

Sharable Representation of Clinical Guidelines in GLIF: Relationship to the Arden Syntax

**Mor Peleg, Ph.D.¹, Aziz A. Boxwala, M.B.B.S., Ph.D.², Elmer Bernstam, M.D., M.S.E.¹,
Samson Tu, M.S.¹, Robert A. Greenes, M.D., Ph.D.², and Edward H. Shortliffe, M.D.,
Ph.D.³**

¹Stanford Medical Informatics, Stanford University School of Medicine, Stanford, CA

²Decision Systems Group, Harvard Medical School, Brigham & Women's Hospital, Boston,
MA

³Department of Medical Informatics, Columbia University, New York, NY

The relationship between GLIF and Arden

Mor Peleg, Stanford Medical Informatics, MSOB x-208, 251 Campus Drive, Stanford, CA 94305-5479

Abstract

Clinical guidelines are intended to improve the quality and cost effectiveness of patient care. Integration of guidelines into electronic medical records and order-entry systems, in a way that enables delivery of patient-specific advice at the point of care, is likely to encourage guideline acceptance and effectiveness. Among the methodologies for modeling guidelines and medical decision rules, the Arden Syntax for Medical Logic Modules and the GuideLine Interchange Format version 3 (GLIF3) emphasize the importance of sharing encoded logic across different medical institutions and implementation platforms. These two methodologies have similarities and differences; in this paper we clarify their roles. Both methods can be used to support sharing of medical knowledge, but they do so in complementary situations. The Arden Syntax is suitable for representing individual decision rules in self-contained units called Medical Logic Modules (MLMs), which are usually implemented as event-driven alerts or reminders. In contrast, GLIF3 is designed for encoding complex multi-step guidelines that unfold over time. As a consequence, GLIF3 has several mechanisms for complexity management and additional constructs that may require overhead unnecessary for expressing simple alerts and reminders. Unlike the Arden Syntax, GLIF3 encourages a top-down process of guideline modeling consisting of three levels that are created in order: Level 1 comprises a human-readable flowchart of clinical decisions and actions. Level 2 comprises a computable specification that can be verified for logical consistency and completeness, and level 3 comprises an implementable specification that includes information required for local adaptation of guideline logic as well as for mapping guideline variables onto institutional medical records. A major emphasis of the current GLIF3 development process has been to create the computable specification that formally represents medical decision and eligibility criteria. We based GLIF3's formal expression language on the

Arden Syntax's logic grammar, making the necessary extensions to the Arden Syntax's data structures and operators to support GLIF3's object-oriented data model. We discuss why the process of generating a set of MLMs from a GLIF-encoded guideline cannot be automated, why it can result in information loss, and why simple medical rules are best represented as individual MLMs. We thus show that the Arden Syntax and GLIF3 play complementary roles in representing medical knowledge for clinical decision support.

Keywords: GLIF, Arden Syntax, Clinical Guidelines, computer-interpretable guidelines, knowledge modeling

1. Methodologies for modeling shared computer-interpretable guidelines and medical decision rules

Implementation of computer-interpretable guidelines in decision-support systems that are used at the point of care has been proposed as a way to improve health care safety, quality, and efficiency. Researchers have devised many approaches to this task. Many of the developments have occurred in the context of specific host information-systems platforms. Other workers, who wish to foster sharing and reuse of medical knowledge in guidelines and other decision-support systems, have concentrated on specifying formal methodologies for representing the knowledge.

Hripcsak and colleagues began developing the Arden Syntax for Medical Logic Modules [1, 2] in 1989 to facilitate sharing of single-step alerts (e.g., alert a physician that his patient has hypokalemia) and reminders (e.g., remind a physician to order a test), termed *medical logic modules* or MLMS. The Arden Syntax is now a standard of the American Society for Testing and Materials (ASTM), originally published as such in April 1992 as ASTM E1460-92. Jenders and colleagues developed and published a second version of the Arden Syntax, Arden 2.0

(<http://www.cpmc.columbia.edu/resources/arden/>), under **Health Level Seven, Inc. (HL-7)**, an ANSI-accredited Standards Developing Organization that operates in the health-care arena.

Researchers have built on a variety of approaches to handle representation of clinical guidelines for computer-based decision support. For example, Lobach and colleagues describe a relational database schema for automating the delivery of multi-step guidelines [3]. Another model, *PROforma*, is a logic language with an object-oriented model. In *PROforma*, guidelines are modeled as constraint-satisfaction graphs, where nodes represent *tasks* (i.e., clinical actions, decisions, enquiries, or complex *plans* that are tasks composed of other tasks). Arcs connect tasks within *plans*. The ordering of tasks reflects logical, temporal, or other constraints [4]. Another guideline-modeling methodology, Asbru [5], emphasizes guideline intentions (e.g., maintain normal blood glucose level), rather than only action prescriptions (e.g., give insulin). Asbru is an expressive language for representing time-oriented actions, conditions, and intentions in a uniform fashion. Asbru models guidelines as plans that can be hierarchically decomposed into (sub)plans or actions. Yet another guideline model, EON, enables the specification of a guideline through a combination of modeling primitives, such as different types of decision-making mechanisms, control-flow constructs, actions, activities, and abstractions [6].

Most of the guideline-modeling methodologies cited have been developed independently from one another. As developers of the GuideLine Interchange Format (GLIF) since the mid-1990s, we have pursued a different goal: to create a shared representation for computer-interpretable guidelines that incorporates features important for a wide variety of applications and can be used as a basis for implementation of those applications in diverse clinical-systems environments. The GLIF guideline-modeling methodology [7], therefore, was intentionally based on a desire to

leverage the work invested in other approaches. Like the Arden Syntax, GLIF is being further examined as a basis for a guideline-representation standard; both the Arden Syntax and GLIF are foci of the Special Interest Groups of the Clinical Decision Support Technical Committee of HL7.

