TimeWrap - A Method for Automatic Transformation of Structured Guideline Components into Formal Process-Representations

Katharina KAISER and Silvia MIKSCH
Institute of Software Technology & Interactive Systems
Vienna University of Technology
Favoritenstrasse 9-11/188
A-1040 Vienna, Austria, Europe
`{kaiser, silvia}@asgaard.tuwien.ac.at`

Abstract. Guideline and protocol representation languages have reached a level of complexity where auxiliary methods are needed to support the authoring of protocols in the particular language. Several approaches and methods exist that claim high knowledge about both, the medical context and the formal requirements. Therefore, we need knowledge-based methods to facilitate the human plan designer and create the protocols of the particular language as automated as possible. We present a three-step wrapper method, called TimeWrap, to extract information, in particular temporal issues, out of semi-structured data and integrate it in a formal representation. We illustrate our approach using the guideline-representation language Asbru and examples from guidelines to treat conjunctivitis.

1 Introduction

For better supporting the medical staff during their diagnostic and therapeutic steps, clinical guidelines and protocols (CGPs) shall proceed in a computer-supported way. Hence, a transformation of the CGPs in a (semi-)formal representation that will be executed in an application is required. Various guideline-representation languages, like *Asbru* or *GLIF* (compare the next section), are available for this reason.

However, clinical guidelines and protocols exist often only in free text. Guideline-representation languages have accomplished a state of complexity where the generation of such protocols is a very challenging venture. As a result we can say that the transformation from text to a (semi-)formal representation is mostly either missing or burdensome and time-consuming, but urgently needed to proceed with the task of computer-supported treatment planning.

Our aim is to facilitate the generation of computer-supported protocols and in series to support the creation of parts of protocols in *Asbru*. Asbru is a very complex guideline-representation language and the creation of Asbru protocols is a very sophisticated process. We have analyzed clinical guidelines to figure out which parts of the guidelines can be used to easily extract information as automated as possible and convert as well as transform it into Asbru. Figure 1 illustrates our approach. By means of a domain- and a time ontology relevant information is extracted from the clinical guidelines. We are not performing any natural language understanding task to capture the content of guideline components. Afterwards it is integrated into different kinds of intermediate representations and transformed into the formal representation of a guideline-representation language, e.g. Asbru. The application of intermediate representations is chosen to better structure the content of the CGP and to represent it in a concise form, as e.g. only special aspects, like temporal flows, are represented. Furthermore a progressive refinement process can be passed through.

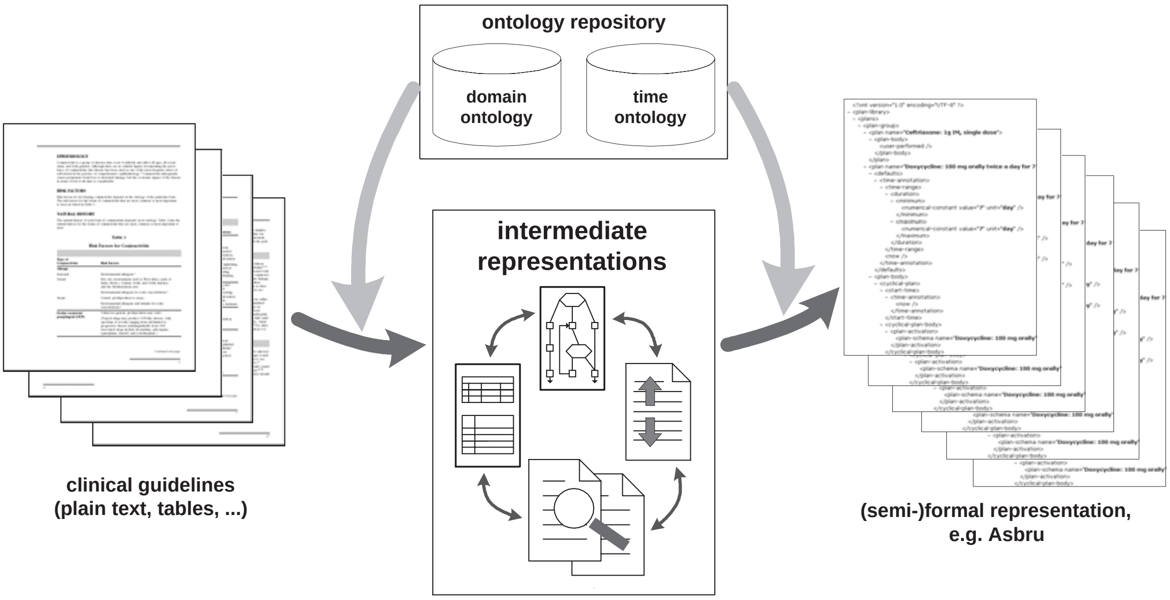
ontology repository
domain ontology
time ontology
intermediate representations
clinical guidelines (plain text, tables, ...)
(semi-)formal representation, e.g. Asbru

Figure 1: Idea of the method for creating a formal representation of clinical protocols.

