 1) is to provide a single, logical service that can replace innumerable discrete decision support modules, resulting in:

- Reduced clinical variation—by sharing common logic through a central service, and by recommending best clinical practices and policies from a single source.
- Increased agility—by standardizing architectures and application programing interfaces (API), and by leveraging expressive, off-the-shelf rules management systems to speed the development of new and increasingly complex clinical rules.
- Lowered maintenance costs—by sharing and re-using clinical rules specified in English-like languages, and by replacing rules hard-coded in multiple programing languages that are inaccessible to knowledge engineers.

As a demonstration of the feasibility of this approach, the ECRS is being tested and has been implemented in four distinct, point-of-care decision support projects (table 1):

1. CDSC guidelines: a set of reminders related to the management of diabetes, coronary artery disease, and hypertension, called from four disparate ambulatory systems at PHS in Massachusetts (self-developed), Wishard Health Services (Wishard) in Indiana (self-developed by the Regenstrief Institute), WVP Health Authority in Oregon (NextGen), and University of Medicine and Dentistry of New Jersey (GE Centricity).[14 15]
2. Centers for Disease Control and Prevention adult and pediatric immunization scheduling recommendations:[16] These will be deployed operationally at PHS in 2013, and are under consideration for use by other consortium members.
3. CT imaging recommendations for pediatric traumatic brain injury patients:[17] This research study of the pediatric emergency care applied research network will provide real-time decision support to emergency room clinicians at two of 10 pediatric emergency centers using the Epic electronic health record.
4. Integration of external decision support with the substitutable medical applications, reusable technologies (SMART) platform under development at Boston Children's Hospital, including a

Table 1 Acronyms for terms related to the enterprise clinical rules service (ECRS)

Acronym	Term
BRMS	Business rules management system
CCD	Continuity of care document
CDSC	Clinical decision support consortium
ECRS	Enterprise clinical rules service
ODM	IBM operational decision manager
PF	Patient factory
PIM	Patient information model
RES	Rules execution service

repurposing of the previously described immunization scheduling rules.[18]

BACKGROUND AND SIGNIFICANCE

In 1993, researchers at BWH implemented one of the first successful computerized provider order entry (CPOE) systems, with a key design feature to 'support order feedback and alerts', demonstrating the usefulness of CDS in improving care efficiency, care quality, and patient safety.[19] Synchronous and asynchronous decision support introduced into the Brigham integrated computer system over the next dozen years fell into one of three general types: (1) rules-based CDS defined and run by a clinical alerting system; (2) API calls passing input data to modules that performed specific CDS, ie, drug–drug interaction checking; and (3) individual program code, frequently developed for a single research study. The most flexible of these approaches, the clinical alerting system or 'event engine,' is a rules-based system that includes event processing, a workflow engine, logic templates, rules, notification, and alert presentation with actionable recommendations.

The event engine provides reusable code modules for fetching data, evaluating rule logic, notifying recipients, building alert presentations, logging results, and linking recommendations to actions in the CPOE system.[20 21] However, the event engine lacks the capability for the forward chaining of rules; it can only evaluate relatively simple rules one at a time. Although its modularity provides the ability to create many different rules, the system has significant dependencies on BWH data structures and databases; thus, limiting its wider use throughout PHS.

In contemplating a central decision support service (DSS) to replace disparate CDS modules throughout PHS, we hypothesized that contemporary business rules management systems (BRMS) would be expressive enough to implement clinical rules, and the required number of rule patterns would be sufficiently limited in order to warrant a single service. In previously described work, we decoupled backend decision support from existing applications in a series of prototypes using ILOG JRules (now part of IBM operational decision manager (ODM)), and were able to reproduce similar expressivity and attain adequate performance.[22] We subsequently conducted an extensive analysis of the PHS rule library[23]—7210 rules, comprising 181 rule types—and found only 42 taxa among four functional dimensions: trigger, input data elements, interventions, and offered choices—the set of actions 'offered' to the system. Based on this finding, we concluded 'that a very large amount of decision support can be accomplished with a fairly small number of functional constructs across a finite set of categories'.[24]

MATERIALS AND METHODS

Design objectives

We identified the following nine design goals for the ECRS from our earlier CDS development and studies:

