A Framework for a Distributed, Hybrid, Multiple-Ontology Clinical-Guideline Library and Automated Guideline-Support Tools

Yuval Shahar, Ohad Young, Erez Shalom, Maya Galperin, Alon Mayaffit, Robert Moskovitch, and Alon Hessing

Medical Informatics Research Center
Department of Information Systems Engineering
Ben Gurion University, Beer Sheva, Israel 84105
http://medinfo.ise.bgu.ac.il/medlab/
{yshahar, ohadyn, erezsh, mayafit, robertmo, hessing, gmaya}@bgumail.bgu.ac.il

Abstract. Clinical Guidelines are a major tool in improving the quality of medical care. However, most guidelines are in free text, not in a formal, executable format, and are not easily accessible to clinicians at the point of care. We introduce a Web-based, modular, distributed architecture, the *Digital Electronic Guideline Library* (DeGeL), which facilitates gradual conversion of clinical guidelines from text to a formal representation in chosen target guideline ontology. The architecture supports guideline classification, semantic markup, context-sensitive search, browsing, run-time application, and retrospective quality assessment. The DeGeL *hybrid meta-ontology* includes elements common to all guideline ontologies, such as semantic classification and domain knowledge; it also includes four content-representation formats: free text, semi-structured text, semi-formal representation, and a formal representation. These formats support increasingly sophisticated computational tasks. The DeGeL tools for support of guideline-based care operate, at some level, on all guideline ontologies. We have demonstrated the feasibility of the architecture and the tools for several guideline ontologies, including Asbru and GEM.

Keywords: Clinical guidelines, Digital libraries, Knowledge representation, Ontologies, markup

1 Introduction: Clinical Guidelines

Clinical guidelines (or **Care Plans**) are a powerful method for standardization and uniform improvement of the quality of medical care. According the Institute of Medicine's (IOM) definition, *clinical guidelines* are "systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances" [1]. For our purposes, clinical guidelines are a set of plans, at varying levels of abstraction and detail, for screening, diagnosis or management, over limited or extended periods of time, of patients who have a particular clinical problem, need, or condition (e.g., fever of unknown origin; or therapy of insulin-dependent diabetes). We are focusing here mainly on guidelines for management of patients over extended periods, namely, on management of chronic patients. **Clinical protocols** are typically highly detailed guidelines, often used in areas such as oncology and experimental clinical trials. **Reminders** and alerts can be viewed as "mini guidelines", useful mostly for representing a single rule that needs to be applied whenever the patient's record is accessed, as opposed to representation of a long-term plan [2].

Extensive evidence exists that conforming to state-of-the-art guidelines improves the quality of medical care, and sometimes even the survival of patients, a fact that had been rigorously demonstrated [3-5], while reducing its escalating costs. Clinical guidelines and alerts are most useful at the point of care (typically, when the care provider has access to the patient's record), in particular, when a specific care plan is prescribed by the care provider.

Several of the major tasks involved in guideline-based care, which would benefit from automated support, include knowledge-modeling tasks, such as *specification* and *maintenance* of clinical guidelines, and runtime tasks (which often need also patient data), such as runtime *application* of guidelines and retrospective *assessment of the quality* of the application of the guidelines. Other related tasks are *search*, *retrieval*, *browsing*, and *visualization* of relevant guidelines, examination of the *eligibility* of one or more patients for a given guideline, or the *applicability* of one or more guidelines to a given patient. Thus, computer-based techniques could greatly assist in performing the tasks involved in guideline-based care.

Most clinical guidelines, however, are text-based and not easily accessible to care providers, who need to match them to their patients and to apply them at the point of care. Similar considerations apply to the task of assessing retrospectively the quality of clinical-guideline application. Some improvement had been made by publishing several of the clinical guidelines in an electronic format, such as HTML or PDF files. Yet, care-providers, overloaded with information, rarely have the time, nor the computational means, to assist them in utilizing the valuable knowledge, encoded in the guidelines, during treatment. Therefore, there is a pressing need to facilitate *automated* guideline specification, dissemination, application, and quality assessment. Since analyzing unstructured text-based guidelines is not feasible, due to limitations of current technologies, such automation requires formal representations of clinical guidelines that can be parsed and executed by machines. We call such representations *machine comprehensible*. (The term "*comprehension*" is used here in a strictly formal sense, not a cognitive one.)

Thus, the need to automate guideline-based care implies, in practice, *a need to convert the mass of free-text guidelines into machine-comprehensible representations.* The architecture we propose here is greatly motivated by this need.

1.1 Automated Support to Clinical Guideline-Based Care

During the past 20 years, there have been several efforts to support complex guideline-based care over time in automated fashion.

Several simplified approaches to the task of supporting guideline-based care encode guidelines as elementary state-transition tables or as situation-action rules dependent on the electronic medical record, as was attempted using the Arden syntax [6]. An established (ASTM) medical-knowledge representation standard, the *Arden Syntax* [7], represents medical knowledge as independent units called *Medical Logical Modules* (*MLMs*), and separates the general medical logic (encoded in the Arden syntax) from the institution-specific component (encoded in the query language and terms of the local database).However, rule-based approaches typically do not include an intuitive representation of the guideline's clinical logic, have no semantics for the different types of clinical knowledge represented, lack the ability to easily represent and reuse guidelines and guideline components as well as higher, meta-level problem-solving knowledge, cannot represent intended ambiguity (e.g., when there are several options and several pro and con considerations, but no single action is, or should be, clearly prescribed) [2], and do not support application of guidelines over extended periods of time, as is necessary to support the care of chronic patients. On the other hand, as Peleg et al. [2] also point out, such approaches do have the advantage of simplicity when only a single alert or reminder is called for, and the heavier machinery of higher-level languages is uncalled for and might even be disruptive. Thus, they might be viewed as complementary to complex guideline representations. In the current research, we have indeed focused on the case of chronic patients whose long-term care requires complex guidelines.

Examples for systems focusing on such longitudinal care include ONCOCIN [8], T-HELPER [9], DILEMMA [10], EON [11], Asgaard [12], PRO*forma* [13], the guideline interchange format (GLIF) [14], the European PRESTIGE project, the British Prodigy project [15], and, on the commercial side, Epic Systems Corp.'s Active-Guidelines model [16]. A useful guideline ontology, GEM [17], enables structuring of a text document containing a clinical guideline as an *extensible markup language* (*XML*) document, using a well-defined XML structure, although it does not include a formal computational model. The feasibility of creating an implementation framework for GEM-encoded guidelines has been demonstrated [18], although it does not seem to support extended care over significant time periods, due to the lack of persistent-memory mechanisms, interaction with an electronic patient record, and complex control structures.

Most of the approaches mentioned above can be described as being *prescriptive* in nature, specifying *what* actions need to be performed and *how*. However, several systems, such as Miller's Attending system [19, 20], have used a *critiquing* approach, in which the physician suggests a specific therapy plan and gets feedback from the program.

An excellent comparative review of most current approaches to the support of complex guideline-based care is provided by Peleg et al.'s [21] comprehensive paper.

1.2 The Guideline Conversion Problem

The existence of automated architectures for guideline representation makes the question "*How will the large mass of free-text guidelines be converted to a formal machine-readable language*?" be a most pertinent one.

The core of the problem is that expert physicians cannot (and need not) program in guideline-specification languages, while programmers and knowledge engineers do not understand the clinical semantics of the guidelines. In addition, some of the guideline's knowledge is of *implicit* nature, clear only to the expert physician authoring the guideline; this knowledge must become *explicit* during the conversion process. Thus, converting guidelines into machine-comprehensible formats must capitalize on the relative strengths of both expert physicians and knowledge engineers. The conversion process must also support and facilitate *collaboration* amongst these two very different types of users, and the *iteration* inherent in such a process.

Another aspect we need to consider is that machine-executable representations are crucial for providing computerized assistance, such as automated *execution* of the guideline, while for some other tasks, such as *search and retrieval* of relevant guidelines, text-based representations are of a significant importance (and indeed, might be more useful).

Finally, we need to keep in mind that the procedural knowledge encoded in the guideline, such as laboratory tests or clinical procedures, should be truly sharable and generalizable across multiple sites. Thus, the conversion process must support embedding in the guideline's representation terms from *standardized medical vocabularies,* which are well-understood everywhere.

2 The Hybrid Guideline-Representation Model

To gradually convert clinical guidelines to machine-comprehensible representations, we have developed a *hybrid* (i.e., one that has multiple representation formats co-existing simultaneously), multifaceted representation, an accompanying distributed architecture, and a set of Web-based tools. The specification tools incrementally and in iterative fashion transform a set of clinical guidelines through several intermediate, semi-structured phases, eventually arriving at a fully formal, machine comprehensible representation of the guideline.

Our guiding principle in the research project to be described here is that expert physicians (if possible, throughout the world) should be transforming free-text guidelines into intermediate, semantically meaningful representations, while knowledge engineers should be converting these intermediate representations into a formal, executable representation.

2.1 The gradual conversion process

Underlying the various modules and tools we will be describing further on, is the guiding principle mentioned above: *Expert physicians* use the tools to classify the guidelines along multiple semantic axes, and to semantically *markup* (i.e., label portions of the text by the semantic labels of the target ontology) existing text-based guidelines, thus creating a *semi-structured* format (which is still text-based). The expert physicians might even further structure the guideline, possibly with a knowledge-engineer's assistance, into a *semi-formal* structure, which includes ontology-specific control-flow knowledge. *Knowledge engineers* convert the marked-up text, or the semi-formal structure, into a *formal*, *fully structured*, machine-comprehensible representation of the target ontology, using an ontology-dedicated tool (Figure 1). All of the hybrid guideline-representation formats co-exist and are organized in the DeGeL library within a unified structure, the *hybrid representation.* Part of the hybrid representation, shared by all hybrid guideline ontologies, is the *hybrid meta-ontology* (Section 3).

Note that different parts of the same guideline might exist at different levels of specification (e.g., eligibility conditions might include also executable expressions, thus supporting automated eligibility determination, although the guideline's procedural aspect is still only semi-structured or in a semi-formal format). In addition, all specification levels are optional. Finally, if needed, new representation levels can be added.

Since expert physicians can rarely program (our experience over the past 3 years also indicates that they do not find control structures, such as sequential or parallel subtask execution, very intuitive), while knowledge engineers rarely understand all the hidden subtleties underlying the clinical guideline, it is necessary for both types of experts to interact at some point in the guideline-specification process. This interaction usually happens when the domain experts creates the semi-formal representation level, which includes specification of the control structures, assisted by the knowledge engineer. Thus, our hybrid-specification process, which merges several grades of increasing formalization, intertwines the expertise of both types of experts to gracefully convert clinical guidelines into a machine-executable format. The conversion process is performed gradually using the following representation formats:

1. *Semi-structured Text* – snippets of text assigned to top-level target-ontology knowledge-*roles, such* as the eligibility criteria for applying the guideline, or the guideline's objectives. These roles would have different names in different guideline ontologies, of course.
2. *Semi-formal representation* – further specification of the structured text, adding more explicit procedural control structures, performed jointly by the knowledge engineer and expert physician, such as specification in explicit fashion of whether the actions are to be carried out sequentially or concurrently.
3. *Formal representation* – final specification performed by the knowledge engineer, resulting with the guideline converted to a machine-comprehensible format, executable by an appropriate runtime execution module specific to the chosen target guideline ontology. Thus, the output of our authoring tool(s) is a *hybrid representation* of a guideline which contains, for each guideline, or even for different sections (knowledge roles) within the same guideline, one or more of the above three formats.

These three current levels of hybrid structuring (or four, including the original free text) are in principle possible within all guideline-representation languages. For example, they were easily implemented within the context of the Asbru language, which happens to be the default guideline ontology in our architecture (see Section 3.2).

