Semantic Rules Representation in Controlled Natural Language in FluentEditor

Anna Wróblewska[1,2], Paweł Kapłański[2], Paweł Zarzycki[2], Iwona Ługowska[3]
[1]Institute of Computer Science, Warsaw University of Technology, Warsaw, Poland,
[2]Cognitum, Warsaw, Poland,
[3]Department of Soft Tissue and Bone Sarcoma and Melanoma, M. Sklodowska-Curie Cancer Centre/Institute of Oncology, Warsaw, Poland,
awroblew@ii.pw.edu.pl, p.kaplanski@cognitum.eu, p.zarzycki@cognitum.eu

Abstract. **The purpose of this paper is to present a way of representation of semantic rules (SWRL) in controlled natural language (English) in order to facilitate understanding the rules by humans interacting with a machine. The rule representation is implemented in FluentEditor – ontology editor with controlled natural language (CNL). The representation can be used in a lot of domains where people interact with machines and use specialized interfaces to define knowledge in a system (semantic knowledge base), e.g. representing medical knowledge and guidelines, procedures in crisis management or in management of any coordination processes. Such knowledge bases are able to support decision making in any discipline provided there is a knowledge stored in a proper semantic way.**

***Keywords*: controlled natural language, decision support, ontologies, semantic web rule language.**

I. Introduction

Controlled Natural Language (CNL) aims to support a human – computer communication and thus improve easiness and efficiency of these interactions.

The top-level goal of this research is to support decision-making in any domain assuming that we have a proper knowledge representation and suitable algorithms to enable reasoning.

Such a decision support system comprises of: a semantic knowledge base, a program for integrating domain-specific information with the knowledge base and user interfaces to allow interacting with the system and get the right information needed to make the right decision at the right time [1].

Our first assumption is that we have a knowledge base, e.g. *Ontorion Knowledge Server* [2], where the knowledge is stored/ modeled using semantic technologies (that is ontologies, semantic rules, reasoning engines).

We state that using semantic technologies applied in our knowledge-based decision support system can improve interoperability and effectiveness of decision-making processes. Generally, ontologies are used for sharing knowledge and common understanding of a particular domain of interest, which makes communication between various beings possible and unambiguous. The various actors may be human users with different levels of expertise or computer programs (agents). The semantic technologies allow for easy adaptation to changes in the modeled knowledge and procedures without the need for intervention in the system architecture and its internal structure. In this case, we need to exchange only the knowledge of the system described using ontologies and semantic rules [3].

Additionally to the whole spectrum of semantic technologies, suitable communication patterns and dedicated interfaces can improve human-computer interactions easiness and efficiency and be more understandable for human users with different levels of expertise. We assume that we need an interface for inputting / modeling the knowledge into the system and an interface using the knowledge and the inferred conclusions (similarly as in [4][5]). Summing up, end-users require less effort to learn how to interact with the system, since the dedicated user interfaces are built upon well-known schema (i.e. natural language).

In the following sections the principal contributions of this paper are: presentation of reasons to use CNL to represent semantic rules (section II), a method to transform SWRL into CNL and vice-versa (section III), simple use cases dedicated to medical diagnostics and crisis management showing the usefulness of the representation and preliminary version of interfaces to input the rules in *FluentEditor* [6] (section IV). In the last section we discuss and conclude the paper.

II. Semantic Knowledge and Controlled Natural Language

A. Semantic Ways of Defining the Knowledge

Modeling domain ontologies and SWRL rules we are able to define a knowledge scheme in any semantic knowledge base. The store for the knowledge base can be implemented in NoSQL technology (e.g. Cassandra [7], Azure Tables [8]) or in RDF data stores (e.g. AllegroGraph [9], Virtuoso [10]).

Relatively simple interface to model ontologies is supported by Protégé [11] or NeOn editors [12]. However, these interfaces are rather simple for experienced knowledge engineers, not for common users that do not know nuances of ontology engineering. On the other hand,

The paper presents controlled natural language and interfaces implemented and being developed in *FluentEditor* by Cognitum (http://www.cognitum.eu/semantics/). The software is developed by P. Kapłański (a main architect), P. Zarzycki and A. Wróblewska.

the simplest way to represent the knowledge is to use natural language. Nevertheless, using natural language is unreachable for the current technology and thus for the machines that should understand this knowledge. The most appropriate solution seems to use controlled natural languages.

