
Marilyn D. Paterno, BMus, BS, Rita D. Zielstorff, RN, MS, Jonathan M. Teich, MD, PhD, Mark Segal, MS, Roberta Fox, MS
Clinical Systems Research and Development, Partners HealthCare System, Boston, MA.

BACKGROUND

The automated alerting system (Event Engine) running at our institution is a well-developed tool for detecting individual clinical events and notifying the clinician with appropriate action suggestions.1 However, protocols, critical pathways, and clinical practice guidelines (CPG's) typically involve more than the detection of a single event. Once an initiating event, e.g., a particular diagnosis or test result, has occurred, the steps followed by the clinician in the management of the problem may vary according to several factors. These include passage of time, such as waiting for a test to be done or a new protocol day; system data, such as the result of the test, or clinician input, such as answers to questions. We have designed an automated processing engine (Navigator) to assist clinicians in managing multiple step clinical algorithms.

Inspired by the Intermed Collaboratory GuideLine Interchange Format (GLIF)2, we determined that an algorithm should include specific descriptive data and instructions for one or more steps to be followed. We developed a method by which an analyst can enter CPG's into the knowledge base.

SYSTEM DESCRIPTION

The first step in defining an algorithm is to take the text version of the CPG and turn it into a flowchart with decision steps and action steps. To do this, the owners of the CPG meet with a medical informatics analyst from our staff who assists them in putting it into flowchart form. This process results in a format that the analyst can follow when entering the algorithm.

When the flowchart is complete, the analyst uses the algorithm editor to add the steps and the overall information, including parameters for clinician notification, to an M database. The editor uses modules from the Event Engine's rule editor to define decision steps. Action step parameters may include queries to the clinician, text recommendations, and executable orders. The user interface is Hyper-M, which is a windows-emulating product used with the M language. The design includes a display area that is updated after each change to show the current version, and a series of controls that activate windows used to update various parts of the algorithm. Entry is modular; it is not necessary for all steps or parameters to be entered at once, or in a particular order. A verification routine checks the definition at each filing for logic and completeness.

EVALUATION

The editor has been designed for use by sophisticated analysts who understand the principles of the Navigator and can follow the flowchart representation of a CPG. We have successfully used it to create a fairly simple algorithm for sleep apnea, and are in the process of defining more complex ones for congestive heart failure and hypercholesterolemia.

CONCLUSIONS

Specification of algorithms is a complex process. Using the algorithm editor, trained analysts can specify clinical algorithms, without any programming, which then are automatically triggered and executed in our environment. We anticipate that this tool will be valuable in facilitating the use of CPG's in our environment. Several clinical groups have already submitted CPG's for inclusion.

References