

# An Implementation Framework for GEM Encoded Guidelines

Peter Gershkovich, MD, Richard N. Shiffman, MD, MCIS  
Yale Center for Medical Informatics  
New Haven, Connecticut

## Abstract

*Access to timely decision support information is critical for delivery of high-quality medical care. Transformation of clinical knowledge that is originally expressed in the form of a guideline to a computable format is one of the main obstacles to the integration of knowledge sharing functionality into computerized clinical systems. The Guideline Element Model (GEM) provides a methodology for such a transformation. Although the model has been used to store heterogeneous guideline knowledge, it is important to demonstrate that GEM markup facilitates guideline implementation. This report demonstrates the feasibility of implementation of GEM-encoded guideline recommendations using Apache Group's Cocoon Web Publishing Framework. We further demonstrate how XML-based programming allows for maintaining the separation of guideline content from processing logic and from presentation format. Finally, we analyze whether the guideline authors' original intent has been sufficiently captured and conveyed to the end user.*

## Background

Translation of published clinical guidelines into a computable format and subsequent integration of guideline knowledge into the flow of medical care remains one of the most challenging tasks in the development of clinical decision support systems. A notion that the current state of technology does not allow for a direct implementation of clinical guidelines without additional processing by humans has triggered the development of various approaches to such transformations.

Many of the current models are focused, however, on secondary extraction of an algorithm, which is sometimes explicitly included and sometimes implied by a guideline. The discretionary nature of such an approach may be one of the reasons for the inconsistency in guideline transformation shown by some studies<sup>1-3</sup>. Therefore, a more comprehensive guideline representation has been developed, which allows for minimal transformation of a

guideline and yet structures guideline knowledge enough that a computer can process it.

The Guideline Element Model (GEM) provides a general document model applicable to clinical guidelines via a hierarchy of more than 100 elements.<sup>4</sup> The model is based on an eXtensible Markup Language (XML) specification, which is a universal format for structured documents and data as defined by the World Wide Web Consortium<sup>5</sup>. Based on the Structured General Markup Language (SGML) that originated in the early 1980s, XML has become one of the most promising technologies for representation of structured data and data interchange. Various associated standards such as Extensible Stylesheet Language (XSL) and Extensible Stylesheet Language Transformations (XSLT) define a set of rules for data processing and content presentation.

XML is - more than other programming tools - instrumental in keeping content separate from programming logic<sup>6</sup>. It would be beneficial for a developer of a knowledge-sharing system to keep logic, content, and format independent, thereby facilitating maintenance of the encoded guideline.

## Objectives

In this paper, we describe a project that demonstrates a potential implementation approach for GEM-encoded guidelines. The work described has the following objectives:

- ✓ To determine the feasibility of direct implementation of GEM-encoded recommendations from current guidelines.
- ✓ To maintain the separation of guideline content from implementation logic and from presentation format.
- ✓ To maintain authenticity of the guideline authors' intent throughout the translation process.

## The Implementation Framework

The system takes advantage of existing networks joined via the Internet. An intended user of the

system must have access to a W3C compliant Web browser, (for example Netscape, MS Internet Explorer, Opera etc.). All processing is done on the server to avoid browser-specific implementation. In addition when a different presentation is necessary for a distinct type of browser, this can be accomplished through browser- or device-specific processing instructions.

The system accepts as input an HTTP request for a GEM-encoded guideline file, in which guideline knowledge has been demarcated with appropriate XML tags. The GEM file has been validated previously and saved as an XML document<sup>4</sup>. The system extracts pertinent information from the GEM file and produces an HTML file that displays the information related to guideline recommendations. Specifically, decision variables and their potential values are displayed. When the user enters patient-specific values for these variables, the system automatically presents guideline-recommended actions.

We selected Apache Group's Cocoon publishing framework<sup>7</sup> for implementation of GEM-encoded guidelines. Cocoon is an XML-aware servlet, which is responsible for processing and formatting of XML documents. Cocoon takes full advantage of current XML technology and provides necessary components for efficient server-side performance. The framework components are Apache Web server, Tomcat servlet container, and Cocoon XML aware servlet. The Apache Web Server and Tomcat servlet container are responsible for correct handling of HTTP requests and for forwarding an XML file requests to Cocoon for processing.

Cocoon provides the following essential components for the implementation model.

- ✓ Producer – takes a request and produces an XML file.
- ✓ Processor – applies the specified logic for processing of an XML file. For example an XSLT processor applies transformation rules to an XML tree.
- ✓ Formatter - produces the output according to Processor instructions.