As far as CNL is concerned, Controlled English was successfully used by large corporations to standardize the language used for internal communication, e.g.: Caterpillar Technical English [13], IBM Easy English [14], Boeing Simplified English [15] etc. The novel approach to CE supported by knowledge representation and reasoning requires that CE has restricted grammar and vocabulary, in order to reduce the ambiguity and complexity inherent in natural language. In the last years, this branch of CE established itself in various application fields (mostly as an interface of knowledge bases) as a powerful knowledge representation language that is readable for humans and processable by computers.

Attempto Controlled English (ACE) [16][17][18] is very expressive CE and it is the one that is mostly used. ACE can be translated into a non-decidable subset of first-order logic. It also provides its subset called ACEOWL [19] that can be translated into SROIQ Description Logic (formal foundation of OWL2). In *Protégé* ontology editor there is a plugin implemented to convert modelled ontology to ACE. However, it has some drawbacks. Among other it has no implementation of SWRL rules.

In *Ontorion Knowledge Server* we use CNL with formal semantic (e.g. description logic, OWL standards, SWRL rules) that will allow us to provide a domain ontology with rules. Then we apply automatic reasoning services and we generate explanations (written in CNL) of the automatically provided implications [5].

B. Examples of Rules Representation

In the following, to make the issue more visible, we present an example of simple rule representation that can be applied in temporary diagnostics that a patient treated with any therapy should have signed a proper consent before.

In Controlled English (CE) in *FluentEditor*, the rule can be written as in Figure 1. The same rule edited in Protégé editor 4.1 (which does not allow defining the rules properly, only to edit them) is shown as in Figure 2. The rule defined in common format of OWL and SWRL (owl/xml) is shown in Figure 3. As it can be easily seen, the rule written in Controlled English has a potential to be more understandable for an average user without any additional knowledge.

```
If a patient signs a consent and a therapy
is-recommended-to the patient then the therapy
is-applied-to the patient.
```

Figure 1 An example of a semantic rule written in Controlled English in *FluentEditor*

```
Consent(?c), Patient(?p), Therapy(?th),
isRecommededTo(?p,?th), signs(?p,?c) -> isAppliedTo(?p,?th)
```

Figure 2 An example of a semantic rule shown in *Protégé* ontology editor

Nevertheless, representation of semantic rules in Controlled English has some limits in respect to unrestricted natural language (similarly as other controlled natural languages). Generally, the sentences should be quite simple without adjectives. Noun phrases should be written together with dashes between words. It requires a little understanding of ontology modelling, i.e. verbs that define semantic relations or concepts and instances should be written as one word (with dashes between words in natural language).

```
<DLSafeRule>
    <Body>
        <ClassAtom>
            <Class IRI="Consent"/>
            <Variable IRI="urn:swrl#c"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="Patient"/>
            <Variable IRI="urn:swrl#p"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="Therapy"/>
            <Variable IRI="urn:swrl#th"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="isRecommendedTo"/>
            <Variable IRI="urn:swrl#p"/>
            <Variable IRI="urn:swrl#th"/>
        </ObjectPropertyAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="signs"/>
            <Variable IRI="urn:swrl#p"/>
            <Variable IRI="urn:swrl#c"/>
        </ObjectPropertyAtom>
    </Body>
    <Head>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="isAppliedTo"/>
            <Variable IRI="urn:swrl#p"/>
            <Variable IRI="urn:swrl#th"/>
        </ObjectPropertyAtom>
    </Head>
</DLSafeRule>
```

Figure 3 An example of a semantic rule shown in OWL/XML format

III. Semantic Rules in CNL

A. Semantic Web Rule Language

The OWL 2 Web Ontology Language is an ontology language for the Semantic Web with formally defined meaning [20]. OWL 2 ontologies in the abstract syntax contains a sequence of axioms and facts, i.e. provide classes, properties, individuals, and data values and can be stored as Semantic Web documents. Ontology axioms may be of various kinds, e.g. *subClass* axioms, *equivalentClass* axioms. Semantic Web Rule Language (SWRL) is proposed to extend the OWL 2 ontologies with rule axioms.

SWRL is based on a combination of the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language [21]. The SWRL standard extends the set of OWL axioms to include Horn-like rules and thus enables Horn-like rules to be combined with an OWL knowledge base.

