SANDS: An Architecture for Clinical Decision Support in a National Health Information Network

Adam Wright\textsuperscript{a}, Dean F. Sittig\textsuperscript{a,b}

\textsuperscript{a} Department of Medical Informatics & Clinical Epidemiology, Oregon Health & Science University, Portland, OR
\textsuperscript{b} Department of Medical Informatics, Northwest Permanente Medical Group, Portland Oregon, USA

Abstract

A new architecture for clinical decision support called SANDS (Service-oriented Architecture for NHIN Decision Support) is introduced and its performance evaluated. The architecture provides a method for performing clinical decision support across a network, as in a health information exchange. Using the prototype we demonstrated that, first, a number of useful types of decision support can be carried out using our architecture; and, second, that the architecture exhibits desirable reliability and performance characteristics.

Background

Electronic medical records are steadily gaining in popularity\textsuperscript{1} and hold tremendous promise for improving the quality and efficiency of healthcare delivered in the United States\textsuperscript{2-4}. However, clinical decision support (CDS) is critical to maximize their value\textsuperscript{5}. There are several types of decision support, including alerts, reminders, documentation templates, as well as tools for facilitating order creation, such as order sets and corollary orders\textsuperscript{6}. While clinical decision support has been shown to be effective in a variety of studies, its adoption is fairly limited outside of a few academic medical centers and integrated delivery networks, particularly decision support targeted at ordering providers\textsuperscript{7}.

One of the major questions of decision support is how to take the interventions that are in successful use at these sites and translate them to sites which have not had as much success with clinical decision support. In another paper we presented a four-phase model for clinical decision support architectures:\textsuperscript{8}

1. \textbf{Standalone systems:} Systems such as MYCIN\textsuperscript{9} or Internist\textsuperscript{10} which operate separately from clinical systems. Such systems are relatively easy to develop and share, but they have significant limitations: they require the user to enter all the information the system needs to make its inference, and since they must be explicitly invoked, they cannot be proactive.

2. \textbf{Integrated systems:} Systems such as LDS Hospital’s HELP\textsuperscript{11} and Regenstrief Institute’s RMRS\textsuperscript{12}, which build clinical decision support into a clinical system. These systems can be proactive and obviate the need for duplicate entry, but sharing decision support content with sites using different clinical systems can be difficult.

3. \textbf{Standards-based systems:} Standardized knowledge representation formalisms, such as Arden Syntax\textsuperscript{13}, GLIF\textsuperscript{14} and GELLO\textsuperscript{15}. Rules are represented in a standardized format so they can be shared by different clinical systems. These standards keep the advantages of integrated systems while enabling sharing, but their adoption has been limited. Also, the kinds of knowledge that can be represented are inherently limited by the constraints of the knowledge representation system.

4. \textbf{Service architectures:} Systems such as SAGE\textsuperscript{16} and SEBASTIAN\textsuperscript{17}, where clinical decision support systems and clinical information systems are separate, but standardized interfaces are used to enable them to operate together. SAGE places a standardized interface in front of a clinical system, allowing a decision support system developer to access clinical data in a standard format. SEBASTIAN takes the opposite approach, placing a standard interface in front of a clinical decision support system which can be called across the network.

Introducing the SANDS Architecture

While all of these approaches have relative advantages and disadvantages, none has achieved widespread adoption and success. In this paper, we present a new architecture for clinical decision support called SANDS (Service-oriented Architecture for NHIN Decision Support).

This architecture draws on the lessons of SAGE and SEBASTIAN, with two major differences. First, it places a standard interface in front of both the clinical system and the decision support system, and second, it explicitly considers the case where a patient’s record is stored across multiple clinical systems, such as in a National Health Information Network (NHIN), and where multiple clinical decision support services are in use. In fact, just as an NHIN is a distributed network for sharing patient information, SANDS is a distributed network for
sharing clinical decision support content. Figure 1 shows a schematic representation of the architecture.

There are several points about the SANDS architecture worth mentioning:

1. The interaction is defined entirely by interfaces: so long as each node (decision support system or clinical system) makes the specified interface available its internal behavior can be arbitrary.
2. Unlike systems like Arden Syntax, there is no required form for knowledge representation, so developers of decision support systems are not constrained in the logic they can implement.
3. Building an architecture like this requires standards. For example, a drug interaction or allergy decision support service might use standards like NCPDP Script, while a reportable disease service might use SNOMED to describe diseases. The specific standards used will ultimately be defined as part of the effort towards development of an NHIN.

Development Methods

To demonstrate the feasibility and advantages of the SANDS architecture we developed and evaluated a working prototype. As shown in Figure 1, there are two interface types required to realize the SANDS architecture: an NHIN Interface and a CDS Interface.

There is, of course, no actual NHIN in use in the United States. However, the Office of the National Coordinator for Health Information Technology has funded the development of four prototype National Health Information Networks. To demonstrate the feasibility of the SANDS architecture we connected to one of these prototypes, developed by Computer Sciences Corporation and the Markle Foundation. This prototype unites local health information exchanges in Indianapolis, Massachusetts and Mendocino, CA.

