

# HELEN, a Modular Framework for Representing and Implementing Clinical Practice Guidelines

S. Skonetzki<sup>1</sup>, H.-J. Gausepohl<sup>2</sup>, M. van der Haak<sup>1</sup>, S. Knaebel<sup>2</sup>, O. Linderkamp<sup>2</sup>, T. Wetter<sup>1</sup>

<sup>1</sup>Department of Medical Informatics, Heidelberg University Medical Center, Germany

<sup>2</sup>Department of Neonatology, Heidelberg University Medical Center, Germany

## Summary

**Objectives:** In order to implement clinical practice guidelines for the Department of Neonatology of the Heidelberg University Medical Center we developed a modular framework consisting of tools for authoring, browsing and executing encoded clinical practice guidelines (CPGs).

**Methods:** Based upon a comprehensive analysis of literature, we set up requirements for guideline representation systems. Additionally, we analyzed further aspects such as the critical appraisal and known bridges and barriers for implementing CPGs. Thereafter we went through an evolutionary spiral model to develop a comprehensive ontology. Within this model each cycle focuses on a certain topic of management and implementation of CPGs.

**Results:** In order to bring the resulting ontology into practice we developed a framework consisting of a tool for authoring, a server for web-based browsing, and an engine for the execution of certain elements of CPGs. Based upon this framework we encoded and implemented several CPGs in varying medical domains.

**Conclusions:** This paper shall present a practical framework for both authors and implementers of CPGs. We have shown the fruitful combination of different knowledge representations such as narrative text and algorithm for implementing CPGs. Finally, we introduced a possible approach for the explicit adaptation of CPGs in order to provide institution-specific recommendations and to support sharing with other medical institutions.

## Keywords

Clinical practice guidelines, representation, acquisition, implementation, local adaptation, execution, decision support

Methods Inf Med 2004; 43: 413–26

## 1. Introduction

Within the past two decades, hundreds of clinical practice guidelines (CPGs) have been developed, aiming at assisting clinical practitioners and patients in the decision about the appropriate treatment in specific clinical circumstances. However, CPGs often fail to affect clinical practice [1-3]. Therefore systematic research is done to find out why physicians do not follow CPGs [4-7]. Besides, several researchers deal with identifying strategies for improving physicians' compliance with the recommendations of the CPG [8-12]. Some of these aspects are:

- **Consideration of different needs caused by different roles and experiences of the users** [13, 14].

This is necessary in order to avoid users finding information on the wrong level of experience, who therefore become more and more frustrated by using the CPG.

- **A flexible implementation strategy based upon the specific focus of the CPG and individual barriers of the target care setting.**

This is necessary due to the wide variety of topics addressed within CPGs (e.g., standardization of diagnostic procedures, assistance during planning of certain therapies, or reducing avoidable risks) and to the fact that barriers in one setting may not be present in others (e.g., organizational constraints, lack of resources) [4, 15].

- **The explicit adaptation of CPGs.**

This is necessary to gain user acceptance and to provide recommendations not only specific to the patient, but also to the clinical setting and to the specific institution in which the treatment takes

place. Local recommendations have to be made due to the fact that for certain illnesses different prevalences should be taken into account, and because every health institution has its own resources and experiences [6, 16-19]. Furthermore the explicit adaptation invites future users to actively participate during an essential part of the development process.

- Providing **patient-specific recommendations** during patient encountering is often recommended but only possible through individually tailored computer-based decision support systems [20 to 22].
- **Rapid prototyping** for effective authoring and testing during the development process [23]. This is a possible way to simulate the usability and effect of the recommendations and to systematically test and refine the achieved evidence foundation.

The HELEN-Project was established to face these problems and to systematically introduce CPGs at the Department of Neonatology of the Heidelberg University Medical Center.

The main focus of this project is to attempt the following:

- to represent existing CPGs from various sources by preserving their current presentation format and structure;
- to provide an editor that easily allows our development team to enter existing CPGs as narrative text and/or as structured algorithms and to apply necessary changes during adaptation in an auditable manner;
- to generate a web-based view for browsing the adapted CPG and the underlying CPG source;

- to automatically generate computer-based decision support components for providing patient specific recommendations, reminders, or alerts based upon this CPG also during patient encountering;
- to facilitate sharing of CPGs with other cooperating institutions, even if these institutions are only interested in parts of the text or algorithms or in the case that they wish to apply minor changes to the CPGs (e.g., different critical values or other preferences for certain therapy options).

Different approaches and tools have been presented in the literature for representing CPGs. Some of them focus initially on the transfer from paper-based to computer-based formats in order to handle and distribute CPGs in an easy way [24-26]. These types of approaches mainly improve the accessibility to the guideline itself.

Other approaches such as GEM additionally focus on computer-interpretable elements for appraisal issues, eligibility determination or on presenting certain parts as single-step guidelines by means of decision variables. [27-29]. With these kinds of approaches it is hard to represent complex plans for diagnosis, therapy or patient management. As a complement to GEM the GLARE project developed a comprehensive representation of clinical algorithms with respect to several aspects of execution such as sequencing, handling of concurrent tasks and errors [30]. Moreover within the GLARE project a flexible engine for executing encoded CPGs was developed. The main problem of GLARE as of the SIEGFRIED representation is the internal storage of the encoded CPG within a relational database system without export capabilities [31]. Therefore, sharing CPGs with other institutions is still problematic.

The Guideline Interchange Format (GLIF) symbolizes an intermediate concept between narrative CPG formats and computer executable formats [32]. GLIF mainly focuses on sharing CPGs among different institutions. Until version 3 the expressiveness of the language used to formulate decision criteria was therefore rather vague [33]. Furthermore GLIF 2.3 encoded CPGs are only executable by extensions as

shown by [34], or [35]. To overcome this weakness following versions of GLIF (3.x) were extended by several new constructs which allow a formal definition of decision criteria, action specification and patient data. Therefore a more expressive object-oriented expression language (GELLO), as well as medical vocabularies (UMLS), a standard medical data model (HL-7 RIM), and elements of the "virtual medical record" (vMR) are now included [36, 37].

As an extension to the GLIF concept, P-CAPE refined amongst others elementary decision criteria resulting in a computer-executable CPG representation, thus overcoming this weakness. Additionally P-CAPE includes a high-level tool for entering guideline parameters in the form of algorithms. The output of this tool are MUMPS code and data structures used within the Brigham and Women's Hospital Information System (BWH) [38]. The core elements of P-CAPE "Navigator" and "Notifier" were specially implemented for the use within the BWH alerting system [39].

Other published concepts for representing CPGs are PRODIGY, Asbru, G-CARE, Opade, Prestige, and PROforma [40-44]. All of these concepts are custom tailored to the requirements of special classes of CPGs, such as clinical protocols, single-step decision rules or reasoning about temporal abstraction data. Therefore they often form additional barriers for sharing CPGs across different institutions.

Therefore EON developed a flexible approach for modeling different classes of CPGs (e.g. one-shot decision, complex multi-encounter CPG, clinical trial protocol). The EON approach is based upon a common conceptual model of time and patient data [33, 45]. The author can choose among different modeling primitives that can be mixed and matched in order to construct processes of decision-making, action sequencing and data interpretation as needed by the different classes of guidelines. In addition both an authoring and execution environment were developed to complete this approach [46]. Unfortunately this modular and flexible approach is not able to deal with narrative guideline elements (e.g., for explanation or educational purposes) nor does it include information needed by

systematic appraisal. Because it is not in the scope of this article we would like to refer to [www.openclinical.org](http://www.openclinical.org) where a comprehensive overview with detailed descriptions of many CPG representation systems can be found. Additionally systematic comparisons of certain features and the interconnections between different guideline representation models can be found in [22, 47], and [48].

## 2. Methods – Development of the Knowledge Repository

Considering the amount of proven representation concepts for CPGs, we initially aimed at using one of these. We made several attempts to rearrange or extend such representation schemes in order to fulfill the requirements of the German Medical Societies and the German Guideline Clearinghouse to which our CPGs have to be submitted [49]. These modifications included a.o. a systematic labeling of the underlying medical evidence for each statement and structuring the considerable amount of additional information which is needed during appraisal, classification, and clearing process of the CPG. Especially our attempts to rearrange and insert new elements without destroying the core system (thus allowing us to benefit from future developments of the underlying system) led to really complex and sometimes confusing tools for our authoring team. Therefore we decided to develop a new representation scheme custom-tailored to the needs within our setting. However we tried to build on proven concepts instead of developing new ones as much as possible.

Considering the aim of developing a flexible, shareable, and computable description of the CPG, we identified the concept of ontologies to be useful. An ontology is a formal specification of terms in the target domain and relations among them [50]. This formal representation can be used to share a common understanding of the structure of information among people or software agents [51].

Within an ontology terms are stated as single elements such as frames or objects

with their associated slots or attributes. These elements are usually structured within a hierarchy. The relations that are used within the hierarchy must be specified by the author (e.g., ‘is-a’, ‘consist-of’, or ‘is-part-of’).

As a methodology for building the ontology we used the process model of the Model-based Incremental Knowledge Engineering Process (MIKE) [52]. This model is based upon the Spiral Model of Software Development as presented in [53].

The main focus of this model lies in the cyclic processing of the following four tasks:

- analysis of requirements
- design or refinement of the ontology
- implementation of an initial knowledge acquisition tool and its extensions
- evaluation by entering examples and test usage of the encoded information.

