HOLON/CADSE: Integrating Open Software Standards and Formal Methods to Generate Guideline-Based Decision Support Agents

Barry G. Silverman, PhD[1], Oleg Sokolsky, PhD[1], Val Tannen, PhD[1], Alex Wong[2], Lance Lang, MD[3], Allan Khoury, MD[3], Keith Campbell, MD, PhD[3], Chen Qiang, MS[1], Arnaud Sahuguet, MS[1]
1-University of Pennsylvania, 2-Sun Microsystems, 3-Kaiser Permanente
Contact: barryg@seas.upenn.edu

ABSTRACT

This paper describes the efforts of a consortium that is trying to develop and validate formal methods and a meta-environment for authoring, checking, and maintaining a large repository of machine executable practice guidelines. The goal is to integrate and extend a number of open software standards so that guidelines in the meta-environment become a resource that any vendor can plug their applications into and run in their proprietary environment provided they conform to the interface standards.

1) CHALLENGE

There is currently a large and rapidly growing body of medical practice guideline documents, both hard copy and electronic, covering every aspect of healthcare delivery. These documents are vital to the extent they help care givers apply the evidence base and consensus knowledge of expert panels of specialists. When followed, such guidelines assure the latest knowledge is used and that prevailing practice is consistent across the healthcare organization and network.

The difficulty is that providers are confronted by a sea of passive guideline documents, and have little time to locate and extract the precise passages that will help them in the current patient episode. Further, even if they find the relevant passages, too often they are incomplete, vaguely worded, and/or not tailored to local practice needs, workflows, and populations. What is needed instead is an active agent approach that anticipates patient episodes, recognizes and retrieves relevant guideline materials, and pushes those materials to the screen at the appropriate interval in a locally useful mode. The challenge is to turn passively searchable guideline repositories into active, useful agent-push approaches.

This challenge is too vast for any one organization to accomplish alone. No one has the resources to program and maintain the many 10s of 1,000s of rules, frames, XML tags, etc. associated with the VERY large knowledge base that guidelines represent. What is needed instead is an environment that utilizes open standards, permits cross-organization authoring of guidelines with appropriate management of copyright and ownership, and provides public interfaces that permit any vendor's decision support tools to display the guidelines within their proprietary interfaces. Our consortium is seeking to develop and validate scalable methodologies and applications that will support this approach. We call this the Computer Aided Decision Support Environment (CADSE).

Its true that most patient record and workflow system vendors today offer methods for entering and displaying rules, templates, evidence, and so on. Yet these applications are invariably monolithic in that knowledge must be authored in their software or it won't run, and often that authoring process is arcane and depends on the vendor to do it for each client separately. Worse yet, none of these vendors address the life cycle of "knowledge management" problems that are the true cost drivers of such a knowledge base – the verification, validation, constant local tailoring, and ongoing maintenance needs. Yet, experience in long-lived software systems (e.g., the military) shows that these items are 50 to 90% of total software cost: e.g., see [2] among others.

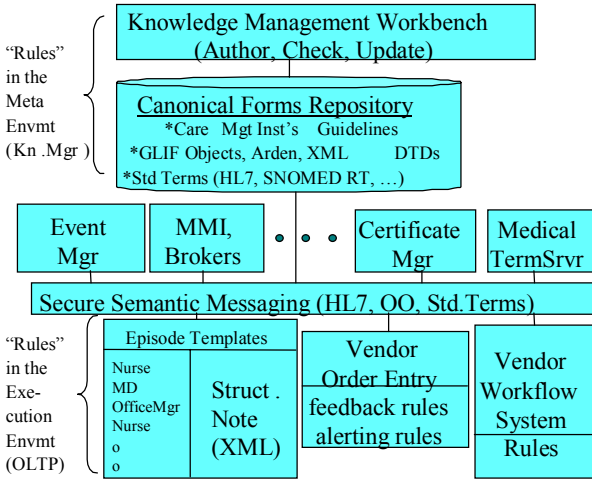
2) KNOWLEDGE MANAGEMENT

In short, one cannot meet the challenge posed here without a knowledge life cycle management environment. This environment is meta-level in that it allows the guideline knowledge to execute in any vendor's software, but permits the management of the life cycle steps (authoring, checking, updating) to be performed by clinicians at a level above the vendor software – in a high level English-like and visual programming mode.

