Generation of Semantic Data from Guidelines for Rational Use of Antibiotics

Qing Hu[1,2], Zhisheng Huang[3], and Jinguang Gu[1,2]

[1] Faculty of Computer Science and Engineering, Wuhan University of Science and Technology, Wuhan 430065, China
[2] Key Laboratory of Intelligent Information Processing and Real-time Industrial System of Hubei Province, Wuhan 430065, China
[3] Department of Computer Science, VU University Amsterdam, The Netherlands

Abstract. Rational use of antibiotics has become an important issue in medical practice and health care. Clinical guidelines are one of the most useful knowledge resources for rational use of antibiotics. As the monitoring of rational use of antibiotics involves complex knowledge of guidelines analysis and management process, traditional way of human intervention is not sufficient to monitor rational use of antibiotics effectively. Therefore, we introduce the semantic technology to semi-automatically transform the knowledge contained in the clinical guidelines and get the semantic data. In this paper, we firstly investigate how to obtain the semantic data from the guidelines knowledge which are described in natural language, then propose an approach to transformation of guidelines knowledge into semantic data, which can be loaded into SeSRUA, a Semantically-Enabled System for Rational Use of Antibiotics. Finally we report how to implement the proposed approach in SToGRUA, a system of Semantic Transformer of guidelines for Rational Use of Antibiotics, as a tool of SeSRUA.

Keywords: Semantic Technology; Antibiotics; Rational use of drug; Knowledge Acquisition

1 Introduction

Rational use of antibiotics is an important issue in medical practice and health care. However, only relying on human intervention is not sufficient to monitor rational use of antibiotics effectively. Therefore, in the monitoring process, a semantically-enabled system for knowledge management can be adopted to generate the monitoring knowledge with the semantic data format. Then the monitoring knowledge will be used in the management of the medical information systems to realize the automatic monitoring of the rational use of antibiotics.

The medical authorities have published the guidelines for rational use of antibiotics. In China, the main medical professional organizations(i.e. The Chinese Medical Association, Pharmacy Administration Commission of Chinese Hospital Association, and Hospital Pharmacy Committee of Chinese Pharmaceutical Association) have promulgated the clinical guidelines for rational use of antibiotics in 2008[1]. The guidelines have become one of the most important knowledge resources for rational use of antibiotics. Therefore, the key issue here is how to transform the guidelines in natural language into semantic knowledge, so that they can be used in a semantically-enabled system.

Computerized Clinical guidelines, alternatively called Computer-Interpretable guidelines (CIGs), implement the guidelines in computer-based decision support systems. Although several formalisms of computer-interpretable guidelines, such as PROforma[2], Asbru[3], EON[4], and GLIF[5], have been proposed to formalize clinical guidelines. Those CIGs rely on an independent system of reasoning and processing. In this paper, we are more interested in a lightweight approach to accommodating guidelines knowledges. Taking into account of the description format of the guidelines being transformed is relatively fixed and the scale of the guidelines knowledge being preprocessed is not so large, we can preprocess the knowledge contained in the guidelines manually to obtain a set of rules with the pattern "if ..., then ...", so that they can be processed by using DCG (Definite Clause Grammar)[4] in the logic programming language Prolog. DCG is a lightweight tool to parse natural language text. With DCG, we can obtain semantic data, without relying on complex natural language processing tools.

In order to get the semantic knowledge contained in the guidelines that can be used in a semantic-enabled system for rational use of antibiotics, a series of work have been done, which include: i) distinguishing three levels of knowledge rules, ii) transforming rules at different levels, and iii) use the semantic data of guidelines in the Semantically-enabled System for Rational Use of Antibiotics system SeSRUA[6].

The rest of this paper is organized as follows: Section2 analyzes the guidelines. Section3 proposes the transformation methods between different levels rules in guidelines and presents the System of Transformer guidelines for Rational Use of Antibiotics(SToGRUA). Section4 evaluates the transformation results. Section5 make the conclusions.

2 Analysis of Knowledge in the Guidelines

Since the diseases caused by bacterial infections are found in various clinical sections, antibiotics become one of the most commonly used drugs in clinical application. While universal use of antibiotics save the lives of many patients, the adverse reactions caused by the abuse of antibiotics appear gradually. Therefore, main medical professional organizations have promulgated the clinical guidelines to manage and monitor the rational use of antibiotics.