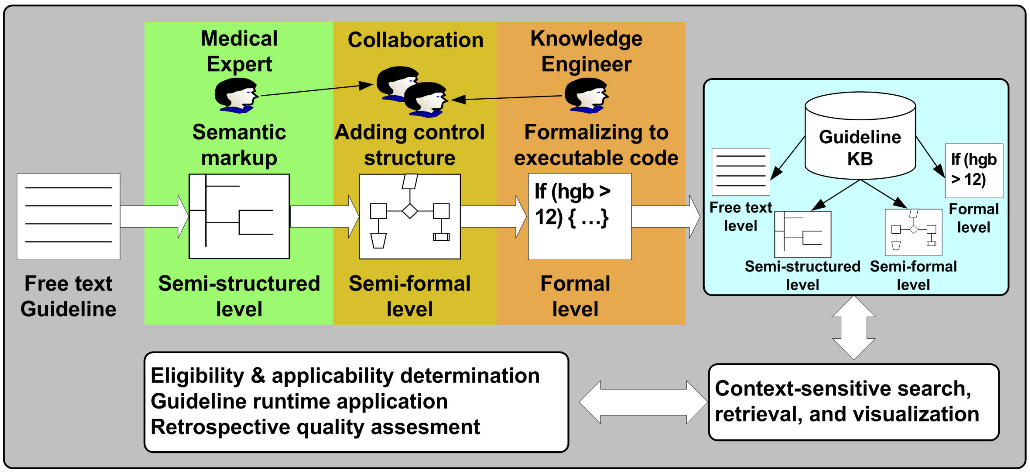


Figure 1: The incremental conversion process in the DeGeL architecture. Input free-text guidelines are loaded into a *markup* editor; expert physicians index and markup (structure) portions of the guidelines by semantic labels from a chosen target ontology, creating semi-structured and, in collaboration with a knowledge engineer, semi-formal guideline representations. Knowledge engineers use an ontology-specific tool to add executable expressions in the formal syntax of that ontology

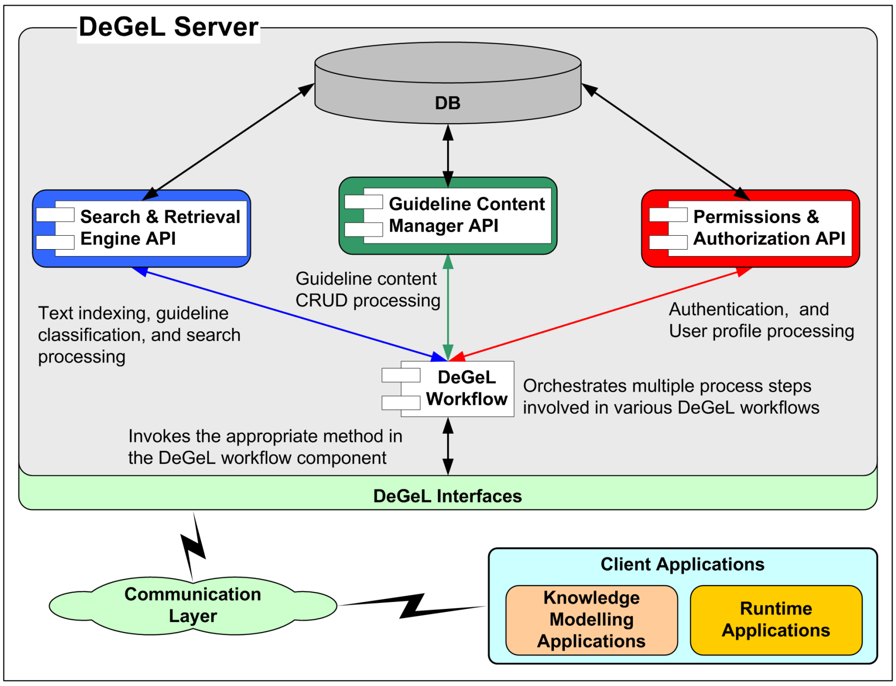

DeGeL Server
DB
Search & Retrieval Engine API
Guideline Content Manager API
Permissions & Authorization API
Guideline content CRUD processing
Text indexing, guideline classification, and search processing
Authentication, and User profile processing
DeGeL Workflow
Orchestrates multiple process steps involved in various DeGeL workflows
Invokes the appropriate method in the DeGeL workflow component
DeGeL Interfaces
Communication Layer
Client Applications
Knowledge Modelling Applications
Runtime Applications

Figure 2: The conceptual architecture of a typical DeGeL server. There are three main components, (1) a *permissions and authorizations* manager component, responsible for generating user-profiles and controlling user access to DeGeL's guideline repository, (2) a *guideline content manager*, responsible for performing *Create, Retrieve, Update and Delete* (*CRUD*) operations on all knowledge entities (e.g., guidelines) stored in DeGeL's repository, and (3) a *search & retrieval engine,* responsible for performing text indexing and store semantic classification of guidelines as well as handling search queries processing. The *DeGeL Workflow* component synchronizes all three components during operations that require use of one or more components. The DeGeL architecture has a single conceptual interface that can be accessed through multiple communication methods (e.g., web services and remote procedure calls), the interfaces to which are part of the *Listener* component.

3 The DeGeL Architecture

We have developed a distributed, web-based architecture, the **Digital electronic Guideline Library** (**DeGeL**), which supports all of the design time and runtime tasks involved in guideline-based care. The DeGeL framework's guideline knowledge-base and various task-specific tools (Figure 2) were designed to handle all of the hybrid guideline representation levels.

The design for DeGeL architecture is not an arbitrary one. It incorporates insights from previous research projects in which the first author was involved, such as EON [11] and Asgaard [12]. It is also important to mention that several medical centers are collaborating in the project, and the design includes insights gleaned from their clinicians. These institutions include the *Stanford University Hospital* (*SUH*), California, The *Palo Alto Medical Foundation* (*PAMF*), California, the *Veterans Affairs* (*VA*) *Palo Alto Health Care System* (*PAHCS*), California, the Soroka Medical Center of Ben-Gurion University in Beer Sheva, Israel, and a non-for-profit organization, the *Clinical Information Center* (*CIC*) in Tel Aviv, Israel, which provides severely ill patients with information regarding their treatment options.

Several of these organizations are have been participating over the past two to three years in evaluations of the DEGEL system and all of its various tools (see Section 7).

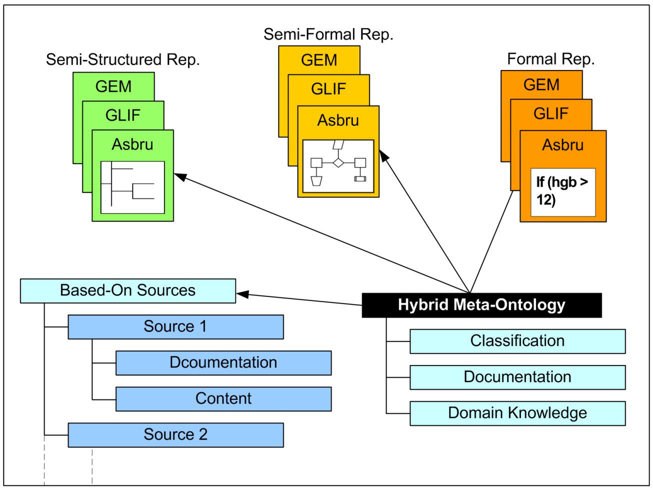


Figure 3: DeGeL's hybrid meta-ontology. The meta-ontology includes pointers to one or more source ontologies of the hybrid guideline, and pointers to its semi-structured, semi-formal, and formal versions of its target ontology (e.g., hybrid Asbru), and several knowledge-roles, independent of the target ontology, that characterize the document (e.g., domain knowledge, semantic indices).

3.1 The Hybrid Meta-Ontology

To support the specification of a guideline in one or more different guideline specification languages, the DeGeL architecture includes a *hybrid* guideline *meta-ontology* (Figure 3) (*meta* is used here in the sense of "*above*").

The meta-ontology is composed of two components:

(1) A *documentation ontology*, which specifies knowledge roles common to all target guideline ontologies, and defines the ontologies of the sources of the guidelines and of the marked-up guidelines (see below).

(2) A *specification meta ontology* for describing a new target ontology, in order to enable adding it into the DeGeL (meta) knowledge base. Thus, we provide an XML schema that describes, for designers of existing or new guideline ontologies, how to generate XML documents that conform to the DeGeL expected structure. These documents are instances of the specification meta ontology and describe particular target ontologies such as GLIF or Asbru.

The documentary component of the hybrid-meta ontology includes several knowledge-roles, such as *documentation*, common to all guideline ontologies. It distinguishes *source guidelines*, which are free-text guidelines uploaded to DeGeL, from *hybrid guidelines,* which are the output of the gradual hybrid conversion process.

Uploading a guideline into the DeGeL library (e.g., a document published by a professional society) creates a *source guideline*. A source guideline can be named, searched, and retrieved, and is annotated using the dedicated *source-guideline ontology*, which documents the source-guideline's details (e.g., authors, date). However, a source guideline cannot be indexed or applied to a patient.

A *hybrid guideline* is a more complex structure, which can be indexed, retrieved, modified, and applied. A hybrid guideline includes one or more source guidelines, several knowledge roles from the hybrid meta-ontology that are common to all target ontologies (e.g., *documentation*, *domain knowledge*), and the semi-structured, semi-formal, and fully-structured (machine-comprehensible) representations of the guideline using the selected target ontology.

The *semi-structured* representation of a hybrid guideline will typically exist for all guideline ontologies. This representation can be processed by all DeGeL tools, without adding any ontology-specific extensions; it corresponds roughly to the top level and intermediate concepts of the target ontology.

The *semi-formal* and *formal* representation levels typically need ontology-specific tools for creation and processing. These levels typically include ontology-specific control structures and low-level expressions regarding patient data or care-provider actions.

For example, marked-up semi-structured temporal queries to the patient record can be semi-formalized by the expert physician, and/or fully formalized by the knowledge engineer.. Indeed, we had created such tools for the Asbru ontology and for the language of our patient-data mediator (see Section 3.4).

3.2 The DeGeL Default Guideline Ontology: The Asbru Language

In the **Asgaard** project [12], the first author and his colleagues had designed an expressive guideline-representation language, **Asbru**. An Asbru specification includes *conditions* (e.g., the *filter condition*, which represents obligatory eligibility criteria, the *complete condition*, which halts the guideline execution when some predefined temporal pattern is true, and allows normal continuation, and the *abort condition*, which aborts the guideline execution when some predefined temporal pattern is true, and returns control to the next higher procedural level); control structures for the guideline's *body* (e.g., *sequential*, *concurrent*, and *repeating* combinations of actions or sub-guidelines), *preferences* (utility functions), expected *effects*, and process and outcome *intentions*. Indeed, a feature initially unique to Asbru is the use of explicit *intentions*, represented as temporal-constraint patterns at multiple levels of abstraction. Using explicit intentions supports intelligent retrospective quality assessment, by representing the guideline designer's intermediate and overall goals regarding care-provider actions and patient outcomes.

We have created a hybrid-Asbru ontology, whose semi-structured level is used by the expert physicians in the first phase of the conversion process. In the *Asbru semi-structured hybrid ontology*, we have included key entities such as *conditions*, *intentions, effects, preferences* and *plan body*, but left out low-level knowledge roles that require deeper understanding of Asbru semantics.

We have also implemented several Asbru-specific tools for supporting conversion into the semi-formal and formal representation levels. (An example of the three representation levels, in the case of an abort condition of a guideline for hypertension therapy, is shown in Table *1* of the Appendix).

