

How can Information Extraction ease formalizing treatment processes in clinical practice guidelines? A method and its evaluation^{*}

Katharina Kaiser^{a,*}

Cem Akkaya^a

Silvia Miksch^{a,b}

^a*Institute of Software Technology & Interactive Systems
Vienna University of Technology, Austria*

^b*Department of Information & Knowledge Engineering
Danube University Krems, Austria*

Abstract

Objective. Formalizing clinical practice guidelines for a subsequent computer-supported processing is a challenging, but burdensome and time-consuming task. Existing methods and tools to support this task demand detailed medical knowledge, knowledge about the formal representations, and a manual modeling. Furthermore, formalized guideline documents mostly fall far short in terms of readability and understandability for the human domain modeler.

Methods and Material. We propose a new multi-step approach using information extraction methods to support the human modeler by both automating parts of the modeling process and making the modeling process traceable and comprehensible. This paper addresses the first steps to obtain a representation containing processes which is independent of the final guideline representation language.

Results. We have developed and evaluated several heuristics without the need to apply Natural Language Understanding and implemented them in a framework to apply them to several guidelines from the medical subject of otolaryngology. Findings in the evaluation indicate that using semi-automatic, step-wise information extraction methods are a valuable

instrument to formalize CPGs.

Conclusions. Our evaluation shows that a heuristic-based approach can achieve good results, especially for guidelines with a major portion of semi-structured text. It can be applied to guidelines irrespective to the final guideline representation format.

Key words: Information extraction and integration, clinical practice guidelines, computer-interpretable guidelines, guideline representation, treatment processes, time-oriented information, otolaryngology

1 Introduction

Computer-supported guideline execution is an important instrument for improving the quality of health care. To execute clinical practice guidelines (CPGs) in a computer-supported way, the information in the guideline, which is in plain textual form, in tables, or represented in flow charts, has to be formalized. Consequently, this means that a formal representation is required in order to make the information computable. Thus, several so called *guideline representation languages* have been developed to support the structuring and representation of various guidelines and protocols and to make possible different kinds of applications (see [1] and [2]).

Many researchers have proposed frameworks for modeling CPGs in a computer-interpretable and -executable format (a comprehensible overview can be found in [1] and [2]). Each of these frameworks provides specific guideline representation languages. Most of these languages are sufficiently complex that the manual for-

* This is an extended and revised version of K. Kaiser, C. Akkaya, and S. Miksch. Gaining Process Information from Clinical Practice Guidelines, Proc. of the 10th Conference on Artificial Intelligence in Medicine (AIME), 2005.

* Corresponding author. Phone: +43 (1) 58801-18839. Fax: +43 (1) 58801-18899
Email addresses: kaiser@ifs.tuwien.ac.at (Katharina Kaiser),
cemakkaya@gmx.at (Cem Akkaya), silvia@ifs.tuwien.ac.at (Silvia Miksch).

malization of CPGs is a challenging project. Thus, research has to be directed in such a way that tools and methods are developed for supporting the formalization process. Currently, using these tools and methods the human guideline developer needs not only knowledge about the formal methods, but also about the medical domain. This results in a very challenging, but time-consuming and cumbersome formalization task.

Thus, we will look for new approaches that can facilitate the formalization process, and support the developer by providing these kinds of knowledge, as well as intelligent methods for a simplified guideline modeling processing.

Within the next section we present related work of guideline formalization tools and Information Extraction (IE) systems. In Section 3 we propose our approach. Section 4 describes our method which is evaluated in Section 5. Our conclusions are covered in Section 6.

2 Related work

In this section, we present a short discussion of some relevant work describing guideline formalization tools as well as some examples of IE systems.

2.1 Guideline formalization tools

To support the formalization of clinical guidelines into a guideline representation language various methods and tools exist, ranging from simple editors to sophisticated graphical applications.

Markup-based tools. *Stepper* [3] is a tool that formalizes the initial text in multiple user-definable steps corresponding to interactive XML transformations. The

result of each step is an increasingly formalized version of the source document. Both the mark-up and the iterative transformation process are carried out by rules expressed in a new transformation language based on XML. Stepper documents all activities. So the transformation process can easily be reviewed by other users. Stepper also provides an interface showing the interconnection between the source text and the model.

The *GEM Cutter* [4] transforms guideline information into the GEM format [5], showing the original guideline document together with the corresponding GEM document and makes it possible to copy text from the guideline to the GEM document.

The GEM Cutter is similar to the *Document Exploration and Linking Tool / Addons (DELT/A)*, formerly known as Guideline Markup Tool (GMT) [6], which supports the translation of HTML documents into an XML language. DELT/A provides two main features: (1) linking between a textual guideline and its formal representation, and (2) applying design patterns in the form of macros. DELT/A allows the definition of links between the original guideline and the target representation, which gives the user the possibility to find out where a certain value in the XML-language notation comes from. Therefore, if someone wants to know the origin of a specific value in the XML file DELT/A can be used to jump to the correlating point in the text file where the value is defined and the other way round. By means of these features the original text parts need not be stored as part of the target representation elements. The links clearly show the source of each element in the target representation. Additionally, there is no need to produce a guideline in natural language from the target representation, since the original text remains unaltered.

Uruz, part of the *Degel* framework [7], is a web-based markup tool, which resembles DELT/A but does not maintain links between different representations of the guideline. It can also be used to create a guideline document without using any source by directly writing into the knowledge roles of a target ontology. Uruz en-

ables the user to embed in the guideline document terms originating from standard vocabularies, such as ICD-9-CM (International Classification of Diseases) for diagnosis codes, CPT-4 (Current Procedural Terminology) for procedure codes, and LOINC-3 (Logical Observation Identifiers, Names, and Codes) for observations and laboratory tests.

Graphic tools. *AsbruView* [8] uses graphical metaphors to represent Asbru [1] plans. It is a tool to make Asbru accessible to physicians, and to give any user an overview over a plan hierarchy.

The *Plan Body Wizard (PBW)* of the *Degel* framework [7] is used by medical experts for defining the guideline's control structure in the Asbru representation [1]. It 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.

