Abstract As part of an enterprise effort to develop new clinical information systems at Intermountain Health Care, the authors have built a knowledge authoring tool that facilitates the development and refinement of medical knowledge content. At present, users of the application can compose order sets and an assortment of other structured and coded knowledge. The flexible nature of the application allows the immediate authoring of new types of documents once an appropriate XML schema and accompanying Web form have been developed and stored in a shared repository. The need for a knowledge acquisition tool stems largely from the desire for medical practitioners to be able to write their own content for use within clinical applications. We hypothesize that medical knowledge content for clinical use can be successfully created and maintained through XML-based document frameworks containing structured and coded knowledge.


In a recent report, the Institute of Medicine listed decision support as one of eight core capabilities that a successful electronic health record should incorporate to promote greater safety, quality, and efficiency in health care. At the heart of most modern decision support systems are two key components: an inference engine and a knowledge base. The success of any contemporary clinical decision support system is a function of the quality of the medical knowledge upon which it is built. The importance of a sound knowledge base in health care has been well detailed in the literature. At Intermountain Health Care (IHC), we are currently developing applications to systematically capture clinical knowledge for an enterprise-wide knowledge base. IHC is a nonprofit integrated delivery network with 21 hospitals, several outpatient clinics, more than 450 employed physicians, and a comprehensive insurance plan.

The breadth and depth of the knowledge base content must meet the needs of the various clinician groups across the enterprise corporation. We are maintaining a strategy to create a knowledge management infrastructure that empowers clinical end users and fosters the alignment of their efforts with those of existing authoritative clinical groups. Given this strategy, the need for two fundamental applications emerged. One would be used primarily for the creation of knowledge content, while the other would serve as a gateway application that facilitates open review and collaboration amongst clinicians. This paper focuses on the development of the former, the “knowledge authoring tool” (KAT). Details about the initial prototype of the review and collaboration application can be found elsewhere.

Background

Key Design Considerations

In creating a strategic plan for developing an enterprise-wide knowledge base, we identified three general principles:

1. The knowledge content should be created by clinicians whenever possible, without requiring the continuous assistance of knowledge engineers.
2. The content should be structured enough to be computable, yet readable enough to be easily understandable by clinicians.
3. The content generation process should enable open collaboration between clinicians with different areas of expertise, representing multiple facilities and services across the enterprise.

Collaborative Clinician-Authored Content

One of the key points agreed on was to develop applications that would facilitate internal knowledge content development for use in clinical information systems by IHC clinicians. This approach was favored for several reasons. First, the creation and maintenance of medical knowledge can be a daunting task, both in terms of scope and complexity. With a relatively small number of knowledge engineers and limited financial resources, it would be highly difficult to generate content that would be both salient and meaningful to an

Affiliations of the authors: Department of Medical Informatics, University of Utah (NCH, RAR, GDF); and Medical Informatics, Intermountain Health Care (NCH, RAR, GDF, RLB, TPH, LKR), Salt Lake City, UT.

Supported by grant 2 T15 LM07124 from the National Library of Medicine (NCH).

The authors thank the members of the Clinical Consistency Project and the members of the Clinical Programs. They also thank the authors from the Urban Central and Urban South regions within Intermountain Health Care for their feedback and efforts in creating clinical knowledge base content. They express their gratitude to Dr. Terry Clemmer, Dr. George Thomsen, and the provider order entry development team at Intermountain Health Care for their insights and continuous support and collaboration. They also thank Samson Tu for his feedback regarding their efforts.

Correspondence and reprints: Nathan Hulse, PhD, Intermountain Health Care, S3E, 4646 Lake Park Blvd., Salt Lake City, UT 84120-8212; e-mail: <nathan.hulse@ihc.com>.

Received for publication: 09/17/04; accepted for publication: 03/19/05.

NATHAN C. HULSE, PHDC, ROBERTO A. ROCHA, MD, PHD, GUILHERME DEL FIOL, MD, MS, RICHARD L. BRADSHAW, MS, TIMOTHY P. HANNA, LORRIE K. ROEMER, MS, RN
Design Objectives

Before deciding whether to use existing authoring tools or develop our own, we identified a set of needs, features, and requirements that were considered essential for the type of authoring environment that we desired. For each need (a reflection of the business, personal, or operational problem that must be addressed), a list of high-level features was compiled. The list of features was expanded into a detailed directory of application requirements (software capabilities needed by the user to solve a problem that will achieve an objective). A summary of these requirements and their relationships is included in Table 1. After validating these needs, features, and requirements with IHC Clinical Program leaders, we performed a literature review to better understand the efforts of other groups who have already built authoring tools.