1. Establish a single logical service to provide CDS.
   The ECRS should provide a single set of services as an entry point for all decision support. While there is a single logical service, the service may be physically deployed as a single instance or partitioned into multiple configurations to meet clinical business requirements.
2. Standardize the input and output requirements for CDS.
   The ECRS exposes a series of service contracts for client applications with the format dependent on the nature of the rule pattern being executed. These contracts specify the formats and types of data required for input as well as the variety of attributes that may be returned as part of a recommendation. Standardization of required inputs and outputs enforces a discipline for adding decision support to clinical applications.
3. Facilitate interoperability with external consumers through the use of healthcare standards.
   Opportunities may include accepting input in standardized structure/content formats, such as the continuity of care document (CCD),[25] or adopting standard interface contracts, such as those being developed by HL7 and the ONC Health eDecisions effort.[26]
4. Support multiple rule execution patterns.
   The design of the ECRS should not limit the kinds of rule patterns that the service might execute. Rule execution patterns come in many varieties; examples include stateless or stateful interactions and point-of-care interactions versus long-running guidelines. Complex rule patterns may include decisions that require a rule set to be executed in order to determine the rules necessary to compute an ultimate recommendation, or rule sets that dynamically chain together in order to create a final recommendation.
5. Maintain separation between data inputs and underlying inference models.
   The design of the ECRS should shield its underlying inference models from service consumers in order to avoid maintenance obligations. A variety of model implementations are possible in order to support inference by a rule execution engine. For example, relationships between entities, such as a patient and her problems, may be modeled by alternative methods in a programing language, such as by aggregate collections or by modeling the associations as objects themselves.[27] However, if the underlying inference model is tightly coupled to the service input, applications will be required to update service calls every time changes are made to the inference model.
6. Leave the presentation of recommendations to client systems.
   The recommendations returned by the ECRS should be usable in many different contexts by many different systems. Recommendations should be declarative in an application-interpretable format and not prescribe how the recommendations are to be presented to an end user. A single decision support rule set may be used to present recommendations back to patients or providers, may be called by text or graphic-based systems, or may be used to provide a textual response, coded response, or a list of choices for the user to specify a further action.
7. Support highly scalable deployments.
   The ECRS has been implemented to support the high volume of real-time CDS transactions that occur daily in a

typical large healthcare enterprise. Similarly, given its interoperable interfaces, it may be desirable to scale the ECRS to meet the needs of a state or other geographically large region. The design of the ECRS should scale gracefully with the addition of processing nodes, and should provide fault-tolerant behavior required of mission-critical production services.

8. Emphasize the creation and maintenance of decision support content by knowledge engineers, thus minimizing the need for software engineers.
   A major non-functional requirement of the ECRS design is to minimize the need for software engineers in CDS content development and limit their efforts to development of the runtime framework. In earlier work, in which we created a large set of rules for a SmartForm application, we found that using software engineers to translate detailed content specifications into executable rules resulted in a high number of translation errors, prolonged content development time, and rule content that was not understandable by domain experts.[28]
9. Leverage off-the-shelf rules management systems.
   In the 20 years since the introduction of CPOE and CDS at BWH, the field of general-purpose production rules systems has matured. There are multiple vended systems as well as open-source implementations that are used across a wide variety of industries. Such systems generally provide English-like rules languages with a rich set of operators, content management components, and scalable rule execution components. There are no barriers to re-expressing healthcare-specific expression languages such as GELLO[29] on top of these general-purpose systems. The availability of such commodity components can be leveraged to reduce development time for runtime frameworks such as the ECRS.

Performance evaluation

We evaluated the baseline performance of the ECRS in two ways: examining component performance under increasing load, and retrospective analysis of component performance from trial data. Using LoadRunner (Hewlett-Packard Inc.) load testing software, we made sequential calls to the ECRS from one to five concurrent clients, and then five to 30 concurrent clients, increasing by five clients at a time. Each client made continuous, sequential calls to the ECRS, and we ran each interval for 5 min to reach steady state. We replaced the data-fetching cycle with a static CCD so that data remained local to the ECRS, which eliminated dependencies on factors outside the control of the decision support system. We ran the tests on a quiescent network to eliminate any effect of external network traffic.

In order to evaluate ECRS performance during the CDSC trials, we reviewed data from the 1-year period, between 1 September 2011 and 31 August 2012, consisting of transactions from ambulatory clinics at PHS in Massachusetts and Wishard Health Services in Indiana. At PHS, decision support transactions occurred in real time, with in-line CCD creation, and were triggered by opening of the chart. Transactions were timed-out at 3 s and backstopped by native decision support. At Wishard, decision support transactions occurred in near real time, with transactions triggered by patient registration. We instrumented ECRS components to capture time stamps at runtime, record the times used for instantiating and populating patient objects, and execute downstream service calls used by ECRS, including classification services and executing rules. We excluded times associated with data retrieval and document assembly necessary for CCD creation because they occurred externally to the DSS.

RESULTS

System description

The ECRS is implemented in enterprise java as a series of stateless enterprise java beans, deployed to an application server such as Red Hat JBOSS or IBM WebSphere. Each bean is delegated a set of responsibilities that are further described in the following sections (figure 1).