We have to pay special attention on temporal aspects of CGPs. To model and to present them in *Asbru* is a very complex task. It claims both for comprehension of the CGP and good knowledge about *Asbru* - especially the representation of temporal flows. On this account we try to automize the modeling of flows. As a first step we have chosen an area of treatment planning: the drug administration. We want to demonstrate this by means of a simple example.

The next section describes various approaches related to our solution and explains their benefits and limitations. In Section 3 we describe requirements regarding the time annotations of plans and especially cyclical plans in Asbru and in Section 4, we introduce our solution to the semi-automatic transformation of text to guideline components. We illustrate the usability of our contribution by a case study in Section 5. Finally we conclude with the discussion of the most important issues and future developments.

2 Related Work

In the last years various kinds of guideline and protocol representation languages were developed. Thus, the need to support guideline and protocol acquisition and authoring was emerging and different types of intelligent acquisition methods and tools were developed. In the next subsection we illustrate these two development steps.

2.1 Guideline and Protocol Representation Languages

The major challenges in representing clinical protocols in a computer readable form are to provide a clear, precise representation with defined semantics and to handle the complex forms of uncertainty which are common in the medical domain. There are several approaches to formalize guidelines or protocols in a computer readable way, e.g., Asgaard/Asbru, GLIF, EON, Prestige, PRO*forma*, Guide (A comprehensive overview can be found at [8]).

Some of these approaches lack a formal definition of their semantics. Often they provide a clearly defined framework but the frames are filled with free text. Such a protocol can therefore only be interpreted by a human and not by a computer. Also execution or verification can only be performed by humans who have to interpret each part of free text and decide its precise meaning – an unreliable and often not reproducible process. But there are numerous notations of logic which provide clear formal semantics. However, the task of modeling a protocol in such a notation is simply impossible to achieve. In particular, intertwined processes which develop over time and which involve uncertainty are hard to model in formal logic from scratch. The plan-representation language *Asbru* [7, 10] developed within the *Asgaard* project has clearly defined semantics and complex language constructs to represent uncertain and incomplete knowledge.

2.2 Guideline and Protocol Acquisition - Intelligent Knowledge Acquisition

In the last years, several methods to acquire and extract information from clinical guidelines have been proposed. Such acquisition tools range from simple editors to sophisticated visual wrappers.

Markup-based tools. Guide-X [12] is a methodology that describes a way to translate a guideline into a computerized form. An implementation of this methodology was done in Stepper [13]. The formalization process is divided into several steps, whereas each step has an exactly defined input and output.

The GEM Cutter [9] transforms guideline information into the GEM format. It shows 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 our Guideline Markup Tool (GMT) [14], which supports translating guidelines in free text into the Asbru language, by providing two main features: (i) linking between a textual guideline and its formal representations, and (ii) applying design patterns in the form of macros.

These markup-based tools all have in common that the creation process for the computerized guidelines has to be done manually by a human plan editor.

Graphic tools. A graphical approach was used in AsbruView [5] which was developed to facilitate the creation, editing and visualization of guidelines written in the language Asbru. To be suitable for physicians, AsbruView uses graphical metaphors, such as a running track and traffic control, to represent Asbru plans.

Two tools are available to translate guidelines into PRO*forma* [4] - both make heavy use of the same graphical symbols representing the four task types in PRO*forma*. AREZZO is designed to be used on client-side only, whereas TALLIS [11] supports publishing of PRO*forma* guidelines over the World Wide Web.

These graphic-based tools have in common that they can only be used for design from scratch.

Wrapper tools. Finally, different kinds of wrappers were developed to transform an HTML document into an XML document and deliver the extracted data content in XML format with a DTD (for example, XWRAP [6] or LiXto, which provides a visual wrapper [3]).

These methods and tools are very useful in case highly structured HTML documents are used or simple XML files should be extracted. However, clinical protocols are more complex and XML/DTD files that are more structured are needed in order to represent them.

Our approach considers the limitations mentioned above and tries to support the plan generator of guideline components by automating parts of the development process. It is important to phrase that we are using semi-structured guideline components as source and we are not aiming towards an automatic solution of the transformation process.