A critically important feature of Cocoon is its ability to determine the type of device or browser used to make an HTTP request. This allows the system to select appropriate processing instructions for each device. Additionally,

Cocoon provides its own method for generation of dynamic XML documents via eXtensible Server Pages (XSP)<sup>10</sup>. This open standard provides users with a powerful method for dynamic data integration. We use XSP to compile a new XML document dynamically as will be shown further. XSP allows access to external data (other than XML documents), such as databases or server-side files and components.

The interaction diagram shown in Figure 1 describes the consequent activation of system components.

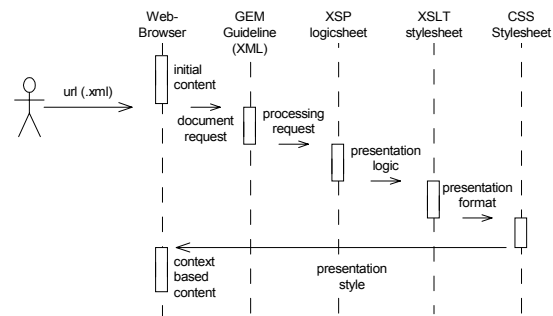


Figure 1. System Interaction Diagram.

The Cascading Stylesheet (CSS) specification (the last processing instruction on the diagram) is applicable when dealing with HTML output and adds a convenient method to define a presentation style (for example, color of the text or font size) which can be passed to the browser.

We also used GNU Emacs<sup>11</sup>, an extensible, customizable, self-documenting real-time display editor, for editing of XML documents and writing XSLT and XSP programs. GNU Emacs has many convenient features (for example, an SGML editing mode), which are useful when working with XML documents.

### Guideline Implementation System

We selected the guideline on diagnosis, treatment, and evaluation of initial urinary tract infection (UTI) in febrile infants and young children from the American Academy of Pediatrics (AAP) and the guideline on management of latent tuberculosis infection (LTBI) from the Centers for Disease Control and Prevention (CDC) for implementation.<sup>8,9</sup> These guidelines were produced by respected authorities and have been widely disseminated. They are representative of the wide variety of guidelines produced by national organizations. Both guidelines were transformed using the implementation framework we describe.

Together the guidelines supplied 12 recommendations, comprising 15 conditionals and 2 imperatives. 24 unique decision variables and 17 suggested actions/directives.

When a guideline is first requested the <identity>, <developer>, and <purpose> elements of the document model are presented to the user. The screenshot shown in Figure 2 demonstrates the context-independent (default) presentation style. This general view of a guideline is appropriate when the system is unaware of context (such as clinical case details). The GEM-encoded guideline is transformed using an XSLT stylesheet to show general guideline characteristics and to provide options for further interactions. The browser window displays:

- Guideline title
- Options on how to proceed
- General information about the guideline and developer

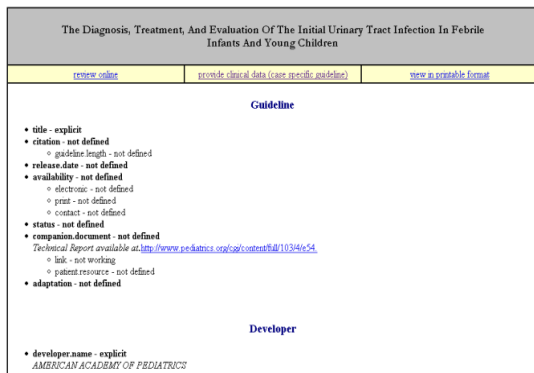


Figure 2 Default view of UTI guideline.

This view is generated via an XSP logicsheet responsible for context analysis. If no context parameters are provided, the default XSLT stylesheet is applied. Finally, a cascading stylesheet is applied to format the presentation output.

Three options are expressed as hyperlinks and available at this point: (1) review the guideline on-line, (2) review it in printable format, or (3) make the guideline-case specific by providing clinical data. The on-line version of the guideline has convenient links to various portions of the guideline; it is usually used to navigate through the document content on the Web. An important distinction, however, is that each view is dynamically produced per user request and has the same underlying source – the GEM-encoded guideline. The same underlying source is used when the user selects a printable copy of the document, which again is generated dynamically.

The caching mechanism of Cocoon takes care of performance issues and makes sure that processing of a document occurs when it is first requested or when it has changed.

If the user proceeds to enter case-specific data, the XSP logicsheet recognizes the supplied parameters and a different stylesheet is applied to process the same GEM-encoded guideline. All conditional decision variables are processed and recommendations are presented to the user based on the value (state) of the variables. The screenshot shown in Figure 3 demonstrates this.

