Tool Support for GLIF3: Status Report (10/27/2000)

Tool-support is a crucial issue for guideline development and use. We are planning to build tools to support browsing, authoring, validation, downloading, and execution of GLIF-encoded guidelines. Following is a short description of the tools we are building already and/or planning to build.

1) Authoring tools
At Stanford, we have using the Protege knowledge acquisition tool developed by Mark Musen's group (http://protege.stanford.edu) to author guidelines in GLIF. The guideline flowcharts (algorithms) are graphically displayed by Protege's diagram widget, that lets the user create diagrams by dragging and dropping nodes and connectors. The other GLIF objects are entered using Protege's forms that are automatically created by the Protege tool from the GLIF ontology. We envision the process of authoring to take 3 steps: authoring by a domain expert, authoring by an informatician, and local adaptation. Correspondingly, GLIF3 has 3 levels of specification: the conceptual flow-chart that is created by the medical domain expert (a physician, usually), the computable specification that is created by the informatician, and the implementable level that still needs to be worked out. I am writing a document that is a "walk-through" of the authoring process using Protege, using a management of chronic cough guideline. I will be happy to send it to you when it is ready (within a few days). The Protege team is working (with our collaboration) on a wizard that will guide the authors of the different levels of specification through the authoring process of GLIF.

We have also completed two “widgets” for Protégé for RDF support. One widget can translate guidelines, encoded in GLIF 3.0 and created using Protégé, into RDF format. The other widget can import RDF files containing GLIF-encoded guidelines into Protégé, and automatically lay out the flowcharts diagrammatically on the screen.

The Decision Support Group (DSG) at Harvard is developing an authoring tool, which is specific for GLIF. Many of the editing forms for the authoring tool are complete. A tree-based visualization component for guidelines has been completed and work is underway on a flowchart-based visualization component. The main features of this tool will be:
- Supporting development of guidelines at multiple levels of abstraction with clear separation of abstraction levels
- Supporting authoring of different types of guidelines using templates or *macros*
- Providing multiple displays of the guideline, (e.g., flowchart, timeline)
- Version management of a guideline.

2) A guideline server that will provide the following:
- Guideline authors can submit and edit guidelines over the web, and provide annotations for guideline classification
- Users could browse and download guideline
- Eligibility determination

3) Vocabulary tools

Vocabulary editing and visualization tools have been developed for linking guidelines to the domain ontology.

4) Validation tools
We would like to develop tools that will aid in finding inconsistencies, and incompleteness in guideline specifications. Some of the things that we would like to check for are:
- Syntax checks
- Type checking
- Range checking
- Finding missing "elses" (e.g., a branch with only one branch destination or a decision with only one next step)
- Checking that all the data items that are referenced by a criterion are listed as data items that the guideline contains, and that the Get_Data_Action for variable data items acquires the data beforehand.

Decision criteria are written in a modification of the Arden Syntax logic grammar. An interpreter for this modified language is under development at DSG. The interpreter will be used for validation during authoring and for evaluation of criteria and expressions during execution.

5) Execution tools
Guideline execution would be based on categories of guideline steps. For the current version of GLIF specification, we plan to support execution for action steps, decision steps, and patient state steps. An architecture for the execution engine has been proposed.

DSG have developed a server that provides simple search capabilities. It includes an object-oriented RDF parser.

Proposed additions to the GLIF Model

We are exploring the possibility of integrating into GLIF the GEM model of Richard Shiffman (http://ycmi.med.yale.edu/GEM). In particular, we are interested in classes and attributes that define:
- Evidence: the evidence class is used to describe the scientific documents, which support the suggestion, recommendation, and view of points in medical guidelines.
- Method: The Method class contains all the information about the evaluation of methods used.
- Purpose: To describe the goals of the clinical guideline.