The other interface needed for SANDS is a CDS interface. Unlike the NHIN interface, there is no existing prototype for a CDS interface, so we developed one. In a previous paper, we described a functional taxonomy of clinical decision support which defines:

- **Triggers**: Events in a clinical system which invoke a clinical decision support inference, (e.g., ordering a medication).
- **Data elements**: Patient-specific information used for decision support such as a medication list, problem list or lab result.
- **Intervention**: The action a decision support system takes in response to some clinical situation, such as notifying the user or logging an event.
- **Response Action**: The action a user takes in response to a notification, such as canceling an order or overriding the rule.

This taxonomy attempts to lay out a fairly complete set of items in each of these four groups. We developed an XML-based service protocol which implements the full taxonomy. It is not reproduced here for space reasons, but is available from the authors upon request.

Evaluation Methods

Our prototype of the SANDS architecture consists of two components: a prototype EHR and a prototype decision support network.

The prototype EHR has most of the features of a regular EHR (labs, notes, medications, problems) with one significant difference: instead of having its own internal data store it renders a virtual patient record retrieved from the NHIN prototype.
The prototype decision support network functions according to the protocols and message formats described above and offers a variety of decision support functionality, using both self-developed and existing services. These exact functions are described in the Results section.

Results

We successfully developed a prototype of the SANDS architecture as well as a prototype electronic health record. Figure 2 shows the lab results tab of the prototype client for a sample patient.

To better understand how decision support works in the SANDS architecture, consider a real world problem: Dr. Anderson, a primary care provider, who uses the SANDS prototype EHR; and Dr. Baxter, a gastroenterologist who uses Epic’s EHR. They share a common patient in Frank Jones. Mr. Jones sees Dr. Baxter for management of severe Gastroesophageal Reflux Disease (GERD), but today presents to his primary care provider, Dr. Anderson complaining of a sore throat, which Dr. Anderson diagnoses as streptococcal pharyngitis. Dr. Anderson plans to prescribe erythromycin to treat the infection, but first asks Mr. Jones what medications he’s on. He reports that he is taking Lipitor and aspirin, as prescribed by Dr. Anderson, as well as Prevacid for his GERD, as prescribed by Dr. Baxter. Seeing no danger, Dr. Anderson initiates a prescription for erythromycin in her clinical system, but before the prescription is accepted the decision support network is queried. A message containing the intended prescription, as well as a pointer to Mr. Jones’ record in the NHIN is sent to a drug checking service that Dr. Anderson subscribes to. This service sends a medication list query to the NHIN interface which uses its record locator service to find that Mr. Jones has records in two disparate clinical systems: those of Drs. Anderson and Baxter. The NHIN interface requests the medication lists in these systems, aggregates them and returns them to the drug checking service. This service notices, however, that Dr. Baxter’s medication list indicates that Mr. Jones is actually on Propulsid for his GERD, not Prevacid, as Mr. Jones had indicated to Dr. Anderson. There is a very severe and potentially fatal interaction between Propulsid and Erythromycin, and the system provides this information to Dr. Anderson’s clinical system which raises an alert and blocks the prescription.

Although this may seem like a simple case, it’s important to note that even though the FDA engaged in a significant outreach and public relations campaign to make doctors aware of this interaction, it killed at least 78 people, and is suspected in the deaths of 302 others. In the end, the FDA had to withdraw Propulsid from the market because it was unable to reliably prevent the two drugs from being co-prescribed. A decision support architecture such as SANDS (or another safety mechanism, such as pharmacist verification) may be another way to ensure that such events do not occur.

In order to demonstrate the utility of the architecture, we developed interfaces to a variety of commercially available clinical decision support services. These services illustrate several use cases, including:

- Diagnostic decision support using the Isabel system (Isabel Healthcare, Haslemere, UK).
- Drug interaction checking (Lexi-Comp, Inc., Hudson, OH)
- Information support using UpToDate (UpToDate, Waltham, MA) and Google Co-op (Google, Inc., Mountain View, CA).
- Inappropriate prescribing in the elderly, based on the Beers criteria.
- Automatic reporting of reportable diseases and syndromic surveillance, along with geospatial resolution and mapping (using a service called a geocoder, which resolves addresses to latitude and longitude).
- A personal health record, with support for drug-interaction checking and patient drug information.

One critical success factor of any decision support system is performance. A common goal of clinical system developers is sub-second response time, and one critical question we faced was whether this ideal could be achieved using the SANDS architecture. The SANDS architecture is subject to five kinds of delay which are generally additive.

![Silver, Stephanie](AMIA 2007 Symposium Proceedings Page - 818)
Network latency: The time it takes for a packet to propagate between two hosts on a network. This is a startup cost of transmission – after the first packet, throughput is governed by transmission delay.

Transmission delay: The time needed to transmit a message over the network once latency is overcome. With SANDS’ small message sizes, throughput is usually not a major source of delay.

Patient data fetch: SANDS fetches a patient’s clinical data from the NHIN. The cost of this fetch can be fairly high so SANDS employs a caching strategy at the NHIN interface to reduce this delay.