In this paper, we will focus on the guidelines for rational use of antibiotics in China[1]. The guidelines in Chinese consist of four parts. Part I mainly describes the basic principles of clinical use of antibiotics in therapeutic application, prophylactic application and special pathology or patient physiological conditions application. Part II enacts the management rules of clinical use of antibiotics. Part III emphasizes the corresponding indications and precautions

[4] http://en.wikipedia.org/wiki/Definite_clause_grammar

on concrete antibiotic drugs. Part IV elaborates the treatment principles and pathogen treatment of various types of antibiotic infections.

Those four parts describe the main functions of clinical guidelines for rational use of antibiotics, which include on what kinds of condition a medical professional should use what kinds of antibiotics, on what kinds of clinical indications an antibiotic drug can be used and how to use. These relatively simple knowledge of clinical guidelines are relatively fixed and well structured. For example, the following statement excerpted from the clinical guidelines points out the diagnosis indications for deciding whether or not to use an antibiotic drug.

```
Example 1: According to the patient's symptoms, signs, and blood,
urine and other laboratory examination results, the patient is
preliminarily
diagnosed as bacterial infections and through the pathogen examination
the patient was preliminarily diagnosed as  bacterial infections with
indications for use of antibiotics.
```

It is rather easy for a human to understand the meaning of the guidelines in natural language text, like those in Example 1. However, it is rather difficult for computers to understand those guidelines. Considering the fact that those guidelines are quite limited, we can convert those knowledge contained in the guidelines into a format so that they can be handled by computer systems more easily. Thus, an easy and simple solution is to generate a set of rules with the pattern "if ..., then ..." by manual. For example, the text in Example 1 can be transformed into the rules, which are shown in Example 2.

```
Example 2: According to the patient's symptoms, signs, and blood,
urine and other laboratory examination results, the patient is
preliminarily diagnosed as suspected bacterial infections, then
do the followings.

1.If the patient has blood laboratory test results and has been
preliminary diagnosed as bacterial infections and pathogenic
diagnosed as bacterial infections, then he indeed has indication
to use antibiotics.

2.If the patient has urine laboratory test results and has been
preliminary diagnosed as bacterial infections and pathogenic
diagnosed as bacterial infections, then he indeed has indication
to use antibiotics.
```

The description format of Example 2 represents the logical relationship clearly. However, in the specific diagnosis process, without the specific indication values which can be checked in patient data, those knowledge are insufficient for a judgment whether or not an antibiotic should be used. So we do further processing to transform the knowledge in Example 2 into one with concrete values in patient data, like those shown in Example 3.

```
Example 3: According to the patient's symptoms, signs, and blood,
```

```
urine and other laboratory examination results, the patient is
preliminarily diagnosed as suspected bacterial infections, then
 do as follows:

1.If the patient has blood laboratory test, according to concrete
item and its value(witch refer to the patient examination report),
and has been preliminary  diagnosed as bacterial infections and
pathogenic inspection report show that item and its value (with
refer to the patient examination report), then he indeed has
indication to use antibiotics.

2.If the patient has urine laboratory test, according to concrete
item and its value(witch refer to the patient examination report),
and has been preliminary diagnosed as bacterial infections and
pathogenic inspection report show that item and its value (with
refer to the patient examination report), then he indeed has
indication to use antibiotics.
```

The work converting the rules from Example 1 into Example 3 is called the preprocessing of the guidelines. The main purpose of this preprocessing is to convert the guidelines knowledge into a set of concrete rules which can be used to check the patient data directly. After that preprocessing, we can finally get the fixed format of "if...then..." to describe the clinical guidelines knowledge, just like those with the format of Example 3. After we obtain a set of concrete rules with the pattern "if ..., then ..." in natural language text, we can use DCG in the logic programming language Prolog to parse the preprocessed guidelines knowledge to obtain their semantic data.

In order to distinguish the guidelines rules in every preprocessing stage, we classify the guidelines rules into three different levels of rules, i.e., abstract rules, concrete rules, and semantic rules. The task of the SToGRUA system is to generate the semantic rules that can be used in SeSRUA, a semantically-enabled system for raitonal use of antibiotics.

Abstract Rules Knowledge in the guidelines provides the rules to guide the rational use of antibiotics, so the knowledge in the guidelines that cannot be used to guide our actions directly are called *abstract rules*. In the guidelines, the guide-makers made those abstract rules for professionals only. For this, knowledge in the guidelines described by natural language requires extracting and form-transformation into concrete rules.