The Arden Syntax has achieved a substantial level of acceptance through standardization. Four vendors support the Arden Syntax and two other vendors are currently developing applications for it. At least six health care organizations have implemented MLMs. In the MLM library of the Columbia-Presbyterian Medical Center alone, there were over 150 MLMs, as of March 1997. However, the vast majority of them were not used to implement guidelines, but only unconnected MLMs [see <http://www.cpmc.columbia.edu/resources/arden/>]. Because Arden Syntax is substantially used, we decided to explore using it as a basis for the expression language for decision logic in GLIF. In doing so, we identified many required changes and extensions, and identified directions for future evolution of an expression language. This paper describes in detail the relationship between the Arden Syntax and the current version (3.0) of GLIF (GLIF3), currently still in development and not yet in operational use.

2. Background

We developed GLIF3 by augmenting version 2.0 of GLIF (GLIF2) with components that enable computer-interpretation of encoded guidelines. A major component of GLIF3 is a formal expression language that enables guideline authors to specify decision criteria formally. This expression language is based on the Arden Syntax's logic grammar. In this section, we provide background material on the Arden Syntax, GLIF2, and GLIF3.

The Arden Syntax

The initial version of the Arden Syntax was based largely on the encoding scheme used for generalized medical decision support in the HELP system [8]. The shared medical knowledge is encoded in the form of individual MLMs, each of which represents a single medical decision. Most MLMs are triggered by clinical events (e.g., admission of a patient, or storage of certain medical data in the electronic medical record (EMR)). Once triggered, MLMs evaluate logical decision criteria (e.g., *potassium* < 3.5), and, if the criteria hold, they perform an action, such as sending an alert to a health-care provider. An MLM contains slots, grouped into three categories: *maintenance*, *library*, and *knowledge*. The **maintenance category** is used for knowledge-base maintenance and revision control. The **library category** provides predefined explanatory information and links to the health literature. The **knowledge category** contains the functional components of the MLM. The knowledge category has **evoke**, **logic**, and **action** slots that specify the events that trigger the MLM, the logical criterion that is evaluated, and the action that is performed if the logical criterion holds, respectively. These knowledge-category components define the logical rule that the MLM specifies.

MLMs separate institution-specific entities, such as the mapping of patient data references to fields in an EMR, from the MLM's decision logic. The mappings between the institution-specific terms and the MLM's variables are specified within another component of the *knowledge category*, called the **data slot**. However, only part of this specification is defined by the syntax. The mapping of the MLM variables to the institutional EMR is defined in a section demarcated by curly braces. Because there is as yet no standard terminology or data model for electronic medical records, there can be no standardized syntax for the content within this section, a difficulty that is called the *curly-braces problem* by the Arden community.

Various authoring tools have been developed for writing MLMs. Jenders and Dasgupta created an MLM authoring tool [9] that guides the author in a stepwise manner. MÉDAILLE [10] is an application generated by the PROTÉGÉ-II [11] knowledge-acquisition tool that provides support for entering and editing MLMs. MEDAILLE has a syntax checker and an integrated terminology. Translation of MLMs into an executable form can be done by a number of compilers [12, 13].

GLIF2

We have carried out the development of GLIF [14] through the formation of the InterMed Collaboratory, a consortium of medical-informatics groups at Harvard, Stanford, and Columbia universities. GLIF is designed to allow exchange of computer-interpretable guidelines among institutions and computer-based applications. Unlike the Arden Syntax, it is targeted toward complex guidelines that unfold over time, rather than toward individual medical alerts and reminders. Like the Arden Syntax, GLIF-encoded guidelines are stored as text files for sharing. GLIF has an object-oriented model that consists of a set of classes for guideline entities, attributes for those classes, and data types for the attribute values. GLIF2, published in 1998 [14], represented guidelines as flowcharts of guideline steps such as clinical actions and decisions. However, the attributes of structured constructs were defined only as free-text strings, and such guidelines could not be used for computer-based execution that requires automatic inference.

GLIF3

Recent work by the InterMed Collaboratory has concentrated on the development of the subsequent version of GLIF, called GLIF3, hereafter referred to simply as GLIF [7]

(<http://www.smi.stanford.edu/projects/intermed-web/guidelines/GLIF1.htm>). This version of GLIF

1. Supports three different levels of abstraction, to enable specification at conceptual, computable, and implementable levels
2. Incorporates an expression language based on the Arden Syntax logic grammar for representing logical and temporal decision criteria
3. Defines an ontology for medical concepts and patient data
4. Expands GLIF classes to support representation of several new concepts, such as iteration specification, and grade of evidence
5. Further structures GLIF2 classes by creating hierarchies of action specifications and decision steps to represent more concisely the different forms of clinical actions and decisions found in medical guidelines
6. Introduces a new guideline step called a patient state step, which can be used as a label or entry point into a guideline

We have used GLIF to specify guidelines that differ in their clinical domain, stage of the medical problem and its management (e.g., screening, diagnosis, disease management), multiplicity of encounters, setting (e.g., inpatient or outpatient clinic), time frame (emergency, acute, or chronic), and guideline computability (i.e., algorithmic, guiding, or intermediate) [15, 16]. We specified these guidelines at the conceptual and computable levels.

Currently, there are two authoring tools for GLIF: Protégé-2000 [17], which is a general tool for working with knowledge bases, and a GLIF-specific authoring tool developed by the Decision Systems Group at Brigham and Women's Hospital [18]. Both tools help guideline authors to

visualize the entire guideline as a flow chart, as well as to manipulate the formal specification of the computable representation level. The execution engine, described by Boxwala and colleagues [19], can execute guidelines that are encoded in an enhanced version of GLIF2, and is used in clinical decision-support applications. The enhanced version of GLIF2 included

1. A richer patient data model that supports a number of data types, and permits specification of cardinality and temporal and logical constraints on the values of the data
2. An enhanced action model that allows specifying action-specific parameters for execution of the action
3. A syntax for logical constraints, influenced by Arden Syntax, enhanced in order to support patient data and action model. However, unlike Arden Syntax it did not support temporal and list operators and data types. We are now designing a GLIF3 execution engine.