Within the HELEN project we have chosen a development process with six cycles. The first three cycles focus on certain topics of the domain such as management/appraisal, document structure, and algorithm, whereas the last three cycles are dedicated to the development of the authoring tool, the guideline viewer and the execution engine.

## 2.1 Analysis of Requirements

Each cycle starts with a careful collection of the specific requirements. As the first cycle focuses mainly on management aspects we started with an initial concept for the ontology. This concept is based upon requirements derived from the literature about tools for critical appraisal (e.g., AGREE, SIGN, NGC, AWMF) [54-58], the Guideline Element Model (GEM) [59], and from our own requirement analysis at the Department of Neonatology [60].

The second cycle focuses on identifying elements for preserving the document structure of existing guidelines as published by e.g., SIGN [61] and DEGAM [62]. This was necessary to enable the mapping of existing text-based CPGs. The third cycle was dedicated to the requirements of a computable algorithm format according to a.o. EON [33].

One of the intentions of the HELEN-Project was to also include medical domain

experts during the development of computable algorithms. However, one has to consider that medical domain experts usually subconsciously evoke scripts of actions rather than explicitly think in formal rules, necessary for inference machines used within common decision support (e.g., [63]). Therefore, we conceived representation elements based upon concepts as used by procedural programming languages such as C or Pascal.

Within the last three cycles we used standard software engineering methodologies for collecting and analyzing requirements of the developed tools and for recording them within a functional specification. By implementing these tools we also applied systematic refinements to the ontology such as modifications to the forms for entering the CPG (authoring tool), an additional labeling for user-specific presentation (*Guideline Viewer*), or a more detailed description of certain elements for automatic processing (*Guideline Execution Engine*).

## 2.2 Design of the Ontology

After the exploratory process applied for the analysis of requirements we chose a top-down development process for designing the ontology according to [64]. Therefore we started with the definition of the most general concept of each cycle (here guideline, pragmatics, knowledge modules, and adaptation). Subsequently, we refined this concept by means of a hierarchy using ‘is-a’ relations. This kind of relation represents the fact that every subclass inherits the descriptive attributes from its superclass. Consequently, instances of the subclasses can be stated as a specialization of instances of the superclass.

In order to achieve more expressiveness of the ontology we used specialized attributes which refer to other instances based upon the ‘consist-of’ relation. These attributes are labeled with a name followed by ‘\_I’ (e.g. *Developer\_I* which could be interpreted as: ‘developer consists of a list of people, groups or organizations’). To preserve the ability to merge with other ontologies the prefix ‘HELEN\_’\* as a kind of namespace was used for all classes that are specialized to the context of CPGs. For classes not spe-

cialized for the representation of CPGs we used standard names without any prefixes.

Wherever possible values for attributes were based upon existing classification or indexing schemes such as provided by NGC or other stakeholders [57, 65].

After each development cycle we checked the resulting ontology according to the methodology as published in [66]. This task is necessary for ensuring the correctness regarding:

- the correct use of the ‘is-a’ relation within the ontology and ‘consist-of’ relation within specialized attributes;
- avoiding cycles in the class definition like A has a subclass B and at the same time B is a superclass of A;
- analyzing siblings within the hierarchy;
- consistent use of classes, property values and instances.

As a result of the iterative processing of the six cycles we obtained an ontology with five major trees containing 83 classes and over 200 descriptive attributes. As a main class we introduced the class *HELEN\_Guideline* to access the guideline and to instantiate the other classes. *HELEN\_Knowledge\_Module* represents all the content produced during the development process (e.g. text, graphics or algorithm). The life-cycle of the guideline is reflected by the next two classes. *HELEN\_Adaption* contains the localized content produced during the adaption process plus a documentation about this process. *HELEN\_Pragmatics* is a special class containing the necessary accompanying documents of a guideline, the documentation about the development process, and also a set of additional information needed for the clearing process. *HELEN\_Modules* contains several classes which can be understood as complex data types used to organize all the content which is linked with the major classes. These classes are not directly accessible as top level classes since they can only be instantiated from one of the major classes or their components.

\* Names written in italic such as *Guideline Viewer* refer to a developed tool within the HELEN framework, names written in italic and conjunct by the prefix *HELEN\_* such as *HELEN\_Decision* refer to an entity of the developed ontology.

```

<HELEN_Cycle_Diagram id="V_2003_D_04844">

  /* Elements for embedding within superior structures
  <HELEN_Name>Supportive Phototherapy until laboratory results are available</HELEN_Name>
  <HELEN_Next_Step>Instance(V_2003_B_04776 of Cls(Evaluate_automatic))</HELEN_Next_Step>
  <HELEN_Fail_Step>Instance(V_2003_B_04778 of Cls(HELEN_Message))</HELEN_Fail_Step>

  /* Cycle control
  <HELEN_cycle_predelay>...</HELEN_cycle_predelay>
  <HELEN_Cycle_postdelay>Instance(V_2003_D_04906 of Cls(wait_until))</HELEN_Cycle_postdelay>
  <HELEN_Cycle_stop_criterion>Instance(V_2003_D_04908 of Cls(EQ))</HELEN_Cycle_stop_criterion>
  <Guideline_Reference>Instance(V_2003_D_04886 of Cls(Text))</Guideline_Reference>

  /* Cycle steps
  <HELEN_Cycle_Entity>Instance(V_2003_D_04850 of Cls(HELEN_Start_Step))</HELEN_Cycle_Entity>
  <HELEN_Cycle_Entity>Instance(V_2003_D_04855 of Cls(Set_Variable))</HELEN_Cycle_Entity>
  <HELEN_Cycle_Entity>Instance(V_2003_D_04868 of Cls(Evaluate_automatic))</HELEN_Cycle_Entity>
  <HELEN_Cycle_Entity>Instance(V_2003_D_04875 of Cls(Therapy_or_prescription))</HELEN_Cycle_Entity>
  <HELEN_Cycle_Entity>Instance(V_2003_D_04880 of Cls(HELEN_Diagnose))</HELEN_Cycle_Entity>
</HELEN_Cycle_Diagram>

  /* Definition of the cycle postdelay: "wait after each cycle 24 hours"
  <wait_until id="V_2003_D_04905">
  <HELEN_Name>24 h</HELEN_Name>
  <HELEN_wait_until_days>1</HELEN_wait_until_days>
  <HELEN_wait_until_hours>0</HELEN_wait_until_hours>
  <HELEN_wait_until_minutes>0</HELEN_wait_until_minutes>
  <HELEN_wait_until_seconds>0</HELEN_wait_until_seconds>
  </wait_until>

  /* Definition of the stop criterion: "stop if new laboratory data for this patient arrive"
  <EQ id="V_2003_D_04908">
  <HELEN_Name>inspected laboratory results available</HELEN_Name>
  <HELEN_Binary_Arithmetic_Value_A>Instance(V_2003_D_04909 of
  Cls(HELEN_Variable))</HELEN_Binary_Arithmetic_Value_A>
  <HELEN_Binary_Arithmetic_Value_B>Instance(V_2003_D_04912 of
  Cls(Get_Patient_Case_Number))</HELEN_Binary_Arithmetic_Value_B>
  </EQ>

```

**Fig. 1** Definition of a therapy cycle processed until new data arrives

The full annotated ontology and some examples for the usage can be downloaded at: <http://www.med.uni-heidelberg.de/mi/research/dss/helen/helen.htm>.

### 2.3 Implementation of the Initial Knowledge Acquisition Tool

For the knowledge acquisition task we chose the platform-independent and configurable Protégé-2000 toolset [67]. This tool consists of an ontology-based knowledge model, a knowledge acquisition tool, and a process model similar to our own approach. The internal format for the knowledge model is MODEL, a frame-like representation language which is an extension of the CLIPS language. Also part of this toolset is a form generator which allows a flexible and semi-

automatic configuration of entry forms based upon the encoded ontology. This feature especially meets our requirements to reduce the effort for the necessary development of a knowledge acquisition environment during each cycle.

In addition, Protégé-2000 supports several formats such as RDF, Standard Text File, XML as well as the ability to export the acquired knowledge base to an external JDBC Database. This allows a smooth transition from the acquisition to the test use of the encoded knowledge.

### 2.4 Evaluation of the Ontology

A detailed test after each cycle is needed in order to verify whether the target set before has been met. Since the first three cycles focus on representation issues, we

regularly checked the ability to encode all information provided by published guidelines from DEGAM and SIGN. Furthermore we tested whether it was possible to carry out a systematic appraisal based upon the encoded information. Therefore we used the checklist for methodological quality of guidelines as published by [68].

Since there is no normative framework to determine the necessary components and required expressiveness of the computable algorithm format, we derived elementary features both from published descriptions and examples of the above mentioned models as well as from comprehensive comparisons of computer-interpretable guideline models such as [47] and [48]. These comparisons deal with the type, the level of detail, and the number of given primitives used for assembling the algo-

rithms. By examining these, we found similar primitives for representing several types of actions, decisions, branching, and nested subplans. Other dimensions that were compared deal with the expressiveness of the implemented language used for defining decision criteria, setting goals, achieving data interpretation/abstraction, and managing the control flow. Since there is no objective criteria as to which type of expression language is superior to the other we examined HELEN's ability to encode several types of clinical actions (choosing a diagnostic procedure, refining a diagnosis, planning and carry out of an appropriate treatment) and how to use these encoded expressions within several modes of execution (decision support, workflow support,

watchdog, or reminder) during several tests.