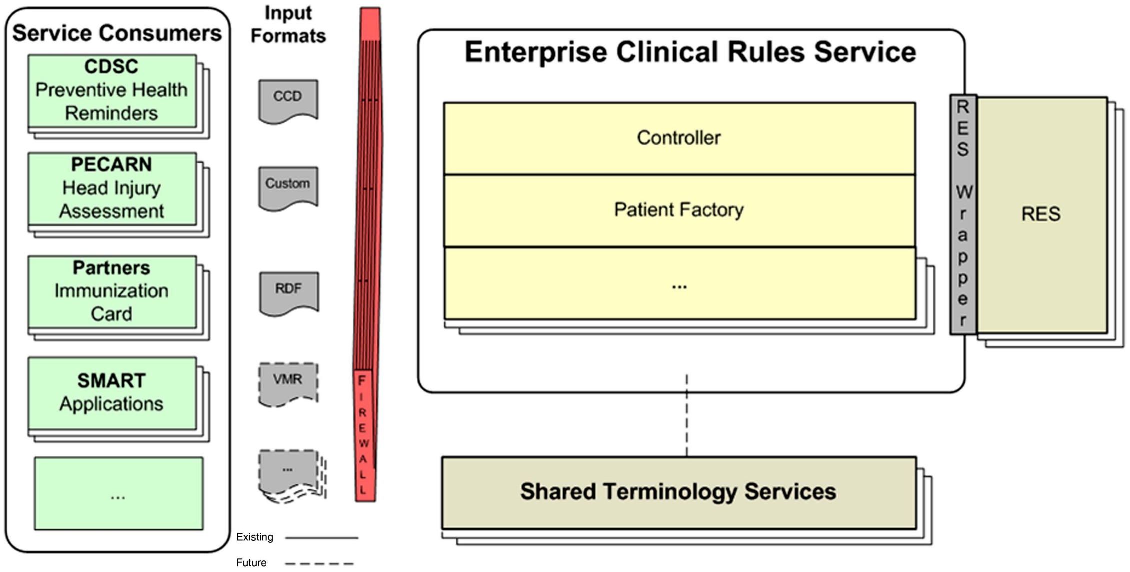

Figure 1 Service Consumers (CDSC, clinical decision support consortium; PECARN, pediatric emergency care applied research network; SMART, substitutable medical application reference technology), input formats (CCD, continuity of care document; RDF, resource description framework; VMR, virtual medical record), and block architecture of the enterprise clinical rules service (ECRS), a modular decision support service. ECRS modules include a controller, a patient factory, which transforms input data into an inference model, a set of shared terminology services for translation and classification, and a backend rules execution service (RES). The RES wrapper abstracts the connection to the RES, allowing different rules engines to serve as the decision processor.

ECRS controller

The ECRS controller provides the entry point to the service and is responsible for managing the control of flow in order to compute a decision for any particular request. The ECRS currently implements a control of flow for computing stateless decision support, but is designed to incorporate additional control flows in a modular way. For example, in a stateful decision, the ECRS controller would need to locate and pass data to the appropriate, previously instantiated rule engine, instead of computing a new decision de novo. The decision to implement a particular control of flow is determined by meta-data associated with the rule set to be executed.

Patient factory

The patient factory (PF) is responsible for the instantiation, population, and elaboration of the inference model evaluated by the rules execution service (RES). The PF provides a modular series of data adaptors and is designed to accept data in a variety of formats. For example, in order to support ECRS consumers outside of PHS, a CCD derivative may be an appropriate form of input, while PHS users may provide data with types native to PHS. Once populated with data, the inference model may undergo further processing, including classification or normalization of data into higher-level categories or the translation of data from one terminology to another.

Rules execution service

The RES manages and provides instances of rules engines—the basic decision-making computational units for the ECRS. The ECRS is designed to maintain high throughput for the RES. Notably, the RES is a scarce resource and potential bottleneck in this architecture; there is finite computing capacity available to the RES, there may be limits according to licensing restrictions, etc. The current implementation of the ECRS avoids external service calls out of the RES because the latency of these calls may exceed decision-making time by an order of magnitude. Our design goal was to maintain an adequate pool of available rules engines and to minimize the time that the rules engine is 'checked-out' to compute a decision.

RES wrapper

The ECRS is designed to be agnostic to the actual RES responsible for decision-making computations. The ECRS provides an abstraction called the RES wrapper that provides a common interface for a RES. This model provides flexibility as the ECRS may be configured with one or more RES. For example, one decision support project may require a high-performance vended RES, while another project may require collaborative decision content development, and may be best suited to the use of an open-source RES. The current implementation of the ECRS contains an implementation for ODM.

Shared external services

Rule evaluation relies on the elaboration of data, including problems, medications, and allergies in order to normalize primary sources into higher-order classes. The ECRS uses external classifiers to derive problem, medication, and allergy class relationships from primary data. Because service calls are generally expensive during the rule evaluation process, classification is performed as a post-processing step once the PF has assembled the primary data.