The SWRL rules (semantic rules) are of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold.

Both the antecedent (body) and consequent (head) consist of atoms. Atoms in these rules (Figure 4) can be of the form *C(x), P(x,y), sameAs(x,y), differentFrom(x,y)* or *builtIn(r, x, ...)* where *C* is an OWL description or data range, *P* is an OWL property, *r* is a built-in relation, *x* and *y* are either variables, OWL individuals or OWL data values, as appropriate. In the context of OWL Lite, descriptions in atoms of the form *C(x)* may be restricted to class names. The main safety condition of SWRL is that only variables that occur in the antecedent of a rule may occur in the consequent.

```
axiom ::= rule | …
rule ::= 'Implies(' [ URIreference ] { annotation }
                                antecedent consequent ')'
antecedent ::= 'Antecedent(' { atom } ')'
consequent ::= 'Consequent(' { atom } ')'

atom ::= description '(' i-object ')'
       | dataRange '(' d-object ')'
       | individualvaluedPropertyID '(' i-object i-object ')'
       | datavaluedPropertyID '(' i-object d-object ')'
       | sameAs '(' i-object i-object ')'
       | differentFrom '(' i-object i-object ')'
       | builtIn '(' builtinID { d-object } ')'
builtinID ::= URIreference
i-object ::= i-variable | individualID
d-object ::= d-variable | dataLiteral
i-variable ::= 'I-variable(' URIreference ')'
d-variable ::= 'D-variable(' URIreference ')'
```

Figure 4 A grammar of SWRL

B. Controlled English for SWRL

We propose own specially designed methods for representation of SWRL in Controlled English (SR-CE) in predicative editor *FluentEditor* [6] (that is an interface to the *Ontorion* knowledge base). The editor prohibits one to enter any sentence that is not grammatically or morphologically correct and actively helps the user during sentence writing (Figure 5).

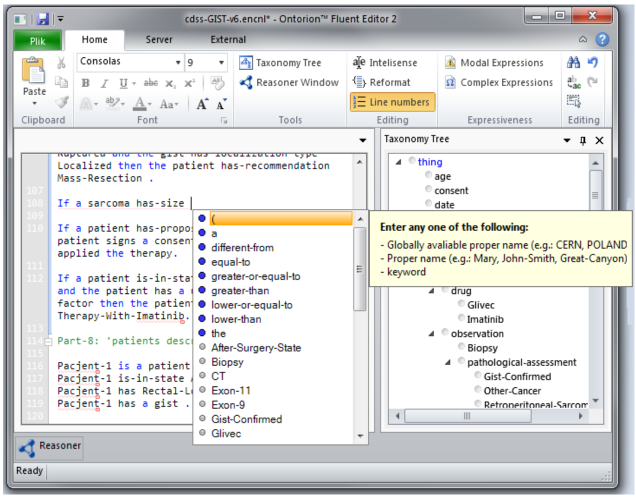

Figure 5 *FluentEditor* interface helps the user during sentence writing

The very first version of CE was introduced in [22]. Up to now, the CE language was extended, specified to be more logical and natural [23]. Additionally, we added representation of semantic rules. The EBNF grammar[1] of the CE is designed to be as close as possible to natural English and can be translated to OWL2 and back easily. Grammar of CE was implemented using context-free grammar parser. All the mechanisms were implemented in *FluentEditor*.

CE language deals with concepts, roles and instances that can be represented by symbols in form of extended words[2]. Concept and role identifiers should start with small letter in opposite to instances and cannot be any of keywords. The morphology is applied if needed by separated dictionary based module. Additionally it is required that each sentence starts with upper-case letter and ends with „full stop" sign. There are five general groups of sentences that are allowed in *FluentEditor*:

1) regarding concept specifications, e.g. subsumptions, complement, equivalence, disjointness, value partitition, more complex concept definitions,
2) instance specifications,
3) role (object properties) specifications, e.g. inclusion, equivalence, disjointness and other axioms (symmetry, transitivity),
4) data attributes (properties) specifications and their data ranges,
5) semantic rules specifications.

In the following we consider only the semantic rules. However, in the next section we introduce some examples considering the whole spectrum of the designed Controlled English.

