Executing Clinical Practice Guidelines using the SAGE Execution Engine

Prabhu Ram PhD[a], David Berg BA[a], Samson Tu MS[b], Guy Mansfield PhD[a], Qin Ye MD MS[a], Robert Abarbanel MD PhD[a], Nick Beard MD MSc[a]

[a]*Health Informatics, IDX Systems Corporation, Seattle, WA, USA*
[b]*Stanford Medical Informatics, Stanford University School of Medicine, Stanford, CA, USA*

Abstract

We report the first successful test of an interoperable guideline execution engine that interprets encoded clinical guideline content and executes that content via functions of a target clinical information system (CIS). For this test, an exemplar immunization guideline was encoded in the SAGE guideline model using standards-based information models and terminologies. This guideline content was subsequently executed using the prototype SAGE guideline execution engine, which interacts through standards-based VMR/Action services to instantiate real-time guideline recommendations via existing functions of the target CIS. In this paper, we describe our test implementation and highlight the significance and implications of each component of our deployment architecture.

Keywords: *clinical practice guidelines, decision support, computer-interpretable guidelines, virtual medical record*

Introduction

Evidence-based clinical practice guidelines have been widely recognized as a primary vehicle to standardize clinical care process, improve quality of care, and reduce unwanted variation and associated costs [1]. Despite this potential, past research efforts have indicated that passive distribution of text-based guidelines has had minimal improvement effect on clinical care [2]. Recent studies suggest the impact of clinical guidelines may best be realized via guideline-driven computerized decision support that is integrated into the real-time care workflow [3].

The SAGE (Standards-based Sharable Active Guideline Environment) project is a collaborative research and development project among research groups at IDX Systems Corporation, the University of Nebraska Medical Center, Mayo Clinic – Rochester, Intermountain Health Care, Apelon Inc., and Stanford University to develop a standards-based comprehensive technology infrastructure that will enable encoding and dissemination of computable clinical practice guidelines. Key deliverables of the SAGE project are: a standards-based guideline representation model; a guideline authoring/encoding workbench; and an interoperable guideline deployment system. Key objectives of the SAGE project are: interoperability of encoded guideline content across disparate CIS platforms and active rendering of guideline content via real-time interaction with existing CIS application functions.

The SAGE Approach

Guideline Knowledge Representation

In the past decade, several guideline models have been developed to formalize clinical guidelines in computer interpretable format (Asgaard [5], EON [6], DILEMMA [7], GLIF [8], GUIDE [9], PRODIGY [10], PROforma [11]). A comparison of their conceptual components and effectiveness concludes that "plan organization, expression language, conceptual medical record model, medical concept model, and data abstractions" are the common guideline components among those guideline models [12]. The SAGE guideline model [4,13] builds on this earlier work and is designed to encode guideline content at the level of detail required for execution within the context of a specific care workflow supported by existing functions of the CIS. To this end, guideline content is represented as detailed clinical "recommendation sets" comprising action specifications, decision logic, and the clinical context in which the recommendations are to be active. The SAGE guideline model uses standard information models, constructs, and data-types to express medical and decision-making concepts. A virtual medical record (VMR) information model [18] has been employed and extended for representation of patient data and guideline-driven actions and all medical concepts are referenced using standard medical terminologies (e.g., SNOMED CT, LOINC).

Guideline Processing and Execution

In a typical SAGE deployment environment, we assume the guideline is first imported into the health care delivery institution. Installation of the guideline involves two main steps: (1) the institution may edit the guideline to conform to its organizational and clinical policies prior to deploying the guideline – we call this step *localization*, (2) mapping from the standards-based concepts in the encoded guideline to local CIS data and local CIS calls, a process we refer to as *binding*. Localization efforts span a guideline while the binding process spans multiple guidelines to the entire health care delivery institution. After installation, the guideline is now available for use at the institution and can be deployed into the guideline execution engine.

Once the guideline has been activated, the SAGE execution engine is able to execute the guideline by interpreting the encoded content, obtaining current patient data from the CIS, and invoking functionality within the CIS to implement an action specified in the guideline.

Guideline Execution: Architectural Overview

A SAGE guideline deployment consists of a SAGE execution engine (henceforth referred to as the engine), an event listener, a terminology server, a set of interfaces called VMR/Action services which interoperate with the local CIS. An execution engine can execute multiple guidelines and a health care delivery may run one or more execution engines based on scalability and other considerations.