Information models

Inference model

The patient information model (PIM) is a canonical model, and the data used by the DSS is instantiated into the PIM. The PIM supports only elements commonly used in CDS and is not intended to be a comprehensive data model. The PIM was designed to be easily extensible for the purpose of addressing future needs. Data types represented in the PIM are listed in box 1. Targets are goals assigned to a specific patient; for example, the blood pressure target for the patient. Clinical states are classifications inferred by the rules from interpretations of the patient data. Clinical states may be simple class derivations from external classifiers, for example, 'myocardial infarction' derived from 'anterior wall myocardial infarction', or may represent more complex computations, for example, 'chronic kidney disease stage 3' derived from 'chronic renal insufficiency' in combination with a creatinine clearance calculation. The discovery of a patient's clinical states through execution of clinical state rules, as a first step in a rule operation, allows the PIM to be updated with these states; these states then become available to the engine that is executing the decision support.

Output: action model

The output from ECRS is an XML-based recommendation that can contain many action requests. There are eight different types of action requests listed in table 2. The information returned to a client application will generally include multiple action requests; for example, a message about a care reminder, a request for new procedures, a medication order, or a request for additional information. Recommendations may be designed to support multiple contexts; for example, provide messages for both clinicians and patients. Recommendations are structured data that can be further evaluated by a rule set. The structure of the recommendation is designed to allow the receiving application to construct actionable recommendations for the end-user.

Security

The ECRS is hosted inside the PHS firewall and conforms to PHS security requirements using soap-encoded XML payloads. Security requirements occur at multiple levels.

Box 1 Inference model types

Patient information model (PIM) data types
- Allergies
- Clinical states
- Demographics
- Encounters
- Family history
- Immunizations
- Laboratory results
- Medications
- Physical examinations
- Problems
- Procedures
- Social history
- Systems review
- Targets
- Vital signs

Table 2 Action model types

Action request types	Description	Examples
Encounter	Request for a future encounter to be scheduled	An ophthalmology referral for a patient with diabetes
Knowledge asset	Links to content that can be directed to a clinician or the patient	A link to a website with patient education material
Message	Message (coded or text) directed to a clinician or the patient	A textual reminder
Observation	Request for data collection	A charting of a new blood pressure measurement
Procedure	Request for a procedure to be performed	A foot examination for a diabetes patient
Substance administration	Request for a medication to be administered to the patient	A new medication order or an immunization suggestion
Supply	Request for supply material to be provided to the patient	Oxygen therapy
Clinical state	Classifications inferred by rules from interpreting patient data	An inference of 'chronic kidney disease stage 3' from the problem 'chronic renal insufficiency' and a calculated creatinine clearance

1. Transport level, through HTTPS transport protocol, plus mutual SSL through the use of digital certificates.
2. Authentication, through a limited number of approved security token profiles;
3. Message level, through use of an XML digital signature.

Current PHS deployment

The ECRS is deployed on two clusters, each consisting of two HP Proliant DL380 G5 servers, with four dual core Xeon processors and 3.25 gigabytes of memory. JBOSS application server 4.23 is run in clustered mode on Windows server 2003. The ECRS currently uses ODM V7; one of the clusters is dedicated to the RES components.

General performance

Results for the performance benchmark from one to 30 concurrent users are shown in figures 2 and 3 with classification services in either a non-caching or caching configuration. In the non-caching configuration, execution times for the overall ECRS ranged from 654 ms to 2851 ms, with performance of individual components highly correlated to the performance of classification services. Execution times for the RES remained relatively constant at 49 ms. In the caching configuration, execution times for the overall ECRS ranged from 220 ms to 687 ms. The rule execution service averaged 50 ms, the classification service averaged 4 ms, and the PF service averaged 52 ms. For the peak load, the non-caching configuration performance across 30 concurrent users was 9.7 transactions per second while the caching configuration performance was 28.4 transactions per second (see figure 4).

CDSC trial performance

During the full-year CDSC trial, 693 214 total calls were made to the ECRS (PHS, 678 824; Wishard, 14 390) with an average of 2205/day on weekdays and 306/day on weekends and holidays. Timings were unavailable for 258 347 total failed calls (PHS, CCD creation time exceeding 3-s creation threshold 255 850; PHS, unclassified 2365; Wishard, unclassified 132). Of the remaining 434 867 completed calls, 97% (420 609) were made from PHS and 3% (14 258) were made from Wishard. During the time period studied, the average total round trip was 758 ms. Average time used by the PF was 556 ms, classification services used 173 ms, and rule execution (RES) used 41 ms. PF results included time spent by classification services. Figure 5 shows these averages by month over the year. A change at the beginning of June to flush the classification cache every hour versus every 24 h resulted in longer data preparation times for the PF, as seen in the graph.

DISCUSSION

We have described the architecture and our initial experiences with a centralized, web service-based CDS service. The ECRS is characterized by a modular architecture, an extensible set of data adaptors facilitating the use of differing input standards, and a decision core intended to leverage a third-party BRMS. The modular architecture is designed to support scalability through contemporary clustering techniques.