The main part of the grammar regarding semantic rules specification in CE (SR-CE) is shown in Figure 6 and auxiliary structures are shown in Figure 7. It is more complicated than the SWRL grammar (Figure 4) because of specific characteristics of a natural language. Correspondingly, some constructions in SR-CE cannot be translated directly into SWRL (and vice-versa).

```
<paragraph> ::= { <sentence> }
<sentence>  ::= … | 'If' <antecedent> 'then' <consequent> '.'
<antecedent> ::= { <condition> 'and' } <condition>
<consequent> ::= { <condition-result> 'and' } <condition-result>

<condition>   ::=  <object> <name> <object>
                 | <object> 'is' <name> 'by' <object>
                 | <object> 'exists'
                 | <object> 'is' <object>
                 | <object> 'is-and-is-only' <complex-object>
                 | <object> 'is' 'not' <object>
                 | <object> <name> <abstract-bound>
                 | <object> <name> 'equal-to' <object>

<condition-result> ::=
                   <identified-object> <name> <identified-object>
                 | <identified-object> 'is' <name> 'by' <identified-object>
                 | <identified-object> 'is' <identified-object>
                 | <identified-object> 'is' 'not' <identified-object>
                 | <identified-object> 'is' <not-identified-object>
                 | <identified-object> <name> <abstract-bound>
                 | <identified-object> <name> 'equal-to' <identified-object>
```

Figure 6 The main part of EBNF grammar of SR-CE

In SR-CE rules are presented as if-clause sentences. They consist of antecedent and consequent clauses. The clauses comprises of atoms, likewise in SWRL.

[1] Extended Backus–Naur Form (EBNF) is a family of metasyntax notations used for expressing context-free grammars.
[2] Extended words are a group of words connected with a dash ('-') sign.

Nevertheless, the atoms in antecedent and consequent (*<condition>* and *<condition-result>* respectively) are slightly different. In *<condition>* atom one can use all available objects, however in *<condition-result>* - only identified objects. That is because of the safety condition of SWRL that only variables that occur in the antecedent of a rule may occur in the consequent. SWRL rules cannot create new instances, there can be only new associations (new knowledge/axioms) for the existing instances.

There is only one *<condition-result>* atom, in which one can use not identified object. That is "*<identified-object> 'is' <not-identified-object>"*. It is used to express that an instance, represented with the variable in sentence, is assigned to a new concept, e.g. "*If a person is-year-old greater-or-equal-to 18 then the person is an adult*."

Atoms in antecedent can express:
- OWL property between objects: "*<object><name> <object>"* and "*<object> is <name> by <object>"* for an inversion of the property,
- OWL description (axioms for concepts): *"<object> exists"*,
- *sameAs* axiom to express that two individuals are the same and opposed *differentFrom* axiom: *"<object>is <object>"*, *"<object>is not<object>"*, respectively,
- OWL data range description and data property assignments: *"<object> <name> equal-to <object>"*, *"<object><name><abstract-bound>"*,
- OWL complex description axioms for concepts and data properties: *"<object>is-and-is-only<complex-object>"*.

Atoms in consequence are similar, but without definitional clauses for OWL descriptions. Additionally, they use almost everywhere identified objects, as it was mentioned before.

```
<object> ::= <not-identified-object> | <identified-object>
<not-identified-object> ::= <a> <name>
                          | <a> 'thing'
                          | <a> <name> '(' <natural> ')'
                          | <a> 'thing' '(' <natural> ')'
<identified-object> ::=  'the' <name>
                      | 'the' 'thing'
                      | 'the' <name> '(' <natural> ')'
                      | 'the' 'thing' '(' <natural> ')'
                      | <instance>

<abstract-bound> ::= 'lower-than' <data-value>
                   | 'equal-to' <data-value>
                   | 'different-from' <data-value>
                    …
                   | '(' 'either'<data-value>','
                           <data-value>{','<data-value>}')'
                   | '(''some-integer-value'')'
                   | '(''some-string-value' ')'
                    …
<data-value> ::= <integer> | <float> | <string> | <boolean>
<complex-object> ::=  <a> <name>  | <a> <name> <that>
                    | 'something' | 'something'<that> | <instance>
<that> ::= ['(']{'that'<complex-expression>
           {'and' 'that'<complex-expression>}}}[')']
<instance> ::=  <big-name>
<a> ::= 'a' | 'an'
<name>     ::= <small-letter> { <digit> | <small-letter>
                    | <big-letter> | '-' | '.' | '/' }
<big-name> ::= <big-letter> { <digit> | <small-letter>
                    | <big-letter> | '-' | '.' | '/' }
```