In the following section we will explain temporal aspects in Asbru that are required to model processes and that have to be considered in the development of intermediate representations of processes. In Section 4 we specify our TimeWrap method which tries to overcome the limitations explained above.

3 Temporal Aspects in Asbru

Asbru offers extensive possibilities to define complex temporal dependencies and processes by means of Time Annotations. A Time Annotation specifies four points in time relative to a reference point (which can be a specific or abstract point in time or a state transition of a plan): The earliest starting shift (ESS), latest starting shift (LSS), earliest finishing shift (EFS) and latest finishing shift (LFS). Two durations can also be defined: The minimum duration (MinDur) and maximum duration (MaxDur). Together, these data specify the temporal constraints within which an action must take place (see Figure 2).

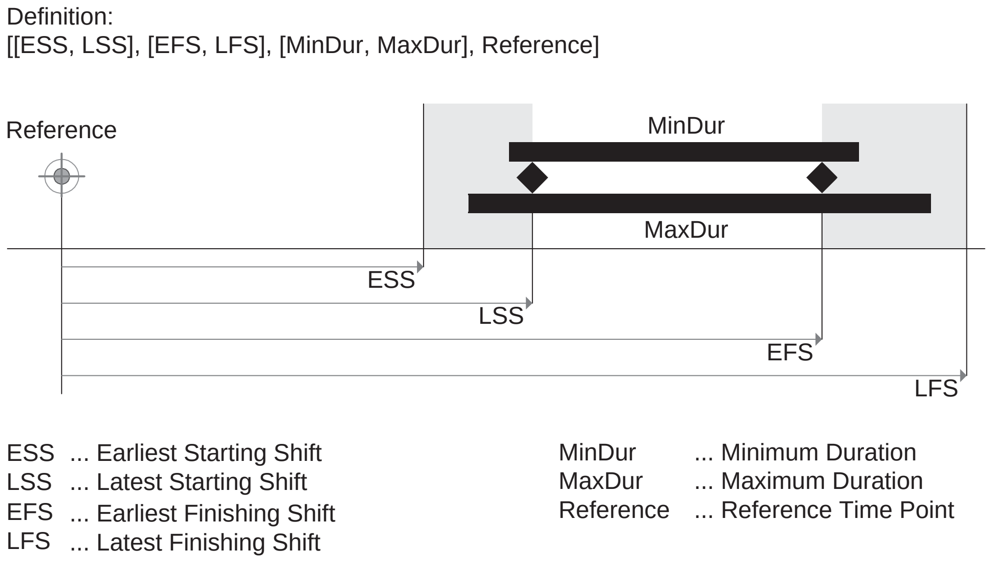
Definition:
[[ESS, LSS], [EFS, LFS], [MinDur, MaxDur], Reference]
Reference
MinDur
MaxDur
ESS
LSS
EFS
LFS
ESS ... Earliest Starting Shift
LSS ... Latest Starting Shift
EFS ... Earliest Finishing Shift
LFS ... Latest Finishing Shift
MinDur ... Minimum Duration
MaxDur ... Maximum Duration
Reference ... Reference Time Point

Figure 2: Time interval in Asbru. The grey areas indicate the periods when the action has to start and accordingly finish.

Asbru offers several different types of plans among other things ’cyclical plans’. A cyclical plan invokes another plan in consistent periods. For this plan additional temporal annotations have to be stated like frequency and possibly the maximum number of cycles. Thereby, the frequency is stated as the period between two iterations that is consistent for the entire cyclical plan (see Figure 3).

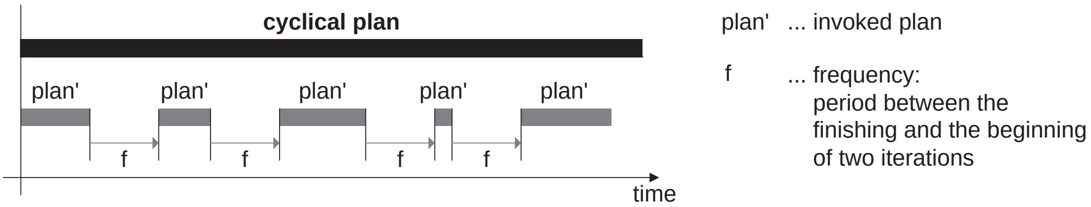

Figure 3: Cyclical plan in Asbru.

4 The TimeWrap Method

The method we have developed facilitates the extraction of information out of semi-structured data and integrates the extracted information into a formal representation. This representation is not ultimate. It is a so called ”intermediate representation” capturing the temporal aspects of a CGP. Other ”intermediate representations” exist that formalize further aspects. Combining and transforming these parts lead to the definit formal representation – in our case *Asbru* [10]. Our method takes text – in this example clinical guidelines – as input.