AREZZO and *TALLIS* [9] support the translation into *PROforma* [10] by means of graphical symbols representing the task types of the language.

Protégé [11] is a knowledge-acquisition tool that supports the translation into guideline representation languages such as EON [12], GLIF [1], or *PROforma* [10]. It uses specific ontologies for these languages, whereas parts of the formalization process can be accomplished with predefined graphical symbols. *AREZZO*, *TALLIS*, and *Protégé* represent the processes by means of flow charts.

But still, in all of the above mentioned cases the modeling process is complex and labor intensive. Therefore, methods are needed that can be applied to automate parts of the modeling task.

2.2 Information Extraction systems

IE is an emerging NLP technology whose function is to process unstructured, natural language text, to locate specific pieces of information, or facts in the text, and to use these facts to fill a database. Similar to IE systems are *wrappers* which aim to locate relevant information in semi-structured data [13] and often do not need to apply NLP techniques due to a restricted grammatical structure of the information resources.

For developing both IE and wrapper systems two approaches can be applied: (1) the Knowledge Engineering approach and (2) the automatic learning approach.

The former is customized manually to a given task (e.g., *FASTUS* [14], *PLUM* [15], *PROTEUS* [16]). But manually generating extraction rules is a cumbersome and time-consuming task. Thus, research has been directed towards automating this task. The automatic approach takes a set of documents and outputs a set of extraction patterns by using Machine Learning techniques. Automatic learning systems can be categorized in three groups:

- (1) *supervised learning systems* require a large set of training data to learn rules using machine learning techniques (e.g., *AutoSlog* [17], *LIEP* [18], *ANNIE*, part of the *GATE* framework [19], or *WHISK* [20]),
- (2) *semi-supervised learning systems* (e.g., *Mutual Bootstrapping* [21], *Snowball* [22], or *EXDISCO* [23]), and
- (3) *unsupervised learning systems*, where rules are learned by a small set of seed rules and an annotated corpus using bootstrapping methods (e.g., *AutoSlog-TS* [24] or *QDIE* [25]).

To cope with the problems of "wrapper generation" and "wrapper maintenance" rule-based methods have been especially popular in recent years. Some techniques for generating rules in the realm of text extraction are called "wrapper induction"

methods. These techniques have proved to be rather successful for IE tasks in their intended domains, which are collections of documents such as web pages generated from a template script [26–28]. However, wrapper induction methods do only extend well to documents specific to the induced rules.

In semi-automatic wrapper generation Machine Learning approaches are applied. Tools may support the design of the wrapper. Some approaches offer a declarative interface where the user shows the system what information to extract (e.g., [13, 28]).

Automatic wrapper generation tools use unsupervised learning techniques. Therefore, no training sets are necessary, just a post-generation tuning (e.g., [29, 30]).

When developing an IE system one has to incorporate numerous criteria to decide which approach to apply [31]. These are the availability of training data, which counts for an automatic learning approach, or the availability of linguistic resources and knowledge engineers, where the Knowledge Engineering approach may be favored. Also the level of performance required and the stability of the final specifications are important factors which may be better fostered by the Knowledge Engineering approach.

3 Our approach: a multi-step transformation process

Most guideline representation languages are very powerful and thus very complex. They can contain many different types of information and data. We therefore decided to apply a multi-step transformation process (cf. Figure 1). It facilitates the formalization process by using various intermediate representations that are obtained by stepwise procedures. The multi-step methodology is necessary, as a one-step or even a two-step modeling process was shown to be not sufficient to the

modeler [32, 33].

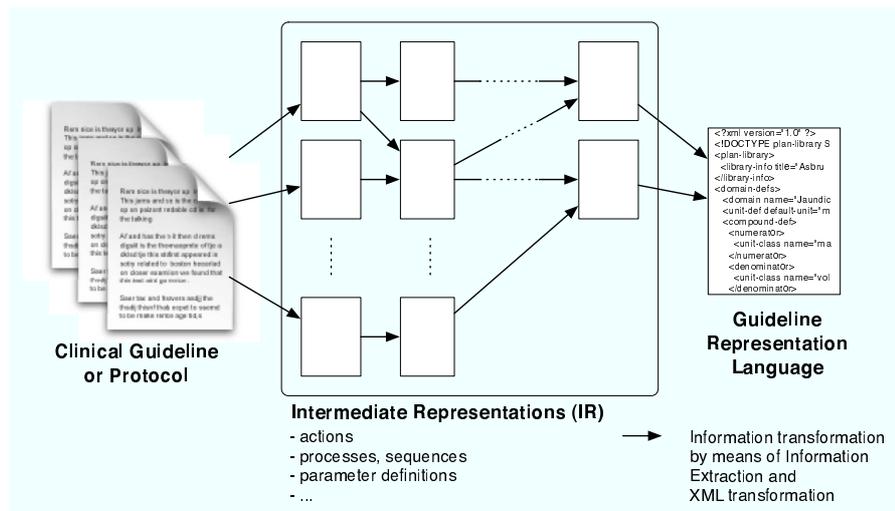


Figure 1. Guideline transformation process. A multi-step process using intermediate representations to transform clinical practice guidelines and (CPGs) into a formal representation language.

The benefits of the intermediate representations are:

- Concise formalization process
- Different formats for various kinds of information
- Separate views and procedures for various kinds of information
- Application of specific heuristics for each particular kind of information
- Simpler and more concise evaluation and tracing of each process step

To process as large a class as possible of documents and information we need specific heuristics. These are applied to a specific form of information, for instance:

Different kinds of information. Each kind of information (e.g., processes, parameters) needs specific methods for processing. By presenting only one kind of information the application of the associated method is simpler and easier to trace.

Different representations of information. We have to take into account various ways in which the information might be represented (i.e., structured, semi-structured, or free text).

Different kinds of guidelines. CGPs exist for various diseases, various user-groups, various purposes, various organizations, and so on, and have been developed by various guideline developers' organizations. Therefore, we can speak about different classes of CGPs that may contain similar guidelines.