During the 4th cycle we customized the Protégé-2000 authoring environment and implemented, if necessary, additional entry-wizards. This task itself checked the consistency and completeness of the ontology again and additionally, it allowed to test the ability to encode CPGs in a user-friendly manner. Within the 5th cycle we developed a web-based tool to browse encoded CPGs. This task checked the correctness of the encoded information with regard to the import of the XML and XSD files which are produced by the authoring environment. We also verified the correct presentation of guideline documents as encoded and the complete presentation of information as

needed for appraisal. The last cycle was dedicated to the implementation of the execution engine. Here we carried out a systematic check of the implementation of each element that is used within an algorithm. Afterwards we checked the correctness of encoded Boolean and arithmetic expressions. Another check was aimed at the correct processing of encoded test algorithms by means of an execution as expected. Finally we applied medical algorithms to test cases derived from real cases at the Department of Neonatology. Some examples about the CPGs which were used for this evaluation can be found under section 4 of this paper. In the next section we will give a short overview of the developed tools of the framework and how they interact.

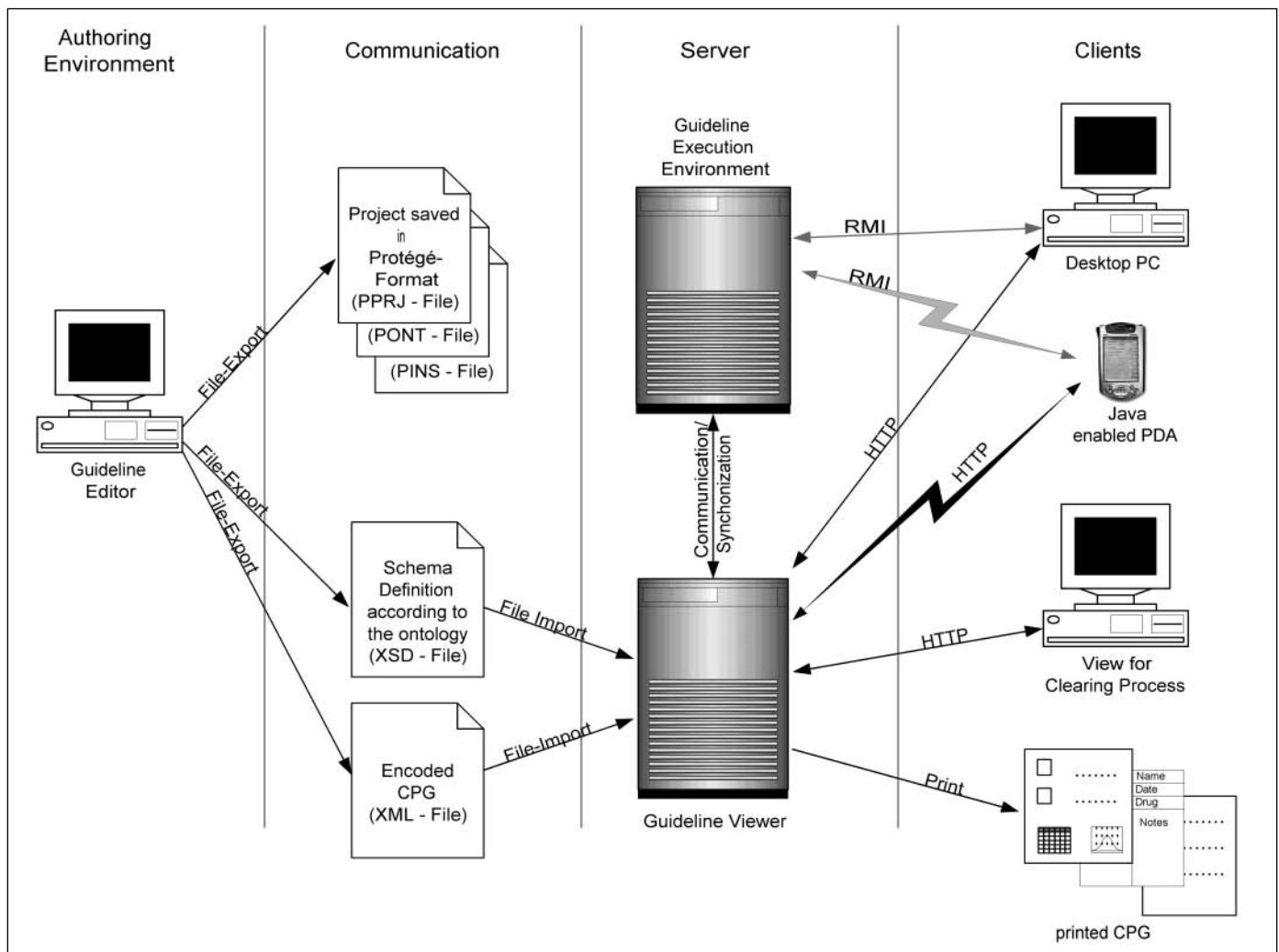


Fig. 2 Major tools for authoring and execution of the HELEN framework and their communication via file, RMI or HTTP

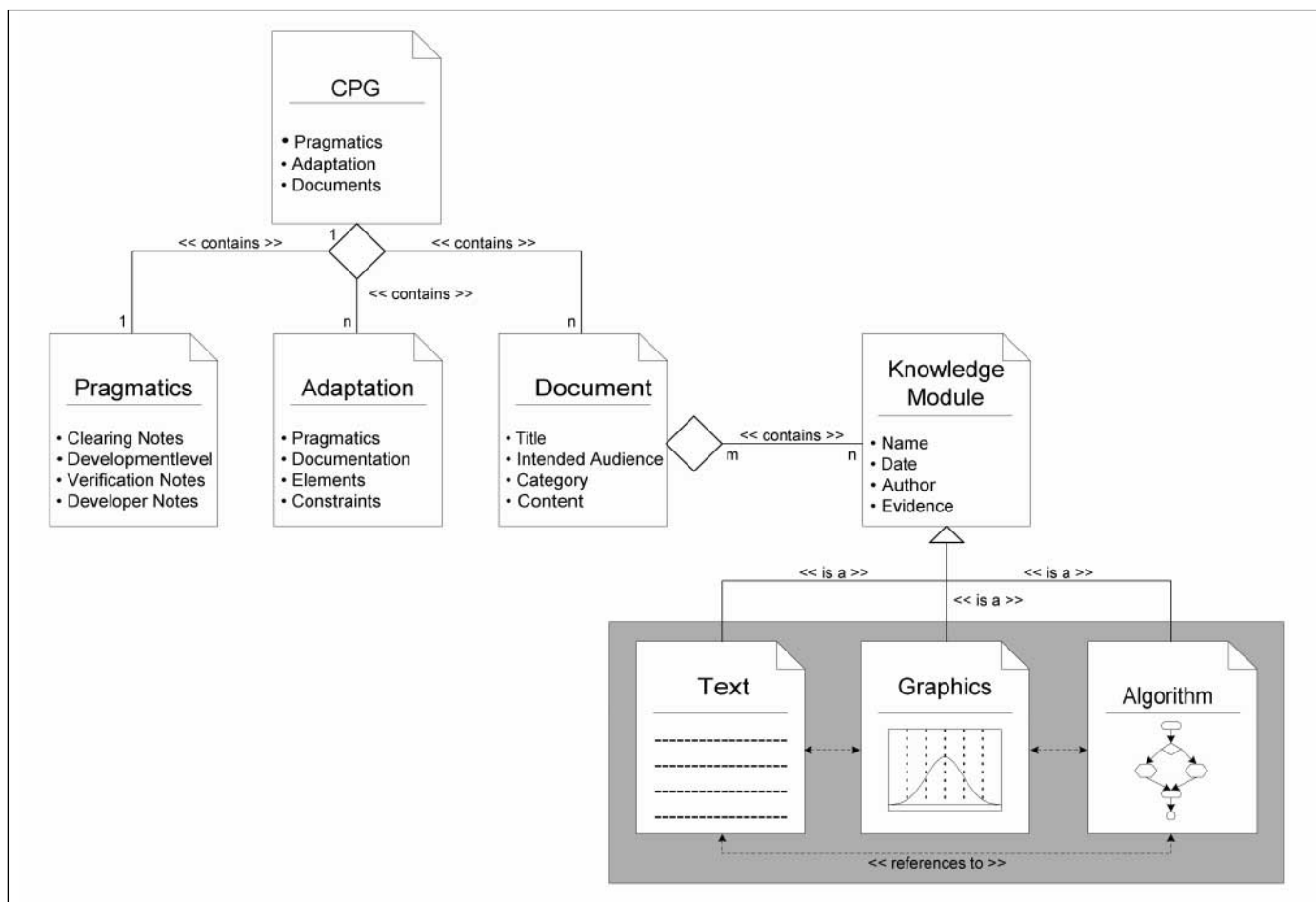
### 3. Architecture of the System

Based upon an assumed life-cycle which consists of authoring a national or international CPG, systematic appraisal, adaptation for local use, and implementation in clinical practice we identified an initial set of three tools (Fig. 2). The authoring environment is called the *Guideline Editor* which is a Protégé-2000 based tool for encoding CPGs according to the *HELEN Ontology*. This *Guideline Editor* helps the authors in structuring, entering, and modifying recommendations during the initial development or adaptation. The result of these processes can be stored either in the native format of Protégé-2000 or as XML-Schema [69]. In order to handle XML/XSD files and other accompanied files such as pictures of the included algorithm in an

easier way, all data can be stored within a zip-archive.

