Proposed Revisions to GLIF

This document lists new requirements for GLIF, some solutions that have been experimented with or are proposed, and a prioritization for approaching the revisions to GLIF.

Requirements for GLIF

The list of requirements for GLIF was developed from experience in various guideline projects at SMI and DSG since the publication of GLIF-2.0. The requirements are categorized into model-level, concept-level, documentation and syntax requirements.

Model-level requirements

Model-level requirements are requirements of the GLIF Model in general, which are not concerned with individual concepts of the GLIF Model.

Guideline adaptation for institutional use

Guidelines being interchanged among institutions will need some adaptation for local use by the recipient institution. For example, a guideline developed at the American College of Physicians is likely to need refinement to support local workflow patterns or practice preferences. As an interchange format, GLIF must facilitate such adaptation. Further, the adaptations to the guidelines must be appropriately documented.

Guideline integration with institutional systems

GLIF-encoded guidelines refer to patient data items in decision criteria and to clinical actions in action steps. For using such guidelines within electronic medical record systems and order entry systems, references to patient data and clinical actions in guidelines will need to be mapped to their instantiations or implementations in the information systems.

Supporting views

Different people can view different parts and features of the guidelines, for different purposes. For example, a patient may want to know during which steps he/she will have to be hospitalized and for how long. Another example, a specialist view of a guideline will be different from the view of a general practitioner. Also, it may prove to be valuable to show the guideline both from the point of view of a single patient and from the top-level view, showing all possible paths of the guideline. Other useful views may include viewing all potentially necessary input patient data for the entire guideline, all possible patient states relevant to a guideline, and all possible exceptions of a guideline.

Model extensibility and reuse

The GLIF model and syntax must provide for easy extensibility for different types of applications. For example, GLIF has been used or is being considered for use in encoding (chronic) disease management guidelines, screening and risk assessment guidelines, clinical protocols, and educational applications. New concepts for representing application-level constructs more directly may be beneficial.

Complexity Management

Since guidelines may be very complex, there is a need for a complexity-management mechanism. This mechanism should also allow representation of a guideline in the context of other guidelines.

Revision Control and Documentation

Guidelines are revised periodically as new knowledge is acquired. GLIF must support marking and documenting of the revisions such that changes to the guidelines and the reasons for the changes can be well understood.

Concept-level requirements

Concept-level requirements are requirements for introducing new concepts into GLIF and defining their syntax and semantics. Some of the concept-level requirements include requirements for changing the syntax and semantics of existing concepts.

Conditional Step

The current conditional step in GLIF uses an extended Boolean model. This makes it cumbersome and error-prone to represent criteria that do not have a true-or-false outcome (e.g., what is the patient’s age category: neonate, infant, toddler, child, adolescence, adult, elderly).

Decision Hierarchy

A Choice Step should be used to represent decisions, which require making a heuristic choice from a set of pre-enumerated alternatives. Each alternative lists its rule-in and rule-out conditions, to help determine whether that alternative may be relevant. For a given patient state, more than one alternative could be taken. For example, a case where the drug given to a patient should be changed, but there are several alternative drugs that can be prescribed given the patient’s state.

Branching

We propose changing the semantics of a Branch Step so that its selection method can be either “all of” or “some of”, but not “one of”. Allowing a Branch Step to have a selection method of “one of” makes the semantics of a Branch Step overlap the semantics of a Conditional Step. Omitting the possibility of the “one of” selection method removes this overlap. This way, in order to represent flow of control from step s_1 to one of mutually exclusive steps, t_i, a Conditional Step will be used to link s_1 to t_i. In order to represent flow of control from step s_1 to several steps, t_i, a Branch Step will be used to link s_1 to t_i.

Iterations

Currently, GLIF does not explicitly support modeling of iterations (analogous to the while and for loops in C). An iteration concept would allow one to represent a repetition of a segment of the guideline until specified events have occurred. GLIF should be able to model repetitive steps (e.g., giving 5 CAF treatments). There should be a way to express while-loops and for-loops, as well as characterize cyclic processes by their cycle/frequency (how much time passes between the start of one iteration of the process and the next iteration) and duration (how long does each iteration last).

Representing activities

Activities are processes that have duration, unlike actions, which are considered instantaneous, using the granularity of the guideline. An example of an activity is a drug regimen: a patient taking a certain dose of a certain medication, at a certain frequency (twice a day) for a specified duration.

Annotation

An annotation makes the guideline more understandable. Consider the ACP guideline for thyroid screening. The guideline recommends TSH and thyroxin tests and based on their results recommends management actions. In GLIF, there is no concept that allows one to represent *hyperthyroidism* as the reason for a particular set of management actions.