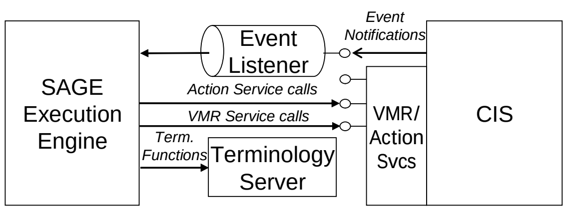

Figure 1 – SAGE Deployment Architecture

The SAGE execution engine is designed specifically to process guidelines encoded using the SAGE guideline model. The engine interprets the content of the context, action, and decision nodes in an encoded guideline, executes workflow and decision logic, and interacts appropriately with the CIS. The event listener is the mechanism by which the engine is notified of state changes in the CIS. As part of conforming to the SAGE engine, the CIS implements the module that forwards events of interest to the event listener. As a reference, the SAGE project currently supplies a Java^{tm} based module that forwards events from the CIS to the listener. The listener is implemented as a web service [14] allowing for broad interoperability and can be used by any conforming CIS to publish events. Most commercial CIS' have terminology services implemented within themselves. In the absence of terminology services within the CIS environment that supports standard terminologies, an external terminology server may be employed in a SAGE deployment. The terminology server encapsulates standard terminologies and implements terminology subsumption that may be used by the engine. The VMR/Action services are interfaces into both patient data and application functionality provided by the CIS. The VMR services are used to get information from the CIS (e.g: *obtain patient's age*) and the Action services are used to initiate actions within the CIS (e.g: *place an order for Hepatitis B vaccine*). The VMR/Action services can be viewed as wrappers around existing CIS data and functionality and support interoperability by presenting a unified view of clinical information systems to the guideline execution engine. The intention of the SAGE project is to align the VMR/Action services interfaces with standards such as HL7 messages and to propose these interfaces as standard access/action mechanisms into CIS data and functional elements. These interfaces could be used by non-CIS systems such as the guideline execution engine to interact with the CIS. It is also the intention of the SAGE project to use the same execution engine and event listener across any CIS that conforms to the specifications of the VMR/Action services.

On detecting a deployed guideline, the execution engine performs validation checks to ensure the correctness of the encoding. Once validated, the CIS events encoded in the guideline are registered with the CIS's event manager, thereby expressing the execution engine's interest in these CIS events. When a relevant event is detected, the engine begins execution.

Guideline Execution: Processing an Exemplar Guideline

Figure 2 shows one recommendation set from a guideline scenario based on the ICSI Immunization Guideline [15]. While real guidelines may include multiple recommendation sets, for simplicity, we will use a single recommendation set to illustrate the basic components of SAGE guideline execution. The clinical scenario for this recommendation set is assessment and recommendations of immunizations for a neonate. The primary logic is as follows: *Check the weight of the patient. If the weight is over 2kgs, check prior medications and determine vaccines that need to be administered. If the medical records do not indicate prior "consent to immunization" then the clinician is to obtain permission to administer the vaccines. If the patient is ill, the vaccines administrations are to be deferred and the deferral reasons documented. If the patient is not ill, the vaccines are to be administered to the patient. During the weight check, if the neonate is under weight (less than 2 kgs), the vaccine administration is to be deferred and the deferral reasons documented.*

In the execution of the exemplar immunization guideline, we assume a patient-guideline association has been established (either through an overt action by a clinician or through an automatic process), such that the patient is "enrolled" into this guideline. The SAGE engine uses this association to filter events that come from the CIS and to maintain guideline state for that patient/guideline combination.

The recommendation set in Figure 2 is represented as an activity graph [4,13]. An activity graph describes the relationship between the activities in the recommendation set as a process model and may contain Context nodes, Decision nodes, Decision Maps, Sub-guidelines, and Action nodes.

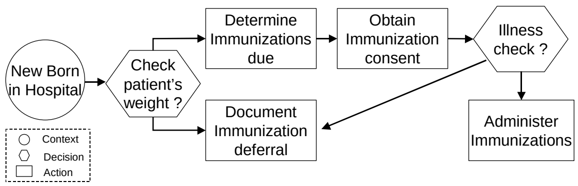

Figure 2 - Neonatal Immunization Activity Graph

Context Handling

Each activity graph segment within a guideline begins with a Context node that serves as a control point in guideline execution by specifying the clinical context for that segment. If the criteria associated with a context node are satisfied, subsequent nodes in the activity graph are executed until another Context node is reached. Context nodes may refer to clinical settings and resources and care-provider roles which have been modeled across the health care provider institution and used across guidelines.