The TimeWrap method consists of three steps:

1. structuring information and representing it in a formal base representation
2. extracting information out of the base representation; and
3. integrating the extracted information into a formal intermediate representation that is the origin for transformation into Asbru. This form of representation can handle temporal uncertainties and other demands that are required for planning.

In the following the three steps will be explained in detail.

Step 1: Structuring Information

We have analyzed various clinical guidelines and protocols written in textual form and found some typical types of styles.

On the one hand, there always exist diagnosis and treatment parts, which are intertwined and on the other hand, the clinical guidelines are using flow charts and multidimensional tables to represent diagnostic and therapeutic knowledge. In our first step of analysis, we have chosen therapeutic parts and tables. One very important component of treatment plans is the prescription of drugs. For administering drugs the following information has to be available:

- Name of the drug, e.g. Ceftriaxone, Erythromycin, etc.
- Value and unit of the dose, e.g. 1 g, 125 mg or values with composed units like 50 mg/kg/day.
- Kind of application, e.g. orally, intravenous, IV, intramuscular, IM, etc.
- Duration, e.g. 7 days, $10-14$ days, etc.

- Frequency of administration, e.g. twice a day, 4 doses a day, etc.

An important part within these definitions for the planning process and in particular for the implementation of Asbru protocols is time-specific data, like the duration of the treatment and the frequency of the drug-administration.

Most guidelines declare the information about the drug administration by the statement of drug that should be administered and the dosage. The dosage is mostly of the form like ’1 g IM, single dose’, ’100 mg orally twice a day for 7 days’, or ’50 mg/kg/day orally divided into four doses daily for 10−14 days’.

This information is extracted from tables and integrated in a formal base representation. The major challange of this step is to cope with a great number of different source formats and to transform them into a unified format.

Step 2: Extracting Information

Time-specific data and information about the dose rate have to be elicited. This is accomplished in three steps which are described in the following paragraphs.

(1) Identifying and Processing of Synonyms and Numeric Values

For simplifying subsequent processing all expressions that were identified as synonyms are converted into a consistent expression. As synonyms identified expressions are differently presented units, like ’days’, ’day’ or ’milliseconds’, ’msecs’ etc., and numeric expressions written in words, like ’single’, ’once’, ’three times’, ’four’. The convertion of numeric expressions into numbers is necessary for subsequent calculations.

(2) Eliciting Data Regarding the Duration and Frequency of Drug Administration

The duration should be identified by an expression commencing with numbers followed by a time-unit (e.g. 7 days), or two value or value-unit blocks connected by a dash (–) (e.g. 4 – 6 days, 5 days – 2 weeks). The latter describe the duration with a minimum and a maximum length.

The frequency can be identified by an expression like ’. . . twice a day . . .’, but also by an expression commencing with numbers followed by a time-unit like in ’. . . every 4 hours . . .’. The latter represents the period between two sequenced actions.

The problem is how to differentiate between two expressions commencing with numbers followed by a time-unit. Which one is the duration? Which one is the frequency? Therefore, we were looking for patterns or methods, which facilitate the differentiation of these expressions. We know that the expression specifying the duration must have a greater unit than the frequency or if the units are both equal the numeric value of the duration has to be greater.

If the expressions were correctly identified as duration and frequency, they are separated into their numeric parts and their unit-parts. If the frequency is stated as ’real’ frequency (e.g. ’twice a day’) it has to be converted into the period between two iterations. That is done by converting the time-unit into the next smaller time-unit and dividing the new interval by the number of occurrences. For example ’twice a day’ is first simplified to ’2/(day)’. Then it is converted to its next smaller unit to ’2/(24 hour)’ and this expression is transformed to ’(24 hour)/2’ = ’12 hour’.

One special case appears if a one-time application is prescribed. This is described by the term ’single dose’. In this case we set the value of the duration to ’1’ without stating a unit.

(3) Eliciting Leftover Data Regarding the Dose Rate of the Drug Administration

Expressions containing information about the dosage of a drug should contain, like already mentioned, the dose rate of the drug, the duration, and the frequency of the administration. Furthermore, the kind of application and additional information that is not specified any more can be stated. The sequence of this data may vary and the specification of the duration, the frequency, the kind of application, and additional information is optional. Hence, the applied procedure is the following:

We try to mark as many terms as possible besides the already found (duration, frequency). Then we elicit the dose rate, possibly the kind of application, omitting the duration and frequency. The remainding terms, if they are not solely stopwords, are added, too. The resulting terms are combined to the dose rate.