Related work

Other research teams have investigated service-oriented decision support as an exploration of modular CDS abstracted from

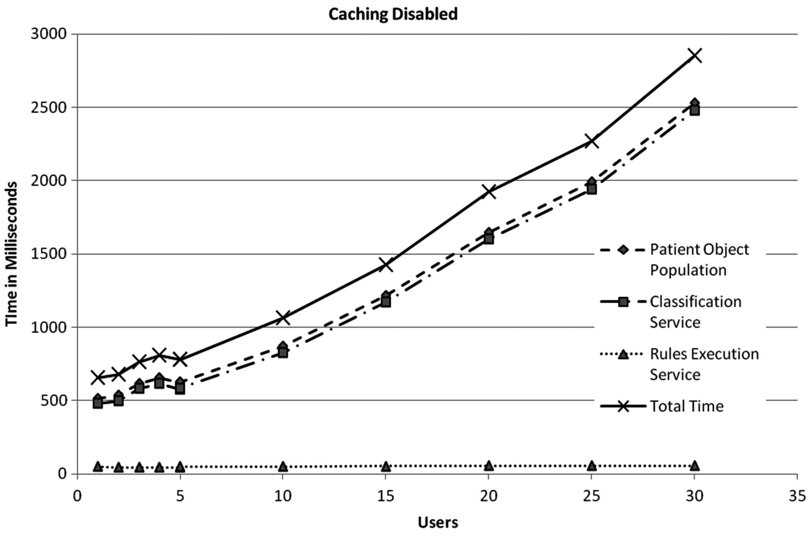

Figure 2 Enterprise clinical rules service module and overall performance with increasing concurrent users, caching of classifications for problems, medications, and allergies disabled.

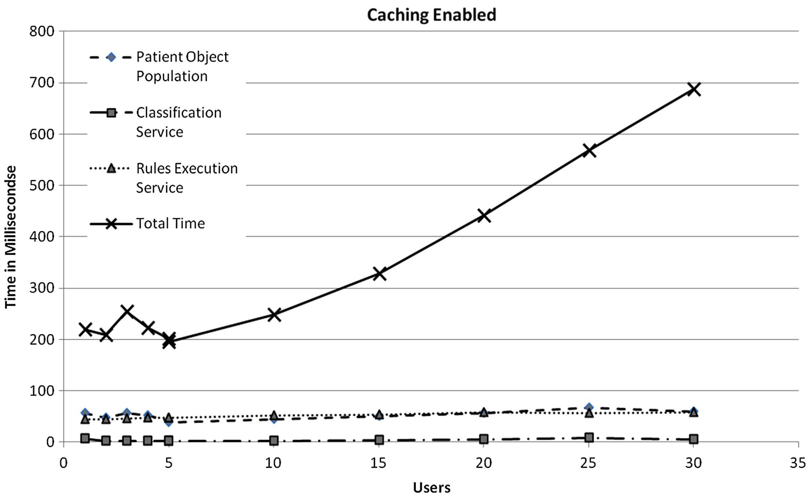

Figure 3 Enterprise clinical rules service module and overall performance with increasing concurrent users, caching of classifications for problems, medications, and allergies enabled.

clinical applications. SAGE, developed by a consortium of academic medical centers, focused on the development of a standards-based guideline infrastructure that could be integrated into the workflow of heterogeneous clinical information systems.[11] Kawamoto and Lobach [30] developed SEBASTIAN, a web service-based system providing synchronous CDS, featuring 'executable knowledge modules', XML-based documents that describe CDS data requirements, patient-specific conclusions, and the logic necessary to reach a conclusion. Wright and Sittig[12] developed SANDS, a decision support model designed to function through two interface facets, a patient data interface and a decision support interface. These services have several key aspects in common with our present study: they support API, attempt to utilize standard terminologies, share data with a decision engine via a data-normalizing information model, and transmit XML-based recommendations to client systems.

Performance

Despite the very acceptable performance of the ECRS in bench-testing and during the trial period, the PHS CCD factory failed to produce a CCD in less than 3 s for approximately 38% of attempted transactions in support of real-time interactions. These failures could be attributed to multiple causes including large datasets, suboptimal data retrieval services, and noisy networks. Anecdotally, our co-investigators noted multi-second creation times on average for CCD, illustrating that data aggregation and retrieval strategies must be carefully considered for applications requiring very low latency, real-time decision support. In our other ongoing evaluations, we are evaluating performance with direct data passing instead of in-line CCD creation. Before production deployment and use, rules undergo both extensive unit testing within the ECRS and end-to-end testing with each consumer. While this trial did not collect data on the correctness of the returned recommendations, to our knowledge, the ECRS produced the expected output to all given sets of inputs.

