

SAGEDesktop: An Environment for Testing Clinical Practice Guidelines

David Berg BA¹, Prabhu Ram PhD¹, Julie Glasgow MD¹, Jonathan Castro MD²

¹Health Informatics, IDX Systems Corporation, Seattle, WA, USA

²Biomedical and Health Informatics, University of Washington, Seattle, WA, USA

Abstract—Clinical practice guidelines have begun to appear in electronic form with the expectation that they will be used to provide automated clinical decision support. Systems capable of executing these clinical practice guidelines are emerging concurrently. A few research projects have developed guideline authoring tools that allow encoding electronic clinical practice guidelines. However, we are not aware of tools that allow testing of electronic clinical practice guidelines by their authors. We have developed a tool called SAGEDesktop for the testing of electronic clinical practice guidelines. SAGEDesktop does not require external infrastructure to carry out this testing. It includes: 1) an embedded guideline engine which can execute electronic clinical practice guidelines 2) functionality that emulates the capabilities of a clinical information system necessary for the execution of the guideline, and 3) terminology server functionality that encapsulates standard clinical terms. Coupled with an authoring tool, SAGEDesktop provides a complete development environment in which precise and robust electronic clinical practice guidelines can be encoded and tested. We describe the features of SAGEDesktop, narrate our vision of the tool's use, and discuss SAGEDesktop's strengths and weaknesses.

Keywords— clinical decision support, clinical information systems, clinical practice guidelines, computerized patient records, guideline encoding, guideline testing, healthcare infrastructure.

I. INTRODUCTION

Evidence-based clinical practice guidelines (CPGs) 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 brought minimal improvement to clinical care [2]. Recent studies suggest the impact of CPGs may best be realized via guideline-driven computerized decision support that is integrated into the real-time care workflow [3].

Several research efforts [4] have focused on developing models for representing CPGs. Some of these projects have developed authoring tools to encode CPGs. While these tools allow the encoding of CPGs into a computable format, utilities have been lacking that allow testing of electronic CPG (eCPG) by their encoders during the encoding process. Rather, this testing is typically done in the presence of a fully functional clinical information system (CIS) and guideline execution automation. This approach is

impractical for its high monetary cost, high resource usage, complexity, and deployment and maintenance difficulties.

In this paper we report on a tool called the SAGEDesktop, which we have developed to enable eCPG authors to test their encoding in the absence of an explicit guideline execution engine, a CIS and a terminology server. SAGEDesktop allows the encoder to rapidly and easily test the encoding, analyzing its behavior in the presence of encoder-controlled clinical events and clinical data. This allows iterative modification of the eCPG to correct errors and unintended side effects. Coupled with an authoring tool, SAGEDesktop forms a complete integrated development environment in which the encoder creates, tests, and improves eCPGs without incurring a prohibitive resource overhead.

We begin with a brief description of a project called SAGE (Standards-based Sharable Active Guideline Environment), for which the SAGEDesktop was developed. We present a brief narrative of how eCPGs are authored, and describe how these eCPGs are tested by their encoders in the SAGE project using SAGEDesktop. We describe SAGEDesktop's functionality and how it circumvents the need for complex infrastructure components for testing eCPGs. We conclude with a discussion of the benefits and weaknesses.

II. The SAGE Project

The SAGE project is a collaborative research and development effort among software vendors, clinical institutions and academia to develop a standards-based comprehensive technology infrastructure that will enable encoding, dissemination and deployment of eCPGs. The SAGE project partners are: Apelon Inc., IDX Systems, Intermountain Health Care, Mayo Clinic - Rochester, Stanford Medical Informatics, and the University of Nebraska Medical Center.

A SAGE deployment consists of a SAGE guideline engine server (for execution of eCPGs), a CIS (for its user interface and storage of patient data) and a terminology server (for standard clinical terminology reference, and other services such as terminology subsumption). End-user interaction with the guideline is done through the CIS, for example when a SAGE-recommended medication order is placed through the CIS order entry system, or a SAGE-generated alert is delivered to a clinician through the CIS' Inbox mechanism. For additional information on the SAGE project, please see the SAGE website [5].