To transform information by applying IE methods, we generated specific templates that can present the desired information. The IE methods detect relevant information which is filled into the templates' slots for subsequent processing. In the next section we present a method that extracts process information from clinical guidelines for otolaryngology using heuristic algorithms. The output of this method is a unified format, which can be transformed into the final representation. Detailed information as well as information about the further processing of the resulting representation to the *Asbru* representation is described in [34].

4 Extracting process information of clinical practice guidelines

CPGs present effective treatment processes. One challenge when authoring CPGs is the detection of individual processes and their relations and dependencies. We try to detect these using IE. CPGs consist of semi-structured and free text. The resulting output can subsequently be processed to yield refined representations, leading ultimately to the representation in a specific guideline representation language.

Our main goal is to acquire treatment processes from CPGs. Each process is described by at least one sentence. This means that a sentence, for instance, '*Take acetaminophen or ibuprofen.*', presents only one process and not a selection of two processes. The rules are extraction patterns which are based on syntactical and semantical constraints as well as delimiters.

In order to gain rules to extract the process information, we first had to choose guideline documents which are then used to obtain the rules and to test these rules

with other CPGs. We have chosen guidelines from the National Guideline Clearinghouse (NGC)¹ repository using several criteria. These criteria are the guideline category of treatment and management, the evidence-based quality of guidelines, the existence of treatment instructions featuring temporal aspects of flows, the document structure enabling the detection of text modules such as tables, lists, and paragraphs, and the clinical specialty. We obtained several guidelines from various clinical specialties and have chosen guidelines of otolaryngology. These resulting 18 guideline documents were developed by ten organizations (see Table 1). We then divided the set into a training set of six guidelines (see Table 2) and a test set of twelve guidelines.

Table 1

Developing organizations of the guidelines used for development and testing.

Organization	Guidelines
American Academy of Family Physicians	13
American Academy of Otolaryngology	13
American Academy of Pediatrics	13, 15
Centers for Disease Control and Prevention	1, 14
Cincinnati Children’s Hospital Medical Center	9, 10
Finnish Medical Society Duodecim	17
Institute for Clinical Systems Improvement	2, 3, 4, 7, 8, 16
Practice Guidelines Initiative	18
Scottish Intercollegiate Guidelines Network	6, 11
University of Michigan Health Systems	5, 12

4.1 *Obtaining extraction rules*

From the training set of guidelines we developed rules based on patterns for IE using the knowledge engineering approach. Patterns are defined on three levels, whereas patterns at a certain level serve as concept classes in the preceding levels: (1) phrase level patterns, (2) sentence level patterns, and (3) discourse level

¹ <http://www.guidelines.gov>

Table 2

Set of guidelines used for development of the rules.

#	Title
2	Acute pharyngitis
4	Acute sinusitis in adults
7	Diagnosis and treatment of obstructive sleep apnea
10	Evidence based clinical practice guideline for medical management of otitis media in children 2 months to 6 years of age
12	Otitis media
16	Rhinitis

patterns. Pattern rules were designed using the atomic approach [31]. Thereby, a domain module is built that recognizes the arguments to an event and combines them into template structures strictly on the basis of intelligent guesses rather than syntactic relationships. In doing so domain-relevant events are assumed for any recognized entities, leading to high recall, but much overgeneration, and thus low precision. Further development would result improving filters and heuristics for combining the atomic elements, improving precision.

Medical terms (i.e., drug agents, surgical procedures, and diagnostic terms) are based on a subset of the Medical Subject Headings (MeSH)² of the United States National Library of Medicine. We adapted them according to missing terms, different wordings, acronyms, and varying categorization.

Phrase level patterns. They are used for identifying basic entities, such as *time*, *dosage*, *iteration*, and *condition* expressions, which build the attributes of actions. They are defined by regular expressions.

Sentence level patterns. They use phrase level patterns, medical terms, and trigger words for the medical terms to identify medical actions and their attributes. The trigger words are mainly verbs and indicate the application of a therapy (e.g.,

² <http://www.nlm.nih.gov/mesh/>

the administration of a drug agent or the implementation of a surgical procedure) or the avoidance of a therapy. Sentence level patterns are delimiter-based and use syntactic constraints. We can categorize the patterns in two groups: (1) patterns for free text and (2) patterns for telegraphic text.

The former are applied to free text, which has a grammatical structure and is usually identified in paragraphs, but also in list elements. These patterns indicate that therapy instruments (i.e., agent terms and surgical procedures) combined with trigger terms (e.g., 'activate', 'indicate', 'perform', 'prescribe') appearing in the same clause identify relevant sentences. The particular clauses must not be condition clauses. Phrase level patterns, such as <dosage>, <duration>, <condition>, and so on can be arbitrarily combined with <therapy instrument> <trigger> pairs. But information concerning a treatment recommendation can be distributed in several sentences. These sentences including additional information (e.g., '*The standard dose is 40 to 45 mg/kg/day.*') neither contain a therapy instrument nor a trigger term, but also have to be identified by sentence patterns.

Telegraphic text patterns are applicable in list elements. In these elements often ungrammatical text is formulated and therefore, there is no need for trigger terms. Often, only a therapy instrument indicates the relevancy of an element. Other patterns exist for list elements indicating that these elements are relevant if within their context or in the paragraph preceding the list special terms appear. These terms (i.e., 'remedy', 'remedies', 'measure', 'measures', 'medication', 'medications') are important, because they specify actions that may not contain therapy instruments in the form of agent terms or surgical procedures (e.g., '*Maintain adequate hydration (drink 6 to 10 glasses of liquid a day to thin mucus)*').

Discourse level patterns. They are based on *sentence level patterns*, but are augmented to consider the structure and the layout of the documents. They are used to categorize sentences, merge them to actions, and find relationships between actions

to structure them. To accomplish the latter task we analyzed treatment processes contained in the guidelines and detected the following processes, whereas some of them are identified by discourse level patterns:

- Processes without temporal dependencies
- Sequential processes
- Processes containing subprocesses
- Selections of processes
- Recurring processes

4.2 Gaining process information

To extract processes from CPGs we proceed in several steps which serve to filter segments of text containing treatment instructions from the documents and to generate processes. We propose a two-step approach (see Figure 2) to gain a representation that is independent of the subsequent guideline representation language.