After this step we can generate an intermediate representation that can subsequently be transformed into Asbru. We will describe this task in the following section.

Step 3: Integrating the Extracted Information

For the administration of drugs, two types of plans are used that exist in Asbru, too:

- A plan that specifies the adminstration of a single dose of the drug. This administration is not further described.
- A plan that is running during a specified period activating a single dose plan in cyclical intervals.

If neither duration nor frequency is specified in the dosage-expression or ’single dose’ is specified, only the first plan is used, otherwise both plans are used.

A cyclical plan is characterized by

- the frequency of the invocation of the subplan,
- the duration,
- a starting shift,
- a finishing shift, and
- the number of the iterations

whereby only the first item is mandatory.

We have defined a schema for this intermediate representation that can represent different types of plans. These plans can be linked together with other plans in sequential or hierarchical order or in an iterative or cyclical order. Additionally, these plans may have time annotations that may contain uncertainties regarding the begin, the end, and the duration of the plan. Time annotations regarding the beginning and the ending are referring to the beginning or finishing of another plan that is explicitely stated. It is possible to state multiple time annotations and different reference plans for the beginning and finishing. In cyclical plans there is also a declaration regarding the frequency that specifies the time period between the finishing of the last iteration and the beginning of the subsequent iteration. This is particularly important in drug administration, whereby the application in short periods in a row is inhibited.

5 Case Study

For evaluating our TimeWrap method we used guidelines containing instructions for the administration of drugs from two different sources. The first guideline is the Preferred Practice Pattern (PPP) of the American Academy of Ophthalmology (AAO) for providing guidance for the pattern of practice for diagnosis and treatment of the patient with conjunctivitis [1]. The second guideline is a Clinical Practice Guideline of the American Optometric Association (AOA) for the care of patients with conjunctivitis [2].

Both documents contain instructions for drug administration, which are mainly represented in the form of tables. Tables can present data and information in a compressed form maintaining a concise and structured way. In doing so, a classification of certain data is already comprehensible and concise.

For further processing, the data cannot be used in the available form. It has to be transformed into an ”intermediate representation” as shown in Figure 1, in which the information is also machine-readable. One possibility for such an intermediate representation is the presentation in XML. At present we are fine-tuning an application that implements an existing method for representing information of a table in a semi-structured way by assigning semantics. We have obtained an example file for evaluation and testing which is shown in Listing 1.

Listing 1: Structured Information: example file about drug administration.

```
<?xml version=”1.0” encoding=”UTF−8”?>
<!DOCTYPE treatment SYSTEM ”treatment.dtd”>
<treatment>
    <cause name=”Gonococcus” person=”adult”>
        <drug dosage=”1g IM, single dose”
              name=”Ceftriaxone” />
    </cause>
    <cause name=”Chlamydia” person=”adult”>
        <drug dosage=”100 mg orally twice a day for 7 days”
              name=”Doxycycline” />
    </cause>
    <cause name=”Chlamydia” person=”child”>
        <drug dosage=”50 mg/kg/day orally in 4 divided doses for 10
           −14 days”
              name=”Erythromycin base” />
    </cause>
    <cause name=”Ophthalmia neonatorum” person=”neonate”>
        <drug dosage=”25−50 mg/kg IV or IM, single dose, not to
           exceed 125 mg”
              name=”Ceftriaxone” />
    </cause>
</treatment>
```

The discrete entries cover possible classes of dosage indications. The XML-file is parsed and every ’drug’-element is analyzed.

We start with analyzing the value of the dosage-attribute of the first drug-tag. We simplify discrete words and detect and convert synonyms into a consistent term. In the present expression no synonyms are detected, but the word ’single’ is converted to ’1’. Now we are trying to elicit the duration, but no numeric value followed by a time-unit is found. The same applies for the frequency. The only useful expression found is ’1 dose’ which indicates a nonrecurring plan. Therefore, eliciting the dose rate is not necessary, as the complete term for dosage including ’single dose’ is more significant. The resulting intermediate representation is shown in Listing 2.

Listing 2: Intermediate representation for administering a single dose.

```
<plan name=”Ceftriaxone: 1g IM, single dose”
      plan-id=”plan55131512” />
```

In the second drug-tag, the dosage-attribute contains the value ’100 mg orally twice a day for 7 days’. After identifying synonyms and numeric values written in words the term is converted to ’100 mg orally 2/ day for 7 day’. The duration is extracted with a value of ’7’ and the unit ’day’. The frequency is constituted as ’2/ day’ and therefore has to be translated to the length of the interval between two actions. The time-unit is detected with ’day’, hence the next smaller time-unit is ’hour’, whereas ’24 hour’ correspond to ’1 day’. The new interval of ’24 hour’ is now divided through the number of occurrences ’2’ and thus the result is a value of ’12’ with the unit ’hour’.