By screening the contents of the guidelines, we extract the knowledge and preliminary transform them into "if ...then ..." form, which can be dealt by a computer. We define the "if ... then ..." form of knowledge as Abstract Rules. Example 1 is an example of abstract rules. First, make sure knowledge can be converted into the "if ... then ..." form.

Concrete Rules Concrete rules are relative to abstract rules. It means to make abstract rules in the guidelines more concrete so that hyponymy, entailment and other logical relationship between the knowledge will become more clear. The extracted abstract rules include No-doing rules, How-to-do rules and Experience-based rules. It is obvious that the first two rules are instructive, while the third one is not so instructive, namely, the concrete index values are not clearly stated in the rules. In such kind conditions, the participation of medical experts will be needed to make the rules more concrete so that they can be used for a checking with computers. The concrete processed rules are called concrete rules.

With the participation of medical professionals, based on the values in hospital examination reports and the relationship between the values (such as "and" or "or" relationship), the abstract rules can be specified. Concrete rules are basic for the description of logical relationship.

Semantic Rules Semantic web technology provides a common framework for data sharing and reusing. The experience of antibiotics selection need be shared. There are enormous amount of accumulated medical and experiential data in antibiotic selection. These scattered information resources in different hospital system are beyond practitioners' learning ability and it is impossible for practitioners to fully acquire them. Thus, hospitals have to rely on computera to process the data. In order to help the computer to understand the knowledge, semantic web technology can be used to deal with the representation of those knowledge and data.

The community of the semantic technology has developed several international standards for the knowledge representation language of semantic data, such as RDF (Resource Description Framework) and OWL (Web Ontology Language), which have been found many applications in biomedical domains. Therefore, in order to achieve automatic monitoring of rational use of antibiotics, those standards (e.g. RDF/RDFS) will be used to describe pharmacological knowledge and pharmacokinetics-based rational antibiotic-use method. The natural language description in documents and guidelines that related to rational use of antibiotics can be transformed into their semantic description. By transforming, hyponymy, entailment and other logical relationship between the semantic knowledge will become so clear that they can be mapped to corresponding ontology knowledge graph[7].

Semantic rules are knowledge description that transformed from concrete rules. Knowledge described in nature language will be transformed into semantic technology-based descriptions that can be handled by computer directly.

3 Semantic Transformer of Guidelines for Rational Use of Antibiotics (SToGRUA)

SToGRUA is the system of semantic transformer of guidelines for rational use of antibiotics. Through the SToGRUA system, we get the semantic rules of the knowledge in the guidelines of rational use of antibiotics. Of course, it is not the case that all of the guidelines knowledge can be represented as a RDF/OWL data. Some of those knowledge may need a higher level of knowledge representation language, like those in rule-based language. In this paper, we will focus on the knowledge which can be represented in those lightweight languages, i.e., RDF/RDFS/OWL. We leave the rule-based formalisms of the guidelines knowledge in the future work.

In order to get those semantic data (i.e., semantic rules of the guidelines), firstly, we transform the abstract rules into concrete rules. Then we transform the concrete rules into semantic rules.

3.1 Transformation from Abstract Rules to Concrete Rules

Abstract rules in the guidelines contain many instructional rules for using antibiotics rationally and the rules expression form is in the "if...then..." format. With the help of medical experts, this form of knowledge can be further transformed into concrete rules in following two steps:

Some of the abstract rules are specified while some are not. The specified abstract rules are concrete rules. Therefore, the first step is to distinguish the concrete rules from abstract rules. For instance, if we extract the indication knowledge in the guidelines, the abstract rules will be specified and do not need transformation. However, as Example 2, we need the participation of medical experts since they can turn the abstract medical test results into concrete medical rules. Based on corresponding values on the medical examination reports, we can then transform the abstract rules into the concrete rules in the format of "if ... then ...".