We will use the Asbru ontology in this paper for demonstration of the current architecture's various aspects. It is the default ontology we are currently using for the guideline specification process. Note that the services supplied by the DeGeL framework are the same for all hybrid guideline ontologies with respect to the meta-ontology and the semi-structured text representation level. For example, we have also marked up guidelines using the GEM ontology, as well as by guideline ontology, specific to the needs of the CIC organization. Furthermore, the overall guideline-specification workflow is essentially independent of the particular target guideline ontology.

3.3 The Guideline Repository Structure

The web-based architecture of the DeGeL framework implies that it should concurrently support multiple users. Therefore, the database chosen for the implementation of the guideline repository, Microsoft's SQL Server 2000, is a commercial-proven, high-performance, relational database.

The guideline-repository database includes several relational schemas supporting the data and knowledge requirements expected from a central guideline-repository system. For example: (1) storage of the content of the different representation levels generated during the gradual conversion process; (2) the user profiles used by DeGeL's permission and authorization system; (3) the text-indexing tables of each guideline, which are used by DeGeL's search engine; and (4) the semantic axes according to which each guideline is classified.

A problem we encountered due to the use of relational databases was how to store and retrieve efficiently hierarchical data, for example the structure of DeGeL's semantic indices or the hybrid-meta ontology. An intuitive possibility for storing hierarchical data is by using XML files. We have explored the potential of using native XML databases by conducting a feasibility test in which a simpler version of DeGeL was developed using a native XML database. The result of this test was that native XML databases, which are typically geared for search in a large, mostly static database of text documents, are currently not suitable for DeGeL needs such as (1) frequent content updates, and (2) processing individual XML elements. Therefore, we added additional relational schemas for storing the hierarchical data and developed the corresponding software components for handling the transformation of relational data to XML and vice versa (Appendix A).

For example, in order to upload a new guideline-ontology specification to DeGeL, an XML document defining the new ontology is sent to the DeGeL server for processing prior to storing it in the knowledge base. The processing of the document includes checking whether it adheres to the "ontology-specification" XML schema. Similar schemas exist for other knowledge types in the DeGeL framework, such as classification axis. The use of XML inputs to DeGeL enables performing batch operations that involve complex data structures.

4 The Overall Architecture and the Hybrid Design-Time and Runtime Tools

The DeGeL tools could be considered as belonging to two types. Several tools are used mostly to specify and retrieve guidelines, irrespective of a particular patient. Other tools are used mostly at runtime and require automated or manual access to patient data.

To link the runtime tools to the patient data, we have developed separately an accompanying architecture, **Idan** [22], which enables access to any heterogeneous clinical database for purposes of query of both raw clinical data and its abstractions. To support the need for guideline sharing among institutions, and reuse in different environments [23], during the guideline conversion process each free-text concept (e.g., "potassium level") is replaced by a term from a standard medical vocabulary. Therefore, we have developed and are actively using a centralized vocabulary server for supporting guideline specification and for other uses, such as creation of a mapping between the standardized terms and each local clinical-database vocabulary. The vocabulary server currently includes the standard terminologies *International Classification of Diseases* (*ICD-9-CM*) and *Standard Nomenclature of Medicine* (*SNOMED*) for diagnosis codes, *Current Procedural Terminology* (*CPT*) for procedure codes, *Logical Observation Identifiers, Names, and Codes* (*LOINC*) for observations and laboratory tests, and the *National Drug File* (*NDF*) in the case of medications. The vocabulary server also includes a Web-based search and retrieval engine for using these terminologies during guidelines specification (i.e., at design time). At runtime, the mapping created at each local site (i.e., clinical database) enables the DeGeL guideline-support tools to query patient records. The guideline's domain-specific knowledge, required to create guideline-specific abstractions of patient data (e.g., "moderate anemia" in a particular context), is stored in a dedicated knowledge base indexed by the context of each guideline. Thus, each guideline includes pointers to both the raw data terms (in standardized vocabularies) and the abstract concepts (in the abstraction knowledge base) used within that guideline.

In addition, the runtime tools often use the **KNAVE-II** [24] intelligent visualization and exploration client, which uses Idan's computational capabilities, and the fact that the guidelines relevant terms had been mapped to the patient's record, to visually display and explore the patient's raw data or derived concepts. Preliminary assessments of KNAVE-II and its underlying IDAN framework by our clinical colleagues at the PAHCS are highly encouraging, demonstrating significant decrease in time and increase of accuracy to answer queries typical of oncology protocols, when compared to the use of paper charts or Excel [25].

Figure 4 presents an overall view of the DeGeL architecture.

We will now examine the various components of the DeGeL architecture, and how they serve the underlying hybrid, multiple-ontology framework. All of the tools were designed to support the various formats implied by a hybrid representation.

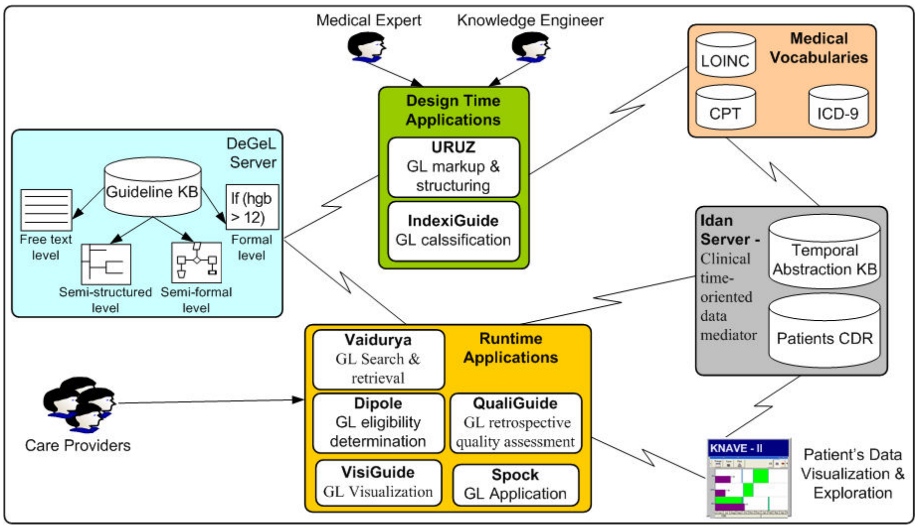
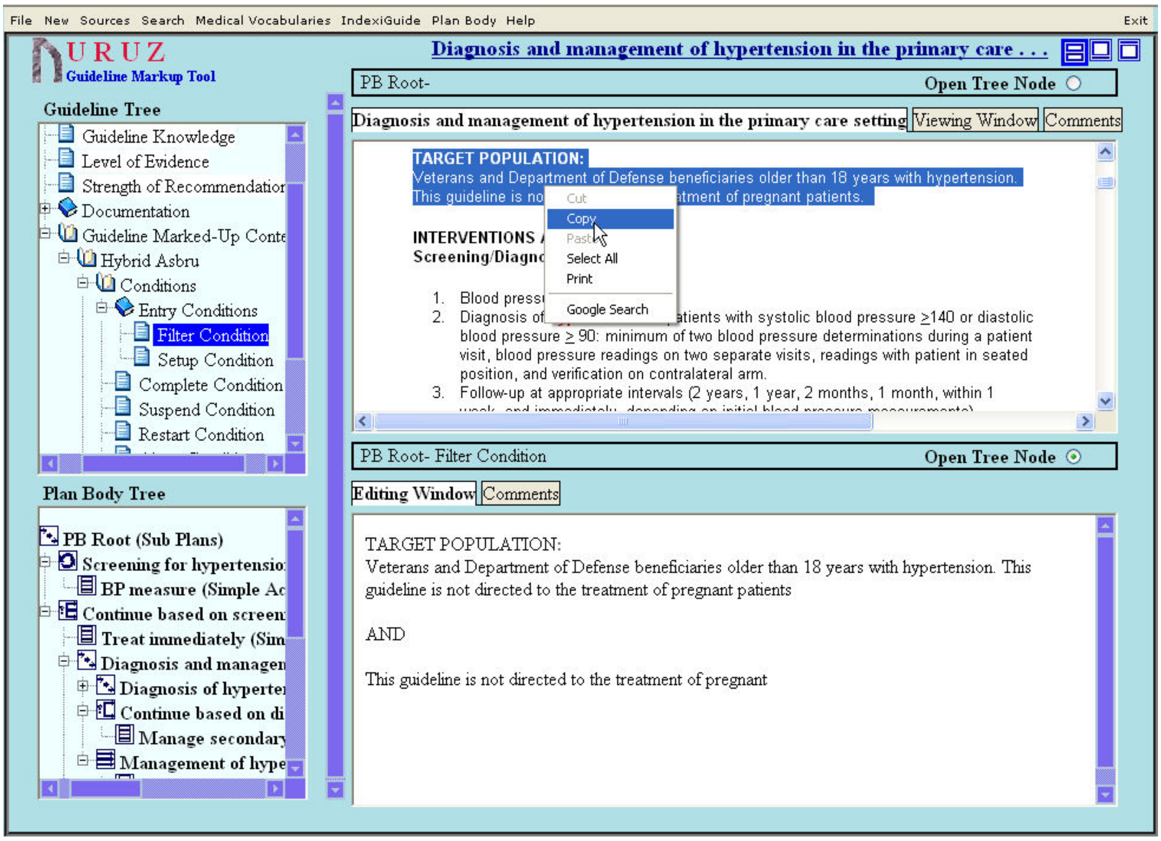


Figure 4: The overall guideline-support architecture, showing both the guideline-representation component (the *DeGeL* server) and the clinical-data access component (the *Idan* server). Both the knowledge-modeling-time and the runtime tools communicate with the guideline knowledge base, which includes all four hybrid representation formats. The design-time tools, used by the medical experts and the knowledge engineers, access also several controlled-terminology medical vocabularies. The runtime tools, used by care providers at the point of care, access also a *mediator* to the time-oriented patient data (The Idan server), which enables answering of queries at a varying level of complexity, regarding the patient data. The runtime tools also have access to a specialized tool (*KNAVE-II*) for interactive visualization and exploration of raw patient data and various guideline-specific abstractions derivable from them.

4.1 Uruz: Semantic Markup

The *Uruz* Web-based guideline markup tool (Figure 5), which is currently in active use, enables medical experts from the several institutions collaborating in the DeGeL project (see Section 3) to create new guideline documents. A source guideline is uploaded into the DeGeL, and is then used by the Uruz user, a medical-expert knowledge editor, to create a new *guideline document,* marked-up by the semantic labels of one of the target ontologies available in DeGeL. Uruz is sometimes used to create a guideline document *de-novo* (i.e., without using any source) by directly writing into the knowledge roles of a selected target ontology. We are currently developing an Asbru-dedicated tool to add the formal-specification level.

The user of the Uruz mark-up tool browses the *source guideline* in one window, and a knowledge role from the target ontology in the other window. To perform *semi-structured* markup, she labels the source content (text, tables, or figures) by dragging it into the knowledge-role frame. Note that the editor can modify the contents or add new content. This enables turning implicit knowledge into a more explicit representation, further facilitating the task of the knowledge engineer who fully formalizes the guideline. Since the target ontology is selected and read on the fly (in the current implementation, as an XML file created from an XML schema), the semi-structured markup module is independent of the target ontology.



TARGET POPULATION:
Veterans and Department of Defense beneficiaries older than 18 years with hypertension.
This guideline is no atment of pregnant patients.