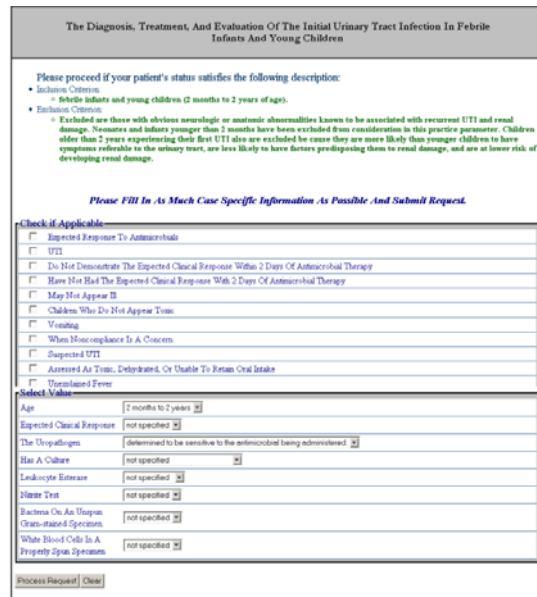


Figure 3 Automatically generated form.

The user is prompted to supply clinical data to the processing engine. Figure X shows the guideline's inclusion/exclusion criteria and all unique variables that exist in the guideline. Some variables with implied true or false values are presented as check boxes and other variables that have defined values are displayed as combo boxes.



Figure 4 Automatically generated recommendations

After a user enters patient-specific data to the system and clicks the “Submit” button, the XSP processor applies processing logic, which was defined in the logicsheet. That logic requires the XSP processor to detect all submitted parameters and add them to a dynamically created XML document. Next, it adds necessary elements (for example a matching <action> element) of the GEM-encoded guideline and sends the final output to the XSLT processor, which in turn applies the appropriate stylesheet. In that case all recommendations (conditional and imperative) are presented to the user as shown in Figure 4. This time the system displays recommendations that satisfy patient-specific criteria and relevant recommendations.

Despite a complex transformation process authenticity of the recommendations is maintained. This is possible because the GEM document model and the implementation framework allow preservation of the original language of the guideline. Here is an example of the original guideline recommendation:

*Recommendation 6*

*If the infant or young child 2 months to 2 years of age with suspected UTI is assessed as toxic, dehydrated, or unable to retain oral intake, initial antimicrobial therapy should be administered parenterally and hospitalization should be considered.*

Transformed according to the GEM document model, the recommendation (shown in Figure 5) is computer processable but somewhat complex to read.

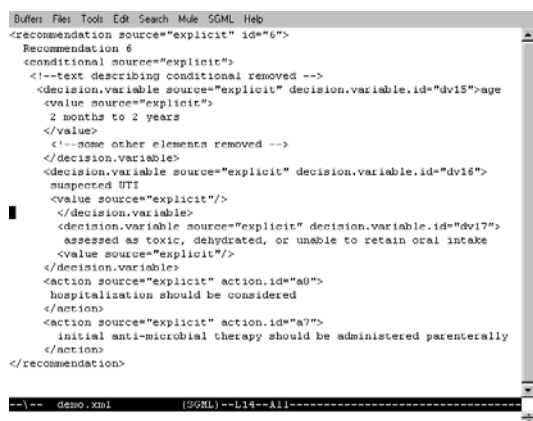


Figure 5. Emacs window showing a fragment of GEM-encoded guideline.

However, as we toggle tag visibility in a GNU Emacs window (as shown in Figure 6), it becomes apparent that there is no semantic

difference between the language of the original guideline and the GEM-encoded version.

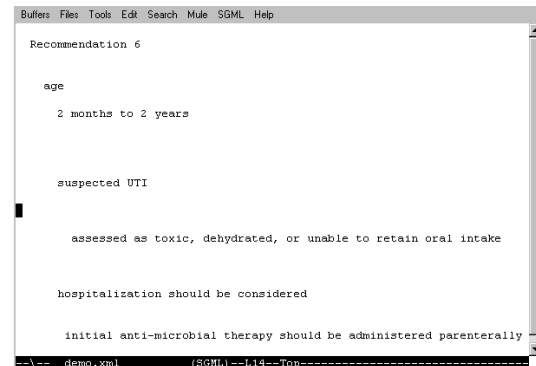


Figure 6. Emacs window showing the same fragment of GEM-encoded guideline as in Figure 5 without tags.