The second tool of this framework is a WWW-Server (*Guideline Viewer*) for browsing encoded CPGs via a standard web-browser. For this purpose we developed a Java Servlet processed within an Apache-Tomcat web server [70, 71]. This servlet reads XML-encoded CPGs from the earlier mentioned zip-archive and produces different views for presenting the content either to the appraisal team or to different groups of health care professionals according to their specific needs. The *Guideline Viewer* naturally focuses on the text-based elements of the CPG, but it is also possible to browse images of included algorithms and to request further information on certain elements such as viewing sub-algorithms, specifications of recommended therapies or browsing through

associated background material. For maintenance purposes, the *Guideline Viewer* additionally provides a web-based interface for uploading and managing CPGs on the server. It also has the capability to synchronize active CPGs with the third tool of our framework, i.e. the *Guideline Execution Engine*. This is necessary to ensure that all active tools carry out the same version of the CPG thereby avoiding mistakes. As a complementary tool to the *Guideline Viewer*, the *Guideline Execution Engine* is specialized in traversing and processing included algorithms for certain patients during encountering. Therefore, the engine administrates additional information necessary for executing CPGs, amongst others data about registered users (e.g., user name, position, role, availability, and period of service). The *Guideline Execution Engine* also communicates via



**Fig. 3** Modular document model of the HELEN Ontology. Beside pragmatics (e.g., information for systematic appraisal) each CPG contains at least one guideline document which is composed of knowledge modules in order to represent the recommendations of the guideline.

RMI (Remote Method Invocation) with several clients for retrieving data, providing choices to the users, and for informing them about necessary tasks. It can also invoke the *Guideline Viewer* to present associated background material suitable for the current task of the algorithm.

In order to ensure the platform independency when connecting the *Guideline Viewer* and the *Guideline Execution Engine* to a hospital information system (HIS) an additional custom-tailored mediator component is necessary. This mediator component is responsible for communicating variable requests between the guideline tools and the local HIS as well as for receiving ADT-data and for generating trigger events for the *Guideline Execution Engine* to activate, abandon, or manipulate the guideline flow control. Despite the explicit task sharing, the software has proved to be a framework which is easily to customize to different scenarios of use by omitting a tool, replacing one with a more sophisticated version, or by introducing a new tool with additional features.

### 3.1 Guideline Authoring

In matching the life-cycle model the *Guideline Editor* aims at assisting authors during the initial development and adaptation. Therefore the editor allows authors to create new CPGs, to edit existing CPGs, and to store them in Protégé or XML format. A suitable set of additionally developed plug-ins and widgets assists the authors in formulating guideline recommendations as well as in checking the completeness, the syntactic correctness, and within narrow confines also the semantic correctness of entered values.

One of the advantages of the underlying Protégé-2000 environment in combination with the developed ontology is the fact that the developer team can choose a top-down, as well as a bottom-up or a middle-out approach to develop the guideline. By using the bottom-up approach, the authors generate suitable knowledge modules which represent the evidence found in the literature. By structuring these knowledge modules, documents of the CPG are generated. In combination with the pragmatic's sec-

tion and in certain cases with the adaptation section these documents form the complete CPG (Fig. 3).

Alternatively the authors start with the top-level class *HELEN\_Guideline* and instantiate from this class the necessary documents in a top-down approach. During the following step each document must be refined by the use of suitable knowledge modules. Finally the development team completes the pragmatics section with information about the development process and the life-cycle of the guideline. The middle out approach starts with the most important and ends with the least important elements. Here the authors instantiate classes of *HELEN\_Documents* and *HELEN\_Knowledge\_Module* which they identify to be important and form the CPG by adding missing parts.

In order to support the authors during encoding and to provide them with an easy to use tool we customized the pre-configured Protégé entry forms for each class. For certain slots of these forms (attributes of the classes) we used specialized widgets as provided by the web site of Protégé for viewing images or for preventing the author from repetitive mechanic routine or fault prone tasks such as entering the date and the author for each created or edited instance [72].

Special attention is given to the customization of the entry forms for the *Knowledge Modules*. As mentioned earlier we distinguish between text, graphics and algorithms.

Beside standard text formats such as ASCII or HTML, the HELEN tools support a slightly modified version of HTML called *HELEN\_HTML* in order to repre-

5.1.2 SIDE EFFECTS (type=Text, name=V\_2003\_B\_ABS\_05547)

Title of the text element  
5.1.2 SIDE EFFECTS

Aim of the text element  
Consider most most frequent psychostimulant side effects as reported in short term studies like insomnia, reduced appetite, abdominal pain, headache and dizziness.

Intended audience  
Experienced physician  
Physician

The most frequent psychostimulant side effects in short term studies are insomnia, reduced appetite, abdominal pain, headache and dizziness; less frequently, anxiety, irritability, or proneness to crying <sup>115, 116</sup>. Some side effects may, on further analysis, represent part of the underlying ADHD/HKD condition and be noted in the pre-treatment or placebo states. Some common psychostimulant treatment-related difficulties and suggested responses are listed in Table 1 <sup>103, 106, 116, 117</sup>.

**Table 1: Psychostimulant side effects and suggested management options**

Side effect	Management options
Anorexia, nausea, weight loss	Monitor carefully, give medication with meals, give calorie supplements
Growth concerns	If significant (rare in long term) or causing parental anxiety, attempt weekend or vacation medication breaks

```

</td>
<td valign="top" width="66%">
<p>
<font face="Arial, Helvetica, sans-serif">Reduce or discontinue medication (discontinue if thought disorder or psychosis suspected &#8211; this is rare)</font>
</p>

```

Good Practice Point  
- When psychostimulants are first introduced and being titrated, regular contact between the family and the clinician is important to deal with questions and to make any necessary adjustments.  
- Blood testing should be carried out at the discretion of the supervising clinician and only when clinically indicated.

Statement Of Evidence  
Ila

Grade Of Recommendation  
B

HELEN Date  
Mon Jun 09 13:29:28 CEST 2003

MIME-Type  
HELEN\_HTML

HELEN Module Based On  
HELEN Author  
SIGN Executive Authoring Team

**Fig. 4**  
Protégé form for entering a text-based knowledge module by using the *HELEN\_HTML* syntax

sent narrative text elements. This version of HTML contains three new tags for linking to the other *HELEN\_Knowledge\_Modules* within a CPG. Consequently we developed a specialized edit widget for *HELEN\_HTML* to prevent the author from manually encoding text in this format (Fig. 4).

Although HTML supports the representation of images we used a specialized knowledge module instead. This decision was made due to the fact that images could be used within non-html documents. Furthermore, as they are knowledge resources, images must directly be accessible and annotated with descriptive attributes corresponding to other *HELEN\_Knowledge\_Modules*. The acquisition form for images is the same

as shown in Figure 4 despite the fact that the editor for *HELEN\_HTML* is replaced by an image-viewer widget.

The third knowledge module represents an algorithm which can be used to illustrate the constellation of clinical actions and decisions on a high level of abstraction. Alternatively executable algorithms can be modeled by specifying sequences and dependencies between algorithmic elements on a lower level of abstraction. For editing such diagrams we used the built-in diagram widget of Protégé. This widget provides an easy to use graphical interface that allows encoding of certain parts of the formal guideline representation by means of icons such as diamonds, circles and rectangles as shown in

Figure 5. For several reasons we developed a special slot widget which produces an image file from the content of the diagram widget and stores the corresponding filename in a specialized slot. These images are beneficial during the development process for communication and presentation and are also used by the *Guideline Viewer*.

Finally we developed an XML-Schema export filter called *HELEN\_XML\_Backend* in order to integrate the *Guideline Editor* within the presented framework. This backend extends Protégé with the ability to produce an XML-Schema definition according to the ontology and to store the encoded CPG as a valid XML-file.

## 3.2 The Guideline Viewer

One basic use of encoded CPGs is viewing them via a standard web browser (Fig. 5). Therefore we developed the *Guideline Viewer* which also assists the systematic appraisal of CPGs. This is a Java Servlet which can be carried out by a servlet container such as Apache-Tomcat [71]. In order to incorporate the *Guideline Viewer* in daily routine, some elementary features beside displaying, searching and navigating must be provided. Such features address management tasks such as uploading, updating or deleting the encoded CPG. Therefore the *Guideline Viewer* comes with a separate password protected entry point which is provided via a generic URL. This URL can also be used to access the viewer by different roles such as physician, nurse, or patient. Depending on the chosen entry point the system presents a pre-selected view with a list of guidelines and included documents dedicated to the corresponding user role as encoded within the CPG. By generating such user specific views the system has also taken care of adapted documents and knowledge modules. Therefore the provided web-pages are either based on adapted documents or modules or, in case that these do not exist, on original elements. For each displayed document or knowledge module the *Guideline Viewer* also provides an *INFO*-Button. By using this button additional background information as well as the underlying original resource will be displayed.