INTERVENTIONS
Screening/Diagno

1. Blood press
2. Diagnosis of atients with systolic blood pressure ≥140 or diastolic blood pressure ≥ 90: minimum of two blood pressure determinations during a patient visit, blood pressure readings on two separate visits, readings with patient in seated position, and verification on contralateral arm.
3. Follow-up at appropriate intervals (2 years, 1 year, 2 months, 1 month, within 1

TARGET POPULATION:
Veterans and Department of Defense beneficiaries older than 18 years with hypertension. This guideline is not directed to the treatment of pregnant patients

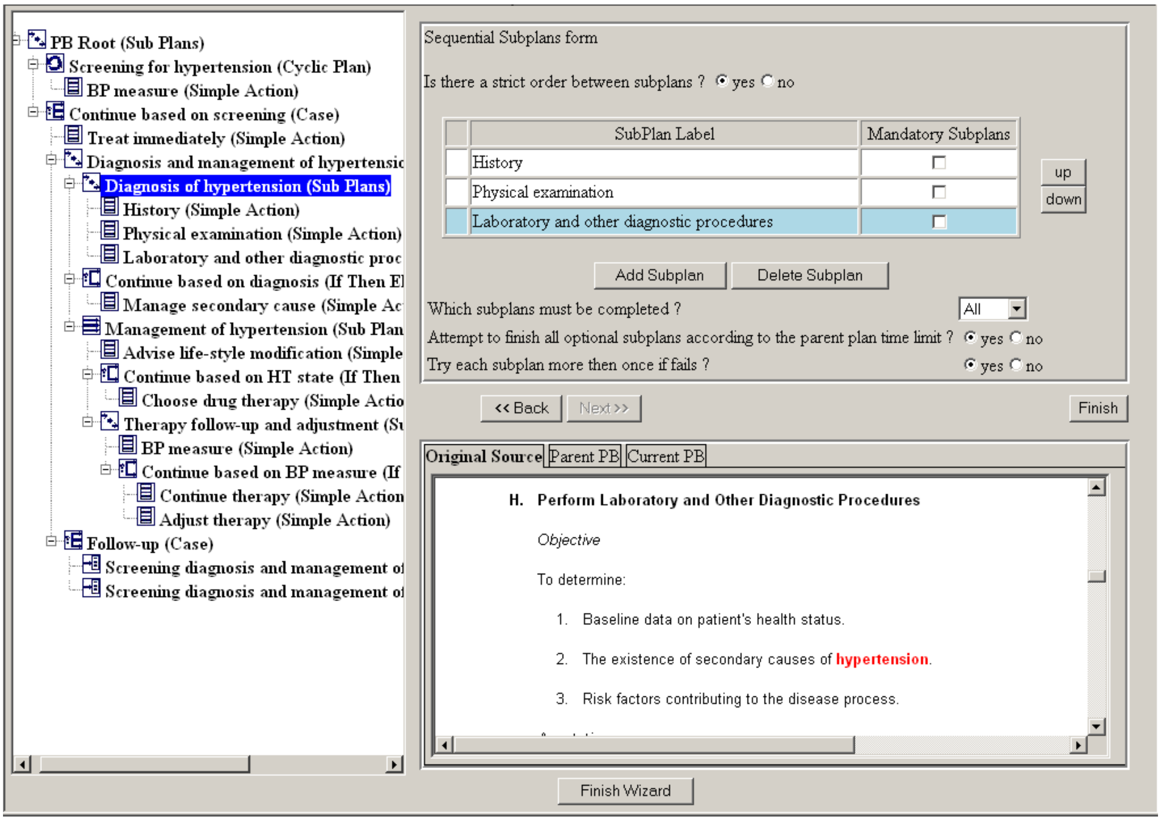
AND

This guideline is not directed to the treatment of pregnant

Figure 5: The Uruz Web-based guideline markup tool. The tool's basic semi-structuring interface is uniform across all guideline ontologies. The target ontology selected by the medical expert, in this case, Asbru, is displayed in the upper left tree; the guideline source is opened in the upper right frame. The expert physician highlights a portion of the source text (including tables or figures) and drags it for further modification into the bottom frame's *Editing Window* tab labeled by a semantic role chosen from the target ontology (here, the Asbru *filter condition*). Note that contents can be aggregated from different locations in the source. The *Comments* tab, the tab next to the *Editing Window* tab, stores remarks on the current selected knowledge-role, thus supporting collaboration among guideline editors.

Uruz supports also adding a semi-formal Asbru representation. Semi-Formal Asbru is a simplified version of Asbru, with similar semantics to the full version, but with a somewhat less complex syntax. The main reason for using Semi-formal Asbru is to improve the collaboration between the expert physicians and the knowledge engineers during the guideline conversion process, specifically after an expert physician structured the guideline and before a knowledge engineer converts it to Asbru. In addition, the semi-formal format still supports text-based retrieval of procedural knowledge, unlike the fully formal format. Finally, a semi-formal structure is obligatory when an electronic medical record is unavailable, since interaction with the clinical user is imperative. This property is exploited to an advantage by our hybrid runtime application module. Semi-formal Asbru has all of Asbru's knowledge-roles, such as conditions (e.g., eligibility, completion, and abort conditions), branching constructs (e.g., if-then-else or switch-case), various synchronization constraints of sub-guidelines (i.e., do in parallel, do in sequential) and time-annotations for describing temporal constraints.

Instead of using Asbru's complex notion of (plan) arguments, each guideline in semi-formal Asbru has a list of patient-related data, *obtained-values,* defined during design-time. Temporal-patterns, the building blocks of a guideline in Asbru, are expressed with combinations of text and time-annotations instead of Asbru's complicated formal expressions. The semi-formal version syntax is defined using an XML schema.



```
PB Root (Sub Plans)
  Screening for hypertension (Cyclic Plan)
    BP measure (Simple Action)
  Continue based on screening (Case)
    Treat immediately (Simple Action)
    Diagnosis and management of hypertensic
      Diagnosis of hypertension (Sub Plans)
        History (Simple Action)
        Physical examination (Simple Action)
        Laboratory and other diagnostic proc
      Continue based on diagnosis (If Then El
        Manage secondary cause (Simple Ac
      Management of hypertension (Sub Plan
        Advise life-style modification (Simple
        Continue based on HT state (If Then
          Choose drug therapy (Simple Actio
        Therapy follow-up and adjustment (Su
          BP measure (Simple Action)
          Continue based on BP measure (If
            Continue therapy (Simple Action
            Adjust therapy (Simple Action)
    Follow-up (Case)
      Screening diagnosis and management of
      Screening diagnosis and management of
```

Sequential Subplans form

Is there a strict order between subplans ? ● yes ○ no

	SubPlan Label	Mandatory Subplans
	History	□
	Physical examination	□
	Laboratory and other diagnostic procedures	□

Which subplans must be completed ? All

Attempt to finish all optional subplans according to the parent plan time limit ? ● yes ○ no

Try each subplan more then once if fails ? ● yes ○ no

H. **Perform Laboratory and Other Diagnostic Procedures**

Objective

To determine:

1. Baseline data on patient's health status.
2. The existence of secondary causes of **hypertension**.
3. Risk factors contributing to the disease process.

Figure 6: The Uruz Asbru semi-formal plan-body wizard (PBW) module. The module supports creation of an Asbru semi-formal control structure. On the left, the guideline's structure tree is displayed and updated dynamically as the user decomposes the guideline. On the upper right, the user is prompted with wizard-like questions to further specify the selected control structure. In the bottom right, the text of the source, current, or parent guidelines is displayed.

A list of common clinical actions, such as drug prescription, laboratory observation and physical examination, had been added to semi-formal Asbru. These actions can be used as reusable primitive plans during guideline design-time, thus simplifying the process of guideline structuring.

To create an Asbru semi-formal representation, an Asbru-specific module, the *plan-body wizard* (*PBW*), had been embedded in Uruz (such modules can be defined also for other ontologies). The PBW is used for defining the guideline's semi-formal control structure (Figure 6). The PBW enables a user to decompose the actions embodied in the guideline into atomic actions and other sub-guidelines, and to define the control structure relating them (e.g., sequential actions). The PBW, used by medical experts, significantly facilitates the final formal specification by the knowledge engineer. When a knowledge engineer needs to add a formal, executable, expression to a knowledge role, she uses one of the ontology-specific Uruz modules (we are developing one specific to Asbru), which delves deeper into the syntax of the target ontology. For example, in our hybrid Asbru, conditions can include temporal patterns in an expressive time-oriented query language used by all of the application modules.

To be truly *sharable*, and avoid the curly brackets problem [23] when applying the guideline in a new environment, guidelines need to be represented in a standardized fashion. Thus, Uruz enables the user to embed in the guideline document, especially when using the PBW, terms originating from one or more of the standard, controlled vocabularies that our vocabulary server includes, using its built-in search engine. Examples include diagnostic terms from the ICD-9-CM vocabulary, or laboratory tests from the LOINC repository (a mult-axial representation, which we are displaying to clinical users hierarchically). In all cases, the user selects a term when needed, through a uniform, hierarchical search interface to the Web-based *vocabulary server*.

4.2 IndexiGuide: Semantic Classification of Guidelines

To facilitate guideline retrieval, the medical expert indexes the guideline document by one or more intermediate or leaf nodes within one or more *external (indexing) semantic axes* trees, using the *IndexiGuide* tool. Currently, the semantic axes include:

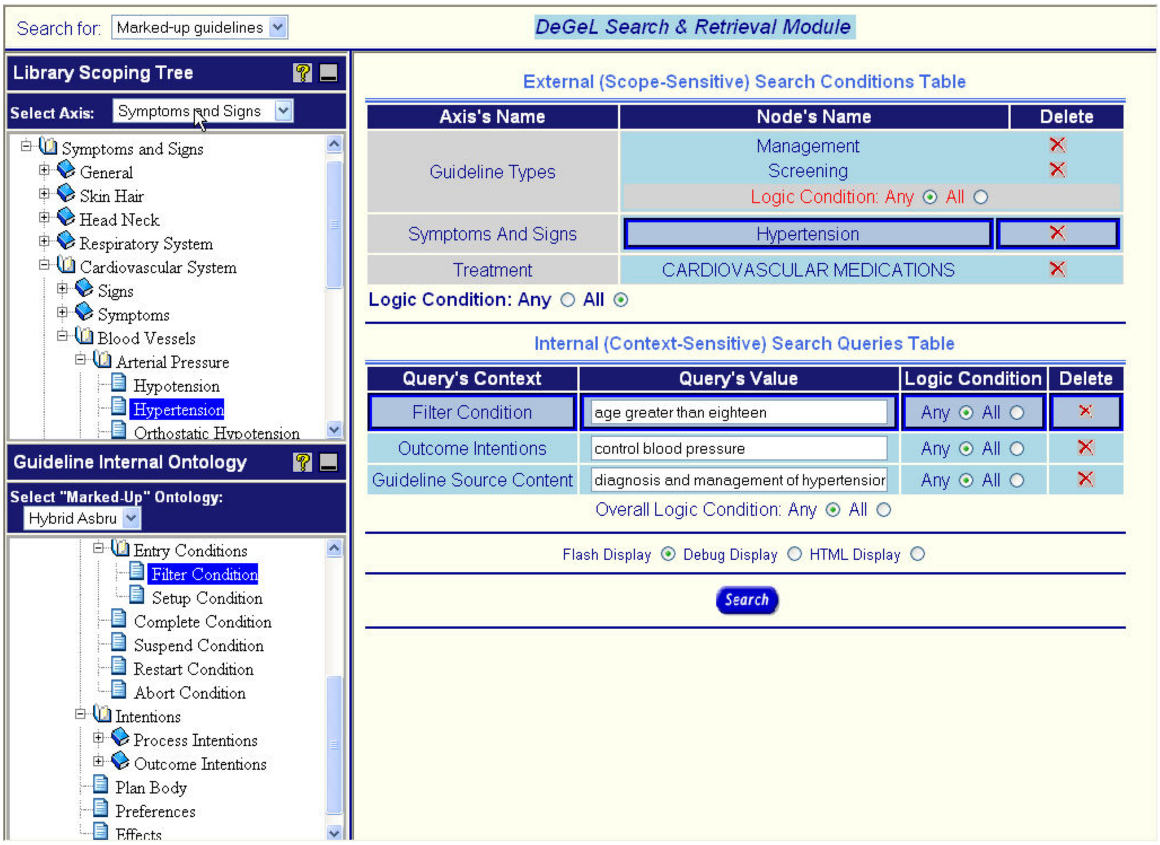
1. The *Symptoms and Physical Signs* axis (e.g., hypertension), which is based on the *Medical Subject Headings* (*MeSH*) standard.
2. The *Laboratory and Special Diagnostic Procedures* axis (e.g., blood-cell counts), which is based on the CPT and LOINC standards.
3. The *Disorders* axis (e.g., endocrine disorders, neoplasms), which is based on the ICD-9 CM standard, a version of ICD.
4. The *Treatments* axis is a combination of a hierarchy of pharmacological treatments (e.g., antibiotic therapy), which is based on the *Veterans Administration NDF* (*VA-NDF*) standard, and a hierarchy of other treatments (e.g., Surgery, special therapeutic procedures, anesthesia), which is based on the CPT standard.
5. The *Body Systems and Regions* axis (e.g., pituitary gland), which is based on the MeSH standard.
6. The *Guideline Types* axis (e.g., screening, prevention, management).
7. The *Medical Specialties* axis (e.g., Genetic).

