Published in final edited form as:
J Biomed Inform. 2008 December ; 41(6): 982–990. doi:10.1016/j.jbi.2008.03.009.

A Framework and Model for Evaluating Clinical Decision Support Architectures

Adam Wright, Ph.D.[1,2] and **Dean F. Sittig, Ph.D.**[3,4]

1*Clinical Informatics Research and Development, Partners HealthCare, Boston, MA*

2*Division of General Medicine, Brigham & Women's Hospital, Harvard Medical School, Boston, MA*

3*Department of Medical Informatics, Northwest Permanente, PC, Portland, OR*

4*Department of Medical Informatics and Clinical Epidemiology, Oregon Health and Science University, Portland, OR*

Abstract

In this paper, we develop a four-phase model for evaluating architectures for clinical decision support that focuses on: defining a set of desirable features for a decision support architecture; building a proof-of-concept prototype; demonstrating that the architecture is useful by showing that it can be integrated with existing decision support systems and comparing its coverage to that of other architectures. We apply this framework to several well-known decision support architectures, including Arden Syntax, GLIF, SEBASTIAN and SAGE

MESH Terms

Medical Records Systems, Computerized; Decision Support Systems, Clinical; Decision Making, Computer-Assisted; Decision Support Techniques; Hospital Information Systems; Computer Communication Networks/standards; Information Systems/organization & administration/standards; Systems Integration; Evaluation

Background and Introduction

The reasons for using real-time clinical decision support systems (CDSS) are numerous. Such systems have been used to prevent errors, improve quality, reduce costs and save time. The best evidence suggests that such systems, when used, can be extremely effective(1–7). For example, in a recent systematic review, Garg found that, "CDSS improved practitioner performance in 62 (64%) of the 97 studies assessing this outcome, including 4 (40%) of 10 diagnostic systems, 16 (76%) of 21 reminder systems, 23 (62%) of 37 disease management systems, and 19 (66%) of 29 drug-dosing or prescribing systems"(3).

Corresponding Author: Adam Wright, Ph.D., General Medicine, Brigham & Women's Hospital, 1620 Tremont St., Boston, MA 02120, awright5@partners.org, Phone: (781) 416-8764, Fax: (781) 416-8912.
Reprint Requests: Adam Wright, Ph.D., General Medicine, Brigham & Women's Hospital, 1620 Tremont St., Boston, MA 02120

Conflict of Interest: A portion of the architecture described here is based on work done by the Markle Foundation. Subsequent to completing this research, Author Wright was retained by the Markle Foundation in a consulting role on issues relating to Population Health.

Despite the accumulated evidence for the potential of decision support systems to improve quality of care, such systems have not found wide use outside large academic medical centers and integrated delivery systems such as Kaiser Permanente and the VA (8,9).

In formulating technical approaches to improve the adoption of clinical decision support, it is helpful to consider the evolution of architectures for clinical decision support systems. In an earlier paper (10), we described, in detail, the evolution of architectures for clinical decision support systems across four phases which we briefly describe here:

Standalone decision support systems

Many of the first decision support systems, such as the systems of Ledley and Lusted (11), Warner (12), Shortliffe (13) and Miller, Pople and Myers (14), were standalone – they operated separately from other systems, requiring the user to enter findings and other pertinent information. Such systems were relatively well-suited to sharing, since they did not integrate with other systems. However, entering findings was often time-consuming and, by their nature, such systems could not be proactive, since they could only make inferences when explicitly called upon by their users.

Integrated systems

The second wave in decision support architectures was the integration of decision support systems with broader clinical systems. This movement began in the late 1960s with the HELP system at the University of Utah (15), and the Regenstrief Medical Record System in Indiana (16). Such systems could be proactive, and made fewer demands on the user to enter the findings needed for their inferences, since this data could often be accessed from other areas in the clinical system. However, such systems were much more difficult to share since they tended to be very tightly integrated with, and dependent on the idiosyncrasies in, the clinical systems.

Standards-based systems

The third movement in decision support architecture, beginning in 1989, represented attempts to standardize computerized representation of clinical knowledge. The original effort to do so was Arden Syntax, a procedural formalism which is still in development today (8,9). Although a large number of standards followed, none (including Arden Syntax) have yet seen widespread commercial adoption.

Service models

One of the most recent, and most promising, advances in decision support architecture has been the advent of service models, beginning with the Shareable Active Guideline Environment (SAGE) (17) and the System for Evidence-Based Advice through Simultaneous Transaction with an Intelligent Agent across a Network (SEBASTIAN) (18). These efforts once again separate clinical systems and decision support systems, but integrate them using standardized, service-based interfaces. However, these efforts have seen limited adoption due to a variety of issues, ranging from difficulty in standardizing data models to issues regarding patents and intellectual property (19).

Developers and implementers of clinical decision support systems face a formidable choice in determining the method for integrating decision support into broader clinical systems. Not only must they choose between these four classes of architectures, they must also consider which specific instance of any of these classes to choose from. There are literally dozens of knowledge representation formalisms and service-oriented approaches to integrating clinical decision support, and there is no clear consensus on which is most desirable (in fact, most commercial clinical systems implement none of them, or only a proprietary approach). In this paper, we propose a framework for evaluating such architectures, and apply it to several of the more commonly used and studied existing architectures.