The need for authoring tools for building knowledge bases is by no means new. EzMLM was built at Columbia University to allow clinical domain experts with no knowledge of the Arden syntax to author medical logic modules (MLMs). OPAL was created at Stanford to accelerate the encoding of medical knowledge concepts for use within ONCOCIN, a cancer-specific clinical decision support system (CDSS). GEM Cutter was developed at Yale to facilitate the translation of text-based guidelines into the Guideline Elements Model (GEM) schema. The Protégé Project at Stanford has been developing an application designed to aid users in building ontology-based knowledge resources.

These tools have all proven valuable for a variety of specific purposes, but we thought that our requirements warranted building our own knowledge-authoring environment. Specifically, we wanted Web-based, flexible software with native support for XML (see technology decisions below). Both EzMLM and GEM Cutter operate as client-based software focused on building documents specific to one data model (i.e., MLMs, guidelines). This focused approach to authoring allows for very functional, task-specific software but does not provide the flexibility in authoring documents of many different types. Protégé is a generic tool for building authoring environments that allows end users to define and modify domain-specific ontologies, data entry forms, and resulting knowledge instances. However, XML functionality was only recently added to the system (not available at the time our requirements were drafted) and Protégé was never well supported as an applet.

Technology Decisions

We determined that KAT would be most effective if deployed as a Web application rather than client-based software, facilitating its potential integration with a distributed knowledge management environment. The KAT would be easily accessible to users within the IHC network, via the readily available Web browser. Wide-scale deployment would be much easier and new releases of the software would be instantly available to all users, eliminating the need to upgrade client software or provide support for multiple versions. Finally, a Web-based approach would allow us to leverage existing application servers, further reducing the cost of implementation and rollout to the intended audience.

In selecting a modeling language for the knowledge content, we determined that XML would be the most appropriate option. Among the markup languages, it allowed us to avoid the complexity of SGML and the rigid tag set of HTML. The hierarchical structure of XML documents provides for in-context searching of information within the documents themselves. Like SGML, XML allows the user to separate content from presentation. This is ideal since knowledge base content is likely to be used in a variety of contexts and applications. In addition, the specifications for XML schema allow for richly tagged and object-oriented document definitions with a well-defined underlying structure, including locally defined constraints on data types and data constructs. Many of the current initiatives involving the representation of clinical guidelines have also chosen XML as their representation language, including GLIF, GEM, and others.

In addition to these considerations, the functionality of XSLT (Extensible Stylesheet Language Transformations) was a critical factor in deciding to model the knowledge in XML.
Table 1 ■ Needs, Features, and Requirements for the Knowledge Authoring Tool

<table>
<thead>
<tr>
<th>Needs</th>
<th>Features</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supports simple authoring process</td>
<td>Accessible throughout enterprise</td>
<td>Available at terminals throughout the enterprise, regardless of location or</td>
</tr>
<tr>
<td></td>
<td>Easy to use</td>
<td>computer type</td>
</tr>
<tr>
<td></td>
<td>Straightforward process to submit</td>
<td>Adopt LDAP authentication already used in enterprise applications for both</td>
</tr>
<tr>
<td></td>
<td>content for review</td>
<td>security and user convenience</td>
</tr>
<tr>
<td></td>
<td>Quick access for content to be edited</td>
<td>Users are able to use the tool quickly and effectively with minimal training</td>
</tr>
<tr>
<td>Provides distributed, collaborative</td>
<td>Content stored in a central repository</td>
<td>Tool must be integrated with application designed for open review and feedback</td>
</tr>
<tr>
<td>authoring</td>
<td>Reuse of existing content (supports</td>
<td>Users must be able to manage and organize their authoring tasks within the</td>
</tr>
<tr>
<td></td>
<td>modularity)</td>
<td>tool</td>
</tr>
<tr>
<td></td>
<td>Robust document versioning strategy</td>
<td>Users may quickly search, sort, and filter through content to identify</td>
</tr>
<tr>
<td></td>
<td></td>
<td>material for editing</td>
</tr>
<tr>
<td>Flexible, generic tool</td>
<td>Operates independently of the</td>
<td>Information necessary for authoring, processing, and transforming content</td>
</tr>
<tr>
<td></td>
<td>content that it produces</td>
<td>must be stored outside the application</td>
</tr>
<tr>
<td></td>
<td>Able to author multiple, potentially</td>
<td>Must be able to support the authoring of documents from any valid content</td>
</tr>
<tr>
<td></td>
<td>very different types of content</td>
<td>model</td>
</tr>
<tr>
<td>Creates readable, computable content</td>
<td>Output must be structured and coded</td>
<td>Content must be created from well-defined models</td>
</tr>
<tr>
<td></td>
<td>Application must be able to present</td>
<td>Must support use of coded elements directly selected from a terminology server</td>
</tr>
<tr>
<td></td>
<td>content in an understandable, relevant format</td>
<td>Must be able to transform content from the underlying content model to an</td>
</tr>
<tr>
<td></td>
<td>to user</td>
<td>understandable format for the user</td>
</tr>
</tbody>
</table>