III. Electronic Clinical Practice Guidelines

Clinical standards organizations such as Institute for Clinical Systems Improvement (ICSI) and American Diabetes Association (ADA) publish CPGs today essentially as documents to be read by clinicians, who are then expected to carry that knowledge into their clinical practice. With the advent of systems such as the SAGE guideline engine [6], it is expected that CPG-producing organizations will also publish their guidelines in electronic form as eCPGs. Individual institutions (e.g.: hospitals) will download these eCPGs, customize them to fit their local needs, and deploy them to function with the institution's CIS.

The SAGE project provides eCPG authors with a Protégé based Workbench [7] and a Knowledge-Acquisition Wizard called KWIZ [8] to encode eCPGs. The encoder begins with a definitive source guideline, such as the one produced by ICSI for Immunizations [9]. These evidence-based guidelines provide human-consumable protocols and recommendations for the treatment of specific disorders. The major tasks of the encoders in translating a published paper guideline into an eCPG involve 1) disambiguating clinical concepts contained within the guideline, and 2) creating a workflow-based framework into which the guideline logic is integrated.

The clinical concepts used by a guideline range from simple patient data (e.g.: a patient's age) to complex ontological constructs (e.g.: that subset of cardiovascular diseases that are relevant to a patient's Type II Diabetes). All must be encoded using standard representations, which for the SAGE project have been based on SNOMED-CT and LOINC [10, 11]. The source of the patient data used by the guideline logic may be the computerized patient record, the clinician user, or sometimes, both.

Clinical workflow is captured in Protégé using Activity Graphs, which represent sequences of related clinical contexts, decisions, and actions [12]. The guideline logic encoded therein is initiated by context-associated triggers. These triggers reflect events that occur during the course of CIS execution, e.g. a patient checking in to a primary care clinic, or a physician making an entry into a patient's problem list. Figure 1 shows the hypertension management activity graph of the SAGE-encoded Diabetes Mellitus guideline as an example.

IV. SAGEDesktop

eCPGs must be significantly more precise than human-readable CPGs if they are to provide reliable automated clinical decision support. For instance, the ICSI Immunizations guideline lists "progressive encephalopathy" as a contraindication for the DTaP vaccine. While it is straightforward for a clinician to identify a "progressive encephalopathy" in a patient, it is impossible for a computer to do so without a more exact definition. In order to arrive

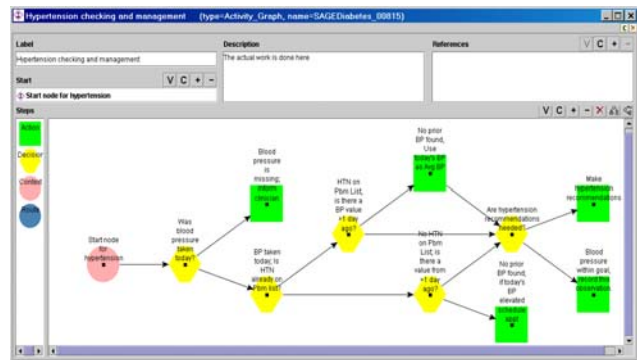


Fig. 1. A SAGE eCPG Activity Graph

at a sufficiently precise encoding, it is crucial that eCPG encoders test guideline functionality as it would be executed. If an eCPG is first tested during deployment (as by a hospital's IT staff), significant problems are likely to be identified which cannot be rectified quickly or easily. Conversely, supplying an eCPG encoder with a guideline engine, a CIS, and a terminology server is not practical because of the complexity and costs involved with deployment.