The choice of the above medical vocabularies for constructing the semantic axes was made based on a trade-off between the expressiveness of each vocabulary, and the need to represent only the top 3-4 levels of each semantic axis, which are typically sufficient for the purpose of classifying the guideline. In addition, we were looking for broadly accepted standards. For example, the LOINC medical vocabulary is not only the most expressive in its clinical domain but also the recommended one by the Health Insurance Portability and Accountability Act (HIPAA).

4.3 Vaidurya: Context-Sensitive Search and Retrieval of Guidelines

The *Vaidurya* hybrid guideline search and retrieval tool [26] exploits the existence of the free-text source, the semantic indices, and the marked semi-structured-text.

Figure 7 shows the Vaidurya query interface. The user, performing a search, selects one or more concepts from one or more external semantic axes, or scopes, to limit the overall search. (e.g., disorders = *hypertension*). The tool also enables the user to query marked-up guidelines for the existence of terms within the *internal context* of one or more target-ontology's knowledge roles (e.g., in the case of Asbru, the filter condition context includes the term *pregnancy*).



Search for: Marked-up guidelines

DeGeL Search & Retrieval Module

Select Axis: Symptoms and Signs

External (Scope-Sensitive) Search Conditions Table

Axis's Name	Node's Name	Delete
Guideline Types	Management	✕
	Screening	✕
	Logic Condition: Any ● All ○	
Symptoms And Signs	Hypertension	✕
Treatment	CARDIOVASCULAR MEDICATIONS	✕

Logic Condition: Any ○ All ●

Internal (Context-Sensitive) Search Queries Table

Query's Context	Query's Value	Logic Condition	Delete
Filter Condition	age greater than eighteen	Any ● All ○	✕
Outcome Intentions	control blood pressure	Any ● All ○	✕
Guideline Source Content	diagnosis and management of hypertensior	Any ● All ○	✕

Overall Logic Condition: Any ● All ○

Flash Display ● Debug Display ○ HTML Display ○

Select "Marked-Up" Ontology: Hybrid Asbru

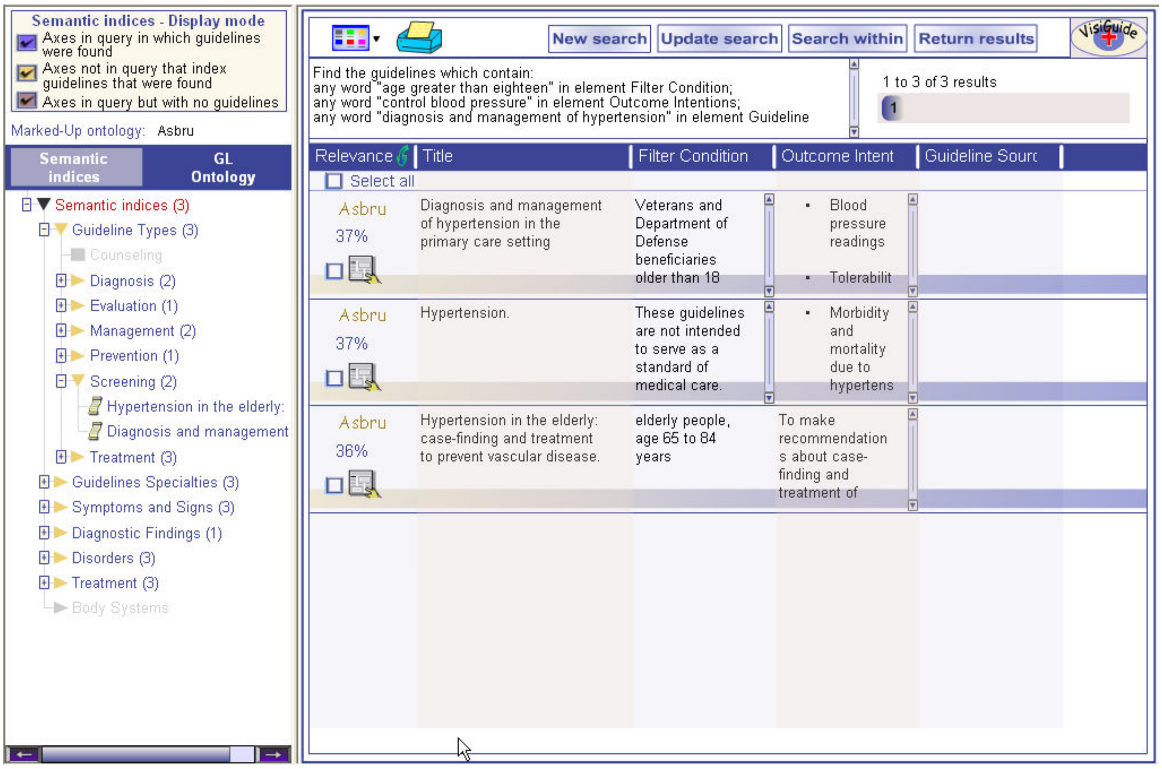
Figure 7: The Vaidurya Web-based, context-sensitive, guideline search and retrieval tool. The user defines the relevant search scope by indicating one or more nodes within the semantic axes (upper left and right frames). The search can be further refined by specifying terms to be found within the source text, and even (after selecting a target ontology such as Asbru), within the context of one or more particular knowledge roles of that ontology (middle right frame).

For search using *external scopes*, the default constraint is a *conjunction* (i.e. AND) of all selected axes (e.g., both a Cancer diagnosis within the disorders axis and a Chemotherapy therapy within the treatments axis) but a *disjunction* (i.e. OR) of concepts within each axis. For *internal contexts*, the default semantics are to search for a *disjunction* of the keywords within each context, as well as among contexts (i.e., either finding the term *diabetes* within the Filter Condition context or the term *hypertension* within the Effects context). The search results are browsed, both as a set and at a single-guideline level, using a specialized guideline-visualization tool.

4.4 VisiGuide: Guideline Browsing

The *VisiGuide* **browsing** and visualization tool (Figure 8) enables users to browse a set of guidelines returned by the Vaidurya search engine and visualize their structure.

VisiGuide is linked to the various DeGeL applications, allowing the user to return one or more selected guideline for use within the Uruz markup tool or the IndexiGuide semantic classifier. Like the Vaidurya search and retrieval tool (see Section 4.3), VisiGuide makes no assumptions regarding the guideline's ontology, and dynamically parses a guideline ontology expressed as an XML schema, although it can have extensions for specific ontologies (e.g., for display of the Asbru semi-formal plan-body, such as acquired by the PBW).



Find the guidelines which contain:
any word "age greater than eighteen" in element Filter Condition;
any word "control blood pressure" in element Outcome Intentions;
any word "diagnosis and management of hypertension" in element Guideline

1 to 3 of 3 results

☐ Select all

Relevance	Title	Filter Condition	Outcome Intent	Guideline Sourc
Asbru 37% ☐	Diagnosis and management of hypertension in the primary care setting	Veterans and Department of Defense beneficiaries older than 18	• Blood pressure readings • Tolerabilit	
Asbru 37% ☐	Hypertension.	These guidelines are not intended to serve as a standard of medical care.	• Morbidity and mortality due to hypertens	
Asbru 36% ☐	Hypertension in the elderly: case-finding and treatment to prevent vascular disease.	elderly people, age 65 to 84 years	To make recommendation s about case-finding and treatment of	

Figure 8: An example of the VisiGuide Interface in the *multiple-ontology* mode. In this mode, multiple guidelines, typically retrieved by Vaidurya search engine, are displayed within the various semantic axes indexing them (left frame); the contents of knowledge roles relevant to the user are displayed and compared as a table (right frame). The "Return Results" button returns selected guideline back to the requesting application (e.g., to the Uruz markup tool). The left-hand icon in the top menu enables the user to select among several preset views, such as an *eligibility* view (which in the case of the Asbru ontology displays, for all guidelines, along with the guideline's title, the *filter* and *setup* conditions).

VisiGuide organizes guidelines along the semantic axes in which they were found, distinguishing between axes that were requested in the query (e.g., disorders = *breast carcinoma* and treatments = *chemotherapy*) and axes that were not requested but which where originally used to classify a retrieved guideline (e.g., treatments = *radiotherapy*). Axes that were requested in the query but in which no guideline was found are highlighted (differently) as well.

In the *multiple-guideline* display mode, a table listing the content of desired semi-structured knowledge roles for all retrieved guidelines or for all guidelines that are indexed by a certain semantic axis can be created on the fly by simply indicating the interesting knowledge roles in the target ontology by which the guideline was marked (semi-structured), thus enabling quick comparison of several guidelines. Several preset default views exist for quick selection of a group of knowledge roles to display, such as the *eligibility determination* and the *quality assessment* views, in the case of the Asbru ontology (see Figure 11).

In the *single-guideline* display mode, a listing of the content of each of the knowledge roles or any combination can be more deeply examined. Thus, supporting actual application or quality assessment. The ontology by which the guideline that is being browsed was marked up is displayed, and the user dynamically selects which knowledge roles she is interested in. These knowledge roles then appear on screen and their content can be examined.

5 DeGeLock: Authorization and Permission

Due to practical and legal considerations, any digital guideline library must include a comprehensive *authorization* model. The hierarchical model used in DeGeL uses the notions of *virtual expert groups* and of the different *functionalities* inherent in the hybrid meta-ontology model, which imply different levels of authorization. Guideline editors are *members* of one or more (*editing*) *groups* (Figure 9) and have different authorizations in each group.

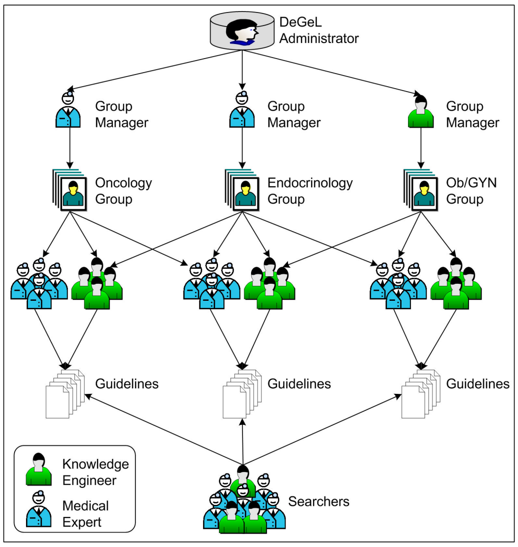


Figure 9: The DeGeL authorization and collaboration model. All guideline editors (medical experts and knowledge engineers) are *members* of one or more *groups*. Group members edit guidelines in their group. Most users are *Searchers* (the minimal level of permission) who only browse the library. *Group managers* can add group members (medical experts or knowledge engineers) and edit their permissions. *System administrators* can create new groups and might have special permissions, such as editing semantic axes.