Parsing overhead: SANDS is based on XML protocols and there is overhead in parsing the XML. Given the size of the messages used in SANDS, the overhead is fairly small.

Inference time: This is the actual time that the inference takes to run. This is not overhead added by SANDS. Even tightly integrated decision support systems face this delay.

To assess the robustness of the SANDS architecture and the delays associated with it we conducted a timing and reliability study. We set up the system to automatically poll each service in the SANDS implementation every 5 minutes (24 hours/day, 7 days/week) over a continuous four week period (a total of 13,440 requests). Table 1 shows the results of this study. The first column after the service name, Time, gives the mean response time in milliseconds for each service. The next four columns give the frequency with which various results occurred: if the query failed, if it returned within 1 second, if it returned within 5 seconds, or if it was successful but took 5 or more seconds to complete. Seven of the nine (78%) SANDS decision support modules had subsecond response times > 97% of the time.

**Discussion**

Overall, these results indicate that it is not only possible, but highly reliable and fast to perform a variety of useful clinical decision support interventions using the SANDS architecture with sub-second response time. Most (7/9) services consistently (>97% of the time) serviced requests in under one second. The syndromic surveillance and prescribing in the elderly services, which were self developed, had outstanding performance because they were optimized for efficiency and located locally so a request to them did not have to traverse the Internet. The Drug-drug interaction service never completed within 1 second, but almost always (99.83%) completed within 5 seconds. This is likely due to the data structures used by the developer of the service. The last three services, called geocoders, are worth further comment. The syndromic surveillance use case has a mapping capability, where cases of reportable diseases are displayed geospatially. To do this, a service called a geocoder is used. A geocoder takes an address and finds the corresponding latitude and longitude. Three different free geocoding services were tested and the variation in performance was striking – Google’s geocoder never failed, and returned within 1 second 99.9% of the time. Yahoo’s geocoder, doing the same task, failed 0.31% of the time and took, on average, nearly 2.5 times as long. Clearly, system reliability and performance will be an important consideration for any user of the SANDS architecture. While the architecture itself imposes only minimal latency and overhead, the performance of the same task, implemented by different developers, may vary greatly so it will be important to carefully evaluate performance before choosing services.

An architecture like SANDS has a number of advantages over other approaches. Unlike Arden Syntax, it does not constrain the type of logic that can be represented. Because the services are centralized, knowledge maintenance is easier and the effort required on the part of clinical system implementers to enable decision support is reduced. Further, an architecture like SANDS would enable a variety of business models to flourish – one might imagine that some decision support types might be offered for free by, say, medical specialty societies, while others, such as drug interaction databases, could be offered by subscription. Because SANDS reduces the effort required to switch between services, we might expect competition to be robust. For example, the three geocoding services discussed differentiate themselves

<table>
<thead>
<tr>
<th>Service</th>
<th>Time (ms)</th>
<th>Failed</th>
<th>&lt;1s</th>
<th>&lt;5s</th>
<th>&gt;=5s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagnosis (Isabel)</td>
<td>495</td>
<td>0.07%</td>
<td>99.41%</td>
<td>100.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Information (Google)</td>
<td>169</td>
<td>0.00%</td>
<td>99.79%</td>
<td>100.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Information (UpToDate)</td>
<td>787</td>
<td>0.80%</td>
<td>90.33%</td>
<td>99.58%</td>
<td>0.42%</td>
</tr>
<tr>
<td>Prescribing in the elderly</td>
<td>73</td>
<td>0.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Drug-drug interaction</td>
<td>1281</td>
<td>0.00%</td>
<td>0.00%</td>
<td>99.83%</td>
<td>0.17%</td>
</tr>
<tr>
<td>Syndromic surveillance</td>
<td>74</td>
<td>0.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Google geocoder</td>
<td>124</td>
<td>0.00%</td>
<td>99.90%</td>
<td>100.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Yahoo geocoder</td>
<td>315</td>
<td>0.31%</td>
<td>97.05%</td>
<td>99.34%</td>
<td>0.66%</td>
</tr>
<tr>
<td>Geocoder.us</td>
<td>216</td>
<td>0.10%</td>
<td>99.83%</td>
<td>99.97%</td>
<td>0.03%</td>
</tr>
</tbody>
</table>
primarily on speed, but one might imagine that clinical services could differentiate based on accuracy, utility or price as well. Finally, the SANDS architecture is promising because it aligns the efforts of decision support with efforts at health information exchange. As terminology and standards issues are resolved to enable information exchange, SANDS also benefits.

Future Work
Performance and reliability are only two characteristics by which a system like SANDS could be evaluated. In future work, we intend to conduct a comparative evaluation of SANDS with a number of other architectures for representing and sharing decision support content to learn the relative advantages and disadvantages of each approach. We also intend to integrate more services and more health information exchange prototypes into the architecture to determine its ability to scale and grow.

Conclusions
This paper has introduced SANDS, a new architecture for clinical decision support and demonstrated that, first, a number of useful types of decision support can be carried out over that architecture; second, that the architecture is reliable; and, third, that it exhibits desirable performance characteristics, enabled decision support interventions with sub-second response time.

References