3. The GLIF process of guideline modeling

GLIF enables a top-down creation of a guideline specification. We envision a multi-step process for authoring guidelines. First, a medical expert starts with an informal textual description of the problem, and a set of assembled evidence. She then creates an imprecise, potentially incomplete **conceptual** view of the guideline, in the form of a flowchart that includes guideline steps linked to each other in a temporal order. The guideline steps represent clinical actions (e.g., prescribe aspirin), decisions (should the patient should be placed on a diet?), and patient states (e.g., the patient is receiving a single anti-hypertensive drug). In addition, branch and synchronization steps can be used to express parallel execution; a branch step can lead to multiple subsequent steps that converge in a synchronization step, thus supporting parallel execution of the steps contained in the different branches (e.g., ordering pre-operative laboratory-tests and scheduling operating-room time). The information specified in the guideline steps is largely in the form of

unstructured text strings. In a second encoding stage, a team consisting of an informatician, aided by a medical expert, refines the flowchart representation to produce a fully detailed, precise, and **computable** specification, in which decision and eligibility criteria, patient data, action specifications and control flow are formally defined. In the third and final stage, an encoding team creates an **implementable** specification, in which data and action specifications are mapped onto specific data and procedures used by the implementing institution. The encoding team performs contextual adaptation of encoded guidelines for local health-care setting, and maps encoded guideline variables, concepts, and action specifications to the local clinical information systems. This mapping is not yet supported by GLIF3.

GLIF3 is related to the Arden Syntax in three ways. First, GLIF3 logical and temporal decision criteria are specified using an expression language called Guideline Expression Language (**GEL**) that is derived from and enhances the Arden Syntax logic grammar [20]. Second, GLIF's Message_Action action specification is based on the Arden's *write statement*, which enables specification of a text string and its email destination. Third, GLIF has a special construct, called an MLM-macro, that guideline authors can use to map guideline declarations onto a procedural specification that defines evoke, logic and action slots, which correspond to the respective Arden MLM slots.

We use the American College of Cardiology/American Hospital Association/American College of Physicians-American Society of Internal Medicine (ACC/AHA/ACP-AISM) guidelines for the management of patients with chronic stable angina [21] to demonstrate how we can use GLIF to model a complex guideline, and how GLIF uses constructs derived from the Arden Syntax. Figure 1 shows the top-level flow chart view of the guideline that we created (in this case using Protégé-2000) from the textual description, and the accompanying flowchart supplied by the

guideline authors. When a guideline specification is created in GLIF, the flowchart is usually a first approximation of the guideline specification. At this point, the specification is still potentially imprecise and incomplete, as the information contained in the guideline steps is only free-text.

The top-level flow chart view of the guideline includes steps that represent high-level actions and decisions. GLIF's **nesting** mechanism enables expansion of such steps through the use of subguidelines. For example, the "Empiric Therapy" action step in Figure 1 can be nested into a subguideline that contains the steps of the treatment algorithm. The *nesting* mechanism is useful in managing the complexity of large, multistep guidelines that contain many pathways.

We refine the conceptual flowchart representation into a *computable* representation by adding a structured specification that corresponds to the text within the guideline steps. Action steps that were described in free text are refined to include well-defined tasks, such as scheduling a procedure, referring a patient, or ordering a test. GLIF has a hierarchy of action specifications that guideline authors can use to model both programming-oriented actions (e.g., sending a message to a user) and medically-oriented actions (i.e., actions that rely on medical concepts such as ordering a clinical test). Medical concepts and patient-data elements that need to be referenced by action and decision steps are specified in GLIF's **domain ontology**, which contains the clinical meaning of the concepts and patient data (i.e., by referencing controlled medical vocabularies, such as UMLS) as well as their structure, or data model. Defining medical concepts in relation to standard medical vocabularies allows the guideline encoding to contain concepts that are institution-independent. Mapping to institution-dependent Electronic Medical Record codes and procedures can therefore be specified in another level, which is not yet supported.

The default medical data model, used by GLIF's domain ontology, is HL-7's Reference Information Model (RIM) [22]. The clinical part of HL7's RIM version 1.0 (<http://www.hl7.org/library/data-model/RIM/C30100/rim0100h.htm#EL10003>) can model medical knowledge and patient data uniformly. All clinical data are specified as *Act* objects, or *Acts*. Acts have attributes that provide information about clinical concepts they represent and their timing. Relationships are used to model the Act circumstances and allow grouping the Acts and reasoning about them. Acts are specialized into procedures performed on a patient, observations about the patient, medications given to the patient, and other kinds of services. Different sub-classes contain additional attributes that help characterize an Act. For example, Observation has a *value* attribute, whereas Medication has attributes about dosage and route of administration. Act objects have a *mood* that distinguishes the ways in which they can be conceived: as an event that occurred, a definition, intent, order, etc.

When decision or eligibility criteria are used, they are specified in GEL, and can reference medical concepts and data that are defined in the domain ontology. These refinements turn the *conceptual* specification into a *computable* specification. An example of a refinement is the following GEL *criterion* that refines the abstract decision "Conditions present that can cause angina? (severe anemia, hyperthyroidism, ...)" that was stated in text at the top-level view of the guideline specification (Figure 1).

(SevereAnemia *is in* ProblemList) OR (Hyperthyroidism *is in* ProblemList) ...

ProblemList is a list of *concepts* that represent all of the known current problems, retrieved from the EMR. The GEL operator "*is in*" checks for membership of the left argument in the right argument, which is a list. Unlike the Arden Syntax's "*is in*" operator, the GEL operator can work with complex objects, such as Concept, which has three attributes: the concept name; the

concept's unique identifier, taken from a controlled vocabulary; and the unique identifier that represents the controlled vocabulary itself. In this example, we check whether the concepts *SevereAnemia* or *Hyperthyroidism* occur within the list of current problems that are associated with this patient.

