Medical Event Representation and Reasoning for Chinese Clinical Guideline

Zhonghua Zhu, Pingfang Tian
Wuhan University of Science and Technology, Wuhan 430065, Hubei, China
Intelligent Information Processing and Real-time Industrial System Hubei Province Key Laboratory, Wuhan Hubei, China
webartisan@yeah.net, 24958320@qq.com

ABSTRACT: Event is the hot topics in the study of clinical guidelines. In this paper, we define medical event narrowly from the view of medical operation and represent the event of clinical guideline in a format which can be understood by computer. Meanwhile, we mine hidden relationships between events with user-defined rules in Stardog. Then, we reason the next executable event by preforming SPARQL queries. We take a segment of Chinese clinical guideline as example and construct an event ontology to verify the feasibility of this framework. Furthermore, event knowledge is not only computerized in this framework, but also can link to drug ontology and disease ontology. So the event ontology can be used to some medical application such as intelligent drug recommendation system.

1 INTRODUCTION

Clinical guidelines is the multidisciplinary clinical guidance developed systematically, which can help clinical doctors, patients and other stakeholders to make the appropriate treatment, choice and decision making with specific clinical problems[1]. The guidance is usually presented by clinical actions should be taken. Meanwhile, these clinical actions will lead to different states of diagnosis and treatment, which we call clinical events or events for short.

It is an important way for the formal representation of guideline knowledge to be applied in computer applications. However, operation and event are important elements of clinical guideline, so event is of great significance for the research of clinical guideline. In order to make the guideline knowledge computer-comprehensible, scholars all over the world have done depth research on it. Then, several excellent framework and tool appear, such as Arden Syntax [2], EON [3], Asbru [4], GLARE [5], Prestige [6], GLIF3 [7], SAGE [8]. Medical knowledge of clinical guideline can be well represented in these frameworks and tools, but the study on event isn’t enough.

In this paper, we focus on how to transform the Chinese medical event descripted with natural language into semantic data which can be understood by computer, and mining the hidden relationships between events. We take a narrow sense of event and event in guideline is defined as all kind of operations and behaviors. Different form SNOMED CT, in which the definition of event is various special status should be avoided, namely accident [9]. The main study of this paper is representing the event in clinical guideline and reasoning the relationships between them.

Relevant conceptions and definitions about medical event are introduced in the second section of this paper. In the third section, we propose a framework to represent event knowledge in clinical guideline. In forth section, we choose a segment of antimicrobial drugs guideline as the object to be handled, representing and reasoning events of it. At last, the advantages and disadvantages of the framework are discussed and we put forward challenges about further work.

2 MEDICAL EVENT REPRESENTATION

2.1 Definition of medical event

The first definition of event can be dated back to philosofy and artificial Intelligence. Some well-known mechanisms appear in the research process of event, such as Event Calculus [10], Situation Calculus [11], and ALAN [12]. The definition of event is controversial in this field all the time. Francois Bremond defined event in paper [13] like this: an event is one or several change(s) of state at two successive time instants or on a time interval. Different from event in these mechanisms, elements
of medical event usually don’t contain subject, time and place. In clinical guideline, some treatment operations can be regarded as events, as well as clinical actions, such as “oral amoxicillin”, “intravenous drip ceftriaxone”. In SNOMED CT, medical event is accident. However, the medical event we discuss in this paper is clinical operations performed in guideline.

2.2 Temporal relationship

13 fundamental temporal relationships are mentioned by Francesco Mele in paper [14], with which the binary relationship between events is presented. The figure of temporal relationships is as following.

![Figure 1. 13 temporal relationships.](image)

In figure 1, we can see that the start or finish time of event A and event B are different, the different temporal relationships are described. However, there is no explicit time for medical event, and general expression of event is in the timeline, such as “during …”, “before …”, “after …”. So, it is unnecessary to define an exact time property for time point, and we just use a binary relationship to demonstrate the order of priority between two events. We incorporate time instant or interval into the next event as its precondition when the time expression appears.

2.3 Logic relationship

There are logic relationships between medical events, such as “AND”, “OR”.

“AND” is usually used to describe two events need to be performed in one therapy link. The effect of treatment will be counterproductive if anyone of the two events isn’t be executed. It means that two events with “AND” are indispensable. For example, in the expression “during the use of chloramphenicol, review hemogram periodically and monitor poisonousness of blood”, we can exact three medical events, “use chloramphenicol”, “review hemogram periodically”, “monitor poisonousness of blood”. To ensure the effect of chloramphenicol, all of these three events should be performed.

“OR” demonstrates two events are optional, and the expected effect will be achieved with any one of the two. For example, in the treatment of acute bacterial pharyngitis and tonsillitis, we can inject procaine penicillin into muscle or take penicillin V orally. It means that both “inject procaine penicillin into muscle” and “take penicillin V” are independent. And either of them can take expected effect for acute bacterial pharyngitis and tonsillitis.