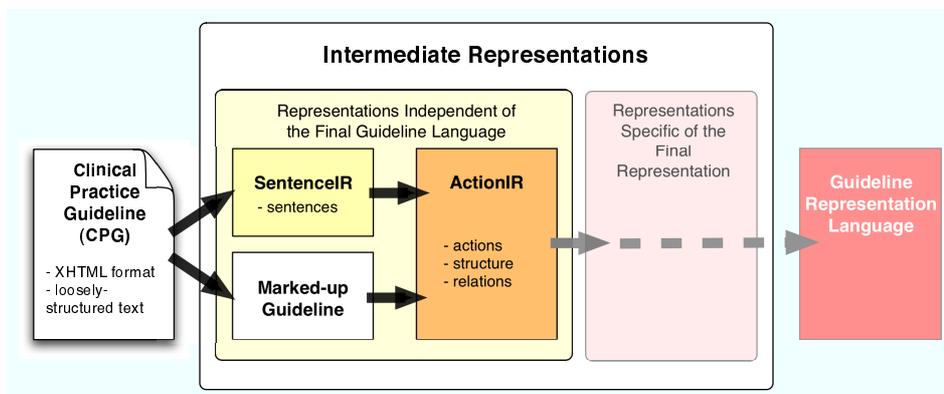


Figure 2. Steps to obtain a semi-formal representation of CPGs. To gain process information from a CPG the first two steps are accomplished in order to have a representation independent of the final guideline language.

The first step is to extract the relevant sentences containing treatment instructions by marking-up the original guideline document. This is explained in Section 4.2.1.

The subsequent step is to combine several sentences to one action and to structure the actions and detect relations among them. This step is described in Section 4.2.2.

These two steps should provide a basis for the subsequent transformation of the process information into any guideline representation language.

4.2.1 *Extracting relevant sentences*

This task is a first step towards our final guideline representation. We will achieve it by two modules: (1) the segmentation & filtering module and (2) the template generation module (see Figure 3 for an overview).

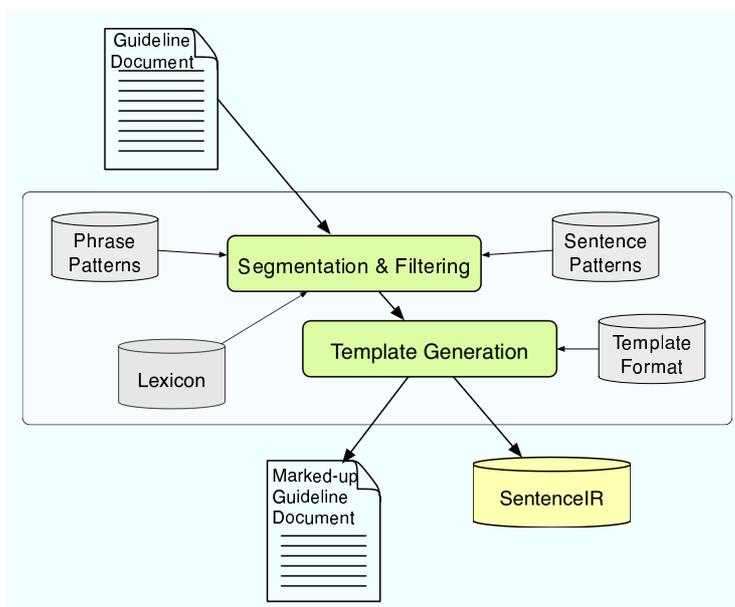


Figure 3. Detecting relevant sentences. We split this task into two modules: (1) segmentation & filtering module and (2) template generation module.

This first intermediate step is especially important as not the entire content of a guideline contains processes, which are to be modeled. Although health care consists of the three stages observation, diagnosis, and therapy [35] we only want to model the control flow regarding the therapy. Only about 20 % of sentences of a guideline are of interest for modeling these processes. On this account it is important to select the relevant sentences for modeling.

Thus, this task performs an automatic mark-up of sentences that are utilized to process the subsequent steps.

Segmentation & filtering. Detecting relevant sentences is a challenging task, which we undertake in two steps: (1) detecting irrelevant text parts to exclude them from further processing and (2) detecting relevant sentences. Irrelevant text parts (i.e., sections, paragraphs) are associated with diagnosis, symptoms, or etiology, relevant sentences describe actions of a treatment processes.

The first filtering occurs at the section level. Sections in the document with captions indicating diagnosis or symptom declarations will be omitted in further processing. We can identify these captions by keywords such as 'history', 'diagnosis', 'symptom', 'clinical assessment', 'risk factor', and so on.

Detecting relevant sentences is not a trivial task. First, we parse the entire document and split it into sentences. Then we process every sentence with regard to its context within the document and its group affiliation. Thereby, the context is obtained by captions (e.g. '*Acute Pharyngitis Algorithm Annotations | Treatment | Recommendations:*') and a group contains sentences from the same paragraph or the same list, if there are no sublists. Each sentence is then checked for relevance by applying *sentence level patterns*.

Template generation. After having collected the relevant sentences from the guideline, we can proceed with generating the intermediate representation *SentenceIR*. We generate two files: one file listing all relevant sentences and the marked-up guideline document (Listing 1 shows the source of a marked-up guideline document). Both are linked by applying the same id to the same sentences. The presentation of the template file and the guideline document are as simple as possible in order to support the user by detecting all relevant sentences.

Listing 1. Excerpt of a source listing of the marked-up guideline document "Evidence-based clinical practice guideline for children with acute bacterial sinusitis in children 1 to 18 years of age". Relevant sentences are enclosed by HTML-like "a" tags.

```
1 <li>
2   <a id="delta:8">In children with risk factors for Streptococcus pneumoniae,
   it is recommended that Amoxicillin, high dose (80 to 90 mg/kg/day) or
   Augmentin (with high dose amoxicillin component) be utilized as
   first-line therapy.
3   </a>
4   <ul type="disc">
5     <li>
6       <a id="delta:9">Note: Failure with amoxicillin is likely to be due to
       resistant Streptococcus pneumoniae, Haemophilus influenzae, or
       Moraxella catarrhalis.
7       </a>
8       <a id="delta:10">High dose amoxicillin will overcome Streptococcus
       pneumoniae resistance (changes in penicillin-binding proteins).
9       </a>
10      The clavulanic acid component of Augmentin is active
11      against resistant Haemophilus influenzae and Moraxella
12      catarrhalis (B-lactamase enzyme).
13    </li>
14  </ul>
15 </li>
```

4.2.2 *Extracting required information and finding processes*

The information contained in *SentenceIR* and the marked-up guideline document are the input for the next task (see Figure 4 for an overview). Its goal is to structure relevant sentences and find relationships between sentences. Again, the output of this task should be represented in a format that is independent of any desired guideline representation format.