XSLT uses XPath (a W3C recommendation for data access within XML documents) to create meaningful representations of the data in XML. It allows for the transformation of XML data into different output formats, such as HTML, PDF, Java source code, or even modified XML. This is vital throughout the knowledge creation process, as end users need to be able to preview and use the document during the authoring and review phases, often in an environment patterned directly after the clinical application for which the knowledge content is designed. In addition, XSLT facilitates the migration of content from older models to newer ones, as the schema definitions mature over time. Given the set of functionality that XML and its related technologies provide, we thought that it best matched our needs for building the enterprise knowledge base. Others have found the combination of XML and XSLT to be quite powerful in building Web-based applications.

Architecture
We opted to construct our knowledge base patterned after the paradigm of a “document framework.” A document framework consists of a set of loosely coupled servers, databases, and software components that are integrated through standardized interfaces. It addresses the problem of content...
fragments by allowing for the creation and sharing of robust, reusable XML documents for electronic transactions both within and external to an organization. At IHC, a knowledge repository (KR) serves as the central hub for all knowledge base activity within such a framework. The KR includes a database to house the knowledge base content as well as a series of services that allow external applications to interact with the content. Units of knowledge (logical, modular divisions of knowledge content) in the KR comprise individual XML documents that contain both knowledge content and implementation metadata. Each document represents an instance of a specific document category. A category is ultimately defined by an XML schema model that specifies the valid structure of documents specific to that group, be it an order set or an antibiotic monograph.

Knowledge engineers define these document categories and interact with the KR by uploading models, Web forms, and data entry templates specific to each category. They can construct these XML schemas from scratch or import common components from low-level models (directories and libraries) already in existence within the KR. These directories and libraries contain a common set of complex data type definitions, in essence, building blocks that can be shared across data models. They provide a consistent way for common structures (such as coded and linked elements) to be defined and accessed. This minimizes redundancy (and the maintenance associated with it) across the document category definitions in that common elements are defined in one place.

Authors interact within the framework by creating new content or updating existing content and “publishing” these XML documents to the KR. They can also assign different statuses to documents within the KR (“active,” “under review,” “inactive”) that specify the extent to which other individuals and applications can access the content. Clinical applications request data from the KR as needed at runtime. The framework and its main components are illustrated in Figure 1.

The different classes of knowledge content supported by the authoring tool will vary in the levels to which they contain XML markup, encoded elements from reference terminologies, and decision support logic. We envision a continuum of different levels of sophistication among these three axes. Minimally structured documents contain XML tags only to delimit general sections within a document like headings, subheadings, and hyperlinks. They contain neither coded elements nor decision support logic. The next level includes highly structured documents, which contain extensive tagging at the paragraph and sentence level as well as some coded elements. This level of structured content is required in more sophisticated clinical applications, such as the “infobutton,” where more extensive metadata about the content is necessary. Infobuttons are information retrieval tools that automatically generate queries to electronic resources using context-sensitive information and patient data extracted from the electronic medical record (EMR). The highest level of knowledge content sophistication includes “fully structured” richly tagged documents in which there are minimal narrative data. These documents are “executable” in the sense that they are fully encoded as needed and support decision logic. We anticipate using two mechanisms to support decision logic, one in which we link via a pointer to external objects that contain logic and can be activated (e.g., MLMs, rule bases, Java objects, stored procedures) and the other where the decision logic is actually specified using XML, patterned after schemas such as RuleML or XML-based Arden syntax. The differences as to the level of tagging and coding within the different document categories are reflective of the iterative process required to convert free-text information to the richly tagged and coded data necessary for use within computer information system modules. Initiatives such as the HL7 Clinical Document Architecture have recognized this need for iterative development as evidenced by the different phases proposed in the standard. A simple digoxin order within an order set illustrates how knowledge content can fit across the different levels in this continuum (Fig. 2).

**System Description**

**Development**

The initial development of KAT proceeded from early January to July 30, 2003. Two programmers were involved in developing the software during this initial phase: one responsible for the core KR database services and the other for the application code. In addition, four knowledge engineers worked with domain experts to create XML schemas, Web forms, and XSLT transforms for different clinical documents that are being authored in KAT. The tool is currently available for use within IHC to anyone with authorized Internet access.