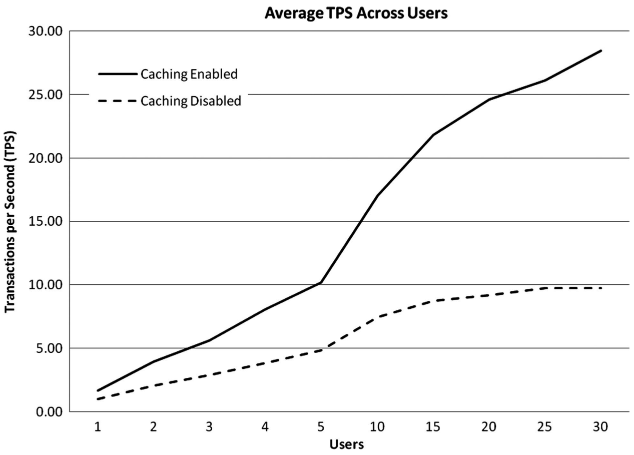

Figure 4 Enterprise clinical rules service overall performance in transactions per second (TPS) with increasing concurrent users, caching of classifications enabled and disabled.

Model preprocessing and elaboration

The ECRS is characterized by extensive preprocessing and formulation of an inference model in order to minimize processing time required by the decision core to compute the requested recommendation. The input model is subjected to exhaustive classification before decision processing that recognizes class-level membership for problems, medications, and allergies. This architecture consciously trades off the cost of preprocessing the inference model in order to maximize the availability of decision processing units as the scarce resource in the system. An alternative design could integrate classification computations into decision processing to lower the cost of preprocessing at the expense of the availability of the decision-processing units. For example, given a RES execution time of 50 ms for the piloted rule set, a classification web service call averaging 100 ms would increase RES execution time by 200%. An efficient design would ultimately require the tight integration of a classification facility with the decision processor.

Data adapter architecture

The ECRS has the capability to accept data in multiple formats. Because the system was intended to be used with both clients internal to PHS, as well as with as-yet-to-be-identified external clients, we chose a design that would minimally allow internal clients to use proprietary data formats while allowing external clients to use shareable standards, such as the CCD, or evolving formats, such as the resource description framework payload employed in the SMART platform.[31] This philosophy minimizes the burden on client application developers and conserves content assets as new data adaptors are developed or old adaptors are modified, without requiring modification of internal decision support logic.

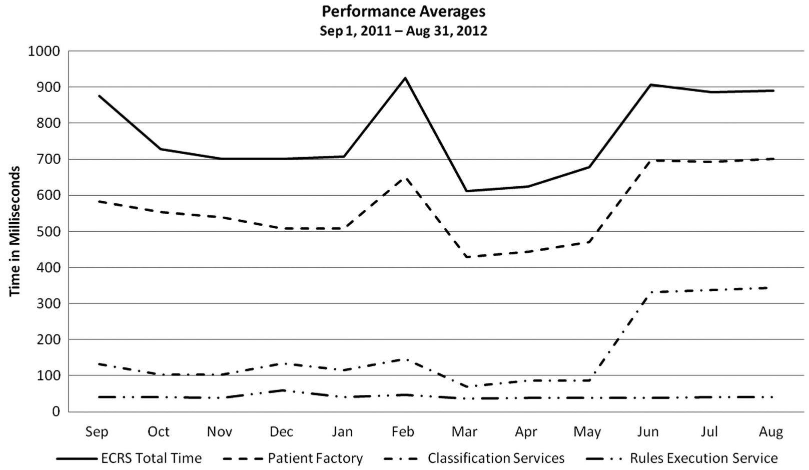

Figure 5 Actual enterprise clinical rules service module (ECRS) and overall performance during the full year clinical decision support consortium trial, sharing preventative alerts and reminders between Partners Healthcare and Wishard Health Services.

Backend decision services

The ECRS has been designed and implemented to use a third party general-purpose BRMS in order to provide a production rule execution environment. A contemporary BRMS provides general-purpose facilities to evaluate rule sets over data models implemented in a programing language, and may provide additional convenience tools such as decision tables and decision trees. In previous and current work, we have found the expressivity of general-purpose BRMS to be adequate for CDS. In addition, we have found that the function libraries of these tools are easy to extend; for example, in supporting temporal reasoning.

Models and relationships to standards

As we have been developing the ECRS, standards relating generally to interoperability and specifically to decision support have continued to evolve. We have benefited greatly from the CCD standard, which we had adopted before the onset of the meaningful use regulation. We believe our use of a CCD as a data payload for the purpose of sharing executable alerts and reminders is the first large-scale demonstration of interoperable decision support leveraging this standard. The work of the CDSC demonstrates the great promise of interoperability standards, and the opportunity to disseminate CDS to areas where information technology resources are lacking. ECRS functionality is comparable to that delineated in the HL7 DSS specification,[32] differing in terms of concrete interfaces as proposed by the standard. Notably, the ECRS implements the functionality of the DSS EvaluateRules interface, but differs in the concrete implementation of the interface, the underlying inference model, and the schema of the recommendation object returned. Given the newness of this standard, with limited implementation experience, we should continue to encourage experimentation with service features and models to gain the experience and evidence necessary to establish meaningful and useful standards.