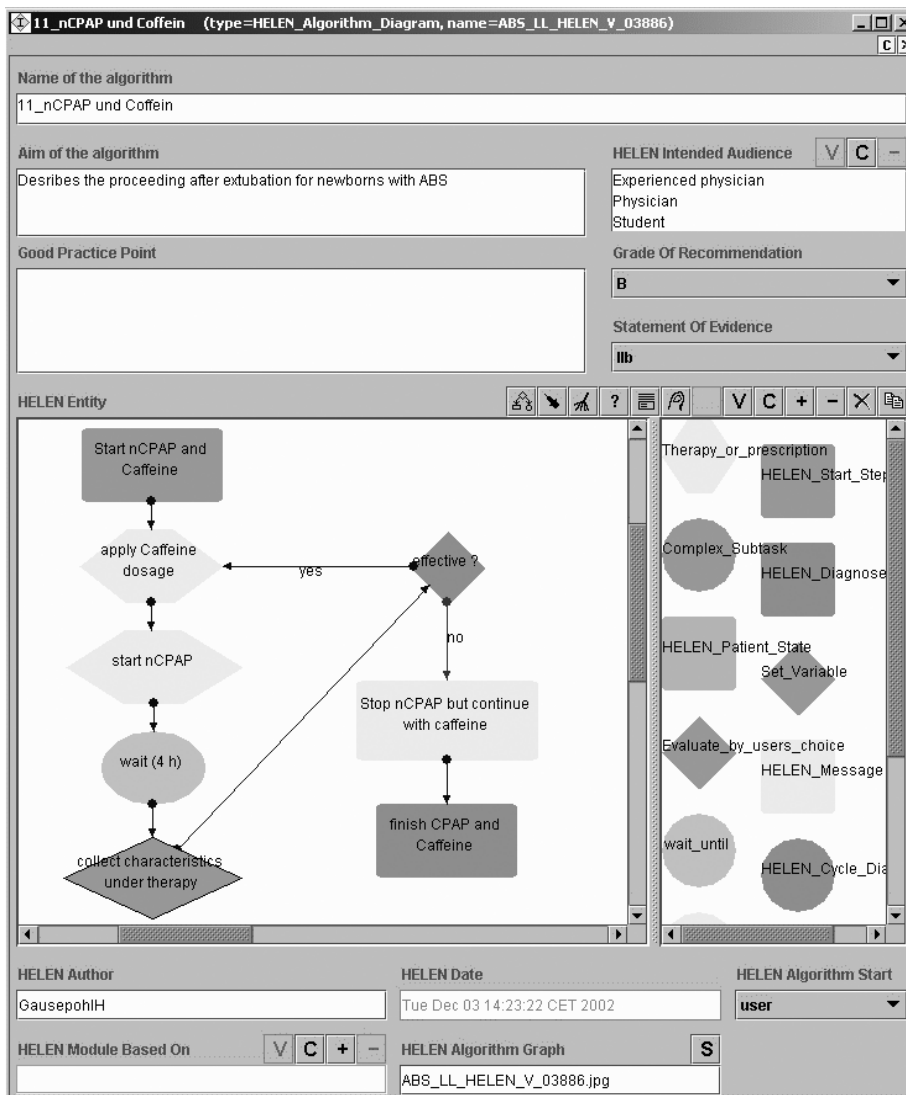


Fig. 5 Protégé form for entering an algorithm-based knowledge module by means of diamonds, circles and rectangles



In order to contribute to the framework the *Guideline Viewer* provides a dedicated entry point additionally to the already mentioned synchronization feature. This feature can be used by the hospital information system and other tools to invoke the viewer for displaying requested *Knowledge Modules*. It will be used by e.g., the *Guideline Execution Engine* providing background information about certain decisions or recommended tasks. Another use of this functionality can be found in triggering the printing of patient handouts or paper-based prescription forms via an algorithm. In return the *Guideline Viewer* can invoke the *Guideline Execution Engine* to start a displayed algorithm for a certain patient-physician combination.

### 3.3 The Guideline Execution Engine

The third tool of our framework is dedicated to the execution of algorithms as part of CPGs. The intended use of this engine is to execute one or more CPGs for a certain patient by traversing encoded algorithms and communicating with physicians, nurses, and patients about decisions, recommendations, or scheduled actions.

As a server-based tool the *Guideline Execution Engine (GEE)* is basically responsible for traversing the algorithms, sequencing clinical actions, and evaluating contained logic or arithmetic statements. Additionally, administration tasks such as reading the encoded CPG, management of information about users (e.g., username, role, responsibilities), and management of information about the patients (e.g., demographic and ADT-data) are necessary and performed by the GEE.

Due to the fact that the GEE interprets flexible programming logic it supports several modes of use which depend on the encoded algorithm. A possible application can be an algorithm for sequencing clinical actions depending on information about the patient, external trigger events, and user's choices. Other applications can be reminders, several kinds of decision support, or triggering the *Guideline Viewer* as context sensitive information resource.

**11\_nCPAP und Coffein**

```

graph TD
    A[Start nCPAP and Caffeine] --> B[apply Caffeine dosage]
    B --> C[start nCPAP]
    C --> D[wait 4 h]
    D --> E{collect characteristics under therapy}
    E --> F{effective?}
    F -- yes --> B
    F -- no --> G[Stop nCPAP but continue with caffeine]
    G --> H[finish CPAP and Caffeine]
  
```

**5.1.2 SIDE EFFECTS**

The most frequent psychostimulant side effects in short term studies are insomnia, reduced appetite, abdominal pain, headache and dizziness; less frequently, anxiety, irritability, or proneness to crying. <sup>115, 116</sup> Some side effects may, on further analysis, represent part of the underlying ADHD/HKD condition and be noted in the pre-treatment or placebo states. Some common psychostimulant treatment-related difficulties and suggested responses are listed in Table 1. <sup>103, 106, 116, 117</sup>

**Table 1: Psychostimulant side effects and suggested management options**

Side effect	Management options
Anorexia, nausea, weight loss	Monitor carefully, give medication with meals, give calorie supplements
Growth concerns	If significant (rare in long term) or causing parental anxiety, attempt weekend or vacation medication breaks
Sleep difficulties	Monitor carefully, reduce or omit late afternoon or evening medication (but note that some patients improve with added evening medication)
Dizziness and headache	Monitor carefully (check blood pressure), ensure

Fig. 6 Using a standard web browser to access documents within the CPG as provided by the 'Guideline Viewer'

Algorithms are represented as directed graphs assembled from predefined entities which are connected by 'followed-by' relations. Each algorithm strictly requires a *HELEN\_Start\_Step* and each leaf of the algorithm must be terminated by a *HELEN\_Diagnose*. Such diagnoses can – depending on their use – be understood as 'intermediate result', 'return to the

calling algorithm', 'final achievement', or 'abandoning the execution'. Beside these essential entities the author can use *HELEN\_Actions*, *HELEN\_Messages*, or *HELEN\_Decisions* to produce the required functionality. An action can be a physical examination, history taking, laboratory request, technical examination, therapy or a prescription. Decisions are used

branching steps where either the user can choose (*HELEN\_Evaluate\_by\_users\_choice*) or where the system automatically decides depending on logic statements (*HELEN\_Evaluate\_automatic*) which step to take next. Beside basic Boolean logic statements we extended the language to compare variables and constants also with the results of arithmetic expressions or specialized functions accessing certain information such as current time or date (*HELEN\_Binary\_Arithmetic\_Operator*).

Additionally, a specialized set of entities is dedicated to control the execution. These entities are obtained from procedural programming languages to generate conditional, synchronizing or wait statements. Besides, several types of cycles such as 'do-while', 'repeat-until' or cyclic statements are

possible. Another concept obtained from procedural programming is the precept utilization of variables and constants. Constants are used amongst others to achieve a maintainable set of thresholds. The systematic use of variables also provides a flexible and easily adjustable architecture for accessing information about the patient stored in external databases, hospital information systems, or other systems as long as the value is accessible by a database connection (Fig. 7). In case the value is not stored yet the *GEE* provides the ability to request the value just in time when it is needed by sending a predefined question to a stated single user or a group of professionals. For this purpose we also developed a JAVA-based communication tool available for desktop systems as well as for PDAs to as-

sist the involved health care professionals (Fig. 8). This tool can be understood as a kind of e-mail client where the user receives and answers messages. Therefore the *GEE* pushes requests, hints and all the other information on a server-based agenda.