Six subsequent releases of KAT have been deployed since the initial production go-live in July 2003. Features included in these releases were a cleaner user interface, the ability to transfer document ownership to other users, a medical spell
to author, organize, and document management features to help authors organize their knowledge content. A flexible utilization-monitoring infrastructure has also been implemented to track usage patterns as authors develop their knowledge content using KAT. The monitoring infrastructure logs time-stamped events and their associated parameters, tracking events such as log-in, log-out, navigation, and the various editing features within the XML editing screens. The monitoring infrastructure also allows us to track the duration of each authoring session in KAT. An authoring session is defined as the time interval (in minutes) between log-in and log-out, or if the user exits the Web browser without logging out, the time interval (in minutes) between log-in and the last recorded event.

The KR was built upon an Oracle 9i database, with special support for XML-based functionality. A series of services were written that allow for the storage, retrieval, validation, and searching of documents within the repository. These services also contain business logic for a detailed document versioning strategy. The versioning strategy allows multiple versions of a document to be present since revisions and updates occur on a routine basis as authors modify existing documents. As document updates are “published” using KAT, the new documents are stored in the KR, and older versions are “inactivated” yet remain in storage.

KAT was designed to directly address the needs and requirements detailed during the planning phase of development (Table 1). Among the most notable of these were that the application provided (1) quick, manageable access to content for authors; (2) flexible, generic addition of new content types; (3) support for content reuse; and (4) rapid availability of knowledge content in clinical applications.

The XML editing functionality of KAT was implemented using a plug-in distributed by Altova, known as Authentic®. A plug-in is a software program that extends the capabilities of another program in a specific way; in this instance, it provides XML editing capabilities to Microsoft’s Internet Explorer (versions 5.5 and higher). It operates as an ActiveX control embedded within HTML pages. It allows for a form-based approach to XML authoring in which users can add, remove, and modify form elements that have been defined within a given XML schema. The forms rendered are very similar to typical HTML-based Web forms (as evidenced in Figure 3) but create XML rather than posting individual data elements to a server. Authentic® relies on XML schemas to define the valid document structure and the data types for each field used in the form. The XML editor also requires an XML template for editing and a proprietary Web form that defines the layout of the form in which the document will be edited.

The selection of Authentic® was made for several reasons. First, the forms rendered within the XML editing plug-in are customizable, in terms of both appearance and the extent to which XML elements are visible and available for editing within the form. The plug-in also has a documented application programming interface (API) that is fully scriptable within Web pages. The API allows for a wide variety of scripted events, including data manipulation, extraction, and validation against the reference XML Schemas. Finally, it is freely distributable to all end users. The creation of schemas and form layouts does require the purchase of licensed software, but this need is limited to knowledge engineers, while KAT end users can freely use the plug-in.

**Key Features**

The most important features of KAT were designed to directly address the needs and requirements detailed during the planning phase of development (Table 1). Among the most notable of these were that the application provided (1) quick, manageable access to content for authors; (2) flexible, generic addition of new content types; (3) support for content reuse; and (4) rapid availability of knowledge content in clinical applications.

Within KAT’s main page, we created two separate tables to allow for quick access to documents of interest to the author (Fig. 4). The “Work in Progress” table contains documents that are currently under development but not yet published for open review within the KR. The second table, the “Published Documents” table, contains all the published documents in the repository that were authored by the user. Both tables can be sorted and filtered via quick client-side scripting to allow for ready access to a desired document. Hyperlinks in each row of the two tables allow each document to be edited, updated, previewed, published, or transferred to another author.

To keep the application generic and flexible, we created a “document mapping” table within the KR. The purpose of this table is to provide the data that KAT needs to facilitate XML authoring of a given document category as well as its transformation into an output format (typically HTML and/or PDF). The table contains unique identifiers specific
to each document category available within KAT, including the XML schema, the XSLT document, the data template, and the associated Web form. The KAT loads the data from this table into memory on start-up and uses it to access the necessary associated documents required during the authoring process from the KR. As a result, the application remains independent of the content that it produces. Any additions to the “document mapping” table are instantly available in KAT, without the need to re-deploy the application or recompile its source code.

The KAT was designed to maximize knowledge content reuse during the authoring process. This was a key design feature for two main reasons: first, to reduce the workload on the authors themselves by preventing the need for multiple documents with the same content (and thus unnecessary duplication of effort), and, second, to foster collaboration and knowledge sharing among the authoring groups. Two main features in KAT were included to support content reuse among knowledge authors, namely “save as” and “nesting.”