A Framework for Evaluation

We conducted a background review of the clinical decision support and medical informatics evaluation literature and consulted with experts to determine useful classes of informatics evaluations. A cornerstone of our literature review was a prior framework for informatics evaluation published as “Designing medical informatics research and library-resource projects to increase what is learned” by William Stead and the members of the Biomedical Library and Informatics Review Committee (BLIRC) (20). The BLIRC is the board that reviews grants submitted to the National Library of Medicine and, as a group, developed a consensus framework to “increase what is learned from information access, information systems, and medical informatics research projects” by improving the quality of evaluation in the field of informatics. The BLIRC framework is designed to work across the field of informatics, so it is necessarily quite general. To supplement the framework, we turned to a more recent publication, the “Roadmap for National Action on Clinical Decision Support”, commonly referred to as the CDS roadmap (21). The CDS roadmap was published by the American Medical Informatics Association at the request of the United States Department of Health and Human Services’ Office of the National Coordinator for Health Information Technology. It lays out a research and engineering agenda for clinical decision support based on the work of an assembled panel of several dozen national experts in clinical decision support. While not a general evaluation framework, such as the BLIRC’s, its single focus on decision support made it extremely useful in the development of our evaluation framework.

Based on our review of the literature and discussions with experts, we propose a four-phase framework for evaluating clinical decision support architectures. The four phases are:

1. **Feature determination:** We develop a set of desirable features for decision support architectures based on a review of literature and expert opinions and then evaluated the relative ability of decision support architectures to exhibit these desirable features.
2. **Existence and Use:** We develop a four level spectrum of the existence and use of clinical decision support architectures, ranging from theoretical discussion at one end to widespread adoption and use at the other extreme.
3. **Utility:** We evaluate each architecture’s ability to implement a wide range of decision support use cases.
4. **Coverage:** We measure each architecture’s ability to cover a large knowledge base of decision support content.

Elements of both the BLIRC evaluation framework and the CDS roadmap are easily visible in our own model. For example, the system development categories of the BLIRC framework inform our existence and use spectrum, and the CDS roadmap substantially guided our work in both the feature determination and utility categories.

Selecting Architectures for Comparison

In addition to putting forward a framework for evaluation of clinical decision support architectures, we felt it was important to apply it to the four general classes of decision support architectures described previously, and to the most widely used and studied specific implementations of these architectures. All four of the architectural approaches were considered in the review: standalone, integrated, standardized and service-oriented. For standalone and integrated architectures, hypothetical best-in-class were considered – if it is possible for any system developed in that architectural paradigm to exhibit a desirable feature, that feature is credited to the hypothetical system for that paradigm. For standards-based systems, Arden Syntax (8,9,22) and GuideLine Interchange Format (GLIF) 3.5 (23–25) were used. GLIF offers a choice of expression languages, with GELLO (a loose acronym for Guideline Expression Language Object-Oriented) being the recommended choice. We used GELLO for three reasons: first, because it is recommended, second, because it is the most expressive expression language for GLIF, and third, because there is some extremely promising work underway on developing a GELLO toolkit and authoring environment (26). For service-oriented systems, SEBASTIAN (18,27) and SAGE (17) were evaluated. It's important to note that there is a fundamental difference between knowledge representation formalisms (like Arden Syntax and GLIF) and approaches like SEBASTIAN and SAGE. While the knowledge representation formalisms are focused on how to encode and represent knowledge, approaches like SEBASTIAN and SAGE extend to include the actual method of integrating this knowledge into clinical systems. Further, each of these architectures and formalisms can be considered either in the abstract, or in a specific implementation. For example, Arden Syntax is both an HL7 and ASTM standard, but it has also been implemented in commercial clinical systems from Siemens, Eclipsys and McKesson. In each case, the specific commercial implementations contain changes and extensions from the standard. For the purposes of this evaluation, we consider only the formalisms and architectures in their pure forms – not specific implementations. We made this decision because these implementations are often not well documented (or are considered proprietary by their implementers) so they could not be fully evaluated.

Results

Feature Determination

Based on a review of the history of decision support systems and architectures (28), as well as a review of some best practices for decision support, a set of desirable features for a decision support architecture was identified:

- **Avoids vocabulary issues:** It is desirable for decision support systems to avoid conflicts relating to terminology and vocabulary. Standards like Arden Syntax create vocabulary issues by leaving the vocabulary specification to the user. Systems like GLIF and SAGE deal with vocabulary issues by specifying a clinical information model which includes vocabulary standards – this solves vocabulary problems, but forces clinical system developers to adhere to a specific information model. Stand-alone systems avoid vocabulary issues entirely since they do not interface with other systems.
- **Shareable:** The basic goal of most knowledge representation formalisms is enabling the sharing of decision support content. Each approach, except for fully integrated systems, is shareable. Standalone systems are trivially shareable, since they can simply be copied from one computer to another.
- **Can view decision support content:** Formalisms like Arden Syntax and GLIF specify a format for sharing knowledge that is both human and machine readable. This is generally a good thing because it allows consumers of content to review it. Service-oriented architectures, such as SEBASTIAN, are more challenging in this regard, since the decision support content exists far away from the calling clinical system, and often in a system not under the control of the calling system.
- **Content separate from code:** A common goal of knowledge management and software engineering is the separation of content (decision support rules) from the code required to operate the clinical system. All of the architectures, except the fully

integrated model, make this separation explicit. Well designed fully integrated systems often also separate content from code, but this is not explicitly required.