As we have extracted a frequency for the flow of the plan, we can reason on a re-occuring action that is implemented by a cyclical plan shown in Listing 3.

Listing 3: Intermediate representation for administering a drug in cyclical periods.

```
<plan name=”Doxycycline: 100 mg orally twice a day for 7 days”
      plan-id=”plan52769441”>
    <cyclical-plan plan-id=”plan5675512”>
        <frequency value=”12” unit=”hour” />
    </cyclical-plan>
    <duration>
        <min value=”7” unit=”day” />
        <max value=”7” unit=”day” />
    </duration>
</plan>
```

```
<plan name=”Doxycycline: 100 mg orally”
      plan-id=”plan54675512” />
```

The third drug-tag contains ’50 mg/kg/day orally in 4 divided doses for 10-14 days’ in the dosage-attribute. We can extract the duration which contains ’10’ as the minimum value and ’14’ as the maximum value, both with the unit ’day’. We cannot find an expression for the frequency, as it is covered by the compound unit of the dose rate. Hence, we do not know the weight of the person when we generate the plan, we cannot calculate the exact dose rate. Therefore, we must generate a plan that is specified more precicely during execution.

The compound unit of the dose rate contains the unit ’/day’. Thus we can set the frequency to ’1/day’ and can calculate the values and units for the intermediate representation: we convert it into the next smaller unit and get ’1/(24 hour)’ that is then calculated to the period between two iterations (’24 hour’). The resulting intermediate representation is shown in Listing 4.

Listing 4: Intermediate representation for administering a drug in cyclical periods.

```
<plan name=”Erythromycin base: 50 mg/kg/day orally in 4 divided
   doses for 10-14 days”
      plan-id=”plan97712431”>
    <cyclical-plan plan-id=”plan84476443”>
        <frequency value=”24” unit=”hour” />
    </cyclical-plan>
    <duration>
        <min value=”10” unit=”day” />
        <max value=”14” unit=”day” />
    </duration>
```

```
</plan>

<plan name=”Erythromycin base: 50 mg/kg/day orally in 4 divided
    doses”
      plan-id=”plan84476443” />
```

The dosage-attribute of the last drug-tag contains ’25-50 mg/kg IV or IM, single dose, not to exceed 125 mg’. Like in the primal tag we find the expression ’single dose’. Thus, we can reason a one-time application and the resulting intermediate representation is shown in Listing 5.

Listing 5: An Intermediate representation for administering a drug in a single dose.

```
<plan name=”Ceftriaxone: 25-50 mg IV or IM, single dose, not to
    exceed 125 mg”
      plan-id=”plan55496632” />
```

After we have finished the generation of the intermediate representation we can transform the data into Asbru plans. Therefore, we created XSLT templates that will do the transformation automatically. Besides templates for cyclical plans we have created templates for plans related in a sequential or hierarchical order, too.

By means of an XSLT processor, like e.g. Xalan[1], we can generate Asbru plans. The resulting XML-file is valid against the Asbru DTD, but is definitely not a complete Asbru plan. It is a subset representing temporal aspects that can be used within an Asbru protocol (see Listing 6) which has to be further augmented to represent a complete CGP.

Listing 6: Asbru protocol after transforming the intermediate representation.

```
<?xml version=”1.0” encoding=”UTF-8”?>
<plan-library>
  <plans>
    <plan-group>
      <plan name=”Ceftriaxone: 1g IM, single dose”>
        <plan-body>
          <user-performed/>
        </plan-body>
      </plan>
      <plan name=’Doxycycline: 100 mg orally twice a day for 7 days
          ’>
        <defaults>
          <time-annotation>
            <time-range>
              <duration>
                <minimum>
                  <numerical-constant value=”7” unit=”day”/>
                </minimum>
                <maximum>
                  <numerical-constant value=”7” unit=”day”/>
                </maximum>
              </duration>
            </time-range>
            <now/>
          </time-annotation>
        </defaults>
        <plan-body>
          <cyclical-plan>
            <start-time>
```