Groups are organized by medical specialty (e.g., oncology). Each *group manager* can accept applications to be a group member, and sets and maintains the authorization configuration of each member in that group. Members of a group can only edit and classify guideline documents based on source guidelines *owned* (uploaded) by a group member, but cannot edit guideline documents owned by another group.

The DeGeL authorization model assumes that each module (e.g., Uruz) enables users to perform several *tasks* (Table 1). Each user is given (within each group) a specific authorization configuration for each module. To facilitate management, we have predefined several common authorization profiles (more can be constructed in similar fashion):

1. *Searcher* (visits the library, performs searches, and views guidelines which have been edited by other users). This is the minimal authorization level for any user.
2. *Classifier* (classifies guidelines alongside semantic axes). Classifiers typically possess medical knowledge to some degree.
3. *Expert Editor* (specifies guidelines' content up to the semi-structured level, using DeGeL's hybrid meta-ontology). Editors are usually medical experts.
4. *Knowledge Engineer* (cannot markup the guideline, but can fully structure the marked-up text up to machine-comprehensible level in the full target ontology). These are experts in the semantics of the guideline’s target ontology.
5. G*roup manager* (manages permissions of their group members); typically, a medical expert; possibly, a knowledge engineer.
6. *System Administrator* (manages users and groups). The top authorization level.

Each user profile targets a specific population of potential users. The majority of physicians will use the library as Searchers; a small number of experts in each specialty will serve as Classifiers or Editors.

Note that the groups are typically organized by medical specialty and not by geographic location (see Figure 9).

Table 1: Examples of several DeGeL modules and the tasks and specific operations they enable users to perform or apply, respectively.

Module	Relevant Tasks
Vaidurya, VisiGuide	Search, Retrieve, Visualize, Browse guideline sources or guideline documents
IndexiGuide	View guideline indices, Classify guideline documents
URUZ	View, Edit, Search within guideline documents
Guideline management	Create, Delete guideline sources and documents
Group management	Add, Remove group members; Modify group members details, authorizations
Axes builder	View, Modify Semantic classification axes
System administration	Add, Remove groups, module-tasks, user types, users

The default configuration profiles for each authorization type are predefined [27]. For example, the *classifier* authorization type grants access to the Vaidurya and IndexiGuide modules for allowing the user to semantically classify guidelines in her group. However, by default, the user is not allowed to modify the content of the guidelines in Uruz.

A group manager can easily assign a new member to a predefined authorization type, possibly modifying the configuration if needed, using a Web-based graphical *permissions-manager* tool, which we had developed for that purpose. The permissions-manager tool is also used by system administrators to manage all DeGeL users, including group managers. Group managers and administrators can view details of group members, authorize addition of new members, and change authorization configurations for existing members.

For example, selecting the *Classifier* authorization type defines a particular default configuration, which authorizes classification in *IndexiGuide*, but not editing in *Uruz.* Similarly when the *Uruz editor* predefined profile is selected (see Figure 14).

Once a predefined profile is selected, the user of the permissions-manager tool can dynamically customize the particular profile of the user in question.

When the permissions-manager tool is being used by a system administrator, additional options are displayed for special types of maintenance operations, such as creating new groups and appointing group managers for them.

6 The DeGeL collaboration model

There are several means of collaboration among DeGeL editors. These support our vision of a Web-based, distributed global community incrementally marking up a large body of clinical guidelines and gradually converting it to executable representations in one or more guideline representation formats. These collaboration facilitators include:

1. The expert group model enables several co-editors to work on the same guideline (e.g., each marking a different knowledge role). Collisions when two users are attempting to modify the same guideline document are prevented by a standard check-out/check-in database model.
2. Information can be shared among editors, using the *element-comments* editing role (see Figure 7).
3. A meta-ontology element (i.e., common to all guideline representation formats), called the *clipboard,* enables editors to create a temporary workspace, which supports sharing any type of free text, figures, or tables, from any source document, thus facilitating the editing process.
4. Editors can copy existing marked-up guidelines (edited by their colleagues), give them a new title, modify them, and thus create a new marked-up guideline. This capability greatly facilitates reuse of an existing semantic markup.
5. Editors can mark-up an existing source (uploaded by a colleague) using a different target ontology than the one used to create the current guideline document using that source.

7 Using the DeGeL Architecture

The DeGeL framework has been fully implemented using a Windows-DNA platform, using an Active Server Pages (ASP) technology, in conjunction with COM+ services. The platform is currently being converted into a .NET Framework technology (including Web-Services interfaces, etc.). The main mode for internal knowledge representation is XML documents, which are structured along predefined XML schemata. Thus, a target ontology (e.g., GEM, Asbru) is an XML file in a predefined DeGeL guideline-ontology meta schema. XML documents are stored in an MS-SQL relational database.

We have been collaborating with clinicians from several medical centers who have been assisting us in designing and, recently, in evaluating the DeGeL framework. These centers were mentioned in Section 3: The SUH, PAMF, and the VA PAHCS medical centers in California, and the Soroka Medical Center and the CIC association in Israel. We will elaborate here a bit more on the nature of these collaborations.

An ongoing NIH-funded project at the VA PAHCS, whose goal is to support guideline-based quality assessment, has enabled an assessment (at least in preliminary fashion and within an academic environment) of all of the DeGeL architecture's modules, as well as a detailed evaluation of the IDAN and KNAVE-II clinical-data access tools [25].

Several clinical domain experts from Stanford University and the VA PAHCS have been using over the past three years the IndexiGuide, Uruz, Vaidurya, and VisiGuide tools, to create a library of about 170 semantically indexed guidelines in DeGeL, about 30 of which are also marked up. The preliminary results have already demonstrated the functionality of the tools and the ability of the experts to work with them, but we are currently evaluating the tools more rigorously.

We are currently conducting several studies to assess the usability and functionality of all tools, in particular, the Uruz mark-up tool. Since inter-editor variability is potentially a serious issue, we are assessing both the general usability of the tool and the significance of such variability. Thus, we are evaluating the mark-up results from several aspects, judging (1) syntactic differences, assessed mainly by a knowledge engineer familiar with the target ontology; (2) semantic differences, assessed mainly by a domain expert familiar with the guideline's domain; and (3) pragmatic differences, assessed by running the resulting marked-up guidelines using the Asbru semi-structured and semi-formal runtime execution module, Spock [28], on a set of simulated patient records.

One initial impression that has clearly emerged from the initial evaluations has been the need to first create and document a *consensus* (typically, text-based) among a committee of domain experts, regarding the meaning of the guideline (preferably, in the terms of a specific target ontology, such as different Asbru conditions). Without such a consensus, there is little point in embarking on a time-consuming mark-up effort, and the variability among editors might well be due to different interpretation of the same guideline. The importance of creating such a consensus before implementing a clinical guideline has been noted previously when using an early version of the GLIF ontology to implement a hypercholesterolemia-management guideline [29] (see additional discussion in Section 8).

The preliminary results of using the Vaidurya search and retrieval engine at several levels (free-text, semantic axes, and context-sensitive search), demonstrating monotonic improvement with the use of increasingly specific queries, have also been highly encouraging [26].

We have also had a fruitful experience with the CIC organization. CIC is a non-for profit association focused on assisting patients suffering from very severe diseases, such as cancer, to access up-to-date information regarding possible treatments relevant to their clinical condition, and to enable the patients and their care-providers to make educated decisions.

In order to supply such a valuable service, the CIC medical consultants create surveys of the most recent information on various diseases and use these for preparing knowledgeable answers to patients' questions.

Over the past two years, we have had an ongoing collaboration with CIC. As part of that collaboration, an ontology for creating CIC surveys was developed in collaboration with the CIC clinicians, and incorporated in the DeGeL knowledge-base. Thus, the DeGeL system is used as the initial clinical information repository of CIC, from which summaries (and later, specific responses for information requests) are created and stored within a separate CIC information system.

Another fruitful collaboration has been ongoing with the Soroka Medical center of Ben-Gurion University, in particular the Obstetrics and Gynecology Division. Apart from getting useful suggestions, mostly for the design-time tools, a major aim of this collaboration is the development of two clinical guidelines in the Asbru language using DeGeL tools, one for treating patients suffering from Pelvic Inflammatory Disease (PID), for the gynecology department, and the second for treating pregnant patients suffering from uncontrolled blood pressure, for the obstetrics department.

Overall, preliminary assessments of the various DeGeL tools by our clinical colleagues are highly encouraging regarding usability and functionality, and formal evaluations of most tools are under way. Preliminary experience has shown the feasibility of marking up, searching, and displaying guidelines in the Asbru, GEM, and CIC ontologies. We intend to add other ontologies, such as GLIF, to the target ontologies available in DeGeL.

7.1 The DeGeL runtime tools

In addition to the various DeGeL specification and retrieval tools, we have also been developing in parallel several Asbru-specific tools for runtime guideline application and for retrospective quality assessment of guideline-based care. The **Spock** runtime-application module [28] is a hybrid Asbru runtime application module, which currently focuses mainly on the semi-structured representation. **QualiGuide** is a retrospective quality-assessment tool, which uses the concept of intention-based quality assessment [12, 30] and the Asbru intentions knowledge role. Besides using the DeGeL knowledge base (i.e., the guideline library itself), both tools use the IDAN mediator and the KNAVE-II visualization tool (see Section 4) to query and explore the patient's data.

8 Discussion

Hybrid representations of clinical guidelines, as described in the current paper, include any combination of free-text, semi-structured text, semi-formal representation, and machine-comprehensible formats in a chosen target guideline ontology. They cater for the different capabilities of expert physicians, who need have only limited knowledge of the semantics of the chosen target ontology, and knowledge engineers, who are expected to have full semantic and syntactic knowledge of the chosen ontology, but do not necessarily have deep knowledge of the guideline's semantics and goals. By incrementally converting free-text guidelines into semi-structured, semi-formal, and then formal specifications, we are gradually enhancing the sophistication of the automated services that the guideline’s representation can support: from full-text search, through context-sensitive search and visualization (sensitive to specific knowledge roles of the target ontology), to fully automated application and quality assessment.

At the same time, the semi-structured view provides an independent value: Search precision has been shown to be significantly improved by marking-up the text of medical documents [31], while displaying documents along a predefined meaningful ontology is highly preferred by users [32].

Furthermore, the tools we are developing for runtime application and quality assessment can exploit the intermediate representation levels. Indeed, only a semi-structured or semi-formal representation is useful when no electronic patient record is available, and the attending physician or quality-assessment nurse is acting as the mediator to the patient record. Thus, our hybrid representation leads to a *graceful degradation* in the level of service experienced by the user, even when neither the guideline nor the medical record is fully machine comprehensible.

Although the advantages are considerable, one obvious limitation inherent in the hybrid-ontology model is the need to provide ontology-specific tools (both for editing and for runtime application) for the semiformal and formal representations. The reason is that, unlike the meta-ontology we suggest, whose elements are common to all guideline ontologies, the procedural semantics of each ontology differ considerably. Proposals to unify additional aspects of guideline ontologies (e.g., a virtual view of the medical record, a common objectives representation, and a generalized hierarchical plan representation) might eventually reduce the magnitude of this problem or completely solve it.