- **Automatic central updates:** One challenge in clinical decision support is keeping decision support systems up to date, particularly since medical knowledge changes quickly. With SEBASTIAN clinical decision support content is held with its developer, so when the knowledge base is updated, the changes are automatically reflected in client clinical systems. The other approaches, which generally require the user to install an update, increase the likelihood that out of date content will be used.
- **Content integrated into workflows:** In the ideal case, decision support is closely integrated into clinical workflows. All of the approaches for clinical decision support, with the exception of standalone systems, provide this capability in theory, although there have been gaps in practice and actual implementations have often left end-users wishing decision support was better integrated with workflow.
- **Supports event driven CDS:** Most clinical decision support is event-driven – some event in a clinical system (such as ordering a medication, receiving a lab result or simply the passage of time) triggers the decision support to fire. All of the approaches support event-driven decision support, except for standalone systems, which the user must explicitly invoke.
- **Supports non-event driven CDS:** Certain cases of clinical decision support are not necessarily event driven – for example, a system which brings up a list of all diabetic patients who haven’t had a hemoglobin A1c test in the past year could be a form of decision support, even though it lacks a specific triggering clinical event. While this use case can be easily handled in standalone and integrated systems, the other approaches have varying levels of support. Arden Syntax is built to handle only event-driven decision support; however, it leaves the specification of clinical events up to the implementer, so non-event-driven behavior can be emulated by, say, specifying an event like “user requests a list of all diabetic patients overdue for hemoglobin A1c”. GLIF has reasonable support for non-event driven logic, and SAGE and SEBASTIAN can be made to perform non-event driven logic.
- **Support decision support over populations:** The normal paradigm for clinical decision support (and for clinical care more generally) is to evaluate and act on one patient at a time. However, it is sometimes desirable to evaluate and possibly treat many patients at a time (for example, ordering the same laboratory test for each patient on a panel of diabetic patients). It is desirable for a decision support architecture to support this approach explicitly. Standalone and integrated systems can be developed to provide this capability, and it is explicitly considered by SAGE as well. Arden Syntax, GLIF and SEBASTIAN are all designed to evaluate one patient at a time, although they may be called repeatedly, allowing logic to be “looped” over a population of patients, emulating this behavior. Also, depending on how curly brace expansion is handled in any particular Arden Syntax implementation, it may be possible to enable this behavior without modifying the Arden Syntax standard. However, we are not aware of any commercial or reference implementations which support this capability.
- **Enables separation of responsibilities:** The people who develop a clinical system (often computer programmers) and the people who develop decision support content (usually clinicians and informaticists) are frequently different. As such, it is helpful if these responsibilities can be separated. This is possible in any of the approaches, but it becomes challenging when decision support content is hard-coded directly into a clinical system, as can be the case in the integrated approach.

- **Enables composition of rules:** It is sometimes the case that one decision support system might want to invoke another decision support system. This is explicitly considered in the service-oriented approaches (SEBASTIAN and SAGE), the formalism-based approaches (GLIF and Arden Syntax) and is also possible with integrated systems. However, this capability can be difficult to achieve with standalone systems unless their inputs and outputs are synchronized.
- **Allows black-box services:** In certain cases decision support developers may consider the logic of their systems to be proprietary. In the case of integrated systems, Arden Syntax and GLIF, the developers must expose their logic to the consumers of their system. However, standalone systems and SEBASTIAN allow for black-box services, where the logic is not exposed. This brings the large caveat that, in a black box service, the consumer cannot review or validate the logic, so choosing to consume such a service must be carefully reasoned, but is, perhaps, justifiable if the service comes from a highly reliable source.
- **Free choice of knowledge representation syntax:** Integrated systems, Arden Syntax and GLIF all require decision support developers to use a specific syntax for representing their content (although interpreters for these syntaxes can, of course, be written in nearly any programming language). In many cases this imposes little limitation, but if the decision support developer is not comfortable working in that syntax, or if the syntax does not support the type of logic that the developer intends to use, this can be a very high cost. It is important to note that both Arden Syntax and GLIF provide some facility for external function calls, allowing this limitation to be bypassed by relying on external code (written in any syntax). This limitation is not faced in standalone systems, since they aren't designed to integrate with clinical systems and it is also avoided in SEBASTIAN which imposes only interface requirements.

Table 1 shows the application of these desired features to the comparison set of currently available decision support architectures. Overall, this analysis indicates that each of the existing architectures has gaps when compared against this list of desirable features. While some of these gaps could be resolved simply by adding features to the existing architectures, others are structural – for example, a fully integrated system, by definition, cannot use a central knowledge service. The fact that no single architecture offers all of these features requires users to make tradeoffs in their selection of an approach.

Existence and Use

The second aspect of this evaluation framework is existence and use. It is generally desirable that a decision support architecture have some degree of wide-spread use. This provides several advantages, including increased likelihood that content sharing partners using the same architecture will be available, as well as increased confidence in the clinical usefulness of a particular architecture, given that it is fairly widely used. We divided gradations of use into a four-step spectrum:

1. The architecture has been proposed and discussed in theory.
2. Prototypes of the architecture and related tools have been developed.
3. More advanced prototypes have been developed, including approaches for sharing content across several sites.
4. The architecture is available in commercially available clinical systems or has been widely adopted (where the definition of wide adoption is, of course, constrained by the relatively limited overall adoption of clinical information systems and decision support systems at present).