At the *implementable* (Third) level, the guideline is mapped onto procedures and data that are used by the implementing institution and local interpretation of high-level guideline recommendations are performed. For example, the stable-angina guideline addresses the evaluation of chest pain, but does not specify which tests should be ordered for a patient judged to be low-risk based on history, physical examination, and initial diagnostic testing. One institution may decide that the relevant test should be serial-troponin levels, whereas another institution may prefer that physicians order either creatine phosphokinase (CPK) or immediate stress testing. Other local implementation considerations may include how guideline advice should be delivered to the user (e.g., alert versus interactive session).

GLIF includes constructs for modeling events, logical criteria, and actions. Therefore, GLIF can represent alerts and reminders. For ease of authoring, modelers who are familiar with the Arden MLM can model alerts and reminders using a special GLIF construct, *MLM macro step*. A **macro step** in GLIF3 is a special class that has attributes that define the information required to instantiate a set of underlying GLIF steps. Those underlying steps represent a pattern that appears in clinical guidelines [23]. In this way, macro steps provide a means to specify declaratively a procedural pattern in a single construct that is realized by a set of GLIF steps. An *MLM macro step* represents a particular pattern of GLIF components: a GLIF *event* (MLM evoke slot), followed by a GLIF *decision criterion* (MLM logic slot), followed by a GLIF *action specification* that can be a *Message_Action*, *Assignment_Action*, or *Event_Action* (MLM action

slot). The guideline author fills in evoke, logic, and action slots of the MLM-macro. As shown in Figure 2, the MLM macro step can be replaced by a sequence of two GLIF guideline steps that do not include macros. The first of these two steps is a decision step that references instances of two GLIF classes: *event* class and *criterion*, which correspond to the evoke and logic slots of the MLM-macro. The decision step is followed by an action step that references instances of the GLIF Action_Specification class that correspond to the action slot of the MLM-macro.

Figure 1 shows that, after a history and physical examination have been obtained, the action step “Diagnostic tests” should be executed. A possible implementation of this action is first ordering tests and then notifying the physician when the test results are back. The notification can be modeled as an MLM macro step, as shown in Figure 3. The action slot of the MLM uses a GLIF Message_Action to specify the message to be sent and the destination. The Message_Action is adapted from the Arden Syntax’s write statement (i.e., WRITE <message> AT <destination>:).

Note that the guideline does not specify which diagnostic tests should be performed to assess the risk of coronary artery disease (CAD). Therefore, we would like to define the evoking event of the MLM-macro as “storage in the medical record of diagnostic results of tests for assessing the risk of CAD.” By this event definition we mean that, regardless of which tests are chosen, when the result of one of them is stored in the EMR, the MLM macro step is evoked. Mapping this kind of event in a standard and portable manner to an institutional EMR is difficult, and has not yet been defined by either GLIF or the Arden Syntax. Therefore, the example in Figure 3 assumes that performing the diagnostic test consists of obtaining serum hemoglobin, fasting glucose, and fasting lipid panel (total cholesterol, HDL cholesterol, triglycerides, and calculated LDL cholesterol).

The alert, shown in Figure 3, is evoked when the results of any of these laboratory tests are recorded in the EMR. Then, a logic criterion is evaluated to determine that all the tests were performed within the past 12 hours, and that the results are of numeric type. These requirements are imposed to ensure that the measurements are current and have legal values (nonempty numeric results). For simplification, (1) no checks were written to verify that the results are within an acceptable range (e.g., $6 < \text{Hemoglobin} < 20$), and (2) the units of measurement were not considered. Note that although all of these tests are needed for the alert to be sent, the evoking of the MLM is triggered when any of these tests are recorded in the EMR. This is because the MLM evoke slot can contain a disjunction of events but not a conjunction.

The *logic criterion* of the MLM macro step is expressed in GEL. The Arden Syntax operators “is number,” “latest,” “where,” “time of,” and “it” are used by GEL. The value TRUE is returned by “is number” if the latter’s argument is a number. The operator “latest” acts on a **query-result** – a list of elements that have data values and primary times (i.e., the time of occurrence) – and returns the query result element that has the latest primary_time. The operator “where” allows expression of a criterion that is applied to every element in the query result. In this example, only elements for which the primary time occurred within the past 12 hours are selected. The “where” clause uses the “time of” and “it” operators. The operator “it” iterates over the elements of the query result. The “time of” operator returns the primary time of a query result element. The *logic criterion* of the MLM macro step uses the GEL function “selectAttribute” to access attributes of a complex object (a legal value type in GEL). In this example, selectAttribute selects the “value” attribute of a query-result element.

The criterion shown in Figure 3 refers to variables that represent query results extracted from records of the laboratory-test results (*Hemoglobin*, *FastingGlucose*, *FastingTotalCholesterol*,

FastingHDLCholesterol, *FastingTriglycerides*, and *CalculatedFastingLDLCholesterol*). The *CalculatedFastingLDLCholesterol* query result is shown in Figure 4(a). Each element of the query result holds a value, which is a number that represents the LDL cholesterol level, and the primary time at which each measurement was made. The values and primary times are taken from *data items*. **Data items**, defined by GLIF's *medical ontology*, refer to codes from controlled medical vocabularies, and to data structures defined by medical data models. As was noted, the default medical data model is HL-7's Reference Information Model (RIM). As shown in Figure 4(b), the data item *CalculatedFastingLDLCholesterolDataItem* refers to a vocabulary code, taken from UMLS, and to an HL-7 RIM observation.

Because GLIF builds on other standards for the medical data model (HL-7 RIM) and expression language (GEL, based on the Arden Syntax logic grammar), we had to reconcile incompatibilities among these standards. These incompatibilities include the Arden Syntax's lack of support for complex data types and time intervals that are part of the HL-7 RIM. In addition, there is a mismatch between Arden's single primary time and the multiple time attributes of the HL-7 RIM [20]. The solution was to allow GEL to support Arden's data types as well as complex data values. Thus, GEL's *query results* and *lists* are like Arden's corresponding types, but their values can be complex types. To map from GLIF's default data model, the HL-7 RIM, to GEL's data model, we defined the *Get_Data_Action* construct. *Get_Data_Action* retrieves patient data from the EMR as lists of HL-7 RIM objects of the patient that correspond to a single medical concept (e.g., LDL cholesterol values) and transforms them to results of type *query result*, which is supported by GEL [20]. A guideline author can use *Get_Data_Action* to specify that an attribute of a complex HL-7 RIM class is the source of data values for the *query result*, and that values of another attribute serve as the primary time in the *query result*. The

Get_Data_Action that is used to retrieve the *CalculatedFastingLDLCholesterol* query result is shown in Figure 4(c). In this case, the values of the query-result elements are numbers that represent the LDL cholesterol level in mg/dL.