The client receives and presents all the messages for which the user takes responsibility. Whereas some of these messages are dedicated for requesting current information about the patient or performed tasks, used to inform about suitable clinical actions or to remind about scheduled tasks, others are used to receive an approval to automatic decisions of the system or to request decisions from the user. In responding to any message the user can also suspend, re-enact or abandon the execution of a CPG for a certain patient.

```
<Therapy_or_prescription id="V_2003_D_04875">
  /* Elements for embedding within superior structures
  <HELEN_Name>Phototherapy</HELEN_Name>
  <HELEN_Next_Step>Instance(V_2003_D_04880 of Cls(HELEN_Diagnose))</HELEN_Next_Step>
  <HELEN_Previous_Step>Instance(V_2003_D_04868 of Cls(Evaluate_automatic))</HELEN_Previous_Step>

  /* Data set with responsible person and background information
  <HELEN_Message_Target_Person>Responsible Nurse - Patient</HELEN_Message_Target_Person>
  <HELEN_Message_urgency>notify this day</HELEN_Message_urgency>
  <Guideline_Reference>Instance(V_2003_D_04861 of Cls(Text))</Guideline_Reference>
  <Guideline_Reference>Instance(V_2003_D_04862 of Cls(Graphic))</Guideline_Reference>

  /* Calculation of the appropriate dosage of the recommended therapy
  <HELEN_Therapy_Dosis>Instance(V_2002_W_03343 of Cls(Mod))</HELEN_Therapy_Dosis>
</Therapy_or_prescription>

/* Definition of the calculation by using the mod-operator
<Mod id="V_2002_W_03343">
  <HELEN_Name>Calculation of the duration of the phototherapy</HELEN_Name>
  <HELEN_Arithmetic_Value_A>Instance(V_2003_D_04860 of Cls(HELEN_Variable))</HELEN_Arithmetic_Value_A>
  <HELEN_Arithmetic_Value_B>Instance(V_2002_W_03344 of Cls(Add))</HELEN_Arithmetic_Value_B>
  <Documentation>Duration of phototherapy</Documentation>
</Mod>

/* Variable for retrieving current bilirubinlevel of the children from a database
/* or if that fails by asking the user
<HELEN_Variable id="V_2003_D_04860">
  <Content_Type>Patient_Data</Content_Type>
  <HELEN_Message_Target_Person>Responsible Physician - Patient</HELEN_Message_Target_Person>
  <HELEN_Message_urgency>notify this day</HELEN_Message_urgency>
  <HELEN_Name>Current_Bilirubin_Level</HELEN_Name>
  <SQL-Select>SELECT Bilirubin_Level FROM neodat WHERE pat_id = Instance(V_2003_D_04912 of
    Cls(Get_Patient_Case_Number)) AND date = Instance(V_2003_D_04917 of Cls(Get_Current_Date));
  </SQL-Select>
  <Unit_Of_Variable>mg/kg</Unit_Of_Variable>
  <User_Input>>false</User_Input>
  <User_Question>Please specify the current bilirubinlevel of the children [mg/kg]</User_Question>
</HELEN_Variable>
```

Fig. 7 Definition of a recommended therapy and calculation of the appropriate dosage

By checking the expiry date of each message at regular intervals the *GEE* supervises the correct processing and intervenes by withdrawing or reassigning a message if necessary. This is only one of the principal tasks to ensure that the guideline can be processed in a proper form even over a longer period of time. Other tasks are necessary to ensure that minor errors within the structure of the algorithms or within formal statements may not lead to an automatic termination of the execution. So far the *GEE* is not able to correct complex errors such as missing 'followed-by' relations nor can it detect deadlocks during execution. For handling such problems we focus on preventing errors within the algorithms such as the mentioned wrong or missing 'followed-by' relations, incomplete or inconsistent specification of entities as well as certain well known typical modeling faults already during authoring of the CPG.

## 4. Applied Material

Following the objectives of our work we applied the developed framework during some projects focusing on implementing CPGs within the Department of Neonatology. As frequently concluded an effective implementation of CPGs should be based on multifaceted interventions [73]. Therefore the presented tools here constitute only one supportive element to our strategy. This strategy is based upon an initial selection of an area with urgent problems, high-risk, high-volume, or problem-prone processes. After careful collection and evaluation of relevant data we tried to determine causes. If indicated we established a guideline authoring team with members of all apparently affected groups. This authoring group is responsible for local tailoring of guideline recommendations by adapting existing or developing new guidelines. Since the *Guideline Editor* is just a tool that assists during encoding guideline recommendations the general strategy to identify, synthesize and formulate evidence is based upon proven concepts from SIGN [61] and DEGAM [62]. To involve a broad range of affected users we presented and discussed

major recommendations during regular staff meetings. These processes resulted in an initial version of the encoded CPG available by means of the *GuidelineViewer*. Closely connected to formulation of recommendations we also focused on changes regarding organization, workflow, technical equipment, and education to promote the implementation. In trying to support these changes we developed several types of algorithms executed by the *Guideline Execution Engine*. Due to the current definition of the HIS used at the Heidelberg University Medical Center, which is basically designed to assist during administrative tasks such as admission, discharge, transfer, and order entry for technical procedures, we could not practice regular data exchange concerning patient-specific clinical parameters [74]. The connection is presently limited to ADT-data and some laboratory results, but within the bounds of a separate project we also focus on a stronger integration by using scheduling, documentation and order entry functionality of the HIS.

Within the next two sections we shall present some examples of our work focusing especially on the usage of framework.

### 4.1 In-depth Investigation

Based upon the developed framework, we implemented two CPGs together with the Department of Neonatology as part of the project. The first one is a guideline for the management of hyperbilirubinemia in the healthy newborn as published by the American Academy of Pediatrics [75]. This is a typical example of a text-based CPG intended for clarifying treatment indications. The text is complemented by a non-executable clinical algorithm used for illustrating the indications of the treatment options. During implementation the adaptation team decided to use this CPG basically as educational material for junior physicians. We also included a separate document with material for nurses who initiate the diagnosis process and carry out several tasks such as the recommended phototherapy. Considering the needs of experienced physicians we developed a single paged summar-

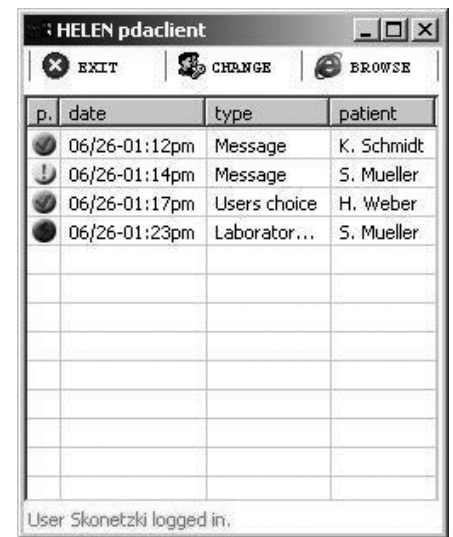


Fig. 8 HELEN PDA client for communicating with the GEE

ization of thresholds for the treatment indications as a printable pocket version. Finally we completed the implementation with an algorithm used as a simple computer-based training application for junior physicians.

In contrast the second CPG (Management of apnea in pre-term newborns [76, 77]) focuses on correct sequencing and timing of clinical actions. The core element is a complex set of nested algorithms which covers several therapy decisions as well as the timing of the execution. We distinguish between five therapy levels ranging from standard care such as food adaptation over caffeine or doxapram medication to several types of mechanical ventilation in emergency cases. Narrative text elements are used here as the explanatory resource of certain decisions, therapy options, and advice with regard to the accomplishment of tasks. The text resources can also be used to give an introduction to the topic or in cases where the physician requests guidance only on an advanced state of treatment. It provides him with alternative entry points to the algorithm. During the execution the system basically acts as a reminder for checking and reconsidering the current treatment for a certain patient. With this CPG we have extensively tested the algorithmic features and the linkage with text-based elements.

## 4.2 In-breadth Experiments

We also worked on several other CPGs which currently are not implemented in all detail but yet modeled on an advanced level. This work has been done in cooperation with experts to verify the flexibility of our approach and to broaden the range of considered modes of use. One example used from ophthalmology is the diagnosis of uveitis [78, 79]. With this example we checked the ability for inserting and connecting a new knowledge module. This new module represents a decision table for a heuristic classification of certain diagnoses. Additionally, we checked the ability to handle a multi-professional (ophthalmologist and rheumatologist) as well as a multi-centered treatment team. The last example to mention here is derived from oncology and deals with the diagnosis and treatment of small-cell lung cancer [80]. The focus here lies on the intended implementation by using algorithms as navigation aids. Fortunately, the representation that we developed is able to deal with this kind of information as well as with the intended use. Slight modifications are necessary to enable the Guideline Viewer to present the CPG by using a frame-based concept. The first frame contains alternating images of the algorithms for navigation and the other frame contains the selected documents and knowledge modules as hitherto.

## 5. Discussion

It is not the intention and it would go beyond the means of the framework presented here to supersede the variety of proposed and implemented approaches for formal representation of clinical guidelines. We rather directly focus on management and implementation specific topics to bring CPGs into clinical practice. As reported by many scientific groups the complexity of the authoring process has proven to be a real bottleneck (e.g. [33]). Therefore it is worthwhile to share such computerized CPGs among different institutions (e.g., [14]). We have taken up both topics and developed a framework for authoring and implementing CPGs that is as easy and flex-

ible as possible to use. During the implementation of the CPGs within the Department of Neonatology a cooperative rapid prototyping process involving medical specialists, future users, and knowledge engineers has arisen and proven itself.

Especially the interconnection between the authoring and execution environment encourages the authors to debate. Besides, it helps to verify unclear or ambiguous topics with test cases, and assists in testing the different representation modules for best suitable descriptions with respect to the special characteristics of each topic. Fortunately this often leads to manifold documents with double tracked descriptions (both text and algorithm) to achieve an optimal accuracy and to satisfy the range of needs of further users. The usual procedure was based upon an initial encoding of the sources as skeletal CPG as provided by national or international medical societies. After that the CPG was completed and adjusted with respect to special requirements of users, environment, and functional organization by using the adaptation feature for each encoded knowledge module. Wherever possible we derived the adapted recommendations from the underlying guideline resources. In case that this wasn't possible we introduced new modules and documents. It is still under examination to what extent the adaptation feature can be useful by sharing the CPG with other institutions.