Stand-alone systems—Level 4. A variety of these systems have been developed, and many are in current use. Some of the most common examples include online risk score estimators, such as the National Cholesterol Education Program's online heart disease risk assessment tool (http://hp2010.nhlbihin.net/atpiii/calculator.asp?usertype=prof).

Integrated systems—Level 4. Directly integrated systems are in fairly wide use. This includes both classical integrated systems, such as the University of Utah's HELP system and the Regenstrief Institute's Regenstrief Medical Records System (RMRS), as well as many commercial systems which include decision support capabilities.

Arden Syntax—Level 4. Several commercial clinical systems vendors (including Siemens and Eclipsys) include Arden compilers, although use of them is limited.

GLIF 3.5 / GELLO—Level 3. Although GLIF is not yet available in any commercial clinical systems, a variety of decision support use cases have been prototyped, and cross-site knowledge sharing has been carried out, particularly as a part of the InterMed Consortium (29).

SEBASTIAN—Level 3. Prototypes of SEBASTIAN have been developed for several different use cases across more than one site in North Carolina, and efforts are currently underway to standardize SEBASTIAN as part of the Health Level 7 (HL7) Decision Support Service specification.

SAGE—Level 3. Prototypes of SAGE have been developed, and a variety of commercial and academic partners are involved in encoding and sharing guidelines (including GE, Intermountain Healthcare, Mayo Clinic, Stanford and the University of Nebraska (30)).

A variety of architectures at Levels 1 and 2 also exist (31,32), but are not covered in detail in this evaluation, which is focused on systems with at least some degree of use.

Utility

The next step beyond existence is utility. In this section, two dimensions of utility are considered: clinical utility (can clinically useful interventions be developed in the architecture) and functional utility (does the architecture allow for a useful variety of different interventions).

Each of the comparison architectures under consideration in this paper easily passes the clinical utility test. A variety of clinically useful clinical decision support systems have been developed in each architecture. Additionally, in many cases, these decision support systems have actually been implemented and evaluated in real-world clinical settings, frequently with very positive results, as described in the background section.

Perhaps more interesting, at least in terms of contrasts, is a different form of utility which we term functional utility – the architecture's ability to be used for use cases with a variety of different properties, including:

- **Developer:** Ideally, a decision support architecture should not restrict who can develop decision support content. For example, it is undesirable for an architecture to have technical or other restrictions which allow only a particular clinical system vendor to develop content. Likewise, it likely is desirable for an architecture to support a variety of decision support developers, such as end users, third-party commercial developers, and open source developers.
- **User:** The architecture should support decision support with a variety of intended users. Although most decision support is targeted at clinicians, the ability to develop

use cases for other potential users, such as public health departments and patients is desireable.

- **Information source:** An architecture should not place limits on the source of knowledge or information for decision support interventions.
- **Clinical purpose:** The architecture should enable development of decision support across a variety of clinical purposes, including diagnosis, therapy, information display and public health.
- **Inference type:** A variety of inference types should be supported – not just simple rule-based systems.
- **Composition:** It should be possible to combine or compose decision support rules or services to form new decision support systems.
- **Business model:** The architecture should enable a variety of business models for providing decision support, including commercial providers, free and open source providers, decision support provided by the government and decision support provided by expert groups and medical specialty societies (who currently provide written guidelines).
- **Pay Status:** There should be a way to support both free and pay models for decision support.
- **Development Status:** A variety of use cases should be demonstrated in the architecture, not just decision support systems developed by the same entities that developed the decision support architecture.

These utility criteria are generally well-supported by the architectures under review here, as shown in Table 2. There are several important exceptions, however. First, Arden Syntax lacks flexibility in the inference type category – Arden is specifically designed for event-driven rule-based decision support, and lacks support for other models (such as Bayesian reasoning, Natural Language Processing (NLP), expert systems and fuzzy logic). Stand-alone systems, integrated systems and SEBASTIAN perform best in this category, because, unlike the other approaches, they do not place any specific limitations on the way knowledge is represented internally. Like Arden, GLIF and SAGE specify a knowledge representation format, which is potentially limiting, but their formats are much more general than Arden's.

A second significant exception relates to the ability to develop commercial and pay services in Arden and GLIF. Although it is possible to develop commercial services in either of these formalisms, both of them require that the knowledge provider provides a human or machine-readable form of their knowledge content to its customer. This has both advantages and disadvantages: the main advantage is that the customer can view the content and validate it; the main disadvantage is that it requires the knowledge provider to expose content which it might consider proprietary. This is an important tradeoff which must be considered, and relates to the "allows black-box services" functional criteria from the Feature Determination section of this model.

Coverage

The final element of the evaluation framework is coverage – the ability of a decision support architecture to encode clinical knowledge in comparison to other approaches. The basis for the coverage metric is a large knowledge base of clinical decision support content described previously (33). This knowledge base spans several hospitals and a large number of outpatient clinics, and contains 181 rule types and 7,120 rule instances and is coded along four axes:

Trigger—The clinical event (such as a new lab result or medication order) which causes a rule to be invoked.