In our example activity graph in Figure 2, the "New Born in Hospital" Context node specifies a clinical setting of *Inpatient Hospital* and a clinical role of *Pediatric Nurse*. This Context node also specifies a triggering event (*Inpatient Admission*). On receiving this event the execution engine processes the Context node. Our example Context node includes a patient specific precondition *Patient age < 7 days*, as a child below the age of 7 days is considered a neonate in the ICSI immunization guideline. Once the triggering event (*Inpatient Admission*) is received through the event listener, the engine evaluates the precondition associated with the Context node. If the precondition evaluates to False, the execution does not proceed any further. If the precondition is satisfied, the SAGE engine resolves the clinical settings and clinical roles associated with this Context node and moves to processing subsequent nodes.

Decisions and Criteria Processing

The second node in the activity graph of Figure 2 is a Decision node labeled "Check patient's weight?". A Decision node [4,13] in the SAGE guideline model represents clinical decision logic by listing alternatives (typically subsequent action nodes), and specifying the criteria that need to be met to reach those nodes. The execution engine will process all alternatives for which criteria are met, thereby allowing concurrent execution of multiple paths in an activity graph.

The SAGE guideline model and execution engine support the following specialized criteria:

- Comparison criterion: typically used to compare an object returned by a VMR service method against a constant. (e.g.: *# of polio vaccines in medical history is 0*)
- Temporal comparison criterion: a variant of the comparison criterion that compares the temporal relationship (before, after, concurrent, etc.) between the time when a VMR instance occurred and some time interval (e.g.: *was MMR vaccine given in the last 4 weeks*).
- Variable comparison criterion: The model allows variables to be defined across VMR service instances and some mathematical functions. This criterion allows comparison between variables and constants. (e.g.: *age > 2 months*, where age is a variable composed of date of birth).
- Presence criterion: checks for the presence or absence of coded concepts in instances of a VMR class within a valid time window (e.g.: *presence of infantile spasm as a Problem in patient's medical record*). Further specializations of this criterion, namely, Intervention presence criterion, observation presence criterion and allergy presence criterion have also been implemented.
- Boolean criterion: specifies True and False.
- N-ary criterion: A boolean combination (AND, OR, NOT) of the other criteria.

The SAGE execution engine will support other criterion types such as Goal Criterion that performs goal testing.

In the example activity graph in Figure 2, the Decision node "Patient's weight check?" specifies two alternative Action nodes ("Determine Immunizations due" and "Document Immunization deferral"), along with the decision criteria that must be satisfied to process those nodes. We will use this section to: a) describe how the guideline engine processes a decision node, and b) illustrate how the SAGE guideline model supports references to standard information models and terminologies. The ICSI immunization guideline [15] specifies that immunizations should be deferred for neonates under 2kg. For simplicity reasons, lets assume that the criteria associated with the nodes "Determine Immunizations due" and "Document Immunization deferral" (the nodes following "Patient's weight check") are *weight >= 2kg* and *weight < 2kg*, respectively.

As specified in the guideline model, the comparison criterion *weight < 2kg* specifies the VMR class (*Observation*), the SNOMED CT code for weight finding (*107647005*), the aggregate modifier for the weight observation (*most recent weight*), the operator to be used for comparison (*less than*) and the value to be compared against (*2 kg*). To evaluate the above criterion, the engine first makes a call to the terminology server to obtain all the terminology subsumption codes for the *weight finding*. The terminology server responds back with all codes that are subsumed by weight observation including the SNOMED CT code for normal weight (*43664005*). A VMR service call is then made to the Observation service of the CIS passing in the SNOMED CT codes, aggregate modifier (e.g: *most recent*), patient identifier, etc. (During the installation/binding process, CIS-specific codes have been "mapped" to the SNOMED CT standard terminology, thereby allowing the standards-based guideline engine side of the VMR service to interface with the parochial CIS side of the VMR service). On a query from the engine, the Observation VMR service returns the weight value and the units it is being returned in. The returned weight, the value specified in the criterion and the comparison operator are passed to an internal predicate evaluator. The evaluator does the necessary unit conversions, performs the predicate evaluation and returns the Boolean result. If multiple criteria are presented through a n-ary criterion, the above process is repeated and the results applied across the AND, OR and NOT operators specified in the n-ary criterion.