The effort invested in creating a consensus of experts before formally representing a guideline has been previously documented by researchers who had used an early version of the GLIF ontology, GLIF-2, for representation of complex clinical guidelines [29]. The research team encoded, using an extended version of GLIF-2, the secondary prevention portion of the National Cholesterol Education Program (NCEP) guideline for management of hypercholesterolemia [33]. The editing and execution framework was the Partner's Computerized Algorithm Processor and Editor (PCAPE) system [34]. Although the expressivity of the representation framework was not an obstacle, the developers invested the bulk of their effort in first creating a consensus regarding the guideline's semantics, and then translating it into an executable format. (It is interesting to note, though, that the major obstacle was in effectively integrating the system into the clinical workflow. The authors' opinion was that without more sophisticated methods for such integration, including outpatient order entry, the benefits of complex guideline systems over simple rule-based reminders will be small.)

Our future plans include modifying the current architecture so as to enable multiple (local) DeGeL sites, increasing the flexibility (e.g., easier access, improved security, and knowledge segmentation) of the DeGeL framework.. A relative difficulty might arise when we implement our vision of multiple DeGeL sites. The task of indexing and searching through all local guideline libraries will become considerably more complex, and necessitate a new, distributed version of the Vaidurya search engine.

An interesting issue regarding the best authorization model for controlling the creation and editing of medical knowledge has arisen due to the distributed nature of the DeGeL library. The DeGeL permission model is based on a virtual, distributed medical-specialty authoring group notion, and on the different functionalities implied by the hybrid-representation model. The model is inspired by the legal and practical considerations involved in editing medical knowledge by multiple experts.

We are currently in the process of development of a new version of the *DeGeLock* permission and authorization module, to improve its flexibility and performance, by focusing on conceptual *entities* (e.g., ontologies, axes, guideline-documents), and *actions* that are performed on every entity (e.g *update*, *delete*, *view*), neither of which necessarily correspond to any particular DeGeL module. We expect the extensions we are planning, namely focusing on entities and actions rather than on specific computational modules, to significantly enhance the *DeGeLock* module's flexibility and support the needs of a global knowledge-editing community.

We also intend to open the DeGeL architecture to facilitate the addition of new modules, by exposing a standard application interface to DeGeL's server-side components, for example to support runtime application modules for guideline languages other than Asbru. Furthermore, we are designing and implementing new client-server based modules (e.g., desktop applications), such as *Uruz-3*, a graphical markup editor for hybrid ontologies, whose interface we consider to be more intuitive than that of the current Web-based Uruz module. In general, we intend to convert modules that require highly-sophisticated graphical interfaces into desktop applications, while maintaining the distributed nature of the architecture through a link to the DeGeL server.

The experience of using the Vaidurya context-sensitive search and retrieval engine has demonstrated that clinical users encounter several difficulties when facing its complex interface. Thus, we are currently adding to Vaidurya customizable and template-based interfaces [26]. These *search templates* represent typical information needs of various user types, and enable customization by the user.

Our vision for the future is a global network of hybrid digital guideline libraries, with a hierarchical (group-based) community of medical experts and knowledge engineers maintaining the knowledge base, using tools of the Uruz type and standardized vocabularies that are not site specific.

We also envision a set of tools for searching, retrieving, browsing, applying at runtime, and assessing retrospectively the quality of application, of guidelines in the library.

Ontology-independent tools, such as Vaidurya and VisiGuide, will be complemented by ontology-specific tools, such as Spock and QualiGuide.

9 Acknowledgments

This research was supported in part by NIH award No. LM-06806. We thank Samson Tu and Mor Peleg for useful discussions regarding the need for supporting the use of multiple guideline ontologies. Drs. Richard Shiffman and Bryant Karras assisted us in using their GEM ontology. Drs. Mary Goldstein, Susana Martins, Lawrence Basso, Herbert Kaizer, Aneel Advani, and Eitan Lunenfeld, were extremely helpful in assessing the various DeGeL tools.

10 References

1. Field, M. and Lohr, K., *Clinical Practice Guidelines: Directions for a New Program*, ed. I.o.M. Committee to Advise the Public Health Service on Clinical Practice Guidelines. 1992.
2. Peleg, M., Boxwala, A., Bernstam, E., Tu, S., Greenes, R., and Shortliffe, E., *Sharable representation of clinical guidelines in GLIF: Relationship to the Arden syntax.* Journal of Biomedical Informatics, 2001. **34**: p. 170-181.
3. Grimshaw, J. and Russel, I., *Effect of clinical guidelines on medical practice: A systematic review of rigorous evaluations.* Lancet, 1993. **342**: p. 1317-1322.
4. Qualigini, S., Ciccarese, P., Micieli, G., and Cavallini, A., *Non-Compliance with Guidelines: Motivations and Consequences in a case study.* Proccedings of the symposium on Computerized Guidelines s and Protocols (CGP 2004), 2004: p. 75-87.
5. Micieli, G., Cavallini, A., and S, Q., *Guideline Compliance Improves Stroke Outcome - A Preliminary Study in 4 Districts in the Italian Region of Lombardia.* Stroke, 2002. **33**: p. 1341-1347.
6. Sherman, E., Hripcsak, G., Starren, J., Jenders, R., and Clayton, P., *Using Intermediate States to Improve the Ability of the Arden Syntax to Implement Care Plans and Reuse Knowledge.* Proceedings of the Annual Symposium on Computer Applications in Medical Care (SCAMC-95), 1995. **19**: p. 238-242.
7. Hripcsak, G., Ludemann, P., Pryor, T., Wigertz, O., and PD, C., *Rationale for the Arden Syntax.* Computers and Biomedical Research, 1994. **27**: p. 291-324.
8. Tu, S., Kahn, M., Musen, M., Ferguson, J., Shortliffe, E., and Fagan, L., *Episodic Skeletal-plan refinement on temporal data.* Communications of The ACM, 1989. **32**: p. 1439-1455.
9. Musen, M., Carlson, R., Fagan, L., and Deresinski, S., *T-HELPER: Automated Support for Community-Based Clinical Research.* Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care, 1992: p. 719-723.
10. Herbert, S., Gordon, C., Jackson-Smale, A., and Renaud Salis, J., *Protocols for clinical care.* Computer Methods and Programs in Biomedicine, 1995. **48**: p. 21-26.
11. Musen, M., Tu, S., Das, A., and Shahar, Y., *EON: A component-based approach to automation of protocol-directed therapy.* Journal of the American Medical Information Association, 1996. **3(6)**: p. 367-388.
12. Shahar, Y., Miksch, S., and Johnson, P., *The Asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines.* Artificial Intelligence in Medicine, 1998. **14**: p. 29-51.
13. Fox, J., Johns, N., and Rahmanzadeh, A., *Disseminating medical Knowledge: the PROforma approach.* Artificial Intelligence in Medicine, 1998. **14**: p. 157-181.
14. Peleg, M., Boxwala, A., Omolola, O., Zeng, Q., Tu, S., Lacson, R., Bernstam, E., Ash, N., Mork, P., Ohno-Machado, L., Shortliffe, E., and Greenes, R., *GLIF3: The Evolution of a Guideline Representation Format In Overhage M.J., ed.* Proceedings of the 2000 AMIA Annual Symposium, 2000: p. 645-649.

15. Johnson, P., Tu, S., Booth, N., Sugden, B., and Purves, I., *Using scenarios in chronic disease management guidelines for primary care.* Proceedings of the 2000 AMIA Annual Symposium, 2000: p. 389-393.

16. Tang, P. and Young, C., *ActiveGuidelines: Integrating Web-Based Guidelines with Computer-Based Patient Records.* Proceedings of the 2000 AMIA Annual Symposium, 2000: p. 843-847.

17. Shiffman, R., Karras, B., Agrawal, A., Chen, R., Marenco, L., and Nath, S., *GEM: a proposal for a more comprehensive guideline document model using XML.* Journal of the American Medical Informatics Association, 2000. **7(5)**: p. 488-498.

18. Gershkovich, P. and Shiffman, R., *An Implementation Framework for GEM Encoded Guidelines.* Proceedings of the 2001 AMIA Annual Symposium, 2001: p. 204-208.

19. Miller, P., *Expert Critiquing Systems: Practice-Based Medical Consultation by Computer*. 1986, New York, NY.: Springer-Verlag.

20. Miller, P., *A Critiquing Approach to Expert Computer Advice: ATTENDING*. 1984, London: Pitman.

21. Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R., Hall, R., Johnson, P., Jones, N., Kumar, A., Miksch, S., Quaglini, S., Seyfang, A., Shortliffe, E., and Stefanelli, M., *Comparing Computer-Interpretable Guideline Models: A Case-Study Approach.* Journal of the American Medical Informatics Association, 2003. **10(1)**: p. 52-68.

22. Boaz, D. and Shahar, Y., *Idan: A distributed temporal-abstraction mediator for medical databases.* Proceedings of the 9th Conference on Artificial Intelligence in Medicine-Europe (AIME), 2003: p. 21-30.

23. Pryor, T. and Hripcsak, G., *Sharing MLM's: an experiment between Columbia-Presbyterian and LDS Hospital.* Proceedings of the Seventeenth Annual Symposium on Computer Applications in Medical Care, 1993. **399-403**.

24. Shahar, Y., Goren-Bar, D., Galperin, M., Boaz, D., and Tahan, G., *KNAVE II: A distributed architecture for interactive visualization and intelligent exploration of time-oriented clinical data.* Notes of the 7th International Workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-2003), 2003: p. 103-110.

25. Martins, S., Shahar, Y., Galperin, M., Goren-Bar, D., Boaz, D., Tahan, G., Kaizer, H., Basso, L., and Goldstein, M., *Evaluation of KNAVE-II: A tool for intelligent query and exploration of patient data.* Proceedings of the 2004 Medinfo, 2004.

26. Moskovitch, R., Hessing, A., and Shahar, Y., *Vaidurya - A concept-based, context-sensitive search engine for clinical guidelines.* Proceedings of MEDINFO-2004, the Eleventh World Congress on Medical Informatics, 2004.

27. Shahar, Y., Shalom, E., Mayaffit, A., Young, O., Galperin, M., Martins, S., and Goldstein, M., *A distributed, collaborative, structuring model for a clinical-guideline digital-library.* Proceedings of the 2003 AMIA Annual Fall Symposium, 2003: p. 589-593.

28. Young, O. and Shahar, Y., *Spock: A Hybrid Model for Runtime Application of Asbru Clinical Guidelines.* Proceedings of MEDINFO-2004, the Eleventh World Congress on Medical Informatics, 2004.

29. Maviglia, S., Zielstorff, R., Paterno, M., Teich, J., Bates, D., and Kuperman, G., *Automating complex guidelines for chronic disease: Lessons learned.* Journal of the American Medical Informatics Association, 2003. **10(2)**: p. 154-165.

30. Advani, A., Musen, M., and Shahar, Y., *Medical quality assessment by scoring adherence to guideline intentions.* Proceedings of the 2001 AMIA Annual Fall Symposium, 2001.

31. Purcell, G., Rennels, G., and Shortliffe, E., *Development and Evaluation of a Context-Based Document Representation for Searching the Medical Literature.* International Journal of Digital Libraries, 1997. **1**: p. 288-296.

32. Pratt, W., *Dynamic Organization of Search Results Using the UMLS.* Proceedings of the 1997 AMIA Annual Fall Symposium, 1997: p. 480-484.

33. *Summary of the second report of the National Cholesterol Education Program (NCEP) Expert Panel on Detection, Evaluation, and Treatment of High Blood Cholesterol in Adults (Adult Treatment Panel II).* Journal of the American Medical Association, 1993. **269(23)**: p. 3015-3023.

34. Zielstorff, R., Teich, J., and Fox, R., *P-CAPE: A high-level tool for entering and processing clinical practice guidelines.* Proceedings of the 1998 AMIA Annual Symposium, 1998: p. 478-482.