3.2 Design for Semantic Transformation of Concrete Rules

A considerable part of the knowledge in the guidelines involves indications and contraindications in the antibacterial instructions. For instance, "If infection caused by hemolytic streptococcus, then penicillin should be used" and "If infection caused by Streptococcus pneumoniae, then penicillin should be used". These knowledge can be used directly in detecting the data on the electronic medical records. There exist some well-known pharmacological and pharmaceutical semantic data set, such as Drugbank[5], which are usually represented as the RDF Ntriple format. For example, the description of a drug's indications uses a predicate (e.g. Indication) followed by the indication descriptions in natural language as follows:

```
<http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugs/DB00007>
<http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/indication>
"For treatment of prostate cancer, endometriosis, uterine
fibroids and premature puberty" .
```

Obviously, we need more fine-grained knowledge about indications, because the indication knowledge in a text cannot be used directly in the system. We have to transform them into the semantic data. Hence, a particular predicate "indications" is introduced to represent the set of all the indications.

[5] http://www.drugbank.ca/

Meanwhile, for each disease symptom description involves its causes description, we introduce the predicate 'CausedBy' and the additional parameters predicate 'Modifier' in the Indication description to cover various conditions. Therefore, a complete indication of semantic rules should be like the followings:

```
<http://ontoweb.wust.edu.cn/medicine#ZSH121216D6369005>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://ontoweb.wust.edu.cn/medicine#DrugBook>.
# The above triple description set an internal concept
# identifie for certain drug (e.g. penicillin)

<http://ontoweb.wust.edu.cn/medicine#ZSH121216D6369005>
<http://wasp.cs.vu.nl/apdg#Concept>
<http://www.ihtsdo.org/SCT_6369005>.
# The above triples map the concept into the medical ontology SNOMED

<http://ontoweb.wust.edu.cn/medicine#ZSH121216D6369005>
<http://ontoweb.wust.edu.cn/medicine#Indications>
<http://ontoweb.wust.edu.cn/medicine#id_hzs121216g6369005>.
# The above triples define a collection of indications

<http://ontoweb.wust.edu.cn/medicine#id_hzs121216g6369005>
<http://ontoweb.wust.edu.cn/medicine#Indication>
<http://ontoweb.wust.edu.cn/medicine#id_hzs121216g6369005_1> .
#The above triples define the first indication description

<http://ontoweb.wust.edu.cn/medicine#id_hzs121216g6369005_1>
<http://ontoweb.wust.edu.cn/medicine#Symptom> "infection" .
<http://ontoweb.wust.edu.cn/medicine#id_hzs121216g6369005_1>
<http://ontoweb.wust.edu.cn/medicine#CausedBy> "Hemolytic streptococcus" .
```

Of course, some concrete rules, especially those involving multiple steps detections of antibiotics clinical rational use, cannot be simply described in RDF Ntriples. We need to put the knowledge into a clinical diagnosis decision support system and describe them in higher-level rules (such as rules description about time and event). Because RDF is not expressive enough to represent the temporal knowledge and the event knowledge, more powerful and expressive formalism should be used to formalize that kind of higher-level rules. The workflows built on the LarKC Platform [8] can be used to solve the problem partially.

3.3 Rules Parsing through DCG in The SToGRUA System

The concrete rules have a unified "If ... then ..." format. That is a fixed format of "premise and conclusion" and can be defined into a template easily. Hence, based on the templates they can be automatically transformed into semantic rules without relying on a complex natural language processing tool.

The reasons why we are using the logic programming language Prolog for the processing are based on its distinguished features. Prolog is a rule-based language. It can be used to describe and generate knowledge representation for rules[9,10]. Prolog provides DCG support and then it can be used as a convenient syntactic parsing tool. Comparing with the natural language processing system, parse of DCG is relatively simple, it works quite well to deal with the concrete rules with a fixed format rules paring. Using Prolog, we implement the SToGRUA system (Semantic Transformer of guidelines for Rational Use of Antibiotics). First, we load the specified guide knowledge as a text file into the Prolog system, then it will be parsed by the SToGRUA system with DCG. Finally, corresponding semantic triples are generated. The main process is just as follows:

```
start :- initial_id_number(N),
        working_on_drug(Drug),
        working_on_ontology(Ontology),
        file_name(File), open(File, read, In),
        output_file_init(Drug, Ontology, Out),
        set_stream(In, encoding(utf8)),
        rules_processing(In, Out, N, M),
        close(In), close(Out),
        format('~w rules have been processed.~n',[M]).
```

The predicate *rules_processing* realizes the DCG processing, i.e. Using the following rules to get the corresponding information of cause-result-state-drugname. The corresponding triples are generated by the predicate *write_ntriples*.

```
rule_processing(Line, Out, N):-
        phrase(rule(Cause Result, Status, Drug), Line, Rest),
         atom_codes(Cause1, Cause),
         atom_codes(Result1, Result),
         atom_codes(Drug1, Drug),
         atom_codes(Rest1, Rest),
         write_ntriples(Out, N, Cause1, Result1, Status, Drug1, Rest1).
```