The “save as” feature in KAT is similar to that available in many text editors today. Specifically, authors can search the KR and create a copy of any published document to use as a basis from which to draft their own content. No permanent link remains between the two documents. “Nesting” allows an author to reuse other knowledge documents in the repository in a different way. Unlike the concept of “linking,” that is typical in HTML documents, nesting provides for the full incorporation of referred content and not navigation to referred content at runtime. For example, an order set may contain references to “nested order sets.” “Nested order sets” are defined as small groups of orders that are reusable enough to be useful in many different order sets. During the editing process, KAT allows authors to search the KR and insert appropriate references to nested content at very specific locations within the XML documents. The master document itself contains only a numerical identifier specific to the nested content, but when the document is previewed or used in a clinical application, all the content that the identifier refers to is immediately imported into the document. Services from the KR and document retrieval functionality afforded by XSLT facilitate this process. Nesting allows for modularity in document design; as nested content is changed and updated, these changes are immediately reflected in all documents that point thereto.

Knowledge base content is moved to production systems by “publishing” the content. While knowledge content is being developed, either as new content or a working copy of existing content for revision, the XML is classified as a work in progress and remains in the author’s “Work in Progress” table. Although an author can transfer ownership of the document to another author, no one else can access, edit, or use that document in its current state. When the author responsible for the content has completed and validated his or her work, he or she can proceed to publish the content to the KR. This process moves the document to production systems and makes it immediately available for all to use in clinical applications. If the author is publishing updates to an existing document, this process inactivates the older version, making only the new copy available for use. Currently, each individual clinical author validates the content before activating it for clinical use. Should IHC opt to regulate the publication process more tightly, simple modifications to KAT would allow central figures to publish content rather than individual authors.

Figure 3. An example of knowledge authoring tool (KAT) as it is used to edit an XML-based order set. The Web forms include text fields, combo boxes, tables, and other objects typical in HTML-based Web forms. Buttons across the top allow the user to save, validate, spell check, and search for nested content.
Since its initial release, the number of supported clinical knowledge document categories in KAT has increased from three to 27, ranging from order sets and clinical care guidelines, to laboratory findings and antibiotic monographs. The distribution of knowledge content produced as well as its introduction into the authoring environment to date are summarized in Tables 2 and 3. Several new categories were recently released, including calculations and calculated orders, designed specifically for use within the Provider Order Entry (POE) application. Calculations and calculated orders are the first instances of decision support logic expressed using XML and authored using KAT. Table 4 contains a list of all supported document types and their intended target applications.

Since KAT’s initial go-live, 4732 unique XML documents have been created. The total number of XML-based documents (including all previous versions of a given document) in the KR is 14,035. Only 1375 of the 4732 unique documents in the KR have not been revised since initial publication. Approximately 70% of the content in the KR has been changed at least once since initial publication. One particular order set has been revised and republished for clinical use 40 times since it was initially stored in the KR. The average number of versions per original document, as stratified by document category, ranges from 1.03 to 14.0. The overall average number of versions per original document in the KR is 2.97.

Since the monitoring infrastructure was incorporated in KAT in early May, we have tracked various usage patterns of the application (Table 5). One hundred ten unique users have logged into KAT, with more than 2000 hits since the release in early May 2004. Of this group, 42 different authors have created and published XML-based knowledge content to the KR. This authoring group includes individuals in different professional disciplines, including ten physicians, nine nurses, nine IT staff/coordinators, and 14 knowledge engineers. Daily usage varies from one to 65 authoring sessions per day, with an average of 23 log-ins per day and a median of 24 log-ins per day. Approximately 40 unique users use KAT each month.

The distribution of all authoring session lengths is shown in Figure 5. The mean session length is approximately 41 minutes; the median is 8.5 minutes. The distribution is skewed at both ends, with the vast majority of all authoring sessions lasting either less than 10 minutes or more than two hours. Altogether, users have spent more than 2270 hours logged into KAT since May 2004.

We have also been able to monitor the extent to which content reuse is taking place during the knowledge content development. Authors have used the “save as” feature 324 times since May 2004. By comparison, 1237 original documents were composed from scratch during the same time. Authors have inserted 2232 references to nested documents during this same period.