Last but not least we have to conclude that additional to the already mentioned factors for successful implementations of CPGs we assumed that a confinement to a limited and carefully selected number of urgent areas per department is also a substantial factor for motivation. It is necessary to consider not only the often voluntary and extensive work done during development, adaptation and implementation but also the persistent amount of work for the regular necessary updating of the CPG to keep track with current scientific research as well as with changed basic conditions of providing health care.

### 5.1 Future Work

After developing the basic fundamentals we will focus on enhancing the capabilities

of our framework regarding more flexibility and usefulness. The first problem is how to strengthen and prove the evidence foundation of the guideline. Therefore we consider the presentation of additional information and annotations about the literature used during the development process. Furthermore, intermediate results such as generated evidence tables and information about the process of how the evidence was derived and combined to form a certain statement within a knowledge module could be represented.

By reasoning about this information we attempt an automatic detection of weaknesses within the use of the evidence and whether the evidence foundation is also preserved during adaptation of a certain module. In combination with another field of our ongoing research focusing on attaching rules for detecting typical errors and mistakes made during modeling as well as detecting syntactic and marginal semantic inconsistencies within the guideline this could lead to an advanced detection of contradictions or weak evidence rationale. This might help the authors to keep focused on seeking better evidence. This purpose makes it necessary to include standardized medical terminology and conceptual models (see also [81, 82]). Such terminologies do not only offer a more sophisticated decision support with explicit abstraction of data and actions as well as standardized patient states and clinical diagnosis but are also essential for comprehensive access to electronic patient records, and for our intention of checking advanced semantic rules.

Besides, we would like to spend some effort to extend our framework with a separate tool for an automatic eligibility determination of a CPG for certain patients preferably based upon the yet limited clinical data available within our HIS. Therefore, we have to clarify which information is essentially needed for an automatically generated recommendation to start or abandon a CPG and above all, how to obtain it preferably without additional effort for the health care professionals. A possible solution may lie in the use of methodologies from case-based reasoning. Other extensions concern the automatic export and

import feature to other guideline representation systems.

Besides the technical aspects already mentioned, we consider systematically collecting information on the benefits and efforts by using our CPGs in clinical practice. Last but not least we are focusing on improving the adaptation feature in cooperation with other medical institutions. We consider the introduction of an automatically generated more detailed documentation about necessary adjustments, recurring changes, or extensions to certain elements of the CPG during the adaptation process. Such information might be helpful to optimize CPGs because it offers the initial authoring team structured feedback about imprecise, ambiguous, or infeasible recommendations. By supporting these crucial tasks the adaptation feature hopefully becomes an accepted and useful part of implementing CPGs.

## 5.2 Conclusions

We have developed a working set of tools for authoring, browsing and executing clinical guidelines. This framework provides an open and flexible architecture which can easily be extended by adding new knowledge modules to the ontology and by implementing corresponding classes with the required additional functionality to the existing tools.

We have shown the benefits of combining different knowledge representations such as narrative text, graphic illustrations and algorithms. Finally we have introduced a possible approach for an explicit adaptation process of documented and auditable CPGs in order to provide institution-specific recommendations and to support sharing with other medical institutions.

In the end we would like to point to the fact that all tools were developed under the GNU public licence so that we do not only permit the sharing of formally represented guidelines, but also make use of these guidelines as a shared resource, as other projects did before.

### Acknowledgments

The authors would like to thank Dr. Becker for his contribution in developing CPGs for the Department of Ophthalmology of the University Medical Center Heidelberg. Furthermore we are most grateful to Jan-Peter Klippel and Oliver Merker for their commitment during the development of the HELEN tools.

### References

- Grimshaw JM, Russell IT. Effect of clinical guidelines on medical practice: a systematic review of rigorous evaluations. *Lancet* 1993; 342 (8883): 1317-22.
- Hibble A, Kanka D, Pencheon D, et al. Guidelines in general practice: the new Tower of Babel? *BMJ* 1998; 317 (7162): 862-3.
- Wolf SH, Grol R, Hutchinson A, et al. Clinical guidelines: potential benefits, limitations, and harms of clinical guidelines. *BMJ* 1999; 318 (7182): 527-30.
- Cabana MD, Rand CS, Powe NR, et al. Why don't physicians follow clinical practice guidelines? A framework for improvement. *JAMA* 1999; 282 (15): 1458-65.
- Dahlberg K. Reasons physicians do not follow clinical practice guidelines. *JAMA* 2000; 283 (13): 1686.
- Davis DA, Taylor-Vaisey A. Translating guidelines into practice. A systematic review of theoretic concepts, practical experience and research evidence in the adoption of clinical practice guidelines. *CMAJ* 1997; 157 (4): 408-16.
- Haynes B, Haines A. Barriers and bridges to evidence based clinical practice. *BMJ* 1998; 317 (7153): 273-6.
- Grimshaw J, Russell I. Achieving health gain through clinical guidelines. I: Developing scientifically valid guidelines. *Qual Health Care*, 1993. 2 (4): 243-8.
- Grimshaw JM, Russell IT. Achieving health gain through clinical guidelines II: Ensuring guidelines change medical practice. *Qual Health Care* 1994; 3 (1): 45-52.
- Keffer JH. Guidelines and algorithms: perceptions of why and when they are successful and how to improve them. *Clin Chem* 2001; 47 (8): 1563-72.
- Duff L, Casey A. Implementing clinical guidelines: how can informatics help? *J Am Med Inform Assoc* 1998; 5 (3): 225-6.
- Frankel HL, FitzPatrick MK, Gaskell S, et al. Strategies to improve compliance with evidence-based clinical management guidelines. *J Am Coll Surg* 1999; 189 (6): 533-8.
- Patel VL, Arocha JF, Diermeier M, et al. Cognitive psychological studies of representation and use of clinical practice guidelines. *Int J Med Inf* 2001; 63 (3): 147-67.
- Boxwala AA, Tu S, Peleg M, et al. Toward a representation format for sharable clinical guidelines. *J Biomed Inform* 2001; 34 (3): 157-69.
- Corb GJ, Liaw Y, Brandt CA, et al. An object-oriented framework for the development of computer-based guideline implementations. *Methods Inf Med* 1999; 38 (3): 148-53.
- Silagy C, Weller D, Lapsley H, et al. The effectiveness of local adaptation of nationally produced clinical practice guidelines. *Fam Pract* 2002; 19 (3): 223-30.
- Tierney WM, Overhage JM, Takesue BY, et al. Computerizing guidelines to improve care and patient outcomes: the example of heart failure. *JAMIA* 1995; 2 (5): 316-22.
- Smith WR. Evidence for the effectiveness of techniques to change physician behavior. *Chest* 2000; 118 (2 Suppl): 8S-17S.
- Greco PJ, Eisenberg JM. Changing physicians' practices. *N Engl J Med* 1993; 329 (17): 1271-3.
- Johnston ME, Langton KB, Haynes RB, et al. Effects of computer-based clinical decision support systems on clinician performance and patient outcome. A critical appraisal of research. *Ann Intern Med* 1994; 120 (2): 135-42.
- Shea S, DuMouchel W, Bahamonde L. A meta-analysis of 16 randomized controlled trials to evaluate computer-based clinical reminder systems for preventive care in the ambulatory setting. *JAMIA* 1996; 3 (6): 399-409.
- Shiffman RN, Liaw Y, Brandt CA, et al. Computer-based guideline implementation systems: a systematic review of functionality and effectiveness. *JAMIA* 1999; 6 (2): 104-14.
- Greenes RA, Peleg M, Boxwala A, et al. Sharable computer-based clinical practice guidelines: rationale, obstacles, approaches, and prospects. *Medinfo* 2001; 10 (Pt 1): 201-5.
- Lukoff JY, Dolin RH, McKinley CS, et al. Validation of an XML-based process to automatically web-enable clinical practice guidelines: experience with the smoking cessation guideline. *Proc AMIA Symp* 1999; 311-4.
- Hoelzer S, Schweiger R, Dudeck J. Representation of practice guidelines with XML – modeling with XML schema. *Methods Inf Med* 2002; 41 (4): 305-12.
- Dufour JC, Fieschi M, Fieschi D, et al. A Platform to Develop and to Improve Effectiveness of Online Computable Guidelines. *Stud Health Technol Inform* 2003; 95: 445-51.
- Agrawal A, Shiffman RN. Evaluation of guideline quality using GEM-Q. *Medinfo* 2001; 10 (Pt 2): 1097-101.
- Agrawal A, Shiffman RN. Using GEM-encoded guidelines to generate medical logic modules. *Proc AMIA Symp* 2001; 7-11.
- Gershkovich P, Shiffman RN. An implementation framework for GEM encoded guidelines. *Proc AMIA Symp* 2001; 204-8.
- Terenziani P, Molino G, Torchio M. A modular approach for representing and executing clinical guidelines. *Artif Intell Med* 2001; 23 (3): 249-76.
- Lobach DF, Gadd CS, Hales JW. Structuring clinical practice guidelines in a relational database model for decision support on the Internet. *Proc AMIA Annu Fall Symp* 1997; 158-62.
- Ohno-Machado L, Gennari JH, Murphy SN, et al. The guideline interchange format: a model for representing guidelines. *J Am Med Inform Assoc* 1998; 5 (4): 357-72.
- Tu SW, Musen MA. A flexible approach to guideline modeling. *Proc AMIA Symp* 1999; 420-4.