Actions

An Action node encapsulates a set of work items that must be performed by either a computer system or a person and is an example of a HL7 Reference Information Model Act [16]. Actions as implemented by the execution engine may be synchronous or asynchronous in their interaction with the CIS. In addition, Action nodes may specify preconditions that must be met before they are executed. In our sample activity graph (Figure 2), the Action node is a directive to the clinician (*pediatric nurse*) to obtain consent for immunization from the patient's guardian. The precondition here is an n-ary criterion composed of "*presence of any immunization due conclusions*" and "*absence of immunization consent in patient record*". In our test implementation, this inquiry to obtain the consent is presented to the clinician through the Notification (equivalent to an electronic mailbox) mechanism of the CIS. Since the immunization consent status (SNOMED CT: *243880000*) is the central theme of the inquiry, the valid responses are immunization consent given (SNOMED CT: *310375005*) or immunization consent not given (SNOMED CT: *310376006*) or the codes subsumed by either of the codes. The response from the clinician is recorded in the patient's medical record as an Observation for recording reasons. Furthermore, the engine detects this triggering event and processing continues.

To complete the recommendation set execution portrayed in our example activity graph (Figure 2), another Decision node ("Illness check") is visited and two alternatives are presented. If the patient is ill, the vaccine administration is deferred and appropriate documentation actions are initiated for the clinician to note the deferral reasons. If the patient is not ill, then Action node "Administer Immunizations" is processed. The engine calls the Orders Action service and uses the createPendingOrder method to place "pending" vaccine orders that were due for this patient, generates a Notification to the clinician informing them of the presence of a medication order waiting to be approved. This concludes this activity graph's execution.

Sub-Guidelines

In our sample activity graph, the node "Determine Immunizations due" is a specialized action node that includes a sub-guideline [4,13]. Sub-guidelines are reusable, self-contained statements of guideline logic – somewhat analogous to subroutines in a computer programming language. The sub-guideline in our example embeds a decision map [4,13], which in this case is used to compute which immunizations are due for a patient. The conclusions made during the processing of this decision map are stored in the patient's medical record through Action calls to the CIS.

State Management

The guideline execution engine executes in a stateless manner with some notable exceptions. The guideline encoder provides explicit entry points into activity graphs by marking certain Contexts as starting nodes. These context nodes have a triggering event (CIS or engine originating) that initiates the execution of the activity graph. Once execution begins, additional nodes are visited through the transitions (arrows of Figure 2) encoded in the recommendation set. It should be noted that several activity graphs can execute concurrently and several paths within an activity graph or a decision map may execute concurrently, resulting in multiple threads of guideline execution for a patient at a point in time.

Given this execution model, there are three ways execution can be blocked.

1. Execution can be interrupted when execution reaches a Context node that requires an external triggering event – until this event occurs that thread of execution is blocked.

2. Blocking can occur if a particular node needs to be executed at a later point in time. For example, if the encoding specifies "*Check for the presence of Hepatitis B Immune Globulin test 9 months from date of birth*", the engine would schedule a task to execute at that time and block the current node. When the engine's scheduler module detects the time has passed, it sends an internal event to the event listener. When this event is delivered to the engine, the blocked node is released and execution continues.

3. A thread can be blocked when the engine is waiting for an Inquiry to be answered. For example, let's assume that the Inquiry asks two separate questions "*Has the patient obtained MMR vaccine elsewhere?*" and "*Does the patient appear to be well today?*". Until a response comes back through the CIS for both these questions, that node stays in a PENDING state. As each response comes in, the engine stores the response - so that they may be used to make decisions in future criteria and for documentation reasons. The node state stays PENDING until all the responses are recorded upon which the state is changed to RESOLVED and the thread continues its execution.

Triggering Events

As shown in Figure 1, the event listener is the conduit by which the SAGE execution engine is notified about external events. The events themselves may be CIS generated or engine generated. We are currently defining a standard set of events that the SAGE engine can subscribe to and can be used during guideline encoding. During the installation/binding process, it will be the responsibility of the local care delivery institution to map their internal CIS events to standard SAGE events. This can be done declaratively in the event listener client (that publishes to the event listener) during the binding process of the guideline lifecycle.

Discussion

The challenge faced by the SAGE project is to deliver clinical practice guideline recommendations to clinicians as seamlessly as possible using native CIS applications and user interfaces. This must be achieved in a generic, interoperable manner so that the execution engine need not be rewritten for each CIS it needs to interact with. Moreover, all this must be achieved without requiring large changes to the existing functionality of CISs.

We have taken the first steps in achieving these goals. We have developed a guideline model that is based on standard information models, medical terminologies and HL7 data types. We have developed an engine to execute guidelines encoded using the SAGE guideline model. We have implemented an event listener that feeds the engine with external events. We have implemented the VMR/Action services for a commercial CIS, namely, IDX System Corporation's Carecast[tm], so that guideline interactions can be provided through the interfaces of the CIS. Each of these together forms an infrastructure for us to be able to execute an arbitrary clinical guideline encoded using the SAGE guideline model.