Structure extraction. In this task we obtain the context of each sentence by means of hierarchical groups which is necessary for other subtasks, especially the merging and grouping and the process extraction. Every action is assigned to one group. The context of a sentence defines the affiliation to a group and is defined by the sentence's position in the hierarchal structure. We use the superior headings that establish several context items.

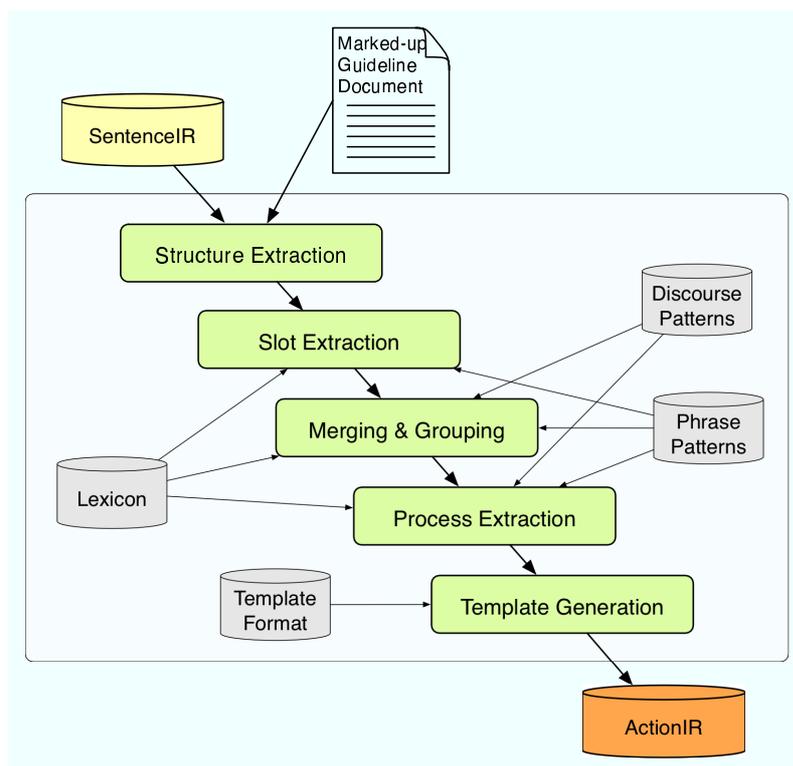


Figure 4. Finding processes and extracting required information. We split this task into five modules (i.e., the structure extraction module, the slot extraction module, the merging & grouping module, the process extraction module, and the template generation module).

Slot extraction. This module is used to extract therapy instruments (i.e., agent terms and surgical procedures), dosage information in case of a drug administration, the duration of the therapy action, the iteration information of the action, as well as conditions which have to be fulfilled to perform an action. It uses both the lexicon and the *phrase level patterns*.

Merging & grouping. In this module we categorize sentences in actions or negative actions and annotations. Annotations always belong to at least one action (or negative action). They cannot exist alone. This module extensively applies *discourse level patterns*.

First, we check whether a sentence describes an action or a negative action. Negative actions are instructions that an action should not be performed, often under

specific conditions (e.g., '*Do not use aspirin with children and teenagers because it may increase the risk of Reyes syndrome.*'). Most guideline representation languages will handle such actions by inverting the condition. Languages may exist which will handle these in other ways. Therefore, we provide a representation for such actions that can be used in a general way.

Furthermore, we identify annotations and assign them to their corresponding actions or negative actions using *name-alias coreferencing* and *definite description coreferencing* based on therapy instruments and their hypernyms. We do not apply pronoun-antecedent coreferencing.

Process extraction. To group actions and to detect relationships between actions we use *discourse level patterns*. We will describe those used by this module below.

The default relationship among processes is that there is no synchronization in their execution. To group actions to a *selection* they must fulfill the following requirements: (1) the actions have to belong to the same group, and (2) agents or surgical procedures must have the same superordinate. For instance, processes describing the administration of *Erythromycin*, *Cephalexin*, and *Clindamycin* within one group are combined in a *selection*, as all these agents are antibiotics. If actions are grouped in a selection, one of these actions has to be selected to be executed.

Furthermore, we try to detect relations between actions that are explicitly mentioned within the text as well as relations that are implicitly given by the document structure. The former is very difficult to detect, as we often cannot detect the reference of the relation within the CPG (e.g., '*After 10 to 14 days of failure of first line antibiotic ...*'). Nevertheless, we found heuristics that arrange actions or action groups if the reference is unambiguously extractable out of the text. These heuristics can be grouped in two categories: (1) detecting sentences describing relations between actions, and (2) detecting actions that are described in the preceding heuristic. A relation is mainly identifiable by a relation term (e.g., 'before', 'after',

'during', 'while'). If such a term appears, we are searching for therapy instruments, as these describe most of our actions. After we have detected these terms, we search for actions containing the particular instruments. If we have found both the source action and the destination action we can create a new relation.

We use patterns of the document structure (e.g., '*Further Treatment*' appears after '*Treatment*' or '*Treatment*' appears before '*Follow-Up*') to detect implicitly given relations. These patterns are part of discourse level patterns to determine relations between several groups.

Template generation. The template of this intermediate representation has to contain actions as well as their relations. It has to be simple and concise and it has to illustrate from which original data the current information was built. We split the new *ActionIR* template in three parts: (1) an area for actions, (2) an area for relations, and (3) an area for the structure illustrating the hierarchy and nesting of groups.