Therefore the only difference between the two is the presentation format. As shown in Figure 4, the presentation can be modified according to the system specifications. In other words, display of recommendations is independent of the guideline and can be presented based on user preferences.

**Discussion:**

The implementation approach presented here demonstrates that it is possible to extract decision variables along with inclusion and exclusion criteria from GEM-encoded guidelines and automatically build a convenient interface for user interaction with the system. It is also possible to retrieve appropriate recommendations from GEM-encoded guidelines based on selected parameters and to display them in a form that indicates guideline-suggested actions. That functionality constitutes the core features necessary to build a more sophisticated system, which would provide data for decision support and evaluation.

The GEM document model provides a comprehensive approach to guideline knowledge representation. We demonstrate that, guideline content does not undergo significant semantic transformation during the GEM encoding process. Arguably, it might be possible to author a guideline, which would not require semantic transformation at all if guidelines were originally developed according to strict standards. The domain expert in such a case would only have to put easily recognizable elements and their attributes in the corresponding nodes in a document model tree. The reality, however, forces us to deal with a variety of guidelines that do not follow strict rules of development.

Therefore, a test of authenticity of transformation should be applied to the encoded guideline document as part of its evaluation. The ability to preserve the language of the guideline and to produce authentic recommendations is essential for preservation of the original intent of the guideline.

Our choice of the XML family of standards was based on its unique combination of features. The Cocoon Publishing Framework was selected for its accessibility to other researchers via open source, the stability of its current version, and its current use in many production applications.<sup>12</sup> The ultimate test for this implementation framework will be a detailed evaluation of its components and must include end-user acceptability.

The combination of XML-based standards and Cocoon allows us to place distinct boundaries between the content of the guideline, the implementation logic, and the presentation format. The ability to maintain such separation facilitates independent development and modification of GEM-encoded guidelines by domain experts and allows programmers and designers to focus on implementation issues without unnecessary cross training.

This work demonstrates the feasibility of GEM document model implementation and helps to define areas that require enhancement or clarification. Another direction for further research is to leverage the device-independent nature of the framework and to demonstrate that different Web-enabled devices can be used successfully with the same GEM-encoded guideline thereby expanding potential uses for the implementation framework.

## References:

1. Ohno-Machado L, Gennari, JH, Murphy, SN, et al. The GuideLine Interchange Format: A Model for Representing Guidelines. *J. Am. Med. Inform. Assoc.* 1998 5: 357-372
2. Patel VL, Allen VG, Arocha JF, Shortliffe EH. Representing Clinical Guidelines in GLIF: Individual and Collaborative Expertise. *J. Am. Med. Inform. Assoc.* 1998 5: 467-483
3. Karras BT, Nath SD, Shiffman RN. A Preliminary Evaluation of Guideline Content Mark-up Using GEM--An XML Guideline Elements Model. *Proc AMIA Symp* 2000. : 413-417
4. Shiffman RN, Karras BT., Agrawal A, Chen R, Marengo L, Nath S. GEM: A Proposal for a More Comprehensive Guideline Document Model Using XML. *J. Am. Med. Inform. Assoc.* 2000 7: 488-498
5. Introduction to XML specification. World Wide Web Consortium (W3C). URL <http://www.w3.org/XML/>
6. Jones KA. Better Way for Web Development. *XML Magazine* 2000:4. Available from URL: <http://www.xmlmag.com/upload/free/features/xml/2000/04fal00/kj0004/kj0004.asp>
7. Introduction materials to Apache Group's Cocoon Project. Available from URL: <http://xml.apache.org/cocoon/index.html>
8. American Academy Of Pediatrics The Diagnosis, Treatment, and Evaluation of the Initial Urinary Tract Infection in Febrile Infants and Young Children (AC9830) *Pediatrics* Volume 103, Number 4 April 1999, pp 843-852. Available from URL: <http://www.aap.org/policy/ac9830.htm>
9. Targeted Tuberculin Testing and Treatment of Latent Tuberculosis Infection. *CDC* 2000 7 49(RR06);1-54. Available from URL: <http://www.cdc.gov/mmwr/preview/mmwrhtml/rr4906a1.htm>
10. XSP specification Mazzocchi S, Rocha R, Working Draft - 2000-01-09 Available from URL: <http://xml.apache.org/cocoon/wd-xsp.html>
11. GNU Emacs.[computer program] Available from URL: <http://www.gnu.org/software/emacs/emacs.html>
12. Web Publishing Frameworks In: *Java and XML*. McLaughlin B. O'Reilly&Associates, Inc., June 2000. p. 232-76.