Data Elements Used—The patient-specific data elements (such as the medication list) used in the inference.

Intervention—The action the decision support system takes in response to its inference.

Choices Offered—Choices offered to the user in response to an intervention.

For example, a drug-drug interaction rule might fire whenever a new medication is ordered (the trigger), compare the medication being ordered to the patient’s medication list (data elements used), pop up an interruptive alert (the intervention) and allow the user to cancel the new order, or discontinue an interacting medication already on the patient’s medication list (the choices offered). Not all architectures can accomplish all types of decision support. Arden syntax, for example, supports making a notification, but does not support offering choices to the user. So any rule in the reference knowledge base which offers choices to the user could not be fully mapped by Arden syntax.

For this evaluation, we looked at the trigger, data elements used, intervention and choices offered for each rule, and determined whether or not each decision support architecture could fully map each rule. Table 3 shows the ability of each reference architecture to perform the various elements of the reference taxonomy. Based on this mapping, we calculated coverage metrics for each architecture: the proportion of the knowledge base that each architecture could cover. It is important to note that many gaps could be easily remediated – for example, if a particular architecture is missing a necessary trigger, the architecture could almost certainly be extended to include that trigger.

For the first two architecture types – standalone and integrated, a hypothetical best-in-class system is considered. For stand-alone, this means a system which can use any kind of data element, but with the significant limitation that decision support can only be triggered by user request. The hypothetical integrated system is actually the gold standard – when decision support is hard-coded directly into a clinical system there is no limit to what can be done (although there are a variety of disadvantages to fully integrated systems, as described previously). For standards-based integrated systems, Arden Syntax and GLIF are used as representatives, since they are the two most widely used and studied approaches. Both of the existing service-oriented approaches, SEBASTIAN and SAGE are also included.

Figure 1 shows the results of this analysis. The gold standard, in this case, is the hypothetical best-case fully integrated system. Because such a system is directly integrated into a clinical system, all data and functional aspects of the clinical system would be, in this hypothetical best case, available to the decision support system, allowing for full expressivity. However, there are two caveats to this strong performance. First, this evaluation is based on a hypothetical best-in-class system which offers the full spectrum of triggers, data elements, interventions and offered choices in the taxonomy. Many commercially available EHRs and CPOE systems, which offer integrated decision support, provide a more limited selection of these taxa and would have correspondingly lower coverage scores. Further, integrated systems have some significant drawbacks, as described in the Feature Determination section of this paper, most notably their inability to be shared with others in a standardized way.

SAGE also performed very well in the coverage evaluation, owing to its very complete virtual medical record and knowledge representation forms. GLIF and SEBASTIAN were able to map slightly less than half the content in the database, largely due to easily remediable limitations on triggers, data elements and interventions. In fact, if the medical and action ontologies in GLIF were extended to include the full taxonomy used here, GLIF would also achieve 100% coverage. Also, as SEBASTIAN is transitioning to an HL7 standard, its data model is being extended to include the full HL7 version 3 Reference Information Model (RIM) which is likewise extremely complete.

Arden Syntax, by contrast, had much lower coverage than the other architectures, mapping only slightly more than 15% of the database. This is largely due to the fact that Arden Syntax only supports simple notifications – it does not allow for offering actionable choices to the user in response to a notification. There has been considerable discussion within the Arden community about extending Arden Syntax to allow for more direct expression of choices, leading to a proposal called the "structured write statement". The structured write statement is currently attached to the Arden Syntax standard as an appendix, which means that it is not part of the normative standard, but is available for use, and may be moved into the standard in a future version. If Arden Syntax is credited with the ability to offer actionable choices to the user, its coverage score increases substantially to 45.9%. Standalone systems, as expected, offer the poorest coverage because they don't support triggers or actionable choices, so only very rudimentary decision support rules (those without a triggering event, and with no response actions) can be fully represented.

Discussion

Implications

This model has implications for a number of different constituencies. First, it provides a method and benchmark for future evaluations of clinical decision support formalisms and architectures. This could be used by researchers, but also by developers of such formalisms, to ensure that their approaches performed as well as possible, and also by potential implementers who wish to choose between these formalisms. In fact, as described in the Coverage section, we believe that many of the gaps observed in existing formalisms could be easily remediated, allowing the formalisms to achieve best-in-class performance on many of the evaluation criteria.

This model may also be of some interest to the Certification Commission on Health Information Technology, who could use it as the basis for a set of certification criteria for clinical decision support, either as a new domain of certification, or within the context of existing certification efforts for ambulatory and inpatient EHRs. Further, the Health Information Technology Standards Panel could use this model as part of its evaluation of decision support related standards in its overall plan to harmonize health information technology standards.

Limitations

Although we believe this model should be broadly applicable and useful, it does have some important limitations. First, it represents one group's thinking, and it is possible that other groups surveying the same landscape could develop divergent evaluation frameworks. Indeed, as described below, we think there is valuable future work in comparing and contrasting this evaluation framework with other frameworks both within Clinical Decision Support and in other aspects of clinical system design and medical informatics. A second limitation is in the set of formalisms and architectures evaluated. We looked at two hypothetical best-in-class approaches and four specific formalisms and architectures, but many more exist. Many of these were excluded from our analysis either because they have only been described in the literature and no reference implementation is available, or because they are no longer in development or use. Also, the systems that were evaluated were evaluated as they exist today – in many cases, there is ongoing work and future versions of these systems which are under development but not yet available for use may correct some of the deficiencies noted.