An *action* contains the action sentence, the assigned annotation sentences, treatment instruments and their MeSH ids, information about the dosage, duration, or iteration of a drug administration, and conditions. If the action is part of a selection, it is stated by the selection id. DELT/A links are inherited from the *SentenceIR* representation in order to provide the traceability of the process from both the original guideline document and the *SentenceIR* document. Listing 2 shows an example instance.

Listing 2. Action instance of an *ActionIR* template for the guideline "Evidence-based clinical practice guideline for children with acute bacterial sinusitis in children 1 to 18 years of age".

```
1 <action id="8" parent="5" group="18" selection="0">
2   <delta-link link-id="8"/>
3   <description>In the child with no risk factors for penicillin-resistant
      Streptococcus pneumoniae standard dose amoxicillin or Augmentin (with
      standard dose Amoxicillin component) may be considered as initial
      therapy.
4   </description>
5   <agents>
6     <agent MeSH="D000658" name="amoxicillin"/>
```

```

7     <agent MeSH="D019980" name="Augmentin"/>
8   </agents>
9   <condition>
10    <item>In the child with no risk factors for penicillin-resistant
        Streptococcus pneumoniae
11    </item>
12  </condition>
13  <annotations>
14    <annotation>Note: Forty-six percent of isolates at Children's Hospital
        Medical Center of Cincinnati, Ohio have intermediate or high
        Penicillin-resistant Streptococcus pneumoniae and local data
        supports that 15% of children locally may fail initial therapy with
        standard dose amoxicillin.
15    <delta-link link-id="9"/>
16  </annotation>
17 </annotations>
18 <context>
19   <item>Antibiotic Treatment</item>
20 </context>
21 </action>

```

Relations are stated by their type (e.g., succeeding, preceding, overlapping) and the concerned actions by their DELT/A ids.

Apart from actions and their relations the *structure* of the document is given illustrating the nesting of the groups and selections.

5 Evaluation

The rules developed using training examples have to reach a state where they are able to extract the correct information from other examples, too. In order to test these acquirements we developed Java applications that generate the intermediate representations.

The particular intermediate representations generated from the test set were evaluated by two persons using the DELT/A tool (see Section 2.1). The participants are computer scientists, who are familiar with guidelines, guideline formalization, and the DELT/A tool, but have no medical background. However, the chosen guidelines do not require specific medical knowledge to evaluate the IE tasks.

We evaluated our rules using recall and precision measures. The *Recall* score mea-

sures the ration of correct information extracted from the texts against all the available information present in the text. The *Precision* score measures the ratio of correct information that was extracted against all the information that was extracted [36].

For the evaluation we provided a test set of twelve guidelines in XHTML format and the necessary language and macro files for DELT/A. The participants generated key target templates for all test guidelines using the DELT/A tool. These were then compared to the templates generated by our system, whereas the input of the second step were the key target templates of the first step. For evaluating the mark-up task we compiled the number of relevant sentences according to the key target template (POS), the number of relevant sentences generated by the system (ACT), and the number of correctly detected relevant sentences generated by the system (COR). Out of these values we were able to compute the recall and precision scores (see Table 3).

In [37] we described a preliminary framework which achieved results of 76 % recall and 97 % precision. However, many users claim that IE systems performing a detection of relevant text parts are only of use if they detect all the relevant parts (i.e., a recall score of 100 %). Otherwise the user has to read the entire document to find the remaining relevant sentences. Thus, we optimized the system according to recall. The resulting values for recall of **90.8 %** and precision of **94.9 %** are promising and point out the benefit of the mark-up task. We are of opinion that even with the current performance a benefit exists, because relevant sentences are not equally spread on the whole document, but mostly constitute clusters which have to be verified then.

To verify the process extraction task we again compared this task's key target templates to the system's output templates. We compiled the number of filled slots according to the key target template (POS), the number of slot fillers generated by the system (ACT), the number of correct slot fillers generated by the system (COR),

Table 3. Evaluation results of the mark-up task.

#	Title	SEN	POS	ACT	COR	REC	PRE
1	Acute otitis media: management and surveillance in an era of pneumococcal resistance	17	5	4	4	0.8	1
3	Acute rhinosinusitis in adults	44	10	10	9	0.9	0.9
5	Allergic rhinitis	75	16	20	15	0.938	0.75
6	Diagnosis and management of childhood otitis media in primary care	57	12	9	9	0.75	1
8	Diagnosis and treatment of otitis media in children	269	49	49	49	1	1
9	Evidence based clinical practice guideline for children with acute bacterial sinusitis in children 1 to 18 years of age	171	32	32	31	0.969	0.969
11	Management of sore throat and indications for tonsillectomy	96	20	18	18	0.9	1
13	Otitis media with effusion	87	7	5	5	0.714	1
14	Pneumococcal vaccination for cochlear implant candidates and recipients	20	4	3	3	0.75	1
15	Reduction of the influenza burden in children	50	6	7	6	1	0.857
17	Sore throat and tonsillitis	120	19	17	16	0.842	0.941
18	Symptomatic treatment of radiation-induced xerostomia in head and neck cancer patients	5	5	3	3	0.6	1
		1,011	185	177	168	0.908	0.949

SEN number of sentences of the guideline document
 POS number of relevant sentences according to the key target template
 ACT number of relevant sentences detected by the system
 COR number of correctly detected relevant sentences detected by the system
 REC ratio of COR to POS
 PRE ratio of COR to ACT

and the number of partially correct slot fillers generated by the system (PAR). Starting from these values we computed the recall (REC) and precision (PRE) scores (see Table 4). The overall scores of this task are **84 %** recall and **86.8 %** precision.

Further analyses of these results (see Table 5 for details) showed that they are mainly based on erroneous extractions of duration and iteration information.

The results of each subtask has to be seen within the context of the benefit of the automatically generated data compared to the manual generation using DELT/A. Thereby, the results still imply that using step-wise IE for generating a semi-formal representation of treatment instructions is a great benefit for both knowledge engineers and physicians.