Figure 1 illustrates the elements of the environment we believe are needed to meet the challenge. This design removes guideline knowledge from vendor control, and makes it a resource and service the vendors obtain from the operating system, much as they obtain other services shown in the middle of Fig.1 (e.g., name servers/brokers, encryption, message routing, terminology service, etc.). Across

the base of Figure 1 are the vendor software systems that use these services and that execute practice rule knowledge sets during patient encounters and note taking, during order entry, and via workflow approaches. For these systems to work effectively, guideline knowledge must be locally tailored and adapted to each clinic's workflow and rules, plus it must be parsed into diverse vendor software.

Figure 1 – CADSE Shown As A Distributed Operating System Facility



To support this parsing, the meta-environment, CADSE, represents the guideline knowledge in an open standards, canonical form for which parsers can readily be deployed to translate it to any vendor's format, although vendors will have to support such interfaces. This canonical form repository embraces several standards that assure semantics, syntax, and terminology of guideline knowledge and message sets are widely understood, easily deployed, and readily maintained. To this end we support the following standards, the integration and extension of which are addressed in the next section: the HOLON architecture [10], the InterMed Guideline Interchange Format (GLIF) –[9], Arden Syntax for procedures, the ANSI HL7 committee's XML/PRA standard for message set and patient record element naming [1], and common medical terminologies embraced by HL7 and the NLM, such as SNOWMED RT, drug codes, and procedure codes, among others: e.g., see [3] among others.

To support the authoring, checking, local adapting, and updating of guideline knowledge the top of Figure 1 shows a web-enabled distributed editing and configuration managing environment. This environment collects and displays guideline knowledge and other forms of evidence via the help of models of generic tasks in guideline authoring [4], skeletal plan refinement [8], terminology-enabled elicitation, visual programming, terminology server support [3], and critiquing [12]. Guidelines so authored, are readily parsed into the canonical forms, and later extended for local adaptation and or updating purposes. A key design idea for this knowledge management workbench is to facilitate parallel authoring of the same guidelines by diverse specialists so as to collect different evidence and viewpoints, place them under configuration control, and resolve conflicts.

3) RESULTS TO DATE

This is the first of a three year effort to construct the workbench, repository structure, configuration management tools, and distributed operating environment that will meet the nation's guideline authoring challenge. Its not our goal to author all the needed guidelines with all their local adaptations. Rather, we are trying to create a robust, scalable, standards-based environment that clinicians and vendors alike will accept as a viable way for them to reach the goal.

To assure our designs are robust, scalable, etc., we need to author, check, adapt, and update some example guidelines and deploy them in a representative sample of execution environments. The Kaiser Permanente's national clinical information system will be the first test environment, slated to begin in 2001. We are actively seeking other partners to further support this approach and test out this environment.

In 1999/2000, our attention is focused on integrating and extending the various tools of the workbench and standards of the repository.

3.1) Constructs for Elciting Knowledge

One of the easiest ways to elicit and check knowledge is if the editor has a structural model and ontology of the domain of guideline authoring that it can use to: (1) delineate the types or flavors of guidelines and for each of these their chapters and sections; (2) assist the writing task via structured-English forms of pickable sentence elements and visual pallets of semantic primitives for flowcharting; (3) speed and further standardize the sentence authoring and flowcharting via the help of standardized pick lists of appropriate vocabulary and terms; (4) organize, structure, and index knowledge authored thus far; (5) detect what knowledge still needs to be added (completeness); and (6) check authored knowledge for coherence issues (consistency, circularities, dead end chains, etc.).

To date we have tested these elements separately. So one of our prototype guideline editors, R2Do2 [7, 11] includes web forms for structured-English

elicitation of IF-THEN-ELSE sentences with pull down menus of available (and extendable) parameters, parameter values, boolean connectors, conditionals, action options, and so on. Users author practice rules one at a time, but get to see the combined results of all rules in a decision table (matrix) showing parameters and actions as column headings and rules as rows. Another section of R2Do2 elicits questionnaires useful for patient interviews. We have also recently added a visual pallet of GLIF semantic primitives that can be used to interconnect decision tables and/or their parts via flowcharts. The R2Do2 environment also includes a parser that converts the structured English, tables, and charts into executable code, including CLIPS rules and XML documents. Not yet integrated, but currently working in various formats are extensions discussed in the following sections.