Future Work

The first and most obvious suggestion for future work based is in the application of this taxonomy to other formalisms and implementations. Although Arden Syntax, SAGE and GLIF are the most commonly used formalisms and architectures, others exist, and the vast majority are cataloged by OpenClinical (34). Further, many of these formalisms have been implemented and extended in various ways. For example, the GuideLine Execution Engine (GLEE) is an execution engine for GLIF3, and may evaluate differently in some aspects than GLIF itself (if, for example, it makes a narrower set of events or data elements available than the full spectrum of GLIF). Also, several commercial clinical systems have implemented Arden Syntax, often with specific proprietary extensions that may result in different performance on various components of the evaluation framework than the standardized version of Arden Syntax evaluated here.

Also, this paper evaluated a hypothetical best-in-class integrated system, but many commercial EHR and CPOE vendors have their own proprietary integrated systems. Each of these systems would almost certainly evaluate differently than the hypothetical system, and an evaluation of several of them would provide a much clearer understanding of the state of the art in clinical decision support as available in commercial systems, which make up the vast majority of the healthcare market, and are much more commonly used than any of the specific formalisms described here. Carrying out such an evaluation of commercial systems would be a valuable contribution to the literature.

Finally, the evaluation framework we propose exists in a broader context of other general and specific evaluation frameworks both within medical informatics and the evaluation literature more broadly. It would be productive to carry out a systematic review and synthesis of these frameworks to develop a better understanding of how and whether they fit together or conflict.

Conclusions

Over the course of this project, we have endeavored to describe a useful model framework for evaluating decision support architectures. We have also applied this framework to several widely used architectures, as well as to some hypothetical best-in-class approaches. Although this evaluation was not intended as a competition, if it were, no clear winner emerged. All of the architectures considered had advantages and disadvantages compared to the others, so, for the time being, decision support and clinical system developers will face trade-offs in selecting a decision support architecture. We are optimistic, however, that, over time, as current architectures mature, performance on these criteria will improve, and that new architectures will emerge with favorable characteristics.

Acknowledgements

Portions of this work were completed as part of AW's doctoral work. We gratefully acknowledge the help of AW's dissertation committee: Holly Jimison, Judith Logan, Kent Spackman, John Halamka and Blackford Middleton. We are also grateful to Dr. Robert Jenders who answered several questions about Arden Syntax, as well as three commercial clinical system vendors who provided feedback about aspects of our model and taxonomy but who did not wish to be named.

Funding: This work was funded in part by NLM Training Grant 1T15LM009461.

References

1. Balas EA, Weingarten S, Garb CT, Blumenthal D, Boren SA, Brown GD. Improving preventive care by prompting physicians. Arch Intern Med 2000 Feb 14;160(3):301–308. [PubMed: 10668831]

2. Cabana MD, Rand CS, Powe NR, et al. Why don't physicians follow clinical practice guidelines? A framework for improvement. Jama 1999 Oct 20;282(15):1458–1465. [PubMed: 10535437]

3. 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 Mar 9;293(10): 1223–1238. [PubMed: 15755945]

4. Grimshaw JM, Russell IT. Effect of clinical guidelines on medical practice: a systematic review of rigorous evaluations. Lancet 1993 Nov 27;342(8883):1317–1322. [PubMed: 7901634]

5. Hunt DL, Haynes RB, Hanna SE, Smith K. Effects of computer-based clinical decision support systems on physician performance and patient outcomes: a systematic review. Jama 1998 Oct 21;280(15): 1339–1346. [PubMed: 9794315]

6. Johnston ME, Langton KB, Haynes RB, Mathieu A. Effects of computer-based clinical decision support systems on clinician performance and patient outcome. A critical appraisal of research. Ann Intern Med 1994 Jan 15;120(2):135–142. [PubMed: 8256973]

7. Kawamoto K, Houlihan CA, Balas EA, Lobach DF. Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. Bmj 2005 Apr 2;330(7494):765. [PubMed: 15767266]

8. Hripcsak G. Arden Syntax for Medical Logic Modules. MD Comput 1991 Mar–Apr;8(2):76. [PubMed: 2038238]8

9. Health Level 7. Arden Syntax for Medical Logic Systems Standard Version 2.6. Ann Arbor, MI: Health Level 7; 2007.

10. Wright A, Sittig DF. A Four-Phase Model of the Evolution of Clinical Decision Support Architectures. Int J Med Inform. 2008in press

11. Ledley RS, Lusted LB. Reasoning foundations of medical diagnosis; symbolic logic, probability, and value theory aid our understanding of how physicians reason. Science 1959 Jul 3;130(3366):9–21. [PubMed: 13668531]

12. Warner HR, Toronto AF, Veasey LG, Stephenson R. A mathematical approach to medical diagnosis. Application to congenital heart disease. Jama 1961 Jul 22;177:177–183. [PubMed: 13783190]

13. Shortliffe EH, Davis R, Axline SG, Buchanan BG, Green CC, Cohen SN. Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the MYCIN system. Comput Biomed Res 1975 Aug;8(4):303–320. [PubMed: 1157471]

14. Miller RA, Pople HE Jr, Myers JD. Internist-1, an experimental computer-based diagnostic consultant for general internal medicine. N Engl J Med 1982 Aug 19;307(8):468–476. [PubMed: 7048091]

15. Kuperman, GJ.; Gardner, RM.; Pryor, TA. HELP: a dynamic hospital information system. New York: Springer-Verlag; 1991.