The primary purpose of SAGEDesktop is to provide an environment in which encoders may test their eCPGs, allowing early correction of unintended behavior. SAGEDesktop is implemented as a Protégé workbench plug-in. In the SAGE project, encoders author their eCPGs in Protégé/KWIZ, using SAGEDesktop to examine the behavior of the encoded guideline. SAGEDesktop embeds a version of the SAGE guideline engine, and provides functionality equivalent to a CIS and a terminology mapper (surrogate for a terminology server). With these SAGEDesktop features, the encoder has the benefit of a complete environment in which to test eCPGs while avoiding the necessity of a fully functional guideline deployment environment (i.e. a guideline execution engine, a full CIS, and a terminology server).

A. Guideline Validation

Before "execution" of an eCPG in SAGEDesktop, the eCPG must be validated to ensure that it conforms to the guideline model constructs implemented by the embedded guideline engine. This validation is analogous to a computer programming language compiler checking the syntax of the program it is attempting to compile. Validation errors and warnings are presented to the user to be addressed. Typical validation errors involve missing information in the encoding, and inappropriate objects used in a guideline construct. Figure 2 shows two validation errors in the SAGE Diabetes Mellitus encoding.

B. SAGEDesktop CIS Functionality

To inspect the behavior of an eCPG, the guideline has

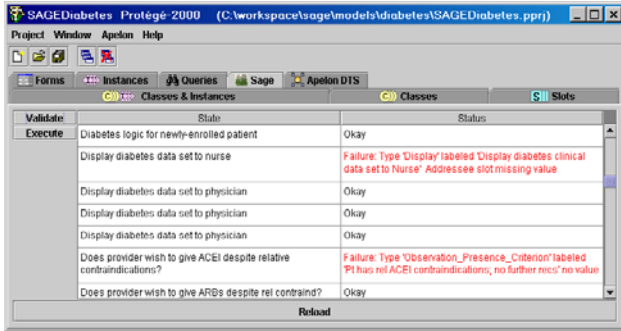


Fig. 2. SAGEDesktop Validation with Errors

to be executed in the context of a patient profile. In a deployed system, such data would reside in the CIS. In addition, the guideline engine must listen for clinical triggering events to activate relevant guideline logic. Furthermore, the guideline may need to obtain clinician responses to guideline inquiries in order to continue execution. SAGEDesktop provides three data entry windows to simulate these CIS interactions.

The first window – labeled Clinical Concepts in Figure 3 – represents the clinical data associated with a patient record (e.g.: a patient’s allergic reaction to Thiazide medications). Each row comprises two fields: an *instance* and a *value*. The instances have been extracted from the eCPG by SAGEDesktop, and defaults filled in for their values. The instances column shows the label of the clinical concept. The values are editable, allowing the encoder to key in appropriate values. The values are unit dependant (e.g.: mmHg for blood pressure readings, kg for weight, etc.), and based on the value type different visual controls are presented to the user. For example, if the patient’s date of birth is sought, a date object (e.g.: 11/18/1970 08:10:00) is presented. If a goal range is to be shown, a range object is presented (e.g.: 45-150).

The second window – labeled Clinical Events in Figure 3 – is a mechanism for simulated clinical events (e.g. an outpatient clinic check-in event) to be submitted to the embedded guideline engine. In SAGEDesktop, the clinical events are presented in a drop-down list which SAGEDesktop has pre-populated from the eCPG. The event chosen by the user is submitted for processing by pressing the *run* button.

The third window – labeled Inquiries in Figure 3 – represents interaction with the clinician that may occur during the course of guideline execution. Depending on the type of inquiry, the responses are presented as radio buttons (e.g.: Tobacco user, Non-smoker, Smoker for an inquiry relating to the patient’s Tobacco use status) or as text boxes for value and unit entry (e.g.: Enter patient’s recent heart rate in beats/min). The user picks or enters their responses and *submits* them to SAGEDesktop. Inquiries appear and disappear in this window based on guideline logic.

C. Execution Results

When an event is submitted to SAGEDesktop, the embedded engine processes the eCPG and the labels of the processed nodes are shown in the Execution Results window (Figure 3). The traces are color coded and prefixed with the kind of guideline object being processed. The traces are indented to show the relative context of the processed nodes, as an encoded guideline may contain sub-guidelines [12].