Appendix A: DeGeL's Knowledge-Base Guideline Repository Structure

A simplified view of the main entities and relations of the guideline repository displayed in the unified modeling language (UML) notation. As explained previously, there are three types of ontologies: (1) the SourceOntology class is related to the SourceGuideline class through the documented-with association; (2) the HybridMetaOntology class consists of knowledge roles that are independent of any particular guideline-specification ontology, and is related to the HybridGuideline class through the documented-with association; , and (3) the SpecificationOntology class is related to the HybridGuideline class through the specified-with association. All the ontology classes inherit common attributes, such as name and creation date, from the abstract base type Ontology. In addition, both HybridGuideline and SourceGuideline classes are related to the ClassificationAxis class through the classified-by association. Finally, the HybridGuideline class is related to additional HybridGuideline classes, through the association decomposed-into, thus fulfilling the requirement of most modern guideline-specification ontologies for enabling the creation of an explicit or Implicit hierarchy of guidelines and subguidelines

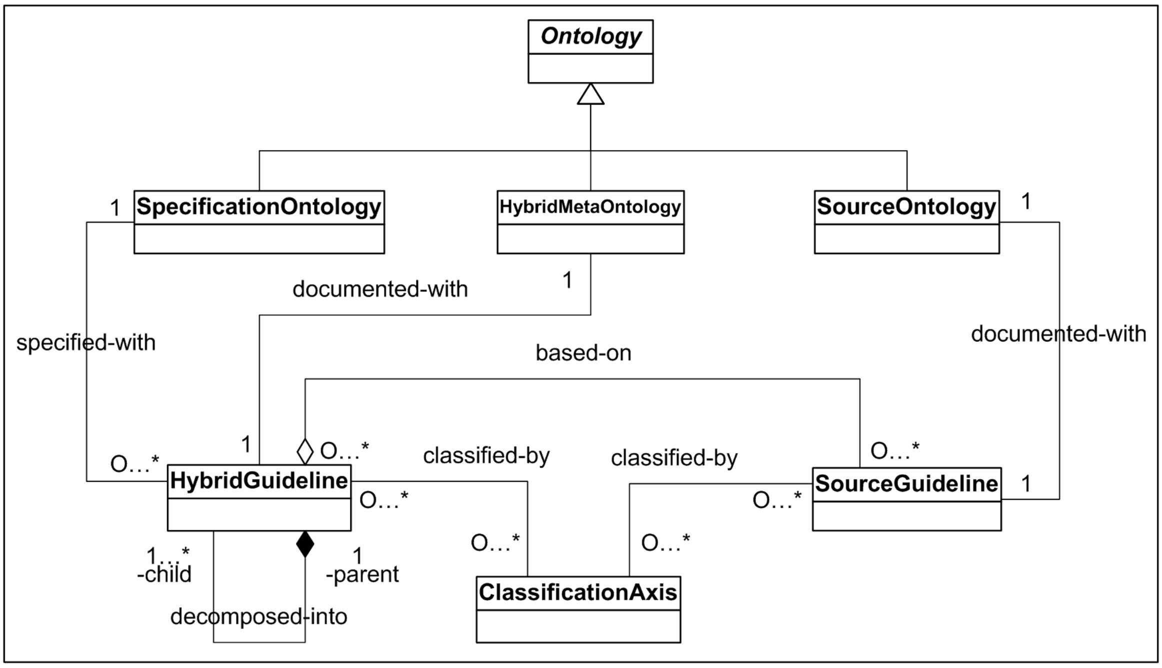


Figure A.1: A simplified view of the main entities and relations of the guideline repository displayed in the *unified modeling language* (*UML*) notation.

Appendix B: A Hybrid-Asbru Representation Example

The following example tracks the life cycle of a Hybrid-Asbru knowledge-role example. The text is taken from a free-text guideline for screening, treating and monitoring patients with hypertension.

The Asbru *abort condition* knowledge role is displayed at all three hybrid representation levels that were created during the incremental conversion process:

1. The free-text source:

> …
>
> Patients should be seen within 1 or 2 months after the initiation of therapy to determine adequacy of hypertension control, degree of patient adherence, and presence of adverse effects. Earlier follow-up may be necessary for patients:
>
> Requiring blood tests
> At increased risk for adverse outcomes from hypertension
> At risk for postural hypotension
>
> Once the patient's blood pressure is stabilized, follow-up at 3- to 6-month intervals (depending on patient status) is generally appropriate. Older persons, diabetics, and those at risk for postural hypotension (with orthostatic symptoms) may require blood pressure measurement in the seated position and, to recognize postural hypotension, after standing quietly for 2 to 5 minutes.
>
> Agents from all of the five major classes of antihypertensive medications are shown to decrease blood pressure. Diuretics and beta-blockers have consistently been shown to decrease morbidity and mortality in the treatment of hypertension and should be considered first-line therapy. Diuretics should be used in low to moderate doses. Alternatively, clinicians may consider alpha-blockers, angiotensin converting enzyme inhibitors, and calcium channel blockers (CCBs) as well as other medications as therapy for selected pre-existing conditions. Clinicians should consider cost where therapeutic effect is equal, and to maximize compliance, should choose medications that keep regimens simple.
>
> If the blood pressure continues to be elevated, clinicians may consider choosing one of the strategies that have proven effective in the treatment of hypertension:
>
> …

Figure B.1: Part of a free-text guideline for screening, treating and monitoring patients with hypertension. This guideline is uploaded into the DeGeL library and stored as a source guideline.

2. The semi-structured representation level:

```
<abort-condition>
  <semi-structured>
    Patients should be seen within 1 to 2 months after the
  initiation of therapy, to determine adequacy of hypertension
  control, and presence of adverse effects. If the blood
  pressure is not controlled (the aim is to decrease BP to less
  than 140/90), clinicians may consider choosing to increase the
  dose of the original medication if no side-effects are present
  during the period of treatment.
  </semi-structured>
<abort-condition/>
```

Figure B.2: A semi-structured representation, showing a hybrid-Asbru *abort condition* marked by the domain expert. The text within this XML element was marked by the expert using the Uruz tool, from a free-text source.

3. *The semi-formal representation level:*

```
<abort-condition>
  <semi-formal>
    <logic-expression     name="HTN     control     adequacy     "
  operator="and">
      <operands>
        <logic-expression name="uncontrolled BP" operator="or">
          <operands>
            <simple-expression        name="systolic        BP"
          stdName="LOINC" stdKey="8512-6">
              <text>systolic blood pressure > 140</text>
            </simple-expression>
            <simple-expression        name="diastolic        BP"
          stdName="LOINC" stdKey="8500-1">
              <text>diastolic blood pressure > 90</text>
            </simple-expression>
          </operands>
          <time-annotation><now/></time-annotation>
        </logic-expression>
        <not-expression name="ACE inhibator side effects">
          <simple-expression   name="Creatinine   side   effect"
        tdName="Idan.TAKB.HTN" stdKey="14.25">
            <text>Creatinine_Clearance IS DECREASING</text>
            <time-annotation><time-range>
                <starting-shift>
                  <minimum   time-unit="days"   include-limit-
                value="true">0</minimum>
                </starting-shift>
                <duration>
                  <minimum time-unit="months">1</minimum>
                  <maximum time-unit="months">2</maximum>
                </duration>
              </time-range><reference>
                <plan-state-transition        state="activated"
              instance-type="first" direction="enter">
                <plan-pointer id="ACE-Inhibator-Administration"
              DeGeLSite="BGU"/>
                </plan-state-transition>
              </reference>
            </time-annotation>
          </simple-expression>
        </not-expression>
      </operands>
    </logic-expression>
  </semi-formal>
```

Figure B.3: A semi-formal representation in the Hybrid-Asbru ontology.

The Asbru *abort condition* marked-up by the medical expert and shown in Figure A.2 was semi-formalized (by the medical expert, possibly in collaboration with a knowledge engineer) using a graphical, Asbru-specific editing tool, which is a part of Uruz. The output of the graphical editing tool is an XML file that adheres to an XML schema that describes the Asbru semi-formal representation level. Part of that XML file (namely, only the *abort condition* knowledge role) is shown here. The XML file shown above specifies the temporal expression inherent in the semi-structured abort condition as an AND/OR tree. Note that:

(1) Components of the AND/OR tree, such as "*systolic blood pressure > 140*" are still in free text, although their logical role is now explicit;

(2) Terms within the expressions, such as "*systolic blood pressure*" are specified using standardized vocabularies, in this case, LOINC 8512-6. The use of standardized terms supports future grounding to the terms of any site-specific patient database

(3) Several of the terms used in the expressions, such as "*creatinine side effect*", are in fact abstract terms defined in the Idan temporal-abstraction knowledge base (TAKB) and indexed by the context of the hypertension (HTN) domain

4. The formal representation level:

```
<abort-condition>
  <formal>
    <LogicPredicate operator="and">
      <ExistsPredicate><QuerySpec><From>
        <Range concept="Idan.TAKB.HTN.bp_state" as="bp"/>
      </From><When><TimePointConstraint relation="equal">
        <TimePoint>      <Start of="bp"/></TimePoint>
        <TimePoint>      <Now/></TimePoint>
      </TimePointConstraint></When>
      <Where><ValueConstraint relation="equal">
          <Value of="bp"/><ValueLiteral is="uncontrolled"/>
        </ValueConstraint>
      </Where></QuerySpec></ExistsPredicate>
      <NotPredicate><ExistsPredicate><QuerySpec><From>
          <Range          concept="ACE-Inhibitor-Administration"
        orderBy="start" top="1" as="ace"/>
          <Range    concept="Idan.TAKB.HTN.Creatinine_Clearence"
        as="cc"/>
        </From><When><TimePointConstraint relation="precede">
            <TimePoint><Start of="ace"/></TimePoint>
            <TimePoint><Start of="cc"/></TimePoint>
          </TimePointConstraint>
          <TimeMeasureConstraint relation="graterThanOrEqual">
            <TimeMeasureGap>
              <TimePoint><End of="cc"/></TimePoint>
              <TimePoint><Start of="cc"/></TimePoint>
            </TimeMeasureGap>
            <TimeMeasureLiteral quantity="1" timeUnit="month"/>
          </TimeMeasureConstraint>
          <TimeMeasureConstraint relation="lessThanOrEqual">
            <TimeMeasureGap>
              <TimePoint><End of="cc"/></TimePoint>
              <TimePoint><Start of="cc"/></TimePoint>
            </TimeMeasureGap>
            <TimeMeasureLiteral quantity="2" timeUnit="month"/>
          </TimeMeasureConstraint>
        </When><Where><ValueConstraint   relation="equal"><Value
      of="cc"/><ValueLiteral is="decreasing"/></ValueConstraint>
        </Where></QuerySpec>
    </ExistsPredicate></NotPredicate></LogicPredicate>
  </formal>
<abort-condition/>
```

Figure B.4: A formal representation in the Hybrid-Asbru ontology.

The Asbru *abort condition* semi-formalized by the medical expert (with the assistance of the knowledge engineer) and shown in Figure A.3, was formalized by an Asbru knowledge engineer, using another graphical editing tool, which is also a part of Uruz. The hybrid-Asbru formalization tool is specific to the hybrid-Asbru ontology, as well as to the ontology of the mediator that is being used to access patient data. In this case, the data-access ontology used is that of the *Idan* temporal-abstraction mediator, which can answer queries regarding either raw data or the abstractions derivable from them, using domain-specific knowledge from the temporal-abstraction knowledge base (TAKB) (see Section 3.4). The XML file shown adheres to the XML schema of formal Asbru and to the schema of expressions in Idan's query language. Note that

(1) Components of the semi-formal AND/OR tree that were in free text, such as "Creatinine clearance is decreasing," are now represented by a constraint on a formal term (decreasing) from the Idan temporal-abstraction knowledge base (TAKB);

(2) Abstractions of data mentioned in the semi-formal representation, such as "systolic blood pressure," are now specified using concepts from the Idan temporal-abstraction knowledge base (TAKB), indexed by the context of the hypertension (HTN) domain.