In a trial, the R2Do2 environment exhibited speed up and enhanced executability of practice rule authoring [11]. More importantly, this trial revealed we need to add to this authoring environment several higher level elements of the structural model of a guideline. So, the apparatus described thus far might be deployed differently for each of the five types of guidelines mentioned in the Inst. Of Medicine report (IoM, 1992), for each of several sections of a given guideline (eligibility criteria, disease severity rating, episode and disease note-taking templates, order entry, results reporting, etc.); for authoring the workflow rules that indicate who are the users of each portion of a guideline (and when); and for collecting enterprise level workflow rules that assure coordination of cross guideline workflows and personnel. We are currently extending R2Do2 in these directions.

3.2) Distributed Configuration Management

Also, we believe the best way to author and manage the updating of guidelines is to provide a distributed authoring and checking tool to collaborative authors across the enterprise(s) and for a central staff to facilitate the integration and conflict resolution of different versions the authors generate. The Convergent Medical Terminology (CMT) project has several years experience supporting just this type of distributed development of description-logic based terminologies [4]. Commercial applications based on methodologies developed for that project are available (vendors with CMT capable tools include Lexical Technology, Ontyx, and IBM). Many of these tools are being used in the development of (1) SNOMED-RT where collaborative work among a consortium of participants is required, and (2) enterprise-specific terminologies where all of the participants in the process are members of the same organization, but may be geographically distributed across the enterprise.

The CMT configuration management environment is a general purpose environment that no longer requires that the items brought under configuration control have a description-logic foundation. Rather it can manage any content that can be represented using a directed acyclic graph (DAG) data structure. We believe that this DAG foundation provides an opportunity for managing decision-support content (rules, guidelines, skeletal plans, etc) and plan to either develop new--or extend existing--decision-support authoring software so that this software is compatible with distributed development processes. We will openly publish these processes so that any software vendor may make tools that are compliant.

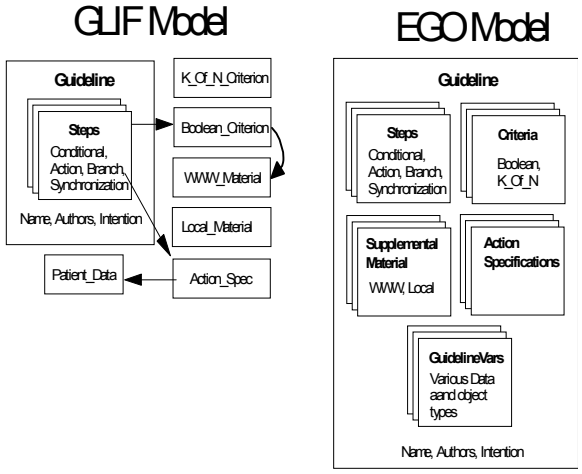
3.3) Extending the GLIF Approach

One of the ingredients that will foster vendor independent guidelines is use of the Intermed Collaboratory's Guideline Interchange Format (GLIF) [9] as a starting point for representing executable decision support logic. GLIF is an object model that is under active development by the Intermed. Its short term objectives centered on formal representation of procedural health care knowledge and associated workflow in an exchangeable format. Long term objectives include machine based execution of guidelines. As such, the GLIF specification served as an excellent starting point for our efforts to develop Executable GLIF Objects (EGOs).

Before embarking on a comparative description of the GLIF and EGO architectures, it is useful to describe some of the roles that EGO is intended to fulfill. First, EGOs must have a means to employ logical formalisms that allow the protocol and algorithms to be dynamically extended in a principled way. Specifically, the formalism we pursue is that of "process algebra", although we are also utilizing classical temporal algebras. Process algebra is a formalism that represents processes as state machines, that is, graphs with labelled transitions. There is a large body of theory and mathematics concerning process specification and analysis [13]. Since guidelines are essentially algorithmic, and consist of a series of actions, they may be represented as graphs with labelled nodes and edges. Although Intermed has not attempted formal methods, GLIF format supports this apporach and there is a natural correspondence between guidelines and processes. By utilizing the process algebraic approach we gain a rigor and reliability in the algorithms (they can be mathematically verified), plus we hope to reuse some of the acyclic graph maintenance tools of the preceding section to help manage a large collection.