XML-based knowledge content is currently being used in three clinical applications at IHC, including POE, an online clinical reference, and the E-Resources Manager application. Order sets, nested order sets, and calculations represented in XML are vital components of our developing POE system. POE end users can select from a filtered list of available order
sets in the knowledge base, as specified by a set of context criteria authored in the XML. These criteria can include items such as the gender, age, and location of the patient. Once this selection has been made, the appropriate XML order set is retrieved and transformed via XSLT into a dynamic HTML-based Web form (Fig. 6). During this process, all nested order sets are imported. Calculations represented in XML are transformed to create JavaScript functions within the Web page that dynamically populate appropriate form fields based off characteristics such as weight and height. Once the appropriate selections and changes have been made to this form (check boxes, pick lists, text boxes) by the clinician, it is posted to a service that converts the posted form data into an XML file containing “instantiated orders” for the patient. Currently, these instance data are used to create a PDF form (via XSL-Formatted Objects) containing the selected orders for the patient. As KAT is developed to more fully support coded orders and as the POE application matures, these data will be used to facilitate electronic order communication. Although still in a pilot phase, the POE project has steadily grown in usage over the past year. More than 2500 unique patients have been treated using POE-based orders since the pilot began in early 2004.

The CPG group has used KAT extensively to recreate their reference library of interdisciplinary patient care standards. Before KAT, their content was stored in an antiquated software-specific format. Viewing the content required client installs of a specialized viewer. The storage format also required the content to be monolithic instead of modular. Due to the burdens associated with these constraints, the CPG team wanted to move the content to a representation language that would allow the content to be used in any of the programs currently under development at IHC. They used KAT to recreate the CPG collection in XML format, modularizing and more richly meta-tagging the data in the process. A Web-based viewer was created for the guideline collection and is currently being piloted at three hospitals within IHC. To date, 1850 hits have been logged during the formal pilot phase of this software (beginning in October 2004). The group’s future plans include using the content for infobutton retrieval in the EMR and nursing work list as well as bedside documentation and patient care plans specific to assigned diagnoses, dynamically pulling the content from the associated CPG documents.

The E-Resources Manager (ERM) program organizes and controls access to internal and external information resources at IHC. It was developed as part of an ongoing effort to provide pertinent, up-to-date clinical information to providers at the point of care. The ERM handles all infobutton requests originating from our clinical information systems. In principle, the ERM is similar to the “infobutton manager” proposed by Cimino et al. Within the ERM, an e-resource profiler uses XML-based profiles specific to each available electronic reference (e.g., PubMed, Micromedex) to characterize important aspects about each reference. Each profile indicates the clinical topics in which an e-resource contains pertinent content.
the context parameters that the e-resource is able to handle, the proprietary query syntax used to express these parameters, and the code systems used to represent the parameter values. These profiles are created and maintained using KAT. This application has received nearly 20,000 hits to date and is scheduled for an enterprise-wide release in the first quarter of 2005.

**Discussion**

KAT’s flexibility in enabling authoring of diverse knowledge content categories is evidenced by the growth in supported content types over the 18 months since its initial launch. We have been able to introduce these document types incrementally in the authoring environment since the authoring tool is independent of the document framework and its models and forms. Once the necessary supporting documents required for each document type were completed (XML Schema, Web form, XML template, and XSLT), they were loaded to the KR, enabling immediate content authoring. This allows knowledge engineers to create and update knowledge content models without requesting code changes from the development team and without the difficulty of being tied to a fixed release schedule. The flexible, open authoring environment that KAT provides makes it a viable solution for groups seeking to capture a wide variety of knowledge content in a structured format.

Demonstrating KAT to various groups within IHC has increased interest in using it. We are encouraged by the amount of content that has been authored as well as the number of contributing authors. The variety of specialties represented among the knowledge base authors suggests that the application is acceptable to individuals of different disciplines. We are currently exploring the acceptability of KAT within our authoring group, studying the motivations ‘for’ and ‘against’ using the tool.

Users have found it important to quickly edit and republish clinical knowledge content to the KR. The large number of authoring sessions lasting less than 10 minutes is reflective of this trend. As evidenced by the quarterly authoring data shown in Tables 2 and 3, many versions of a given document can be created within a short amount of time. This is especially valuable for content tied to clinical applications like POE (such as order sets) due to the highly dynamic nature of clinical treatments. The respective authors receive feedback about their knowledge content through both informal (conversations and interaction with other clinical users) and formal (e-mail, systematic review) channels. The authoring data also seem to suggest that the target application for which clinical knowledge base content is intended is a good predictor of how frequently a document is revised. KR content that is currently being used extensively in clinical applications (such as the order sets, nested order sets, and e-resources