SAGEDesktop also writes out a log file in the background which contains the execution traces. The detail level (1-5) of this log file is set by a configuration parameter, with a larger number providing more details.

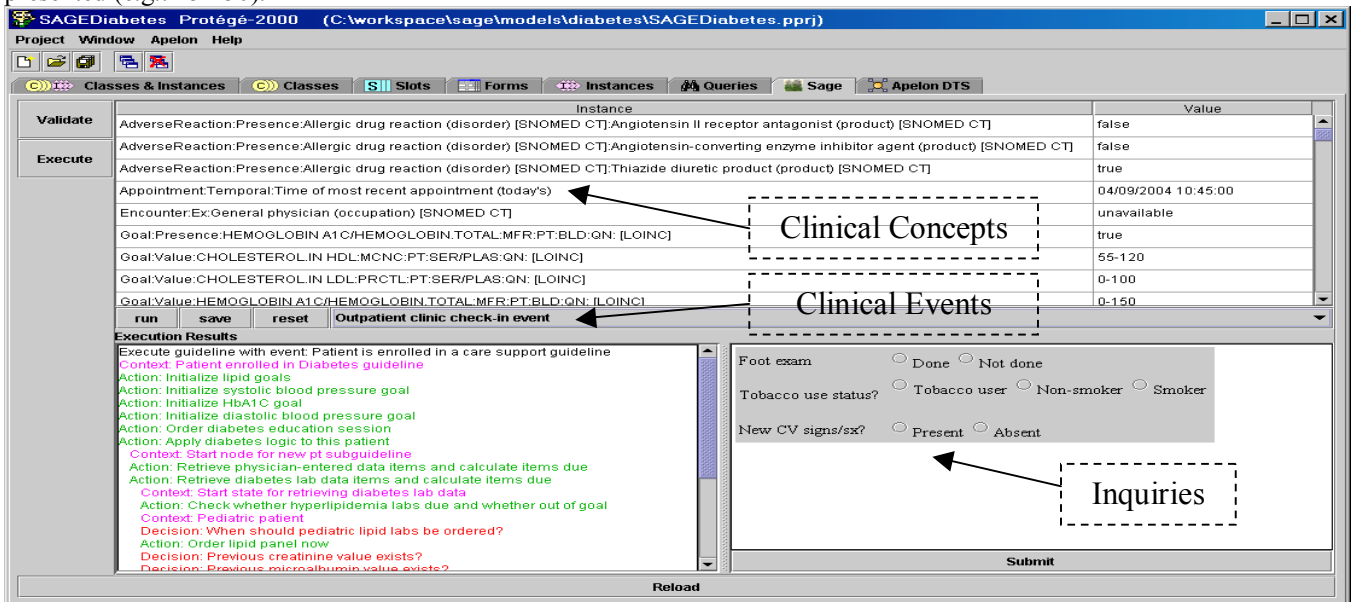


Fig. 3. SAGEDesktop User Interface

D. Miscellaneous Features

A terminology server is used in the deployment environment during eCPG execution to provide clinical terminology references and subsumption services. The SAGE project uses a terminology server called Apelon Distributed Terminology Server [13]. To mimic the terminology server's functionalities, SAGEDesktop implements a terminology mapper that is composed by extracting the standard terms from the eCPG and making them available to the embedded guideline execution engine.

To preserve the entered values in the Clinical Context window for future use, a *save* button is provided in SAGEDesktop. A *reset* button in SAGEDesktop allows the internal state of the embedded guideline engine to be reset. There is a *reload* button that reloads the eCPG into SAGEDesktop. The reload functionality was implemented because it is possible that the eCPG has been modified since it was last read in by SAGEDesktop.