Patient states

Representing patient states will allow a clinician to synchronize the management of a patient to situations handled by a guideline. For example, if a patient returns to the clinic, the clinician will have to match the patient's state to the appropriate step(s) in the guideline that are currently active.

Structuring logical criteria

In order for a guideline to be sharable, the logical criteria should be structured and not be expressed simply as text. Logical criteria should be expressed as logical expressions that refer to medical-domain ontology.

Defining a Medical Domain Ontology

A clinical guideline references many medical concepts. These concepts have to be defined by a medical domain ontology that will be referenced by the guidelines. The domain ontology will reference terms from a specific medical vocabulary. The medical domain ontology would be mapped by the institution, which is using the guideline, to its own medical domain ontology. We need to consider whether the domain ontology will be part of GLIF, or not, and whether it should be based on one specific vocabulary.

Events

Some guidelines may require the notion of events. One such case is when a guideline should be executed in a response to an event (e.g., on the event heart failure some guideline should be executed). Another case is when temporal constraints are to be specified and they refer to the time of an event (e.g., check blood pressure 10 minutes after the administration of a drug).

Temporal constraints

Some guidelines specify temporal constraints. If the guidelines would be executable, then it would be critical that the temporal constraints be specified using special constructs, and not just written as text.

Goals (intentions)

GLIF must provide a means for specifying a declaration of goals to achieve and not only a procedural specification of how to achieve a goal

Exceptions and exception handling

An exception is a deviation from the usual path(s) of the guideline. For example, under what conditions (e.g., severe anemia) should we alter a patient’s treatment? Another example is an adverse event during a protocol.

Patient Data

In GLIF-2.0, patient data are not structured in any way. There is a need to characterize patient data and to define their structure.

In addition, only Action Specifications include reference to Patient Data. Nevertheless, Conditional Steps, Branch Steps, and Synchronization Steps need also to include reference to Patient Data. It should be clear whether certain patient data is used as an input to a step, or result as an output of that step.

Will the definition of patient data be part of the domain ontology?

Also see discussion above on *Guideline integration with institutional systems*.

Actions

ActionSpec needs to be fully defined. Perhaps, classes of different types of actions need to be created.

Keyword didactic

Adding a keyword type of didactic will facilitate searching databases for pertinent supplemental resources.

Documentation requirements

GLIF and its revisions must be accurately and completely documented. Several aspects of GLIF and its concepts are not well understood because of lack of documentation. For example, it is not clear what it means to include a PatientData object in an ActionSpec.

Syntax requirements

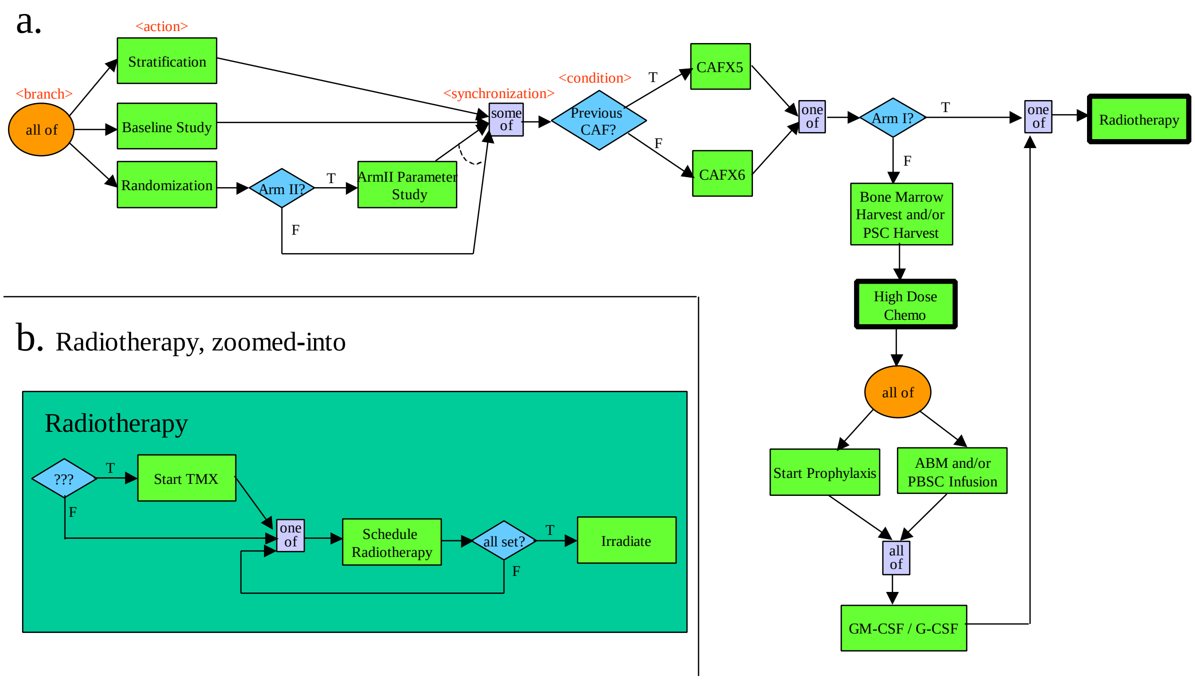
Since GLIF-2.0, XML has become popular as a packaging and transport syntax. Currently, several tools that manipulate XML documents are available that can be useful for parsing, authoring, and validating. We should consider switching from the ODIF syntax to XML syntax.