Figure 7 Auxiliary structures in SR-CE grammar

As far as the auxiliary grammar structures are concerned, one can use not identified and identified object (Figure 7). They represent variables in the semantic rule sentences. Not identified objects are those variables that are used for the first time in a sentence. Identified objects correspond to the variables defined/introduced with not identified object names. The variables of kind of *"a/the <name>"* denote variables assigned to a class represented with its *<name>*. The variables of kind of "*a/the thing"* represent an instance of the top concept *"owl:Thing"* and thus any variable in the SR-CE sentence. Additionally, one can use multiple variables representing instances of the same class in any sentence. It can be achieved using an identifier denoting subsequent variable number, e.g. "*a thing (1)","a thing (2)"*.

There is also a possibility to define complex objects - class and data descriptions, e.g. *"If a person is-and-is-only something that has-child at-least one person then the person is a parent"*.

C. General Implementation Tips

SR-CE grammar is translated to SWRL grammar and thus made available to inferring in a reasoning engine (e.g. *HermiT* [24]). To implement it we used *OWL API* [25], which is a Java API and reference implementation for creating, manipulating and serializing OWL ontologies. A backward implementation of SWRL to SR-CE, e.g. to show the result of reasoning, is also available in *FluentEditor*.

IV. USE CASES

A. Crisis Management Use Case

Crisis management is an activity of public administration which is a part of national security management. Crisis management deals with: crisis prevention, preparation, response and recovery.

In this use case we consider and construct an ontology of crisis management to characterize crises (crisis events, effects, risks and dangers), the studied world (crisis environment: people, natural sites, goods) and treatment system (communication and coordination actors, procedures and tasks and the treatment infrastructure, e.g. resources).

This use case deals with response to a crisis. It concerns a tanker accident and its effects. A tanker truck (containing unknown toxic substance) had an accident. Several children of the near kindergarten (outside when the accident happened) feel sick.

We use the ontology and reasoning engine to find answers to questions: what the dangers can occur and what emergency services are needed. Some semantic rules in the use case are presented in Figure 8 and Figure 9. A fragment of the ontology (also instances modelling the use case), some dedicated semantic rules and provided inferred new knowledge that are written in CE are shown in Figure 10. The developed system infrastructure (*FluentEditor* and *Ontorion Knowledge Server*) can deduce that, e.g. in our use case we need such rescue teams as police, fire brigade and emergency medical services and that the crisis (tanker accident) causes a risk of contamination.

```
If a crisis has-type a crisis-type and an actor reacts-to-type
the crisis-type then the crisis needs the actor.

If an effect causes a potential-danger and a crisis induces
the effect and an actor assumes a procedure and the procedure
reduces the potential-danger then the crisis needs the actor.

If a crisis induces an effect and the effect concerns a
people-group and the people-group has-victims
greater-or-equal-to 1 and an actor is
Emergency-Medical-Service then the crisis needs the actor.

If a crisis induces an event and the event causes a risk then
the risk appears-during the crisis.
```

Figure 8 Semantic rules written in SR-CE and used in crisis management use case

```
Actor(?a), Crisis(?c), CrisisType(?ct), hasType(?c, ?ct),
reactsToType(?a, ?ct) -> needs(?c, ?a)

Actor(?a), Crisis(?c), Danger(?d), Effect(?e),
Procedure(?p), assumes(?a, ?p), causes(?e, ?d),
induces(?c, ?e), reduces(?p, ?d) -> needs(?c, ?a)

Actor(?a), Crisis(?c), Effect(?e), PeopleGroup(?pg),
concerns(?e, ?pg), induces(?c, ?e),
integer[>=1](?v), victims(?pg, ?v),
SameAs(?a, EmergencyMedicalService) -> needs(?c, ?a)

Crisis(?c), Risk(?r), Event(?e), causes(?e, ?r),
induces(?c, ?e) -> appearsDuring(?c, ?r)
```

Figure 9 Semantic rules used in crisis management use case and presented in *Protégé* ontology editor