E. Typical Development Session with SAGEDesktop

A typical encoding session using Protégé/KWIZ and SAGEDesktop progresses as follows. Encoders interpret the knowledge in the paper CPG, modeling the content in Protégé/KWIZ as activity graphs and decision maps using the SAGE guideline model [12]. Once encoding has progressed sufficiently to test the behavior of the eCPG, the validation functionality of SAGEDesktop is used to check for syntactic correctness. After all the validation errors and warnings have been addressed, the encoder begins testing eCPG execution using SAGEDesktop. The tool loads all the instances and default values into the Clinical Context window, and the clinical triggering events into the Clinical Events window. The Inquiries and Execution Results windows are left blank. In the Clinical Context window, the encoder modifies the patient data they expect to have impact on execution of their encoding. They then choose the appropriate clinical triggering event, and begin execution of the eCPG by pressing the *run* button. They continue to modify the values in the Clinical Context window as needed, submitting clinician input via the Inquiries window and triggering additional events in the Clinical Events window as the clinical workflow dictates. They follow the execution of their encoding in the Execution Results window, modify their encoding to correct revealed problems, and repeat the process of testing and modification to refine the eCPG.

IV. DISCUSSION

The current implementation of SAGEDesktop allows it to work only with one eCPG at a time. In deployments, it is likely that multiple eCPGs will be executing and it would be

useful for an eCPG encoder to be able to test interactions resulting from multiple guidelines. The current implementation's eCPG validation functionality does not link errors/warnings directly with the guideline constructs and depends on the encoder using the label of the offending construct in the validation window and locating the node in Protégé/KWIZ to address the error.

Despite these addressable shortcomings, giving the encoder the capability to test and refine the eCPG using the SAGEDesktop significantly increases the encoder's ability to author precise eCPGs. We have seen in the SAGE project that using the SAGEDesktop has been invaluable in improving the correctness of our eCPGs. This has been demonstrated in the SAGE project to deliver more reliable automated clinical decision support.

ACKNOWLEDGMENT

This work has been supported by the U.S. National Institute of Standards and Technology, Advanced Technology Program, Cooperative Agreement Number 70NANB1H3049.

REFERENCES

- [1] Inst of Medicine. Guidelines for Clinical Practice: From Development to Use. Washington, DC: National Academy Press; 1992.
- [2] L.A. Bero, J.M. Grimshaw, E. Harvey, et al. "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] R.D. Zielstorff. "Online practice guidelines: issues, obstacles, and future prospects" *JAMIA* 1998; 3:227-36.
- [4] M. Peleg, S. Tu, J. Bury, et al.. "Comparing Computer-interpretable guideline models: A case-study approach". *JAMIA* 2003;10:52-68.
- [5] The SAGE Project website. <http://www.sageproject.com>.
- [6] P. Ram, D. Berg, S. W. Tu, et al. "Executing clinical practice guidelines using the SAGE execution engine". *Proceedings of MEDINFO 2004* - in press.
- [7] J.H. Gennari, M. Musen, R. Fergerson, et al. "The Evolution of Protégé: An Environment for Knowledge-Based Systems Development" *Int Journal of Human-Computer Interaction*. 2003; 58(1): 89-123.
- [8] R.D. Shankar, S.W. Tu, N. Jones, M. Musen. "Kwiz, a Knowledge-acquisition Framework to Encode Guidelines". *Proceedings of MEDINFO 2004* - in press.
- [9] Institute for Clinical Systems Improvement. "Health Care Guideline:Immunizations". <http://www.icsi.org/knowledge/detail.asp?catID=29&itemID=174>.
- [10] R.A. Cote, D.J. Rothwell, R. Beckett, J. Palotay (eds). SNOMED International. College of American Pathologists, Chicago, 1993.
- [11] C. McDonald, S. Huff, J. Suico, et al. "LOINC, a universal standard in identifying laboratory observations: a 5-year update." *Clinical Chem*. 2003; 49(4):624-633.
- [12] S.W. Tu, M.A. Musen, R. Shankar. "Modeling Guidelines for Integration into Clinical Workflow". *Proceedings of MEDINFO 2004* - in press.
- [13] Apelon website, <http://www.apelon.com>.