EGOs should be convertible to XML documents and vice-versa to facilitate auditing, distributed authoring and the like. Vendor applications need to access EGOs, so interfaces also are needed to support distributed components and wrapped legacy datasets. We call these GLIF Action and Data Managers (GADMs). Finally, EGOs are intended as classes that get instantiated into patient guideline objects. These in turn are then worked on throughout the workflow and by different actors (roles) to fill in the blanks as patient episodes and care occur. For these reasons, EGO requires dictionaries to define the data, action, and role expectations guidelines impose on vendor applications and on healthcare workers.

Fig. 2 – Executable Guideline Objects (EGOs) are a Re-Organization and Extension of GLIF



We have realized several of these preceding goals simply by re-factoring the GLIF design, fleshing out some of the needed class methods, and adding the ExtensibleObject base class. Figure 2 shows how GLIF and EGO differ in their approach to organizing object instances. GLIF's object model centers around the Guideline Object. This Guideline Object is essentially a collection of Steps (and properties which are not shown). Surrounding the Guideline Object are various Criterion objects, Patient Data objects, and SupplementalMaterial. A Guideline is built by establishing references between these objects (e.g. a ConditionalStep references a BooleanCriterion which references a WWW_Material object). The GLIF object model provides no formal mathematical grounding, no terminology, and it ignores the vendor interfaces (GADM and roles paradigms).

In the EGO approach, the Guideline Object functions as a mini-repository of all objects that are used within the guideline. This simple transformation eases the tasks of constructing an EGO, browsing its constituent parts, and most importantly it simplifies the task of containment. Applications containing EGO Guidelines only need to know about a Guideline -- they do not need to know about its constituent parts. Further, as mentioned above but not shown in the figure, each EGO guideline has dictionaries (data, action, roles), terminology enablement, and process graphs.

To do this, we had to extend the GLIF's criterion logic and provide a mathematical grounding based on process and temporal algebra, as mentioned earlier. At present the GLIF has neither primitives for building criteria statements nor terminology standards, but allows free text statements instead. This makes it difficult to manage terminology, to verify logical actions, and to handle data and actions (GADM). In our R2Do2 protoype we adopt a "natural form" to elicit Arden type primitives (parameters, booleans, predicates, valid parameter states, etc.) via pull down menus in a structured English crtierion, conditional, or If-Then-Else type of sentence: e.g., see [7, 11]. In the coming months we hope to also integrate the CMT term server within our parameter name picker menu, and to add mathematically rigorous verification procedures.

In summary, we hope that GLIF and these extensions will lead to several benefits for our project. Authors should simultaneously be grounded in proper semantics, syntax, and terminology yet have the flexibility to author all of the sub-parts of guidelines of differing flavors in a semi-structured English and/or visual representation. To date our authors in R2Do2 create criteria, conditionals, rules, etc. over the web in structured English forms. Reaction to these forms, collected from over a dozen MDs who authored guideline shreds in R2Do2, indicate they are usable and intuitive though incomplete in terms of the range of material that needs to be expressed.

As a guideline is completed, it grows into a rather large set of sub-classes, attributes, and so on. Authors and maintainers need visualization tools to view the hierarchy, navigate and zoom in/out, and work on it. To date, R2Do2 lets users view guidelines within decision tables of criteria, conditions, actions, and rules. These are useful, but user reactions indicate other display forms are also required such as flowcharts, graphs of expandable decision tables, etc.

Another benefit our approach should provide lies in path-free access to patient data and results via the use of standards-based term naming and the GADM interfaces. Path-free access is supported via the HOLON compliance model which includes CORBA IDL, name servers and data brokers, XML and ODL/OQL-based information mediator agents, and open messaging and interfacing standards including HL7/PRA, CMT, and GADM. That is, if all guideline

data elements are standards-based, and all data sources are wrapped to the same standards, then (publish and subscribe-based) brokers and middleware can handle the paths and interfaces. That is the purpose of pursuing HOLON-compliance.