4. Discussion

When the paper describing GLIF2 was published (1998), some observers expressed concern as to whether GLIF was a competitor to the Arden Syntax. As we have shown, GLIF and the Arden Syntax are not competing methodologies, but rather are related, complementary methodologies for the expression and sharing of medical knowledge. Arden Syntax is useful for sharing medical knowledge that can be expressed as individual MLMs that can be used, for example, for generating warnings about drug interactions or alerts regarding abnormal laboratory-test results. GLIF, in contrast, is designed to specify multistep guidelines that unfold over time. Although GLIF can also encode simple decisions, the overhead (i.e., the machinery necessary to use GLIF-encoded knowledge) is much greater than that required for the Arden Syntax. Therefore, if the task can be modeled as a single medical decision, Arden MLMs, rather than GLIF, should be used. Even in the context of a single institution, it seems possible, that both Arden MLMs and GLIF-encoded guidelines could be implemented at the same time. GLIF would be used to execute multistep guidelines, while MLMs could serve as high-level “guards”. The high-level guards could ensure that the interactions among several GLIF-encoded guidelines that are executed for a single patient are desirable. For example, MLMs can alert a physician if a GLIF guideline-recommendation issued an order for a test or medication whereas another guideline issued a similar or conflicting order for the same patient. This way, these safety checks could be reused for any guideline interaction, and would not need to be embedded within the logic of each guideline.

Its developers did not originally intend the Arden Syntax to be used for encoding complex guidelines that involve multiple decisions. However, encoding of multi-step guidelines using the Arden Syntax is possible albeit clumsy, as demonstrated by the implementation of a care plan for the post-operative management of patients following coronary-artery bypass graft surgery [24]. Nevertheless, there is no support in the Arden Syntax specification to aid human understanding of the way MLMs interact with one other. Therefore, the authoring and implementation of multistep guidelines, although possible, is difficult and awkward. Another aspect of guidelines that is not well supported by MLMs is recommendations that concern non-deterministic decisions that cannot be represented by if...then...else rules (e.g., a decision among different options for treatment of CAD). GLIF can model such decisions by listing rules for and against each of the decision's options. For example, a decision on the appropriate treatment of CAD can be modeled as a user choice that has several alternatives (decision options). One such decision option is pharmacological treatment. A rule in favor is "if the patient is at high risk for surgical complications, consider pharmacological treatment"; a rule against it is "If the patient has severe three-vessel disease, do not consider pharmacological treatment". Guideline developers, who are using GLIF, can represent decisions without committing to the way in which the recommendations should be delivered to a user (e.g., ranking the options by the number of favorable rules that they satisfy, or simply presenting to the user the rules for each option). Unlike GLIF, the Arden Syntax has no special constructs for representing non if...then...else decisions in a way that is independent from implementation choices. An MLM can present, to a user, recommendations that are based on non-deterministic choices by specifying a message listing the rule in and out arguments in a certain order, thus committing to some delivery choice.

Because the Arden Syntax has been accepted as a mature standard for delivering alerts and reminders and has been implemented and tested by multiple academic groups and commercial vendors, developers of GLIF wanted to leverage the years of work that have gone into the development of the Arden Syntax. GLIF3 draws on the modeling approaches taken both by GLIF2 and by the Arden Syntax for MLMs. GLIF3 extends GLIF2 models, which represented guidelines in a way that is not computable (i.e., did not support computer-based interpretation involving automatic inference), by defining a formal expression language, GEL, based on the Arden Syntax. GEL, together with formal definitions of medical concepts and data items, action specifications, and control flow, enables GLIF3 specifications to be *computable*. The computable specification reconciles differences between the Arden Syntax's data model and GLIF's object-oriented medical data model. Although GEL supports complex data types, it is deficient in specifying expressions that relate to more than one of an object's attributes (e.g., accessing the primary time, value, value_unit, and method by which an LDL cholesterol value was obtained, see Figure 4(b)) [20]. We are exploring the possibility of designing an object-oriented expression language to overcome this problem.

Guideline authors can use the three layers of representation in GLIF to develop computer-interpretable guidelines in a top-down manner, specifying the abstract flowchart of the clinical decision and actions first, then defining formally the guideline's logic, and, finally, concentrating on implementation details. MLM developers can also carry out a top-down development process. A medical domain expert can create a documentation-level specification by entering the *maintenance* and *library* categories of an MLM. Based on the narrative description of the MLM's function, found in the *purpose* and *explanation* slots of the library category, an informatician can then define the MLM's logic by formally specifying the *evoke*, *logic* and

action slots of the knowledge category in the Arden Syntax's grammar. Then, implementers in local institutions can define the data slot, thus mapping data items and events used by the MLM's knowledge category into institutional EMR codes and procedures. Note that this mapping has not yet been standardized. Even though the top-down modeling process is similar, GLIF3 and MLMs differ in the granularity of the entity they try to model. When a guideline is more complex than a single medical decision, the Arden Syntax does not provide support for maintaining a conceptual view of a guideline that is implemented as multiple interacting MLMs.