Limitations

The ECRS is not intended to be a full, closed-loop CDS application; therefore, it does not offer all of the application features that a consumer may need, such as notification services. Although the ECRS works well for the initial use case, others remain to be tested; including, complex rule patterns, rules that determine what rule sets should be executed against the provided input, stateful rule execution—persistence and retrieval of an inference model over multiple invocations of the DSS—for long running guidelines or protocols, and the use of alternative rules management systems. The framework is not yet in place to test these models, nor is caching of patient data currently available beyond the boundary of a single decision support transaction.

CONCLUSION

Our initial experience with design, implementation and production use of the ECRS reveals that the system performs well both within the context of an individual enterprise, as well as shared among enterprises distributed across the USA. Comparison and prototyping with emerging standards, such as HL7 DSS, indicate that ECRS is sufficiently flexible to meet these standards and to make use of a variety of input models. As we collaborate with additional partners, we will investigate additional CDS patterns that we have designed the ECRS to accommodate and will continue to review and enhance its capability to perform at scale.

Acknowledgements The authors would like to thank the members of the PHS ECRS team, for their effort on and support of this project: Deepika Pabbathi, Richard Boyer, Eugene Shilmayster, and Michael Vashevko.

Contributors All authors meet the ICMJE criteria for authorship. Contributions include the following: HSG: conception and design, analysis and interpretation of data, drafting the article and final approval of the version to be published. HSG is the guarantor. MDP: conception and design, analysis and interpretation of data, critical revision of the manuscript for important intellectual content, and final approval of the version to be published. BHR: conception and design, analysis and interpretation of data, critical revision of the manuscript for important intellectual content, and final approval of the version to be published. MS: conception and design, critical revision of the manuscript for important intellectual content, and final approval of the version to be published. AW: conception and design, critical revision of the manuscript for important intellectual content, and final approval of the version to be published. JLE: analysis and interpretation of data, critical revision of the manuscript for important intellectual content, and final approval of the version to be published. BM: concept and design, analysis and interpretation of data, critical revision of the manuscript for important intellectual content, and final approval of the version to be published.

Funding This project is derived from work supported under contract #HHSA290200810010 from the Agency for Healthcare Research and Quality (AHRQ), US Department of Health and Human Services. The findings and conclusions in this document are those of the author(s), who are responsible for its content, and do not necessarily represent the views of AHRQ. No statement in this report should be construed as an official position of AHRQ or of the US Department of Health and Human Services. Identifiable information on which this report, presentation, or other form of disclosure is based is protected by federal law, section 934(c) of the public health service act, 42 U.S.C. 299c-3(c). No identifiable information about any individuals or entities supplying the information or described in it may be knowingly used except in accordance with their previous consent. Any confidential identifiable information in this report or presentation that is knowingly disclosed is disclosed solely for the purpose for which it was provided.

Competing interests None.

Provenance and peer review Not commissioned; externally peer reviewed.

Data sharing statement The data presented in this manuscript have not been presented in any other publications.