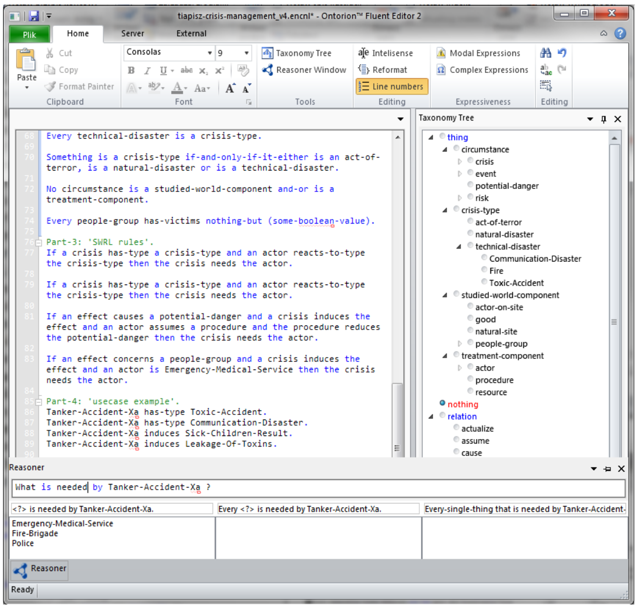

Figure 10 A fragment of crisis management ontology and semantic rules written in CE in *FluentEditor*. At the bottom of the figure/interface inferred new knowledge is shown

B. Oncology Guidelines Use Case

Application of clinical practice guidelines is mandatory in current evidence-base medicine, including oncology [26]. The knowledge contained in the guidelines is dynamic and improved regularly. The SR-CE allows writing the knowledge in a relatively easy way for doctors and read inferred recommendations.

In this use case we consider and build a very simplified ontology dealing with a few rules from gastrointestinal stromal tumour (GIST) guidelines [27]. A sarcoma may have characteristics. A patient may have favourable or unfavourable prognostic factors. The recommended therapy is dependent on the factors and a patient overall state.

Semantic rules in the use case are presented in Figure 11 and Figure 12. A fragment of the ontology (also instances modelling the use case), dedicated semantic rules and provided inferred new knowledge that are written in CE are shown in Figure 13. The developed system infrastructure can deduce that the considered patient has a recommendation for the adjuvant therapy with a drug - imatinib.

```
If a gist has-size greater-or-equal-to 5 then the gist
has-characteristic Large-Gist.

If a patient has a gist and the gist has-characteristic
a characteristic and the characteristic constitutes a
prognostic-factor and the prognostic-factor is
Unfavorable-Prognostic-Factor then the patient
has-prognostic Unfavorable-Prognostic-Factor.

If a patient has a gist and the patient is-in-state
After-Surgery-State and the patient has-prognostic
Unfavorable-Prognostic-Factor then
Adjuvant-Therapy-With-Imatinib is-recommended-to the
patient.
```

Figure 11 Semantic rules written in SR-CE and used in oncology guidelines use case

```
Gist(?g), hasSize(?s), integer[>=5](?s) -> has-characteristic(?g,Large-Gist)

Patient(?p), Gist(?g), Characteristic(?c), PrognosticFactor(?pf),
has(?p,?g), hasCharacteristic(?g,?c), constitutes(?c,?pf),
SameAs(?pf, Unfavorable-Prognostic-Factor) -> hasPrognosticFactor(?p,?pf)

Patient(?p), Gist(?g), has(?p,?g), isInState(?p, After-Surgery-State),
hasPrognosticFactor(?p, Unfavorable-Prognostic-Factor)
-> isRecommendedTo(?p, Adjuvant-Therapy-With-Immatinib)
```

Figure 12 Semantic rules used in oncology guidelines use case and presented in *Protégé* ontology editor