34. Boxwala AA, Greenes RA, Deibel SR. Architecture for a multipurpose guideline execution engine. *Proc AMIA Symp* 1999; 701-5.
35. Greenes RA, Boxwala A, Sloan WN, et al. A framework and tools for authoring, editing, documenting, sharing, searching, navigating, and executing computer-based clinical guidelines. *Proc AMIA Symp* 1999; 261-5.
36. Peleg M, Boxwala AA, Ogunyemi O, et al. GLIF3: the evolution of a guideline representation format. *Proc AMIA Symp* 2000; 645-9.
37. Peleg M, Ogunyemi O, Tu S, et al. Using features of Arden Syntax with object-oriented medical data models for guideline modeling. *Proc AMIA Symp* 2001; 523-7.
38. Zielstorff RD, Teich JM, Paterno MD, et al. P-CAPE: a high-level tool for entering and processing clinical practice guidelines. *Partners Computerized Algorithm and Editor. Proc AMIA Symp* 1998; 478-82.
39. Kuperman GJ, Teich JM, Bates DW, et al. Detecting alerts, notifying the physician, and offering action items: a comprehensive alerting system. *Proc AMIA Annu Fall Symp* 1996; 704-8.
40. Purves IN, Sugden B, Booth N, et al. The PRODIGY project – the iterative development of the release one model. *Proc AMIA Symp* 1999; 359-63.
41. Johnson PD, Tu S, Booth N, et al. Using scenarios in chronic disease management guidelines for primary care. *Proc AMIA Symp* 2000; 389-93.
42. Shahar Y, Miksch S, Johnson P. The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artif Intell Med* 1998; 14 (1-2): 29-51.
43. Barahona P, Azevedo F, Veloso M, et al. A computerised guideline for the management of diabetes. *Stud Health Technol Inform* 2000; 77: 600-4.
44. Fox J, Johns N, Rahmanzadeh A. Disseminating medical knowledge: the PROforma approach. *Artif Intell Med* 1998; 14 (1-2): 157-81.
45. Musen MA, Tu SW, Das AK, et al. EON: a component-based approach to automation of protocol-directed therapy. *J Am Med Inform Assoc* 1996; 3 (6): 367-88.
46. Tu SW, Musen MA. From guideline modeling to guideline execution: defining guideline-based decision-support services. *Proc AMIA Symp* 2000; 863-7.
47. Elkin PL, Peleg M, Lacson R, et al. Toward the standardization of electronic guidelines. *MD Comput* 2000; 17 (6): 39-44.
48. Peleg M, Tu S, Bury J, et al. Comparing computer-interpretable guideline models: a case-study approach. *JAMIA* 2003; 10 (1): 52-68.
49. AeZQ. Checkliste zur Bewertung der Methodischen Qualität von Leitlinien. *Dt Ärzteblatt* 2000; 97 (17): A1170-1172.
50. Gruber TR. A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition* 1993; (5): 199-220.
51. Musen MA. Dimensions of knowledge sharing and reuse. *Comput Biomed Res* 1992; 25 (5): 435-67.
52. Angele J, Decker S, Studer R. Domain and Task Modeling in MIKE. In: *Domain Knowledge for Interactive Systems Design, Proceedings of IFIP WG 8.1/13.2 Joint Working Conference*. 1996. Geneva: Chapman & Hall.
53. Boehm BW. A Spiral Model of Software Development and Enhancement. *ACM SIGSOFT* 1988; 4 (11): 21-42.
54. AGREE. Guideline development in Europe. An international comparison. *Int J Technol Assess Health Care* 2000; 16 (4): 1039-49.
55. Burgers J, Grol R, Klazinga N, et al. International comparison of 19 clinical guideline programs – a survey of the AGREE Collaboration. *Z Arztl Fortbild Qualitätssich* 2003; 97 (1): 81-8.
56. Cluzeau FA, Littlejohns P, Grimshaw JM, et al. Development and application of a generic methodology to assess the quality of clinical guidelines. *Int J Qual Health Care* 1999; 11 (1): 21-8.
57. Bernstam E, Ash N, Peleg M, et al. Guideline classification to assist modeling, authoring, implementation and retrieval. *Proc AMIA Symp* 2000; 66-70.
58. Helou A, Perleth M, Bitzer EM, et al. Methodological quality of clinical practice guidelines in Germany: results of a systemic assessment of guidelines presented on the Internet. *Z Arztl Fortbild Qualitätssich* 1998; 92 (6): 421-8.
59. Shiffman RN, Karras BT, Agrawal A, et al. GEM: a proposal for a more comprehensive guideline document model using XML. *JAMIA* 2000; 7 (5): 488-98.
60. Mayerhoff L. Optimal embedding of a knowledge based system in an existing clinical environment. Specification of requirements and design based upon the CommonKADS methodology. Institute for Medical Biometry and Informatics, Department of Medical Informatics, University of Heidelberg: Heidelberg 2000; 151.
61. SIGN, SIGN 50: A guideline developers' handbook. 1992. Scottish Intercollegiate Guidelines Network, <http://www.sign.ac.uk/guidelines/full-text/50/index.html> Last Access: January 10, 2004.
62. Gerlach FM, Beyer M, Berndt M, et al. The DEGAM-concept – development, dissemination, implementation and evaluation of guidelines for general practice. *Z Arztl Fortbild Qualitätssich* 1999; 93 (2): 111-20.
63. de Clercq PA, Blom JA, Hasman A, et al. An ontology-driven approach for the acquisition and execution of clinical guidelines. *Stud Health Technol Inform* 2000; 77: 714-9.
64. Uschold M, Gruninger M. *Ontologies: Principles, Methods and Applications*. Knowledge Engineering Review 1996; 1(2): 93-155.
65. NGC, NGC Classification for CPGs. 2002, [http://www.guideline.gov/FRAMESETS/static\\_fs.asp?view=about.overview.classify](http://www.guideline.gov/FRAMESETS/static_fs.asp?view=about.overview.classify), Last Access: December 15, 2003.
66. Noy FN, McGuiness DL. *Ontology Development 101: A Guide to Creating Your First Ontology*. 2001, Stanford Medical Informatics: Stanford; 25.
67. Gennari JH, Musen MA, Fergerson RW, et al. *The Evolution of Protégé: An Environment for Knowledge-Based Systems Development*. 2002. Stanford Medical Informatics: Stanford; 32.
68. Ollenschlager G, Helou A, Kostovic-Cilic L, et al. Checklist for methodological quality of guidelines. A contribution to quality promotion of medical guidelines. *Z Arztl Fortbild Qualitätssich* 1998; 92 (3): 191-4.
69. Fallside DC. XML Schema Part 0: Primer. 2001, W3C Recommendation.
70. Sun, JAVA Servlet Technology. 2003, <http://java.sun.com/products/servlet/index.html>. Last Access: 21.01.2004.
71. Jakarta, The Apache Jakarta Project. 2003, <http://jakarta.apache.org/tomcat/>. Last Access: January 21, 2004.
72. Protege, Protégé Plug-ins Library. 2002, <http://protege.stanford.edu/plugins.html>. Last Access: January 21, 2004.
73. Grol R, Grimshaw J. From best evidence to best practice: effective implementation of change in patients' care. *Lancet* 2003; 362 (9391): 1225-30.
74. Gell G, Schmucker P, Pedevilla M, et al. SAP and partners: IS-H and IS-H\* MED. *Methods Inf Med* 2003; 42 (1): 16-24.
75. AAP, Practice parameter: management of hyperbilirubinemia in the healthy term newborn. American Academy of Pediatrics. Provisional Committee for Quality Improvement and Subcommittee on Hyperbilirubinemia. *Pediatrics* 1994; 94 (4 Pt 1): 558-65.
76. Bhatia J. Current options in the management of apnea of prematurity. *Clin Pediatr (Phila)*, 2000; 39 (6): 327-36.
77. Emborn C, Humphrey JA. Apnea of prematurity. 1999. National Association of Neonatal Nurses – Professional Association.
78. Biswas J. Practical concepts in the management of uveitis. *Indian J Ophthalmol* 1993; 41 (3): 133-41.
79. Becker MD, Rosenbaum JT. Essential laboratory tests in uveitis. *Dev Ophthalmol* 1999; 31: 92-108.
80. Becker H, Drings P, Havemann K, et al. Treatment of small-cell lung cancer. In: *Quality assurance in oncology. Diagnosis and therapy malignancy disease, I.f.S.i.d.O. (ISTO)*, Editor: Deutsche Krebsgesellschaft e.V.: Muenchen 1999; 218-229.
81. Beeler GWJ. Taking HL7 to the next level. *MD Comput* 1999; 2 (16): 21-4.
82. Cornet R, Abu-Hanna, A. Usability of expressive description logics – a case study in UMLS. *Proc AMIA Symp* 2002; 180-4.

**Correspondence to:**  
 Dipl.-Inform. Med. S. Skonetzki  
 Institute for Medical Biometry and Informatics  
 Department of Medical Informatics  
 University of Heidelberg  
 Im Neuenheimer Feld 400  
 69120 Heidelberg  
 Germany  
 E-mail: stefan\_skonetzki@med.uni-heidelberg.de