3.4) Execution Layer Interfaces

Parsing EGOs into the execution layer involves developing an interface that vendors can use to access the guideline facility and run the guidelines in their proprietary environments. We are starting with several necessary, but not sufficient assumptions for this interface including: (1) the clinical computing environment includes vendor components for documenting progress notes, for order entry, for workflow management, and for results reporting; (2) episode charting and progress notes occur in a documentation markup format (e.g., XML) that utilizes hierarchies of templates of structured, fill-in-the blanks notes; (3) the vendors author translators that can converse with the GADM specified interfaces to the GLIF objects; (4) the execution environment includes a guideline matchmaker service that identifies which guidelines should fire for a specific patient episode; (5) access to patient data needed by rules, templates, etc. is compliant with GADM, HL7 PRA, HOLON protocols, and CMT; and (6) mediator agents required to translate patient data to the levels useful by guideline rules (e.g, date-of-birth to age, stripping the most recent reading off a list, etc.) are written in Java, make use of the K2 wrapping tool at Penn [14], and are OQL/ODL/HL7-XML/HOLON/CMT compliant.

We are currently constructing a simulation testbed of the clinical computing environment's vendor components in order to test the necessity of each of the previous assumptions. We also hope to discover the sufficiency conditions for these interfaces. Our goal over the next year is to derive the necessary and sufficient set of standards that vendors must conform to if they hope to utilize the guideline facility.

4) CONCLUDING REMARKS

We are convinced of two things: (1) our ability to improve quality and manage costs in healthcare will be determined by how efficiently and effectively we develop/implement decision support for clinicians, and (2) delivering decision support in the form of guideline document repositories is a passive, user-pull approach that quickly loses effectiveness as the number of documents and their complexity grows.

At the same time, we do not blindly believe the agent push approach is a panacea. Managing the life cycle of practice guideline knowledge is a challenging problem that requires a large investment in methodologies, models, standards, formal methods, tools, repository technology, detailed prototype analyses and testing, and scale up if it is to be successful. As this article describes in detail, we are only at the beginning of what promises to be an ambitious journey, one we are not sure we can fully complete. We do know the best way to get there is to increase the rigor and open-ness of our solutions.

ACKNOWLEDGEMENT -- This work was supported partly by a grant from NIST for HOLON, and partly by support from Kaiser Permanente.

REFERENCES

1. ANSI Health Level 7 Reference Information Model (RIM) – XML/PRA Section, avail. at http://www.mcis.duke.edu/standards/HL7/hl7.htm
2. Boehm, B, Software Engineering Economics, McGraw Hill, 1987.
3. Campbell, K., Cohn,S, Chute,C., et al. "Galapogos: Computer Based Support for Evolution of a Convergent Medical Terminology, JAMIA Suppl., Oct. 1996.
4. Bylander, T. et al., "Generic Tasks for Knowledge Based Reasoning: The Right Level of Abstraction for Knowledge Aquisition," IJMMS, v26, 1987, pp 3-28.
5. Field, MJ, Lohr, KN (eds), Guidelines for Clinical Practice, Washington DC: Nat'l Acad. Press, 1992.
6. Hogan, M., "Executable Guideline Objects," U.of Penn Tech Report, Fall 1998.
7. Morales, A., Silverman BG., "HOLON/R2Do2: A Clinician Centered Approach for the Representation and Automation of Clinical Practice Guidelines", Submitted to AMIA'99, 1999.
8. S. W. Tu & M. A. Musen. "Episodic Refinement of Episodic Skeletal Plan Refinement." International J. of Human-Computer Studies 48:475-497, 1998.
9. Ohno-Machado, L., et al., "The Guideline Inter-change Format," JAMIA,
10. Silverman, B.G., et al., Jones, P., Safran, C. et al. "HOLON: Extending the Web Document Model with Objects," JAMIA Suppl,, Nov. 1998.
11. Silverman, BG, Morales, A, Andonyadis, C. "Web based Agents as a Middleware Framework: The Case of R2Do2," J. of AI in Medicine, Dec.98.
12. Silverman, B.G., "Criticism Based Knowledge Acquisition," Innovative Appl. Of AI, AAAI/MIT PRESS, 1992, pp. 291-319.
13. P. Bremond-Gregoire, J-Y. Choi, I. Lee, "A Complete Axiomatization of Finite-state (ACSR) Processes," Information and Computation, Nov 1997.
14. Davidson, S., Overton, C., Tannen, V., et al., "BioKleisli: A Digitial Library for Biomedical Researchers," in S. Letovsky (ed), Bioinformatics, Kluwer Academic Publishers, 1998.