Solutions

Nesting

Nesting is very useful for managing the complexity of guidelines. Nesting enables looking at a guideline from a top-level view and then zooming into/out of some of its parts. Nesting is also useful in representing a guideline in the context of other guidelines. Since nesting allows grouping of parts of a guideline into a single unit, this is a mechanism that can allow model extensibility and reuse of part of a guideline (defining macros), or adaptation of a guideline to a specific institution by replacing specifications for parts of a guideline (i.e., replacing a goal with a procedure).

By use of nesting, steps will be defined as containing other steps. As such, the guideline will be iteratively refined from a coarse level to a fine level of detail. For example, a top-level view may show an action step named “Radiation”, while a zoomed-into view of this action step will show the network of action and conditional steps that fully describe the Radiation action (Figure 1).



b. Radiotherapy, zoomed-into

Figure 1. A top-level and a zoomed-into view of a Breast Cancer guideline. (a) A top-level view of the Breast Cancer guideline. (b) A zoomed-into view of the Radiation action step. Nested steps are shown with a thick border in the top-level view.

Defining Macros

Macros can be a means by which a GLIF encoded guideline, or parts of such a guideline can be reused in other guidelines. A macro is a special kind of nested step in which the contained GLIF steps follow a specific pattern for each type of macro. This pattern of GLIF steps represents an application-specific construct. An example of such a macro is the GEODE state defined by the DSG, and shown in Figure 2.

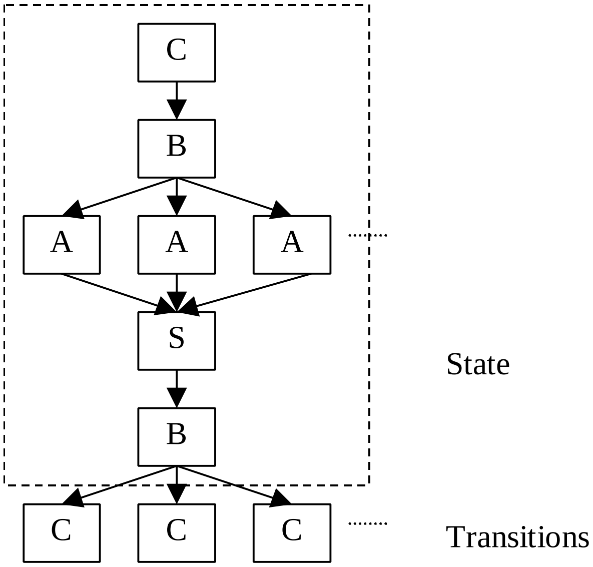


Figure 2. A GEODE clinical state map can be modeled as a pattern of GLIF steps. A state-map contains states and transitions. A state contains criteria for a patient to be eligible for that state and a list of management actions that are performed during that state. The box with the dotted border shows how a GEODE state is defined as a sequence of GLIF steps. The topmost conditional step is the eligibility condition. The subsequent branch step and action steps describe the actions performed in the state. The branch step and conditional steps at the bottom are used to define transition criteria into other states.

[Legends:A=Action Step; B=Branch Step; C=Conditional Step; S=Synchronization Step]

Defining a Medical Domain Ontology

A medical domain-ontology will define medical concepts and procedures. Criteria and action specification could then be expressed as computable expressions that refer to medical-domain ontology, instead of being just textual descriptions.

A medical domain ontology will aid in the adaptation of the guideline by an institution, since it will identify parts of the guidelines that have to be matched to the institution’s definitions, namely terms mentioned in the criteria and action specifications of the guideline.