Both the Arden Syntax and GLIF try to solve the problem of creating a computer-interpretable specification of medical knowledge that could be adapted to local health care settings and integrated into local clinical systems. The Arden Syntax defines specific ways in which MLMs are triggered and recommendations are delivered to users. These implementation-related choices are specified as part of the MLM's *knowledge slot*. When an MLM that is developed by one institution is shared by another institution, two modes of sharing are possible. In one mode, the second institution commits to the knowledge-delivery choices that were made by the developing institution (e.g., issuing a reminder to measure the blood pressure value every 2 hours or issuing an alert when the patient's blood pressure was not measured within the past 2 1/2 hours). In this case, only the mapping to EMR-specific data and events should be done. In the second way, the knowledge that is shared is not committed to any delivery choice, but then, only skeletal code is shared that includes textual descriptions of the actions that need to be done, with no formal encoding. The second institution can then formally specify the procedures that should be done locally. In GLIF, on the other hand, guideline authors can model a guideline on the textual *conceptual-level* and on the formal *computable-level* without making any commitments on the way the recommendations will be delivered to the user. As in Arden MLMs, guidelines encoded

at the conceptual or computable levels can be shared among different institutions, each of which has the freedom to adapt the guideline to their needs at the implementation level. GLIF can use the macro-step mechanism to delineate specific procedures that should be done locally by the implementing institution. Then each local institution must map these procedures in its own clinical information system's functions. We are still developing these mapping mechanisms.

To adapt a guideline to a particular health-care setting, the implementing institution must modify the guideline in accordance with availability of resources and expertise, local workflow, practice preferences, and differences in patient population. Adaptation may simply involve a refinement of generic instructions into concrete procedures, but may also require changes in the guideline's logic [25]. For example, certain decision options may be eliminated because they involve procedures that cannot be performed at the institution; such deletions change the guideline's logic. A generic instruction, such as "perform diagnostic tests for assessing the risk of CAD", can be refined into procedures such as performing specific tests and notifying the physician when the test results are back. As we explained in Section 3, the notification can be done through GLIF's *macro step* mechanism.

Another mechanism that GLIF offers for refining actions is *nesting*. Nesting enables expansion of action and decision steps in a high-level flowchart through the use of subguidelines. Thus, the nesting mechanism, like the macro step, is useful in managing the complexity of large and multibranching guidelines. Nesting also facilitates model extensibility and reuse of part of a guideline, which is encapsulated as a subguideline. Nesting may also be useful in showing the relationship of a guideline to other guidelines by creating a top-level view that makes such relationships explicit.

Achieving an *implementable* specification entails mapping of guideline variables onto institutional data elements and methods. We designed GLIF's *medical ontology* to facilitate mapping of concepts, data items, and medically oriented action specifications to institutional data items and procedures in order-entry systems. The Arden Syntax, on the other hand, does not provide a standard way to map MLM variables to institutional data elements. GLIF's medical ontology structures data items by referring to (1) codes from controlled medical vocabularies, and (2) data structures defined by medical data models. GLIF's default medical data model is HL-7's RIM. In the future, we will structure the medical ontology further to represent mapping between its terms and institutional data elements (i.e., institutional term-dictionaries or procedures used to obtain the corresponding data elements) and methods (e.g., order-entry methods). Currently, GLIF's medical data model abstracts medical concepts and data based on their medical characteristics. We are thinking of facilitating the mapping to institutional data elements by structuring the medical data model to fit more closely with the different data types by which medical data are stored in EMRS (e.g., textual note entries, numerical results).

GLIF-encoded guidelines might be implemented in a variety of ways, among which could be compiling or interpreting the guidelines into executable actions, or translating them into sequences of MLMs, thus taking advantage of the software tools that already exist for implementing MLMs. For the latter, we stress that the process of generating a set of MLMs from a GLIF-encoded guideline depends on many decisions that have to be made along the way, not all of which are technical in nature. For example, we might not want to convert GLIF decision steps intended to be made by users into MLMs because the Arden Syntax does not have special constructs for representing them, as was explained earlier in this section. Also, translating parallel execution in GLIF into a set of interacting MLMs may be difficult. Even when the

guideline developer decides that certain guideline steps should be mapped into MLMs, more decisions are required. For example, a guideline may specify that “the patient’s body temperature must be measured every 2 hours”, but will not tell you how the information (that the collection is to take place) should be given the user. Should care providers be reminded *before* this action needs to take place? Alternatively, should care providers be reminded only if they fail to perform the action on time? The local institutions must make these decisions to implement the guideline. Then, they can add MLM-macro steps to the GLIF-encoded guideline.

Because MLM macro steps contain slots that are analogous to the MLM evoke, logic, and action slots, automatically translating MLM macro steps into Arden MLMs should be easier than translating sequences of GLIF action and decision steps that contain information that is not easily mapped into MLM slots. For example, it is not easy to translate GLIF’s iteration specifications into part of an MLM’s logic slot, or to translate the variety of GLIF action specifications (e.g., `Get_Data_Action`) into MLM actions. Although, by converting MLM macro steps into Arden MLMs, we have the advantage of using existing Arden Syntax tools for MLM implementation, we lose information during the translation process. The main loss is GLIF’s structured definition of complex data items, which can aid in mapping the data items onto the institutional patient data elements, and also assists in encapsulating related data items, such as a medication dose, route, and time of administration. In the translation process, we would have to convert GEL expressions into expressions in the Arden Syntax’s logic grammar that lack complex data items and GEL functions. We would also need to break each complex data item into items that each have a single value and a single (primary) time stamp.

Although GLIF and the Arden Syntax are not code compatible (i.e., we cannot execute Arden MLMs using GLIF machinery), GLIF constructs such as the MLM-macro facilitate the transition

from the Arden Syntax to GLIF should such transition be necessary. However, converting simple MLMs into a GLIF specification may not be worthwhile because GLIF has a much more complex model than is necessary for representing simple rules.

5. Conclusion

The Arden Syntax represents efficiently individual decision rules, but makes difficult the representation of multi-step algorithms that unfold over time. We specifically designed GLIF to enable representation of large, complex guidelines. GLIF may be cumbersome, however, for the expression of individual medical decisions leading to an action, usually an alert or reminder. Recognizing that the Arden Syntax is an established standard that can model medical decisions, GLIF3 builds on Arden's development by basing its expression language and several other constructs on the Arden Syntax, while creating constructs that are suitable for representing multi-step guidelines. GLIF's expression language diverges from the Arden Syntax's logic grammar because of the need to support complex medical data structures, such as those of the HL7 RIM. Developers of both GLIF and the Arden Syntax are involved in the Clinical Decision Support Technical Committee of HL7. Together, we are exploring the possibility of developing an expression language that is consistent with the HL7 RIM.