The DCG rule parsing process is as follows: the corresponding parameters are obtained by matching condition and result in concrete rules. We get the causes and symptoms from the condition and get the status and drug name parameter from the result.

```
rule(Cause, Result, Status, Drug)  -->
          condition(Cause, Result),
          separators,
          conclusion(Status, Drug).
```

We use the following DCG rules to parse conditions of different concrete rules:

```
condition(Cause, Result) -->
              causal_condition_header,
              cause(Cause),
              lead_to,
```

```
               result(Result).


condition(Cause, Result) -->
               condition_header,
               goal_operator(Goal),
               result(Result),
               goal_end,
               {Cause = [goal(Goal)]}.

condition(Cause, Disease) -->
                   condition_header,
                   disease(Disease),
                   patient_operator,
                   goal_operator(Goal),
                   {Cause =[goal(Goal)]}
```

We use the following DCG rules to parse results of different concrete rules:

```
conclusion(indication, Drug) -->
                   conclusion_header,
                   indication_operator,
                   drug(Drug).
......
conclusion(operation(Operation), Disease) -->
                   conclusion_header,
                   disease(Disease),
                   patient_operator,
                   operation(Operation).
.....
conclusion(drug_property(sameas), Drug) -->
                   conclusion_header,
                   drug_pointer,
                  drug(Drug).
```

4 Implementation and Experiments

4.1 A Semantically-Enabled System for Rational Use of Antibiotics

SeSRUA is a semantically-enable system for rational use of antibiotics[6]. It is built on the top of the LarKC[8], a platform for scalable semantic data processing[6]. LarKC uses OWLIM to store and manage semantic data and provides many plug-ins to handle semantic data processing and reasoning. Users can use a Web browser to access the data in the JSON format, which return from the SPARQL endpoint. The user interface of SeSRUA[6] transforms these JSON data into the corresponding visual data and displays them in a user-friendly

[6] http://www.larkc.eu

interface. Therefore, any SeSRUA user will be able to use it even if he/she has no any background knowledge of the semantic technology. We use the APDG (Advanced Patient Data Generator)[11] to generate ten thousand patient data of chronic bronchial in Hubei Province. Those virtual patient data are loaded into the SeSRUA system for the test. APDG uses the domain knowledge to generate virtual patient data, so that the generated patient data look like real ones.

The architecture of SeSRUA is shown in Figure 2.

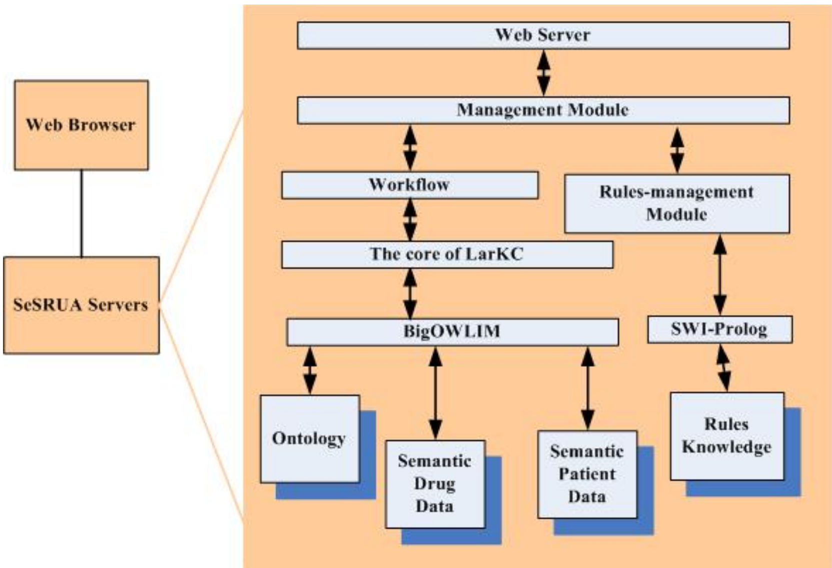

Fig. 1. Architecture of the SeSRUA system.