REFERENCES

1 Garg AX, Adhikari NK, McDonald H, *et al*. Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review. *JAMA* 2005;293:1223–38.
2 Kuperman GJ, Teich JM, Gandhi TK, *et al*. Patient safety and computerized medication ordering at Brigham and Women's Hospital. *Jt Comm J Qual Improv* 2001;27:509–21.
3 Solovy A. The 100 most wired 2005: the quality connection. *Hosp Health Netw* 2005;79:38–50; 2.
4 Bates DW, Cohen M, Leape LL, *et al*. Reducing the frequency of errors in medicine using information technology. *J Am Med Inform Assoc* 2001;8:299–308.
5 Kuperman GJ, Bobb A, Payne TH, *et al*. Medication-related clinical decision support in computerized provider order entry systems: a review. *J Am Med Inform Assoc* 2007;14:29–40.
6 Maviglia SM, Zielstorff RD, Paterno M, *et al*. Automating complex guidelines for chronic disease: lessons learned. *J Am Med Inform Assoc* 2003;10:154–65.
7 Sintchenko V, Coiera E, Iredell JR, *et al*. Comparative impact of guidelines, clinical data, and decision support on prescribing decisions: an interactive web experiment with simulated cases. *J Am Med Inform Assoc* 2004;11:71–7.
8 Blumenthal D. Stimulating the adoption of health information technology. *N Engl J Med* 2009;360:1477–9.
9 Blumenthal D. Launching HITECH. *N Engl J Med* 2010;362:382–5.
10 Kawamoto K, Lobach DF. Design, implementation, use, and preliminary evaluation of SEBASTIAN, a standards-based Web service for clinical decision support. *AMIA Annual Symposium Proceedings; 2005: American Medical Informatics Association*; 2005:380.
11 Tu SW, Campbell JR, Glasgow J, *et al*. The SAGE Guideline Model: achievements and overview. *J Am Med Inform Assoc* 2007;14:589–98.
12 Wright A, Sittig DF. SANDS: a service-oriented architecture for clinical decision support in a National Health Information Network. *J Biomed Inform* 2008;41:962–81.
13 The Clinical Decision Support Consortium website. 2008. [cited 3/11/2011]. http://www.partners.org/cird/cdsc/.
14 Middleton B. The clinical decision support consortium. *Stud Health Technol Inform* 2009;150:26–30.
15 Dixon BE, Simonaitis L, Goldberg HS, *et al*. A pilot study of distributed knowledge management and clinical decision support in the cloud. *Artif Intell Med* Published Online First: 1 Apr 2013. doi: 10.1016/j.artmed.2013.03.004.
16 Centers for Disease Control and Prevention: Immunization Schedules. 2013. [cited March 2013]. http://www.cdc.gov/vaccines/schedules/.
17 Kuppermann N, Holmes JF, Dayan PS, *et al*. Identification of children at very low risk of clinically important brain injuries after head trauma: a prospective cohort study. *Lancet* 2009;374:1160–70.
18 Mandl KD, Mandel JC, Murphy SN, *et al*. The SMART Platform: early experience enabling substitutable applications for electronic health records. *J Am Med Inform Assoc* 2012;19:597–603.
19 Teich JM, Hurley JF, Beckley RF, *et al*. Design of an easy-to-use physician order entry system with support for nursing and ancillary departments. *Proc Annu Symp Comput Appl Med Care*; 1992:99–103.
20 Kuperman GJ, Teich JM, Bates DW, *et al*. Representing hospital events as complex conditionals. *Proc Annu Symp Comput Appl Med Care*; 1995:137–41.
21 Kuperman GJ, Teich JM, Bates DW, *et al*. Detecting alerts, notifying the physician, and offering action items: a comprehensive alerting system. Proceedings: a Conference of the American Medical Informatics Association/AMIA Annual Fall Symposium AMIA Fall Symposium; 1996:704–8.
22 Goldberg HS, Vashevko M, Pastilnik A, *et al*. Evaluation of a commercial rule engine as a basis for a clinical decision support service. *AMIA Annual Symposium Proceedings/AMIA Symposium AMIA Symposium*; 2006:294–8.
23 Wright A, Goldberg H, Hongsermeier T, *et al*. A description and functional taxonomy of rule-based decision support content at a large integrated delivery network. *J Am Med Inform Assoc* 2007;14:489–96.
24 Wright A, Goldberg H, Hongsermeier T, *et al*. A description and functional taxonomy of rule-based decision support content at a large integrated delivery network. *J Am Med Inform Assoc* 2007;14:494.
25 HL7/ASTM Implementation Guide for CDA Release 2 -Continuity of Care Document (CCD®) Release 1. [cited 2013 March]. http://www.hl7.org/implement/standards/product_brief.cfm?product_id=6.
26 S&I Framework: Health eDecisions Project Charter and Members. [cited 2013 March]. http://wiki.siframework.org/Health+eDecisions+Project+Charter+and+Members.
27 Bierman G, Wren A. *First-class relationships in an object-oriented language*. ECOOP 2005-Object-Oriented Programming. Springer, 2005:262–86.
28 Schnipper JL, Linder JA, Palchuk MB, *et al*. 'Smart Forms' in an Electronic Medical Record: documentation-based clinical decision support to improve disease management. *J Am Med Inform Assoc* 2008;15:513–23.
29 Sordo M, Ogunyemi O, Boxwala AA, *et al*. GELLO: an object-oriented query and expression language for clinical decision support: AMIA 2003 Open Source Expo. *AMIA Annual Symposium Proceedings; 2003: American Medical Informatics Association*; 2003:1012.
30 Kawamoto K, Lobach DF. Design, implementation, use, and preliminary evaluation of SEBASTIAN, a standards-based Web service for clinical decision support. *AMIA Annual Symposium Proceedings/AMIA Symposium AMIA Symposium*; 2005:380–4.
31 W3C. Resource Description Framework (RDF): Concepts and Abstract Syntax. [cited 2013 May 28]. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.
32 Del Fiol G, Jenders R, Kawamoto K, *et al*. *HL7 Version 3 Standard: Decision Support Service (DSS), Release 1*. 2011 Health Level Seven International: HL7 Clinical Decision Support Work Group; 2011 August 23. Report No.: ANSI/HL7 V3 DSS, R1-2011.