[1] http://xml.apache.org/xalan-j/index.html

```
        <time-annotation>
          <now/>
        </time-annotation>
      </start-time>
      <cyclical-plan-body>
        <plan-activation>
          <plan-schema name=”Doxycycline: 100 mg orally”/>
        </plan-activation>
      </cyclical-plan-body>
      <cyclical-time-annotation>
        <time-range/>
        <set-of-cyclical-time-points>
          <time-point>
            <numerical-constant value=”0”/>
          </time-point>
          <offset>
            <numerical-constant value=”0”/>
          </offset>
          <frequency>
            <numerical-constant value=”12” unit=”hour”/>
          </frequency>
        </set-of-cyclical-time-points>
      </cyclical-time-annotation>
    </cyclical-plan>
  </plan-body>
</plan>
<plan name=”Doxycycline: 100 mg orally”>
  <plan-body>
    <user-performed/>
  </plan-body>
</plan>
<plan name=”Erythromycin base: 50 mg/kg/day orally in 4
   divided doses for 10-14 days”>
  <defaults>
    <time-annotation>
      <time-range>
        <duration>
          <minimum>
            <numerical-constant value=”10” unit=”day”/>
          </minimum>
          <maximum>
            <numerical-constant value=”14” unit=”day”/>
          </maximum>
        </duration>
      </time-range>
      <now/>
    </time-annotation>
  </defaults>
  <plan-body>
    <cyclical-plan>
      <start-time>
        <time-annotation>
          <now/>
        </time-annotation>
      </start-time>
      <cyclical-plan-body>
        <plan-activation>
          <plan-schema name=”Erythromycin base: 50 mg/kg/day
              orally in 4 divided doses”/>
        </plan-activation>
      </cyclical-plan-body>
      <cyclical-time-annotation>
```

```
              <set-of-cyclical-time-points>
                <time-point>
                  <numerical-constant value="0"/>
                </time-point>
                <offset>
                  <numerical-constant value="0"/>
                </offset>
                <frequency>
                  <numerical-constant value="24" unit="hour"/>
                </frequency>
              </set-of-cyclical-time-points>
            </cyclical-time-annotation>
          </cyclical-plan>
        </plan-body>
      </plan>
      <plan name="Erythromycin base: 50 mg/kg/day orally in 4
          divided doses">
        <plan-body>
          <user-performed/>
        </plan-body>
      </plan>
      <plan name="Ceftriaxone: 25-50 mg/kg IV or IM, single dose,
          not to exceed 125 mg">
        <plan-body>
          <user-performed/>
        </plan-body>
      </plan>
    </plan-group>
  </plans>
</plan-library>
```

6 Results, Benefits, and Limitations

We have shown that by the means of our TimeWrap method time-referenced data of a simple or cyclical recurring process can be extracted from particular data and out of it planning process representations can be created. These processes are first presented in an 'intermediate representation' and afterwards transformed into a formal language, in our case *Asbru*.

Thereby, both the often recurring processing of specifications for drug administration and the troublesome generation of Asbru plans can be prevented. Asbru is a very complex language and not easy to code. Tools that would assist in the process could be very useful. Thus, the knowledge-intensive task of the human plan editor is machine supported, but also the amount of time the process takes can be decreased. By means of the intermediate representation the flows of the clinical protocols can be better structured and presented in a concise form. The intermediate representation can be used to automatic transform them by defined rules to the final representation *Asbru*.

Currently, our method handles simple specifications, which are limited by a particular form of information declaration. That means the limitation to one drug that is administered during a particular interval in invariant distances of time in a constant dose rate. Besides these limitations also other dependencies of the administration of drugs like other medications or treatments or the dependencies of special parameters cannot be processed.

7 Conclusion

We have presented a three-step wrapper method to analyze and structure semi-structured data and information that is used to generate a formal representation. We are aiming to support treatment planning within the medical domain and have therefore illustrated our approach with examples from conjunctivitis and the guideline-representation language Asbru.

It is very important to notice the following issues.

Example of Drug Administration Used As Illustration. The three-step wrapper method presented is illustrated using examples of drug administration. However, TimeWrap can be applied to similar problem characteristics as well. We have chosen the drug administration example because it illustrates our methods more easily.

Automatic vs. Semi-Automatic Transformation. We are not aiming towards an automatic solution to transform different guidelines into formal representations. We are aiming of the automation of defined semi-structured guidelines' components, which can interactively be composed to an overall transformation. However, this last step is done manually.

No Natural Understanding Analysis. We are not performing any natural understanding analysis to capture the content of guidelines. Our starting points are semi-structured guidelines' components, which can be processed without syntactic and semantic analysis in the sense of natural language understanding analysis. We definitely need information about the syntactical shape of the text, but more in a structural sense. Therefore, our methods benefit from simplicity, on the one hand, and utilize the known semi-structure forms of the guidelines' components, on the other hand.