We have implemented the proposed approach in this paper in the system S-ToGRUA. STOGRUA is designed to be a tool of SeSRUA for its data integration. Semantic rules generated by SToGRUA are used in SeSRUA. The relationship between SToGURA and SeSRUAas is shown in Figure 2.

The right frame in Figure 2 is the SToGRUA system. The management component in SToGRUA calls the logic programming language SWI-Prolog to generate semantic rules. The concrete rules are loaded into SToGRUA in the form of text. DCG rules called by the SWI-Prolog system are used to parse concrete rules. Semantic rules of guidelines knowledge which are generated by SToGRUA are loaded into the SeSRUA system, through matching the semantic guidelines data and the patient data to realize the monitoring of rational use of antibiotics.

4.2 Experiments and Evaluation

In the first experiment, we focus on the generation of the semantic data of the guidelines for rational use of antibiotics[1]. We have converted the knowledge of

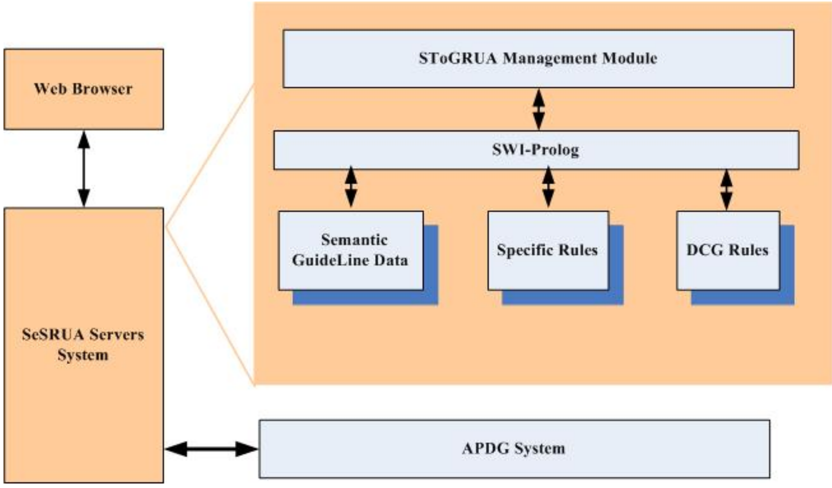

Fig. 2. the Relationship Between SToGURA and SeSRUA.

Part I and Part IV of the knowledge in the guidelines into a set of rules. We would not deal with Part II of the guidelines, because that part of the guidelines concerns only the procedure of hospital management for using antibiotics, which is not the focus of our system. We have not yet converted Part III of the guidelines, because most content of Part III has been covered by the drug manuals of antibiotics, for which its semantic data has been available, by converting the XML-based files into RDF NTriples. Furthermore, some of them have been available at DrugBank. We have used the system SToGRUA to generate the semantic data from those concrete rules.

For the evaluation of the experiment, we firstly assess the quality of transformed abstract rules and concrete rules. We count how many rules have been obtained and how much of the content have been covered by the transformed rules, which is called the coverage ratio (CR), for the assessment of the rule extraction process. We consider a single sentence of the guidelines as the basic unit of the coverage measurement. Thus, the coverage ratio (CR) is the percentage of that transformed rules (TR) with respect to the total number of the sentences number (SN) in the corresponding contents. Namely,

$$CR = TR/SN.$$

The precision of the semantic rules is measured through the amount of the integrated properties in RDF Ntriples. A summary of the experiment is shown in table Table1.

The evaluation results are shown in Table1.

Table 1 shows the experimental results from abstract rules, concrete rules and semantic rules. The precision ratio of transformed semantic rules in Part I and Part IV is 100% and 92% respectively. This initial evaluation of the experiment shows that the guidelines knowledge can be formalized and transformed into

guidelines contents	Abstract Rules			Concrete Rules			Semantic Rules		
	TR	NTR	CR	TR	NTR	CR	TN	TIR	PR
Part I	147	0	100%	247	0	100%	988	100	100%
Part IV	889	19	98%	1351	19	98%	5340	98.80%	92%
Comprehensive experimental data	1036	19	98.80%	1598	19	98.80%	6328	98.80%	92%

Table 1. The exprement results and further optimized contents. TR: Transformed Rule (pieces); NTR: Not Transformed Rule(pieces); CR:Covering Ratio(%); TN:Triple Number (triples); TIR:Triples Integrity Ratio(%); PR: Precision Ratio;

its semantic data. Those generated semantic data are useful for the SeSRUA system. The experiments of the SeSRUA system show that it can judge the 78% percent of the cases for rational use of antibiotics, and 55% percent of the cases for correctly selecting what kinds of antibiotics should be used[6]. A screenshot of the monitoring of rational use of antibiotics in the SeSRUA system is shown in Figure 3.