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Total XML Knowledge Base Content Distribution (Original Content and All Subsequent Revisions Thereof)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interdisciplinary nested order set</td>
<td>42</td>
</tr>
<tr>
<td>Nested order set</td>
<td>93</td>
</tr>
<tr>
<td>Order set</td>
<td>17</td>
</tr>
<tr>
<td>Assessment</td>
<td>154</td>
</tr>
<tr>
<td>Index page</td>
<td>5</td>
</tr>
<tr>
<td>Lab diagnostic findings</td>
<td>1</td>
</tr>
<tr>
<td>Risk factors causes</td>
<td>1</td>
</tr>
<tr>
<td>Symptoms</td>
<td>4</td>
</tr>
<tr>
<td>Antibiotic monograph</td>
<td>12</td>
</tr>
<tr>
<td>Care plan module</td>
<td>344</td>
</tr>
<tr>
<td>Miscellaneous footnote</td>
<td>8</td>
</tr>
<tr>
<td>Literature citation</td>
<td>1354</td>
</tr>
<tr>
<td>Procedure</td>
<td>111</td>
</tr>
<tr>
<td>Tables and tools</td>
<td>479</td>
</tr>
<tr>
<td>Calculated order</td>
<td>40</td>
</tr>
<tr>
<td>Calculation</td>
<td>23</td>
</tr>
<tr>
<td>Clinician reference</td>
<td>16</td>
</tr>
<tr>
<td>E-resources page</td>
<td>2</td>
</tr>
<tr>
<td>Glossary</td>
<td>1</td>
</tr>
<tr>
<td>Index</td>
<td>156</td>
</tr>
<tr>
<td>Interdisciplinary protocol</td>
<td>768</td>
</tr>
<tr>
<td>Patient education</td>
<td>67</td>
</tr>
<tr>
<td>Practice guideline</td>
<td>145</td>
</tr>
<tr>
<td>Problem</td>
<td>999</td>
</tr>
<tr>
<td>Risk for problems</td>
<td>439</td>
</tr>
<tr>
<td>WorkMed examination</td>
<td>20</td>
</tr>
<tr>
<td>E-resources profile</td>
<td>20</td>
</tr>
<tr>
<td>Quarterly content total</td>
<td>152</td>
</tr>
</tbody>
</table>

| Total no. of categories in KR | 3 | 8 | 11 | 14 | 26 | 27 | 27 |

KR = knowledge repository.
material) tends to change more frequently and consistently than the content that is being used as clinical reference material. These document categories are updated more frequently during the initial development phase and then the updates taper off in the months that follow. While some clinical authors who use the knowledge content within clinical applications argue that the ability to quickly change and repost clinical knowledge to the KR is vital to their work flow, others contend that a tighter validation process needs to be implemented to prevent errant data from entering the KR.

The authoring session data regarding KAT suggest that knowledge base development requires a sizable investment in time and resources. The fact that authoring session length is skewed at both ends indicates that most of the authoring sessions involving KAT are either very quick (typical of quick modifications to existing knowledge base content) or somewhat lengthy (as is typical of the generation of novel content). It is important to note that the 2270 hours of KAT session time does not account for the time spent by knowledge engineers and domain experts working together to develop adequate models for knowledge content nor the time spent in building and improving the authoring environment and KR infrastructure. We anticipate that a concerted, ongoing effort by multiple authors will be necessary to expand and maintain the knowledge content in the KR.

The statistics in Table 5 confirm that the authors are using features designed to support content reuse. It is interesting to note that of all original content produced since the monitoring infrastructure went live, 21% was created using existing content as a template. Authors also inserted more than 2200 nested element references in various documents. One particular nested order set is referenced by 18 different order sets, validating the usefulness of content reuse and knowledge modularity.

Some authors hesitate to use nested content since they are essentially referencing content that they may or may not have control over. As new updates occur to the nested content, they are immediately reflected within their own content. This can be disconcerting to some authors and reassuring to others. Authors have the ability to "subscribe" to any document in the repository, so that as soon as updates occur, they are notified via e-mail and prompted to examine the updated material. It is possible that some of the authors are not familiar with the concept of nesting and how it can be leveraged within the context of knowledge base content. These
authors may be creating redundant content or documents that are not modular.

Similar to the knowledge base maintenance approach described by Giuse et al. and Geissbuhler and Miller, we favor knowledge creation and maintenance by distributed practicing clinical experts operating under well-defined work flows. Our approach represents a partial strategy shift in knowledge development from “top down” to “bottom up.” Rather than expecting physicians to rely solely on centrally or externally developed content, we also invite them to participate in the authoring and review of the knowledge content. This approach allows for a broader and more clinically involved authoring base as well as a sense of ownership among clinical knowledge base authors. However, this approach raises issues regarding the flow of knowledge content into the knowledge base. If “anyone” is allowed to author and publish clinical knowledge base content, it is possible that multiple versions of the same knowledge content will coexist. For example, we currently have multiple order sets for “community-acquired pneumonia,” each specific to local practice at different hospitals. It is also possible for incomplete or narrowly focused discipline-specific content to become part of the enterprise knowledge base. If left unconsolidated and unreviewed by individuals of various clinical disciplines (e.g., registered nurses, registered pharmacists, physicians), such content in the knowledge base could hinder the standardization of care that information systems have the potential to foster. These issues underscore the need to incorporate both systematic and ad hoc review and validation routines in knowledge development and maintenance cycles.