6 Conclusions

Modeling clinical guidelines and protocols is a complex task which has to be assisted by both physicians and knowledge engineers. Bearing those two user groups in mind a method is demanded supporting them in their particular fields of functions: the physicians have to be less overcharged by the formal specifications and the knowledge engineers have to be fostered by providing medical knowledge. Apart from this interesting conceptual formulation we have developed a new methodology applying a step-wise IE which might offer distinct benefits. In particular, it automates parts of the modeling process, it disburdens the physicians in the modeling process by providing a medical ontology, it structures the guideline information, it decomposes the guideline into parts containing various kinds of information (e.g., treatment processes, diagnosis methods, definitions), it makes the modeling process traceable and comprehensible, and it is applicable for many guideline representation languages.

Table 4. Evaluation results of the process extraction task.

#	Title	POS	ACT	COR	PAR	REC	PRE
1	Acute otitis media: management and surveillance in an era of pneumococcal resistance	13	15	11	0	0.846	0.733
3	Acute rhinosinusitis in adults	22	21	21	0	0.955	1
5	Allergic rhinitis	24	24	20	0	0.833	0.833
6	Diagnosis and management of childhood otitis media in primary care	27	24	19	0	0.704	0.792
8	Diagnosis and treatment of otitis media in children	107	105	97	2	0.916	0.933
9	Evidence based clinical practice guideline for children with acute bacterial sinusitis in children 1 to 18 years of age	56	60	50	1	0.902	0.842
11	Management of sore throat and indications for tonsillectomy	35	35	28	1	0.814	0.814
13	Otitis media with effusion	12	12	9	1	0.792	0.792
14	Pneumococcal vaccination for cochlear implant candidates and recipients	13	9	7	0	0.538	0.778
15	Reduction of the influenza burden in children	16	14	14	0	0.875	1
17	Sore throat and tonsillitis	38	33	27	0	0.711	0.818
18	Symptomatic treatment of radiation-induced xerostomia in head and neck cancer patients	8	7	6	0	0.75	0.857
		371	359	309	5	0.840	0.868

POS number of slot fillers according to the key target template
 ACT number of slot fillers generated by the system
 COR number of correct slot fillers generated by the system
 PAR number of partially correct slot fillers generated by the system
 REC ratio of COR plus (.5 x) PAR slot fillers to POS slot fillers
 PRE ratio of COR plus (.5 x) PAR slot fillers to ACT slot fillers

Table 5
Evaluation results of the process extraction subtasks.

	Recall	Precision
Agent	0.955	0.944
Dosage	0.846	0.846
Duration	0.542	0.813
Iteration	0.400	0.667
Condition	0.735	0.735
Relation	0.872	0.872
Sentence Categorization & Assignment	0.837	0.837

We have shown that it is possible to semi-automatically model process information from CPGs using IE. Our rules use patterns in the structure of the documents as well as of specific expressions. Thus, we do not need to apply Natural Language Understanding.

We have applied a framework in order to evaluate our rules that can cope with both semi-structured and free text documents. The resulting information is filled in templates which can represent processes and their relations. The information extracted can then be used in further transformations to finally generate a representation in a guideline representation language.

Acknowledgements

This work is supported by "Fonds zur Förderung der wissenschaftlichen Forschung FWF" (Austrian Science Fund), grants P15467-INF and L290-N04.

References

- [1] Mor Peleg, Samson W. Tu, Jonathan Bury, Paolo Ciccarese, John Fox, Robert A. Greenes, Richard Hall, Peter D. Johnson, Neill Jones, Anand Kumar, Silvia Miksch, Silvana Quaglini, Andreas Seyfang, Edward H. Shortliffe,

- and Mario Stefanelli. Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. *Journal of the American Medical Informatics Association (JAMIA)*, 10(1):52–68, Jan-Feb 2003.
- [2] Paul A. de Clercq, Johannes A. Blom, Hendrikus H. M. Korsten, and Arie Hasman. Approaches for creating computer-interpretable guidelines that facilitate decision support. *Artificial Intelligence in Medicine*, 31(1):1–27, May 2004.
- [3] Marek Růžička and Vojtech Svátek. Mark-up Based Analysis of Narrative Guidelines with the Stepper Tool. In Katharina Kaiser, Silvia Miksch, and Samson W. Tu, editors, *Computer-based Support for Clinical Guidelines and Protocols. Proceedings of the Symposium on Computerized Guidelines and Protocols (CGP 2004)*, volume 101 of *Studies in Health Technology and Informatics*, pages 132–136, Prague, Czech Republic, 2004. IOS Press.
- [4] Kristi-Anne Polvani, Abha Agrawal, Bryant Karras, Aniruddha Deshpande, and Richard Shiffman. *GEM Cutter Manual*. Yale Center for Medical Informatics, 2000.
- [5] Richard N. Shiffman, Bryant T. Karras, Abha Agrawal, Roland Chen, Luis Marengo, and Sujai Nath. GEM: a proposal for a more comprehensive guideline document model using XML. *Journal of the American Medical Informatics Association (JAMIA)*, 7(5):488–498, 2000.
- [6] Peter Votruba, Silvia Miksch, and Robert Kosara. Facilitating knowledge maintenance of clinical guidelines and protocols. In Marius Fieschi, Enrico Coiera, and Yu-Chuan Jack Li, editors, *Proceedings from the Medinfo 2004 World Congress on Medical Informatics*, pages 57–61. IOS Press, 2004.
- [7] Yuval Shahar, Ohad Young, Erez Shalom, Alon Mayaffit, Robert Moskovitch, Alon Hessing, and Maya Galperin. DEGEL: A hybrid, multiple-ontology framework for specification and retrieval of clinical guidelines. In Michel Dojat, Elpida Keravnou, and Pedro Barahona, editors, *Proceedings of the 9th Conference on Artificial Intelligence in Medicine in Europe, AIME 2003*, volume 2780 of *LNAI*, pages 122–131, Protaras, Cyprus, 2003. Springer Verlag.