16. McDonald CJ. Protocol-based computer reminders, the quality of care and the non-perfectability of man. N Engl J Med 1976 Dec 9;295(24):1351–1355. [PubMed: 988482]

17. Ram P, Berg D, Tu S, et al. Executing clinical practice guidelines using the SAGE execution engine. Medinfo 2004;11(Pt 1):251–255.

18. 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

19. Nadkarni P, Miller R. Service-oriented Architecture in Medical Software: Promises and Perils. J Am Med Inform Assoc 2007 Mar–Apr;14(2)

20. Stead WW, Haynes RB, Fuller S, et al. Designing medical informatics research and library—resource projects to increase what is learned. J Am Med Inform Assoc 1994 Jan–Feb;1(1):28–33. [PubMed: 7719785]

21. Osheroff JA, Teich JM, Middleton B, Steen EB, Wright A, Detmer DE. A Roadmap for National Action on Clinical Decision Support. J Am Med Inform Assoc 2007 Mar–Apr;14(2):141–145. [PubMed: 17213487]

22. Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden Syntax. Comput Biomed Res 1994 Aug;27(4):291–324. [PubMed: 7956129]

23. Ohno-Machado L, Gennari JH, Murphy SN, et al. The guideline interchange format: a model for representing guidelines. J Am Med Inform Assoc 1998 Jul–Aug;5(4):357–372. [PubMed: 9670133]

24. Peleg M, Boxwala AA, Ogunyemi O, et al. GLIF3: the evolution of a guideline representation format. Proc AMIA Symp 2000:645–649. [PubMed: 11079963]

25. Wang D, Peleg M, Tu SW, et al. Design and implementation of the GLIF3 guideline execution engine. J Biomed Inform 2004 Oct;37(5):305–318. [PubMed: 15488745]

26. Huff SM, Rocha RA, McDonald CJ, et al. Development of the Logical Observation Identifier Names and Codes (LOINC) vocabulary. J Am Med Inform Assoc 1998 May–Jun;5(3):276–292. [PubMed: 9609498]

27. Health Level 7. Patient Evaluation Service Draft Standard. Ann Arbor, MI: 2005.

28. Wright A, Sittig DF. A Four-Phase Model of the Evolution of Clinical Decision Support Architectures (under review). International Journal of Medical Informatics. 2007

29. Peleg M, Boxwala AA, Tu S, et al. The InterMed approach to sharable computer-interpretable guidelines: a review. J Am Med Inform Assoc 2004 Jan–Feb;11(1):1–10. [PubMed: 14527977]

30. Tu SW, Campbell JR, Glasgow J, et al. The SAGE Guideline Model: achievements and overview. J Am Med Inform Assoc 2007 Sep–Oct;14(5):589–598. [PubMed: 17600098]

31. OpenClinical. Methods and tools to support the computerisation of clinical practice guidelines: a short introduction. 2005 [cited 2008 January 3]. Available from: http://www.openclinical.org/gmmintro.html

32. Peleg M, Tu S, Bury J, et al. Comparing computer-interpretable guideline models: a case-study approach. J Am Med Inform Assoc 2003 Jan–Feb;10(1):52–68. [PubMed: 12509357]

33. Wright A, Goldberg H, Hongsermeier T, Middleton B. A description and functional taxonomy of rule-based decision support content at a large integrated delivery network. J Am Med Inform Assoc 2007 Jul–Aug;14(4):489–496. [PubMed: 17460131]

34. OpenClinical. Methods and tools to support the computerisation of clinical practice guidelines: a short introduction. 2005 [cited 2008 Mar 14]. Available from: http://www.openclinical.org/gmmintro.html