Using our infrastructure, we have encoded a version of the ICSI's Immunization guideline [15] and have executed guideline scenarios using the engine and Carecast[tm]. The encoding has three recommendation sets a neonatal immunization scenario, a primary care scenario that handles DTaP, Polio, Pneumococcal, Tetanus-Diptheria, Influenza, MMR, HiB, Hepatitis A and Hepatitis B vaccine administration for children and adult patients, a population-based reminder scenario to remind patients who may have missed their immunizations. The SAGE execution engine executes the above scenarios and we have integrated it against the VMR/Action services of Carecast[tm]. Using this integration we are able to deliver real-time immunization recommendations to clinicians.

The SAGE project has begun work on implementing the Immunization guideline encoding on a different CIS to show interoperability. Additionally, we have begun work on implementing the ADA Diabetes guideline [17] using the execution engine and Carecast[tm].

Acknowledgments

This work has been supported by the U.S. National Institute of Standards and Technology, Advanced Technology Program, Cooperative Agreement Number 70NANB1H3049. The authors also thank the participants of the SAGE project for detailed conversations on the guideline execution engine. SAGE project partners are: Apelon Inc., IDX Systems, Intermountain Health Care, Mayo Clinic - Rochester, Stanford Medical Informatics, and University of Nebraska Medical Center.

References

[1] Inst. of Medicine. Guidelines for Clinical practice: From Dev. to Use. Wash., DC: National Academy Press; 1992.

[2] Bero LA, Grilli R, Grimshaw JM, Harvey E, Oxman AD, Thomson MA:. Closing the gap between research and practice: an overview of systematic reviews of interventions to promote the implementation of research findings. *BMJ* 1998; 317:465-8.

[3] Zielstorff RD. Online Practice Guidelines: issues, obstacles, and future prospects. JAMIA, 1998; 3:227–36.

[4] Tu, SW, Campell J, Musen MA. The structure of guideline recommendations: a synthesis. AMIA Symposium; Washington, DC, 2003.

[5] Y. Shahar, S. Mirksch, P. Johnson. The Asgaard Project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. Artificial Intelligence in Medicine 14, 29-51, 1998.

[6] Musen, MA., Tu, SW, Das, AK, Shahar, Y. EON: A component-based approach to automation of protocol-directed therapy. J Am Med Inform Assoc. 1996; 3(6): 367-388.

[7] S.I. Herbert. Informatics for Care Protocols and Guidelines: Towards a Europeon Knowledge Model. in [7], 27-42.

[8] Ohno-Machado L, Gennari JH, Murphy SN, Jain NL, Tu SW, Oliver DE, Pattison-Gordon E, Greenes RA, Shortliffe EH, Barnett GO. The Guideline Interchange Format: A Model for Representing Guidelines. J Am Med Inform Assoc. 1998; 5(4): 357-372.

[9] Quaglini S, Stefanelli M, Lanzola G, Caporusso V, Panzarasa S. Flexible guideline-based patient careflow systems. Artif Intell Med 2001;22:65–80.

[10]J. Fox, N. Johns, A. Rahmanzadeh, R. Thomson. Disseminating medical knowledge: the PROforma approach. AI in Medicine 14, 157-181, 1998.

[11]Johnson PD, Tu SW, Booth N, Sugden B, Purves I. Using Scenarios in Chronic disease management guidelines for primary care. Proc AMIA Annu Fall Symp 2000:389–393.

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

[13]Campbell, J., Tu, S.W., Mansfield, G., et al. (2003). The SAGE Guideline Model: A Knowledge Representation Framework for Encoding Interoperable CPGs. Stanford Medical Informatics Report SMI-2003-0962 http://smi-web.stanford.edu/pubs/SMI_Abstracts/SMI-2002-0962.html

[14]W3C Web services site http://www.w3.org/2002/ws/

[15]Institute of Clinical Sys. Improv. Immunization guideline http://www.icsi.org/knowledge/detail.asp?catID=29&itemID=174

[16]Schadow, G, Russler, DC, McDonald, CJ. Conceptual Alignment of Electronic Health Record Data with Guideline and Workflow Knowledge, Intl. J. Med. Info 2001; 64:259-274.

[17]American Diabetes Association Diabetes Guideline http://care.diabetesjournals.org/cgi/content/full/25/1/148

[18]Johnson PD, Tu SW, et al. A virtual medical record for guideline-based decision support. Proc AMIA Symposium. 2001; pp. 294-298.

Address for correspondence

Prabhu Ram PhD,
925 Fourth Avenue, Suite 400,
Seattle, WA 98104. USA.
prabhu_ram@idx.com
(206) 609 5971