Structuring patient data

If patient data, which are referenced by criteria and action specifications, are structured, then this can facilitate easier mapping of patient data referenced in the guideline to the institution’s patient records. A hierarchical class model for patient data will be created.

Changing a Conditional Step from representing a Boolean decision to a Switch (case statement)

This will allow representing a choice among more than two alternatives.

Defining a choice hierarchy

The different choice classes are needed to represent determinate choices and non-determinate choices. For example, different weights can be given to different rule-in and rule-out conditions.

Removing the “one of” from selection method from the branch step

This will eliminate the overlap between branch and conditional steps.

Defining an Action/Action-Specification hierarchy and a full Step-hierarchy

This will solve the classification and definition of repetitive actions, cyclic actions, and actions that have duration (activities). It will also define nesting rules that will define how a complex decision or action step will be broken into a sub-graph of steps (i.e., will a compound decision step be broken into sub steps that always end with a decision step?)

Scenario Step

This step will allow annotating the guideline by explicitly representing expected patient states at different stages of the guideline. The scenario step will also allow a clinician to synchronize the management of a patient to situations handled by a guideline.

Structuring Logical Criteria and Temporal constraints using Extended Arden Syntax

In GLIF2.0 logical criteria and temporal constraints are expressed in text. Specifying logical criteria and temporal constraints in a structured way, by using logical expressions that reference the medical domain-ontology will facilitate the mapping of these expressions to the different institutions and the ability to parse these expressions. We propose to use an extended version of Arden syntax. Arden syntax has already been adopted as a standard by the American Society for Testing and Materials (ASTM) and is used for modeling the logic for making (single) clinical decisions in guidelines. It has a rich language for expressing temporal constraints. However, Arden is based on the notion of Event-Condition-Action rules, whereas GLIF2.0 does not support expression of events. We are considering adding the notion of events to GLIF, but many GLIF criteria will not be related to events. Therefore, we may have to use something similar, and not identical, to Arden syntax.

Events

Introducing the notion of events into GLIF will aid in the implementation of the guideline in event-driven systems. It will also allow the expression of event-driven guidelines. Events can be a way to approach modeling of exceptions.

Expressing Goals

We do not yet have a solution for expressing goals. Yuval Shahar's work on intention-based language for representing guidelines may be a useful starting point.

Defining Exceptions

We would like to define exceptions at different levels: single action, sub-guideline, and entire guideline. Including or not including events in GLIF will determine how exceptions will be expressed in GLIF.

Iteration

A consideration in developing the solution is whether to incorporate a step as part of an iteration concept or to have the step contain iteration information.

Solution **Problem**	Nesting	Macros	Medical Domain Ontology + keywords	Structuring Patient Data	Changing a Conditional Step from a Boolean decision to a Switch	Defining a Choice hierarchy	Removing XOR from Branch Step	Defining a hierarchy of actions + Step hierarchy	Scenario Step	Structuring Logical Criteria and Temporal constraints using Extended Arden Syntax	Events	Look at Yuval’ s work on the intention-based language for representing guidelines	Defining exceptions at different levels: single action, entire guideline
Complexity management	+												
Model extensibility and reuse	+	+											
Institution adaptation	+		+	+									
Criteria is expressed as text			+										
Defining action specification more precisely			+										
Choice between many alternatives is not supported					+								
Modeling non determinate choices is not supported						+							
Overlap between Conditional and Branch steps							+						
Iteration of actions, and cyclic actions cannot be expressed								+					
No notion of activities: states with **duration**								+					
Annotation is not possible									+				
Patient States cannot be expresses									+				
Logical criteria expressed as text										+			
Integration with event-driven implementation											+		
Exception expression	+										+		+
Modeling event-driven guidelines											+		

Expression of temporal constraints										+			
Expressing goals												+	

Prioritization

Nesting							AB/MP
	Macros						AB
Iterations							MP
Step hierarchy							AB/MP
	Action hierarchy						MP
	Decision hierarchy						RL
		Goals					?
Branching							MP
Conditional step							RL/MP
Keyword didactic							AB
Structuring logical criteria							OO
	Temporal constraints						OO
Domain ontology							QZ
		Events					?
			Exceptions				AB
			Scenarios				?
				Supporting views			EB

TIME →

1. Initials in right column indicate the person participating actively in the effort (AB=Aziz Boxwala; MP=Mor Peleg; RL=Ronilda Lacson; OO=Omolola Ogunyemi; QZ=Qing Zeng; EB=Elmer Berstam)
2. Rows in shaded in gray indicate groups of tasks, which are dependent on each other (e.g., macros are dependent on nesting).