Our “document-centric” approach to knowledge acquisition is somewhat of a departure from ontology-based systems commonly used in medical informatics. While both approaches define objects and concepts within a given domain, ontologies formally define relationships between these objects, whereas our document framework does not. We believe that a document-centric approach to knowledge acquisition and representation is a more familiar paradigm for typical clinical domain experts, allowing them to contribute more autonomously to the development and maintenance of our clinical knowledge base. Very similar document-centric methodologies, also using XML-based frameworks, have been extensively reported by medical informatics researchers from Justus-Liebig-University. On the other hand, the lack of a formal method of establishing semantic and/or ontological relations across entities has forced the applications
that use KR content to define any necessary relationships using other mechanisms. The latest version of the KR enables formal associations between repository objects, in which XML documents and fragments of documents can be considered repository objects. KAT has not yet been modified to make use of this new KR structure.

We are currently working to integrate KAT more closely with the Healthcare Data Dictionary in use at IHC.\(^5\) In doing so, we hope to allow authors to develop knowledge base content that is more fully encoded and can be tied to existing decision support systems. We are also investigating how suitable our environment will be for building documents that contain conditional logic and work flow information, as is commonly found in guidelines and protocols. Context awareness will be expanded in future versions of knowledge base content, allowing specific sections of documents (rather than the document in its entirety) to be used in a context-sensitive fashion at runtime, especially with regards to the age group, gender, and facility dimensions for which the content was designed. We will also be working to improve the authoring graphical user interface, so that progress in these areas does not come at the expense of poor usability.

We are currently exploring the rationale for knowledge content change and the types of changes present in the content as it evolves. A feedback loop is being developed to inform authors about how frequently and to what extent their content is being used in clinical applications. We plan to explore knowledge content evolution in our environment and identify the channels (informal, formal, or direct feedback) that serve as a direct impetus for change.

We have encountered several limitations in developing KAT and proceeding with knowledge base development in a document framework. Collaborative authoring within development teams using KAT has proven to be difficult since currently only one author can work on a given document at a time. Administrators have voiced their desire to see role-based access control become more important in KAT, allowing managers to designate specifically who can author, publish, or review clinical content within specific projects. Schema maintenance and content synchronization has proved to be a very labor-intensive aspect of our approach. As document category schemas are updated within the KR, subsequent changes may be required for all knowledge content of that category to ensure that it conforms to the changes reflected in the new data model. This requires a batch process that analyzes and updates each knowledge document on an individual basis. Performing content validation testing and simulation using KR content within clinical application modules is not yet possible from within KAT. Users are forced to create and publish content using KAT and then switch to the targeted application to test it. Another limitation lies in KAT’s inability to create process-oriented knowledge using a visual, graphical-oriented tool (similar to Visio, as opposed to the form-based approach currently used). KAT is somewhat limited by the fact that it is browser based; it uses a plug-in and is constrained by all the limitations associated with operating inside a Web-based shell as opposed to a rich client application.

In developing KAT and building a knowledge base upon a document framework paradigm, we have learned three key lessons. First, open and distributed authoring across a wide base of contributing authors allows for rapid creation and refinement of an enterprise-wide knowledge base. Second, software designed to support this type of knowledge development must be flexible enough to accommodate the diversity of both the data models that it uses as templates and the end users that drive the content-creation process. Finally, principles of knowledge reuse must be tightly adhered to, both at the level of defining data models as well as creating knowledge content itself.

Conclusion

We developed KAT, a flexible software application that allows clinicians to create and store XML-based knowledge content to our KR. The resulting knowledge content is structured enough to be computable, yet readable and understandable for the end users. The authoring environment supports content reuse both in terms of nesting modular content and authoring via templates. Present usage patterns indicate both new content generation and extensive revisions of existing documents. By empowering typical clinicians to participate in the capture of medical knowledge, we hope to create a knowledge base whose content is constantly being expanded and maintained.

References

6. Wilkinson SG. Knowledge repository online: an electronic knowledge development and maintenance process. Salt Lake City: Department of Medical Informatics, University of Utah; 2003. p. 156.