References

1. Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden Syntax. *Comput Biomed Res* 1994;27(4):291-324.
2. Clinical Decision Support & Arden Syntax Technical Committee of HL7, inventor Arden Syntax for Medical Logic Systems, version 2.0. Draft revision. USA. 1999 July 7, 1999.
3. Lobach DF, Gadd CS, Hales JW. Structuring clinical practice guidelines in a relational database model for decision support on the Internet. In: Proc AMIA Annu Fall Symp; 1997; 1997. p. 158-62.
4. Fox J, Rahmanzadeh A. Disseminating medical knowledge: the PROforma approach. *Artificial Intelligence in Medicine* 1998;14:157-181.

5. Shahar Y, Miksch S, Johnson P. The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines. *Artificial Intelligence in Medicine* 1998;14:29-51.
6. Tu SW, Musen MA. A Flexible Approach to Guideline Modeling. In: AMIA Symp; 1999; 1999. p. 420-424.
7. Peleg M, Boxwala A, Ogunyemi O, Zeng Q, Tu S, Lacson R, et al. GLIF3: The Evolution of a Guideline Representation Format. In: Proc. AMIA Annual Symposium; 2000; 2000. p. 645-649.
8. Kuperman GJ, Gardner RM, Pryor TA. HELP: A Dynamic Hospital Information System. NY: Springer-Verlag Inc.; 1991.
9. Jenders RA, Dasgupta B. Assessment of a knowledge-acquisition tool for writing Medical Logic Modules in the Arden Syntax. In: Proc AMIA Annu Fall Symp 1996; 1996; 1996. p. 567-71.
10. Bang M, Eriksson H. Generation of development environments for the Arden Syntax. In: Proc AMIA Annu Fall Symp; 1997; 1997. p. 313-317.
11. Musen MA, Tu SW, Eriksson H, Gennari JH, Puerta AR. PROTEGE-II: An Environment for Reusable Problem-Solving Methods and Domain Ontologies. In: International Joint Conference on Artificial Intelligence; 1993; Chambéry, Savoie, France; 1993.
12. Hripcsak G, Cimino JJ, Johnson SB, Clayton PD. The Columbia-Presbyterian Medical Center decision-support system as a model for implementing the Arden Syntax. In: Proc Annu Symp Comput Appl Med Care; 1991; 1991. p. 248-52.
13. Kuhn RA, Reider RS. A C++ framework for developing Medical Logic Modules and an Arden Syntax compiler. *Comput Biol Med* 1994;24(5):365-70.
14. Ohno-Machado L, Gennari JH, Murphy S, Jain NL, Tu SW, Oliver DE, et al. The GuideLine Interchange Format: A Model for Representing Guidelines. *Journal of the American Medical Informatics Association* 1998;5(4):357-372.
15. Bernstam E, Ash N, Peleg M, Tu S, Boxwala AA, Mork P, et al. Guideline classification to assist modeling, authoring, implementation and retrieval. In: Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium; 2000 November 2000; Los Angeles, CA.; American Medical Informatics Association; 2000. p. 66-70.
16. Peleg M, Boxwala AA, Tu S, Greenes RA, Shortliffe EH, Patel VL. Handling Expressiveness and Comprehensibility Requirements in GLIF3. In: to be published in Proc. MedInfo 2001; 2001; 2001.
17. Grosso WE, Eriksson H, Ferguson R, Gennari JH, Tu SW, Musen MA. Knowledge Modeling at the Millennium (The Design and Evolution of Protege-2000). In: Gains BR, Kremer R, Musen M, editors. The 12th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop.; 1999; Banff, Canada; 1999. p. 7-4-1 to 7-4-36.
18. Greenes RA, Boxwala A, Sloan WN, Ohno-Machado L, Deibel SR. A framework and tools for authoring, editing, documenting, sharing, searching, navigating, and executing computer-based clinical guidelines. In: Proc AMIA Symp 1999; 1999; 1999. p. 261-265.

19. Boxwala AA, Greenes RA, Deibel SR. Architecture for a multipurpose guideline execution engine. In: Proc AMIA Symp; 1999; 1999. p. 701-705.
20. Peleg M, Ogunyemi O, Tu S, Boxwala AA, Zeng Q, Greenes RA, et al. Using Features of Arden Syntax with Object-Oriented Medical Data Models for Guideline Modeling. In: Submitted to AMIA Annual Symposium, 2001; 2001; 2001.
21. American College of Cardiology/American Heart Association/American College of Physicians-American Society of Internal Medicine. Guidelines for the Management of Patients with chronic Stable Angina. *J Am Col Cardiol* 1999;33:2092-2197 (<http://www.acc.org/clinical/guidelines/june99/pdf/jun99.pdf>).
22. Schadow G, Russler DC, Mead CN, McDonald CJ. Integrating Medical Information and Knowledge in the HL7 RIM. In: Proc. AMIA Annual Symposium 2000; 2000; 2000. p. 764-768.
23. Boxwala AA, Mehta P, Peleg M, Lacson R, Ash N, Bury J, et al. Modeling guidelines using domain-level knowledge representation components. In: AMIA Annual Symposium, poster presentation; 2000; 2000. p. 645.
24. Starren J, Hripcsak G, Jordan D, Allen B, Weissman C, Clayton PD. Encoding a post-operative coronary artery bypass surgery care plan in the Arden Syntax. *Comput. Biol. Med.* 1994;24(5):411 - 417.
25. Fridsma DB, Gennari JH, Musen MA. Making generic guidelines site-specific. In: Proc AMIA Annu Fall Symp 1996; 1996; 1996. p. 597-601.