Fig. 3. the Screenshot of Monitoring in SeSRUA.

5 Conclusions

In this paper, a systematic approach to transforming the guidelines for rational use of antibiotics in natural language into corresponding semantic data has been proposed. In this transforming process, we convert the text into a set of rules in natural language text, then use the DCG tool in Prolog to generate the corresponding semantic data automatically. These two-step transformation approaches make the system able to handle the vast majority of knowledge in the guidelines and provides basic semantic data for SeSRUA.

These generated semantic data are actually structured knowledge that can be used in the monitoring system of rational use of antibiotics. Using inference engine and workflow management process provided by a semantic platform (e.g. LarKC), the knowledge make it possible to judge and monitor the rational use of antibiotics in terms of various patient data in electronic medical records. Hence, processing and managing the knowledge of rational use of antibiotics automatically and semi-automatically by using the semantic technology.

Based on the existing work, the next work that we will do is to use some new methods to reduce the degree of human intervention in the process of rules' transformation. Currently, the transformation from natural language text to rule is done by the manual works. In the future, we will use the tools of natural language processing for semantic annotation of the guidelines text and develop a system so that we can obtain concrete rules automatically or semi-automatically.

Acknowledgments This work was partially supported by the following projects: The projects of Natural Science Foundation of China under grant number 61272110 and 61100133, the key projects of National Social Science Foundation of China under grant number 11ZD&189 and the project of the State Key Lab of Software Engineering Open Foundation of Wuhan University under grant number SKLSE2012-09-07.

References

1. Chinese Medical Association, Pharmacy Administration Commission of Chinese Hospital Association, Hospital Pharmacy Committee of Chinese Pharmaceutical Association: Clinical guidelines for rational use of antibiotics. Technical report (2008)
2. Fox, J., Johns, N., Rahmanzadeh, A., Thomson, R.: Proforma, approaches for creating computer-interpretable guidelines that facilitate decision support. In: Proceedings of Medical Informatics Europe, Amsterdam. (1996)
3. Shahar, Y., Miksch, S., Johnson, P.: The asgaard project: a task specific framework for the application and critiquing of time oriented clinical guidelines. Artif Intell Med **14** (1998) 29–51
4. Tu, S., Musen, M.: A flexible approach to guideline modeling. In: Proceeding of 1999 AMIA Symposium. (1999) 420–424
5. Peleg, M., Boxwala, A., Ogunyemi et al, O.: Glif3: The evolution of a guideline representation format. In: Proceedings of AMIA Annu Fall Symp. (2000) 645–649
6. Hua, X., Chen, C., Huang, Z., Hu, Q.: Intelligent monitoring on use of antibacterial agents based on semantic technology. Journal of China Digital Medicine **8** (2013) 12–15
7. Cheptsov, A., Huang, Z.: Making web-scale semantic reasoning more service-oriented: The large knowledge collide. In: Proceedings of the 2012 Workshop on Advanced Reasoning Technology for e-Science(ART2012). (2012)
8. Fensel, D., van Harmelen, F., Andersson, B., Brennan, P., Cunningham, H., Della Valle, E., Fischer, F., Huang, Z., Kiryakov, A., Lee, T., School, L., Tresp, V., Wesner, S., Witbrock, M., Zhong, N.: Towards LarKC: a platform for web-scale

reasoning. In: Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2008), IEEE Computer Society Press, CA, USA (2008)
9. Wielemaker, J., Huang, Z., van der Meij, L.: SWI-Prolog and the web. Journal of Theory and Practice of Logic Programming **8**(3) (2008) 363–392
10. Wielemaker, J., Schrijvers, T., Triska, M., Lager, T.: Swi-prolog. Journal of Theory and Practice of Logic Programming **12**(1-2) (2012) 67–96
11. Huang, Z., van Harmelen, F., den Teije, A., Dentler, K.: Knowledge-based patient data generation. In Lenz, R., Mikszh, S., Peleg, M., Reichert, M., ten Teije, D.R.A., eds.: Process Support and Knowledge Representation in Health Care, Springer LNAI (2013)