Our TimeWrap method can be improved according to its ontological foundation. At present, specific expressions and synonyms are defined directly. In the future we will implement this by using an ontology.

In the same way the methods for the calculation of the frequency can be improved. The frequent administration of drugs is not distributed equally through the day. In most cases the application will be in the daytime. For example the administration three times a day will not be every eight hours, but perhaps in the morning, at noon and in the evening. On the other hand, medical domains exist where administration round the clock is necessary.

In the next steps, we will improve our proposed wrapper method and extend the applicability to other typical patterns within clinical guidelines and protocols. The overall goal is to design and develop ontology-based wrapper methods, which are applicable to particular classes of knowledge representation, but guided by the idea of clinical guidelines and protocols.

Acknowledgements

This project is supported by ”Fonds zur Förderung der wissenschaftlichen Forschung FWF” (Austrian Science Fund), grant P15467-INF.

References

[1] American Academy of Ophthalmology (AAO): Preferred Practice Patterns Committee. Cornea/External Disease: Conjunctivitis. American Academy of Ophthalmology (AAO), 1998.

[2] American Optometric Association (AOA): Consensus Panel on Care of the Patient with Conjunctivitis. Optometric Clinical Practice Guideline, Care of the Patient with Conjunctivitis. American Optometric Association (AOA), 1995, revised 1999.

[3] R. Baumgartner, S. Flesca, and G. Gottlob. Visual Web Information Extraction with Lixto. In *Proceedings of VLDB*, 2001. http://www.dbai.tuwien.ac.at/proj/lixto/download.html.

[4] J. Bury, J. Fox, and S. D. The PROforma Guideline Specification Language: Progress and Prospects. In *Proceedings of the First European Workshop on Computer-Based Support for Clinical Guidelines and Protocols (EGWLP 2000)*, volume 83: Studies in Health Technology and Informatics, Leipzig, November 2000.

[5] R. Kosara and S. 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, 2001.

[6] L. Liu, C. Pu, and W. Han. XWRAP: An XML-enabled Wrapper Construction System for Web Information Sources. In *International Conference on Data Engineering (ICDE)*, pages 611–621, 2000.

[7] S. Miksch, Y. Shahar, W. Horn, C. Popow, F. Paky, and P. Johnson. Time-Oriented Skeletal Plans: Support to Design and Execution. In *Fourth European Conference on Planning (ECP'97)*. Springer, September 1997.

[8] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. Greenes, R. Hall, P. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E. Shortliffe, and M. Stefanelli. Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. *Journal of the American Medical Informatics Association (JAMIA)*, 10(1):52–68, Jan-Feb 2002.

[9] K.-A. Polvani, A. Agrawal, B. Karras, A. Deshpande, and R. Shiffman. *GEM Cutter Manual*. Yale Center for Medical Informatics, 2000. http://ycmi.med.yale.edu/GEM/.

[10] A. Seyfang, R. Kosara, and S. Miksch. Asbru 7.3 Reference Manual. Technical Report Asgaard-TR-2002-1, Vienna University of Technology, Institute of Software Technology & Interactive Systems, Vienna, Austria, Europe, 2002. http://www.asgaard.tuwien.ac.at/asbrusyntax.html.

[11] R. Steele and J. Fox. Tallis PROforma Primer - Introduction to PROforma Language and Software with Worked Examples. Technical report, Advanced Computation Laboratory, Cancer Research, London, UK, 2002.

[12] V. Svátek, T. Kroupa, and M. Růžička. Guide-X – a step-by-step, markup-based approach to guideline formalisation. In B. Heller, M. Löffler, M. Musen, and M. Stefanelli, editors, *Proceedings of the First European Workshop on Computer-Based Support for Clinical Guidelines and Protocols (EWGLP 2000)*, volume 83: Studies in Health Technology and Informatics, Amsterdam, 2000. IOS Press.

[13] V. Svátek and M. Růžička. Step-By-Step Mark-Up of Medical Guideline Documents. In G. Surjan et al., editors, *Health Data in the Information Society. Proceedings of Medical Informatics Europe (MIE2002)*, pages 591–595, Budapest, HU, 2002. IOS Press.

[14] P. Votruba, S. Miksch, and R. Kosara. Linking clinical guidelines with formal representations. In M. Dojat, E. Keravnou, and P. Barahona, editors, *Artificial Intelligence in Medicine. Proceedings of the 9th Conference on Artificial Intelligence in Medicine in Europe (AIME 2003)*, Berlin, 2003. Springer.