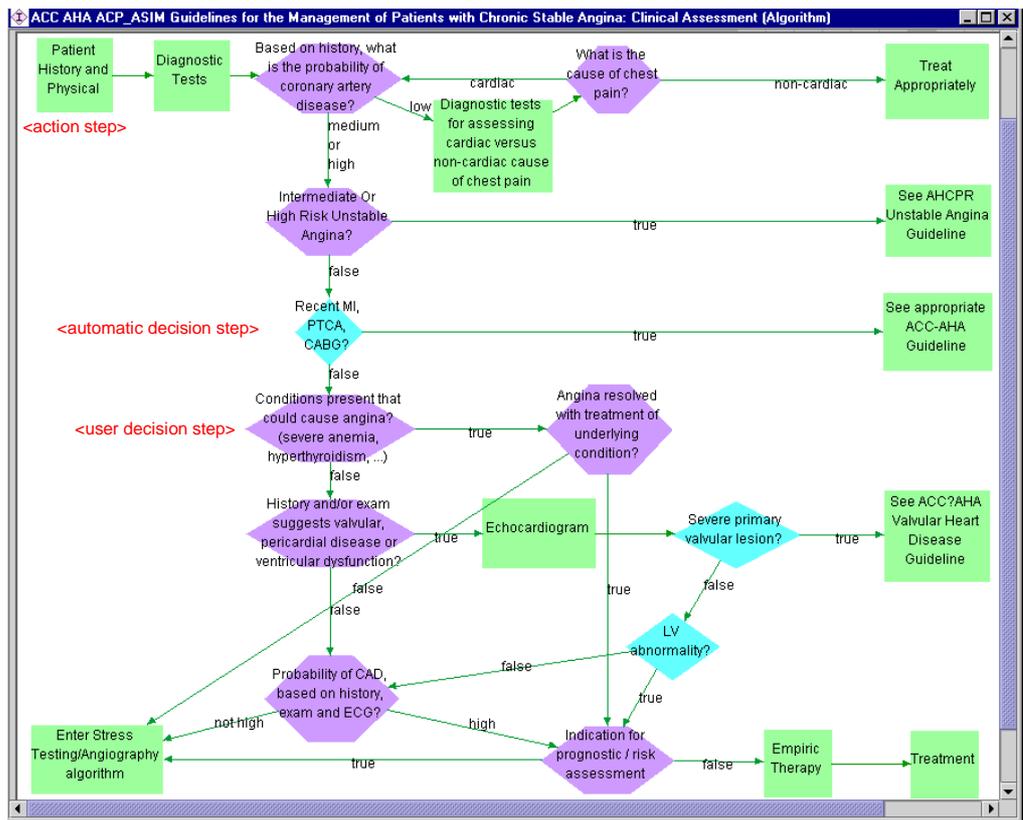


Figure 1. A top-level view of the stable angina guideline, which includes evaluation of chest pain. Boxes represent action steps, diamonds represent automatic decision steps, and hexagons represent user decision steps.

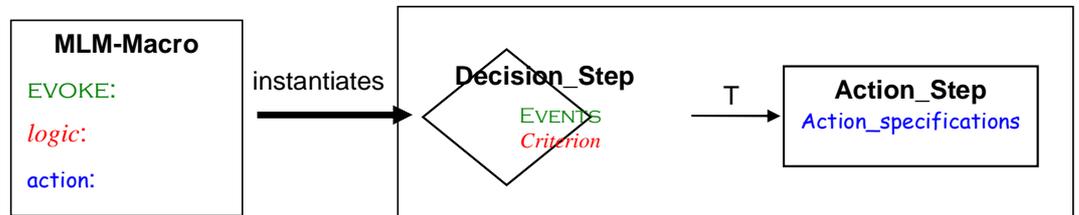


Figure 2. An MLM-macro and its mapping to GLIF steps

Alert: Tests results for CAD risk assessment are back (MLM-macro)	
evoked Event:	event_name: Hemoglobin event_type: patient_data_availability
Event:	event_name: FastingGlucose event_type: patient_data_availability
Event:	event_name: FastingLipids event_type: patient_data_availability
logic: Criterion:	is number (selectAttribute ("value", latest Hemoglobin where time of it >= (now - 12 hours))) and is number (selectAttribute ("value", latest FastingGlucose where time of it >= (now - 12 hours))) and is number (selectAttribute ("value", latest FastingTotalCholesterol where time of it >= (now - 12 hours))) and is number (selectAttribute ("value", latest FastingHDLCholesterol where time of it >= (now - 12 hours))) and is number (selectAttribute ("value", latest FastingTriglycerids where time of it >= (now - 12 hours))) and is number (selectAttribute ("value", latest CalculatedFastingLDLCholesterol where time of it >= (now - 12hours)))
action: Message_Action	message: "Tests results for CAD risk assessment are back."
	destination: <i>mail_dest</i>

Figure 3. Mapping of the guideline encoding to procedures used by the implementing institution.

The institution uses an MLM macro step to create an alert when certain laboratory-test results become available.

(a)

CalculatedFastingLDLCholesterol (Instance of Query_Result)		
value: 150	value: 140
primary_time: 2002-01-08	primary_time: 1999-03-02	

(b)

(c)

Figure 4. The Get_Data_Action and the query result, which holds CalculatedFastingLDLCholesterol data values and their primary times. (a) The

CalculatedFastingLDLCholesterol query result. (b) The
CalculatedFastingLDLCholesterolDataItem from which the query result was extracted. (c) The
Get_Data_Action used to retrieve the query result

Captions:

Figure 1. A top-level view of the stable angina guideline, which includes evaluation of chest pain. Boxes represent action steps, diamonds represent automatic decision steps, and hexagons represent user decision steps.

Figure 2. An MLM-macro and its mapping to GLIF steps

Figure 3. Mapping of the guideline encoding to procedures used by the implementing institution. The institution uses an MLM macro step to create an alert when certain laboratory-test results become available.

Figure 4. The `Get_Data_Action` and its query result, which holds `CalculatedFastingLDLCholesterol` data values and their primary times. (a) The `CalculatedFastingLDLCholesterol` query result. (b) The `CalculatedFastingLDLCholesterolDataItem` from which the query result was extracted. (c) The `Get_Data_Action` used to retrieve the query result