3 EVENT REPRESENTING FRAMEWORK

Event knowledge described in natural language can be transformed into computer-comprehensible format with ontology. Actually, ontology is the formal representation for some conceptions and mutual relationships in special field. Expression of treatment or operation can be regard as a procedure in clinical guideline, of which a series of operations can be summed up in common properties.

In this paper, we choose OWL [15] and define some OWL properties and classes to represent events and relationships. The process is illustrated with a chart below.

![Figure 2. The framework of event representation.](image)

In figure 2, we define some OWL classes and properties as follows:

- **Procedure**: a class is used to represent a therapeutic process or solution, including all components and mutual temporal and logic relationships.
- **Component**: a class contains an operation, condition and extra remark.
- **Operation**: a class represents atomic operation in treatment and it is composed of action and operational objects. Meanwhile, it can link to other ontology, such as Drug and Disease.
- **hasPreviousOp**: an object property used to link component to previous component.
- **hasFollowingOp**: an object property used to link component to next component.
**hasIndispensableOp**: an object property used to link component to its indispensable component.

**hasOptionalOp**: an object property used to link component to its optional component.

**hasOperation**: an object property used to link component to its containing operation individual.

**hasComponent**: an object property used to link procedure individual to component individual.

**condition**: an data property represents the precondition of event including symptom, reaction and so on. Furthermore, it can be used to express time instant or interval.

**action**: an data property represents the way in which event will be operated.

**subject**: an data property show the subject operated in event.

**remark**: an data property used to express extra information.

Among the properties, **hasPreviousOp** and **hasFollowingOp** are mutual inverse. Namely, **hasPreviousOp** has an inverse property that is **hasFollowingOp**, **hasFollowingOp** has an inverse property that is **hasPreviousOp** in turn. With triple expression, \(<A, \text{hasPreviousOp}, B>\) equals to \(<B, \text{hasFollowingOp}, A>\). The property **hasIndispensableOp** and **hasOptionalOp** are symmetric. In OWL, if a property \(P\) is symmetric, and the property relates individual A to individual B then individual B is also related to individual A via property \(P\). Namely \(<A, \text{hasIndispensableOp}, B>\) equals to \(<B, \text{hasIndispensableOp}, A>\) and \(<A, \text{hasOptional}, B>\) equals to \(<B, \text{hasOptional}, A>\). Put another way, the property is its own inverse property.

### 4 EVENT REASONING

Event reasoning means mining the hidden event relationships according existing relationships or reasoning next executable events according current conditions.

Similar to SQL query language, we can know the next executable event with a query on event ontology. For example, if there is a relationship **hasFollowingOp** between event \(E1\) and \(E2\), we can reason the next event is \(E2\) according the relationship **hasFollowingOp**. Meanwhile, we can reason out other related information about \(E1\) and \(E2\).

OWL is a good way to represent object’s properties and relationships, but the biggest weakness is scare reasoning capacity. With OWL, we can just do some simple reasoning. In the framework we proposed, **hasPreviousOp** and **hasFollowingOp** is mutual inverse, **hasOptionalOp** and **hasIndispensableOp** is symmetric. With OWL’s own reasoning ability, there are following tenable reasoning rules.

\[
\text{hasPreviousOp}(E1, E2) \rightarrow \text{hasFollowingOp}(E2, E1), \\
\text{hasOptionalOp}(E1, E2) \rightarrow \text{hasPreviousOp}(E2, E1), \\
\text{hasFollowingOp}(E1, E2) \rightarrow \text{hasOptionalOp}(E2, E1), \\
\text{hasIndispensableOp}(E1, E2) \rightarrow \text{hasIndispensableOp}(E2, E1).
\]

In rules above, \(E1\) and \(E2\) are events, and the word before bracket means the relationships between \(E1\) and \(E2\). Obviously, it is not enough to reason the hidden event relationships. In order to execute complete reasoning and mining relationships fully, it is necessary to define some other user-defined rules. A user-defined rules approach complements the OWL axiom-based approach nicely and increases the expressive power of a reasoning system form the user’s point of view. Fortunately, Stardog is the RDF database that comprehensively supports both axioms and rules. Some problems are simply a better fit form a rules-based approach to modeling and reasoning than to an axioms-based approach. In this paper, we define some rules as follows:

\[
\text{hasPreviousOp}(E3, E2) \rightarrow \text{hasPreviousOp}(E1, E2), \\
\text{hasOptionalOp}(E1, E3). \\
\text{hasPreviousOp}(E3, E2) \rightarrow \text{hasPreviousOp}(E1, E2), \\
\text{hasIndispensableOp}(E1, E3). \\
\text{hasFollowingOp}(E1, E3) \rightarrow \text{hasFollowingOp}(E1, E2), \\
\text{hasOptionalOp}(E2, E3). \\
\text{hasFollowingOp}(E1, E3) \rightarrow \text{hasFollowingOp}(E1, E2), \\
\text{hasIndispensableOp}(E2, E3). \\
\text{hasPreviousOp}(E1, E3) \rightarrow \text{hasPreviousOp}(E1, E2), \\
\text{hasOptionalOp}(E2, E3). \\
\text{hasPreviousOp}(E1, E3) \rightarrow \text{hasPreviousOp}(E1, E2), \\
\text{hasIndispensableOp}(E2, E3). \\
\text{hasFollowingOp}(E3, E2) \rightarrow \text{hasFollowingOp}(E1, E2), \\
\text{hasOptionalOp}(E1, E3). \\
\text{hasFollowingOp}(E3, E2) \rightarrow \text{hasFollowingOp}(E1, E2), \\
\text{hasIndispensableOp}(E1, E3).
\]

We can realize these rules with Stardog rules syntax as follows:

\[
[\] \text{a rule:SPARQLRule} ; \\
\text{rule:content ""
PREFIX \\
gd:<http://www.semanticweb.org/ontologies/Guideline#> \\
IF { \\
?event1 gd:hasPreviousOp ?event2 . \\
?event1 gd:hasOptionalOp ?event3 . \\
} \\
THEN { \\
?event3 gd:hasPreviousOp ?event2 . \\
} \\
""}.
\]

The realization of other rules are similar to this.
5 IMPLEMENTATION AND APPLICATION

Using Guidelines for Clinical Use of Antibacterial (2015) as an example, we intercept a segment from it and represent the events, logic and temporal relationships. The segment is as following:

In early, one can take oral amoxicillin. If local haemophilus influenzae is common, one can also take amoxicillin/clavulanic acid.

If it is ineffective in patients after three days, it can be consider as resistance to penicillin streptococcus pneumoniae infection. One can take oral amoxicillin/clavulanic acid in high dose or intravenous drip ceftriaxone.

Figure 3. A segment of Guideline for Clinical Use of Antibacterial (2015)

From the expression above, we can exact some operations as follows: “oral amoxicillin”, “oral amoxicillin/clavulanic acid”, “intravenous drip ceftriaxone”. Figure 3 shows the relationships between events, where the solid line means the relationships can be marked in OWL and the dotted line shows the relationships reasoned out with rules.

Figure 4. Event relationships in segment.

In Protégé, we define the some classes and properties about event, representation of the example in Protégé is shown in figure 5

Table 1. Before and after comparison about the amount of event relationships.

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Before reasoning</th>
<th>After reasoning</th>
<th>Increasing amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasPreviousOp</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>hasFollowingOp</td>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>hasOptionalOp</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

Meanwhile, we can reason the next executable event according to the given condition. For example, to the condition “ineffective in patients after taking three days”, we execute SPARQL query as follows:

```sparql
select ?action ?drug ?remark where {
  ?s gd:Condition ‘ineffective in patients after taking three days’.
  ?s gd:hasOperation ?op.
  OPTIONAL{?s gd:Remark ?remark}.
  ?op gd:Subject ?drug.
}
```

The result is shown in figure 6.

The results above demonstrate that event and its relationships can be represented well with the event representation framework we proposed and numbers of hidden relationships are reasoned out. Meanwhile, the next executable event can be also reasoned out. It verifies the feasibility of this framework.

6 CONCLUSION AND FUTURE WORK

According to the characteristic of expression in Chinese clinical guideline, we conclude common properties of operations and define medical event narrowly. Meanwhile, we represent events and event relationships with ontology and reason out hidden relationships with Stardog user-defined rules. By preforming SPARQL query on ontology, we can get next executable events. This framework has following advantages: (a) event properties and
relationships can be represented well; (b) numbers of hidden relationships can be reasoned out with user-defined rules; (c) event ontology can be linked to other medical ontologies, such as Drug and Disease; (d) it is easy to develop medical applications with event ontology, such as intelligent drug recommendation system.

However, there are some problems and challenges for this framework, namely the construction of large-scale event ontology. We construct the ontology manually in this paper at present. In future work, we will establish large-scale event ontology with some automated or semi-automated approach and link it to other medical ontologies to form a complete knowledge base.

7 ACKNOWLEDGMENT

This work is partially supported by a grant from the NSF (Natural Science Foundation) of China under grant number 60803160 and 61272110, the Key Projects of National Social Science Foundation of China under grant number 11&ZD189, and it was partially supported by a grant form NSF of Hubei Prov. of China under grant number 2013CFB334. It was partially supported by NSF of educational agency of Hubei Prov. Under grant number Q20101110, the State Key Lab of Software Engineering Open Foundation of Wuhan University under grant number SKLSE2010-09-07, the Teaching Research Project of Hubei Province under grant number 2011s005 and the Wuhan Key Technology Support Program under grant number 2013010602010216.

REFERENCES