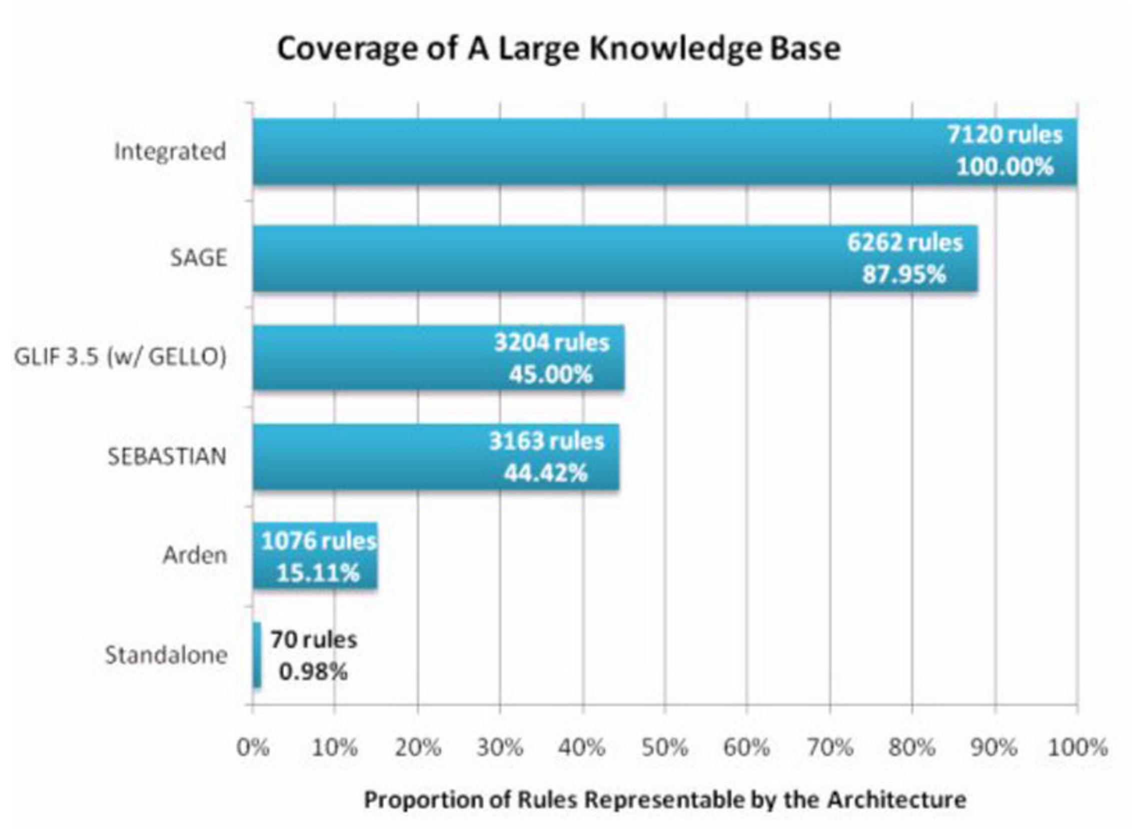

Figure 1.
Proportion of rules in a large clinical decision support knowledge base that could be represented in evaluated decision support architectures and formalisms.

Table 1

Desirable features for decision support architectures and formalisms and their availability in evaluated approaches.

Feature	Standalone	Integrated	Arden	GLIF 3.5 / GELLO	SEBASTIAN	SAGE
Avoids vocabulary issues	x	x				x
Shareable	x		x	x	x	x
Can view decision support content	x	x	x	x		x
Content separate from code			x	x	x	x
Automatic central updates					x	
Content integrated into workflows		x	x	x	x	x
Supports event driven CDS		x	x	x	x	x
Supports non-event driven CDS	x	x	x	x	x	x
Support decision support over populations	x	x				x
Enables separation of responsibilities	x		x	x	x	x
Enables composition of rules		x	x	x	x	x
Allows black-box services	x				x	
Free choice of knowledge representation syntax	x				x	

Table 2

Ability of decision support architectures to implement uses cases along a variety of functional utility dimensions.

	Stand-alone	Integrated	Arden	GLIF3.5 / GELLO	SEBASTIAN	SAGE
Developer	x	x	x	x	x	x
User	x	x	x	x	x	x
Information source	x	x	x	x	x	x
Clinical purpose	x	x	x	x	x	x
Inference type	x	x		x	x	x
Composition role	x	x		x	x	x
Business model	x	x	x	x	x	x
Pay status	x	x	x	x	x	x
Development status	x	x	x	x		

Table 3

Triggers, data elements, interventions and offered choices available in the evaluated decision support architectures and formalisms.

	Standalone	Integrated	Arden	GLIF	SEBASTIAN	SAGE
Trigger						
Order entered		x	x	x	x	x
Lab result stored		x	x	x	x	x
Outpatient encounter		x	x	x	x	x
User request	x	x	x	x	x	x
Time		x	x			x
Admission		x	x	x	x	x
Problem entered		x	x	x	x	x
Enter allergies		x	x	x	x	x
Enter weight		x	x	x	x	x
Data element						
Lab result / observation	x	x	x	x	x	x
Drug list	x	x	x	x	x	x
Hospital Unit	x	x	x	x	x	x
Diagnosis / Problem	x	x	x	x	x	x
Age	x	x	x	x	x	x
Non-drug orders	x	x	x	x	x	x
Gender	x	x	x	x	x	x
Family history	x	x	x	x	x	x
Allergy List	x	x	x	x	x	x
Weight	x	x	x	x	x	x
Surgical history	x	x	x	x	x	x
Reason for admission	x	x	x	x	x	x
Prior visit types	x	x	x	x	x	x
Race	x	x	x	x	x	x
Intervention						
Notify	x	x	x	x	x	x
Log		x				
Provide defaults / picklists		x				x
Show Guidelines	x	x		x	x	x
Collect freetext		x				x

	Standalone	Integrated	Arden	GLIF	SEBASTIAN	SAGE
Trigger						
Get approval		x				
Show data entry template		x				x
Offered choice						
Write order		x		x	x	x
Defer warning		x		x		x
Override rule / keep order		x		x	x	x
Cancel existing order		x		x	x	x
Cancel current order		x		x	x	x
Edit current order		x		x	x	x
Edit existing order		x		x	x	x
Set allergies		x		x	x	x
Write letter		x				x
Write note		x				
Edit problem list		x		x	x	x
Enter weight, height or age		x		x	x	x