- [8] Robert Kosara and Silvia Miksch. Metaphors of Movement: A Visualization and User Interface for Time-Oriented, Skeletal Plans. *Artificial Intelligence in Medicine, Special Issue: Information Visualization in Medicine*, 22(2):111–131, May 2001.
- [9] R. Steele and John Fox. Tallis PROforma Primer – Introduction to PROforma Language and Software with Worked Examples. Technical report, Advanced Computation Laboratory, Cancer Research, London, UK, 2002.
- [10] David R. Sutton and John Fox. The syntax and semantics of the PROforma guideline modeling language. *Journal of the American Medical Informatics Association (JAMIA)*, 10(5):433–443, Sep / Oct 2003.
- [11] John H. Gennari, Mark A. Musen, Ray W. Fergerson, William E. Grosso, Monica Crubézy, Henrik Eriksson, Natalya F. Noy, and Samson W. Tu. The Evolution of Protégé: An Environment for Knowledge-based Systems Development. *International Journal of Human Computer Studies*, 58(1):89–123, 2003.
- [12] Mark A. Musen, Samson W. Tu, Amar K. Das, and Yuval Shahar. EON: A component-based approach to automation of protocol-directed therapy. *Journal of the American Medical Informatics Association (JAMIA)*, 3(6):367–388, 1996.
- [13] Nick Kushmerick, Daniel S. Weld, and Robert B. Doorenbos. Wrapper Induction for Information Extraction. In *International Joint Conference on Artificial Intelligence*, Nagoya, 1997.
- [14] Jerry R. Hobbs, Douglas Appelt, Mabry Tyson, John Bear, and David Israel. SRI International: Description of the FASTUS system used for MUC-4. In *Proceedings fo the 4th Message Understanding Conference (MUC-4)*, pages 268–275, 1992.
- [15] Damaris Ayuso, Sean Boisen, Heidi Fox, Herb Gish, Robert Ingria, and Ralph Weischedel. BBN: Description of the PLUM system as used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 169–176, 1992.

- [16] Roman Yangarber and Ralph Grishman. NYU: Description of the Proteus/PET system as used for MUC-7 ST. In *Proceedings of the 7th Message Understanding Conference: MUC-7*, Washington, DC, 1998.
- [17] Ellen Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 811–816, 1993.
- [18] Scott B. Huffman. Learning information extraction patterns from examples. In *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, volume 1040 of *Lecture Notes in Computer Science*, pages 246–260, London, UK, 1996. Springer Verlag.
- [19] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, July 2002.
- [20] Stephen Soderland. Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning*, 34(1-3):233–272, 1999.
- [21] Ellen Riloff and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 474–479. AAAI Press/MIT Press, 1999.
- [22] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plaintext collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*, 2000.
- [23] Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, Saarbrücken, Germany, August 2000.
- [24] Ellen Riloff. An empirical study of automated dictionary construction for information extraction in three domains. *Artificial Intelligence*, 85(1-2):101–134, 1996.

- [25] Kiyoshi Sudo. *Unsupervised Discovery of Extraction Patterns for Information Extraction*. PhD thesis, New York University, New York, September 2004.
- [26] Ion Muslea, Steve Minton, and Craig Knoblock. A hierarchical approach to wrapper induction. In Oren Etzioni, Jörg P. Müller, and Jeffrey M. Bradshaw, editors, *Proceedings of the International Conference on Autonomous Agents*, pages 190–197, Seattle, WA, USA, 1999. ACM Press.
- [27] Nick Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118(1-2):15–68, 2000.
- [28] Robert Baumgartner, Sergio Flesca, and Georg Gottlob. Visual Web Information Extraction with Lixto. In *Proceedings of the Conference on Very Large Databases (VLDB)*, 2001.
- [29] Robert B. Doorenbos, Oren Etzioni, and Daniel S. Weld. A scalable comparison-shopping agent for the world-wide web. In W. Lewis Johnson and Barbara Hayes-Roth, editors, *Proceedings of the International Conference on Autonomous Agents*, pages 39–48, Marina del Rey, CA, USA, 1997. ACM Press.
- [30] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of the Conference on Very Large Databases (VLDB)*, pages 109–118, 2001.
- [31] Douglas E. Appelt. Introduction to information extraction. *AI Communications*, 12:161–172, 1999.
- [32] M. Balsler, O. Coltell, J. van Croonenborg, C. Duelli, F. van Harmelen, A. Jovell, P. Lucas, M. Marcos, S. Miksch, W. Reif, K. Rosenbrand, A. Seyfang, and A. ten Teije. Protocure: Integrating formal methods in the development process of medical guidelines and protocols. In Katharina Kaiser, Silvia Miksch, and Samson W. Tu, editors, *Computer-based Support for Clinical Guidelines and Protocols. Proceedings of the Symposium on Computerized Guidelines and Protocols (CGP 2004)*, volume 101 of *Studies in Health Technology and Informatics*, pages 103–107, Prague, Czech Republic, 2004. IOS Press.

- [33] Andreas Seyfang, Silvia Miksch, Peter Votruba, Kitty Rosenbrand, Jolanda Wittenberg, Joyce von Croonenborg, Wolfgang Reif, Martin Balser, Jonathan Schmitt, Theo van der Weide, Peter Lucas, and Arjen Hommersom. MHB – a many-headed bridge between guideline formats. Technical Report Asgaard-TR-2004-5, Vienna University of Technology, Austria; Dutch Institute for Healthcare Improvements, the Netherlands; University of Augsburg, Germany; University of Nijmegen, the Netherlands, 2004.
- [34] Katharina Kaiser. *LASSIE – Modeling Treatment Processes Using Information Extraction*. PhD thesis, Institute of Software Technology & Interactive Systems, Vienna University of Technology, 2005.
- [35] Jan H. Van Bommel and Mark A. Musen, editors. *Handbook of Medical Informatics*. Springer Verlag, Heidelberg, 1997.
- [36] Wendy Lehnert, Claire Cardie, David Fisher, Joseph McCarthy, Ellen Riloff, and Steven Soderland. Evaluating an Information Extraction system. *Journal of Integrated Computer-Aided Engineering*, 1(6), 1994.
- [37] Katharina Kaiser, Cem Akkaya, and Silvia Miksch. Gaining process information of clinical practice guidelines using information extraction. In *Proceedings of the 10th Conference on Artificial Intelligence in Medicine in Europe, AIME 2005*, volume 3581 of *LNAI*, pages 181–190, Aberdeen, UK, 2005. Springer Verlag.