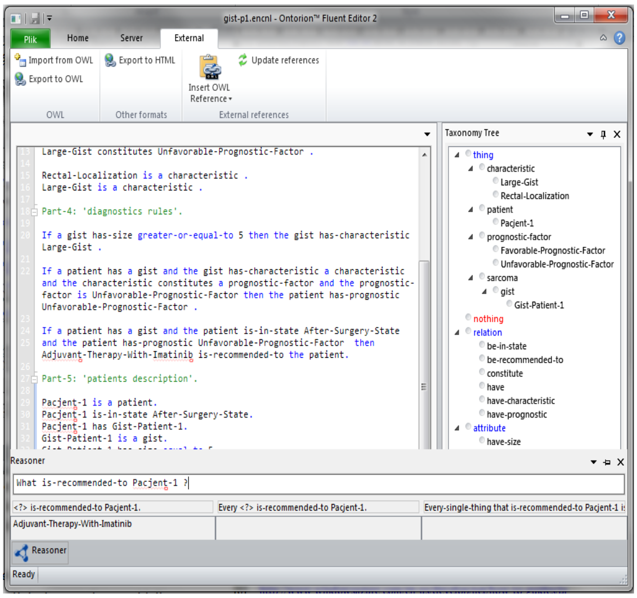

Figure 13 A fragment of simplified oncology guidelines ontology and semantic rules written in CE in *FluentEditor*. At the bottom of the figure/interface inferred new knowledge is shown

V. Conclusion

In this paper we investigated a top-level problem of decision-making support with the use of semantic technology and knowledge bases. We described a concept of using controlled natural language to support human-machine interactions. We presented a grammar of SWRL rules written in controlled English and the results of translation from CE to SWRL and vice-versa. We provided simple use cases showing the usability of the methods implemented in *FluentEditor*.

In the future we plan to elaborate CNL mechanisms for Polish language and improve more usable interfaces for a non-experienced user.

Acknowledgment

We would like to gratefully thank Dr. Aleksandra Zięba, PhD in political science and an expert in crisis management, for her valuable remarks regarding the crisis management use case.

References

[1] Eberhardt J, Bilchik A, Stojadinovic A (2012) Clinical decision support systems: potential with pitfalls. J. Surgical Oncolology 105 (5): 502-510

[2] Ontorion Knowledge Server, http://www.cognitum.eu/semantics/Ontorion/

[3] Sheth A, Ramakrishnan C (2003) Semantic (Web) Technology In Action: Ontology Driven Information Systems for Search, In: "Integration and Analysis, Bulletin of the Technical Committee on Data Engineering", 26 (4): 40-48

[4] R.N. Shiffman, G. Michel, M. Krauthammer, N.E. Fuchs, K. Kaljurand, T. Kuhn: Writing Clinical Practice Guidelines in Controlled Natural Language. Springer LNCS 5972, 2010

[5] A. Wróblewska, A. Zięba, R. Mieńkowska –Norkiene, P. Kapłański, P. Zarzycki: Application of Semantic Knowledge Management System in Specific Areas of Polish Public Administration. Roczniki Kolegium Analiz Ekonomicznych SGH (Annals of the College of Economic Analysis SGH), 2013, in print

[6] FluentEditor, http://www.cognitum.eu/semantics/FluentEditor2

[7] http://cassandra.apache.org

[8] http://www.windowsazure.com/en-us/develop/net/how-to-guides/table-services

[9] http://www.franz.com/agraph/allegrograph

[10] http://virtuoso.openlinksw.com/rdf-quad-store

[11] Protégé ontology editor, http://protege.stanford.edu/

[12] NeON Toolkit, http://neon-toolkit.org

[13] C. Kamprath, E. Adolphson, T. Mitamura, and E. Nyberg. "Controlled Language for Multilingual Document Production: Experience with Caterpillar Technical English". 1998.

[14] A. Bernth. "EasyEnglish: a tool for improving document quality". In: Proceedings of the fifth conference on Applied natural language processing, pp. 159–165, Association for Computational Linguistics, Stroudsburg, PA, USA, 1997.

[15] R. H. Wojcik, J. E. Hoard, and K. Holzhauser. "The Boeing Simplified English Checker". In: Proceedings of the International Conference Human Machine Interaction and Artificial Intelligence in Aeronautics and Space, pp. 43–57, Toulouse: Centre d'Etudes et de Recherches de Toulouse, 1990.

[16] N. E. Fuchs, U. Schwertel, and R. Schwitter. "Attempto Controlled English - Not Just Another Logic Specification Language". In: LOPSTR '98: Proceedings of the 8th International Workshop on Logic Programming Synthesis and Transformation, pp. 1–20, Springer-Verlag, London, UK, 1990.

[17] J.L. De Coi, N.E. Fuchs, K. Kaljurand, T. Kuhn. Controlled English for reasoning on the Semantic Web. In F. Bryand, J. Maluszynski, editors, Semantic Techniques for the Web - The REWERSE Perspective, LNCS 5500: 276-308, Springer, 2009.

[18] Attempto project, http://attempto.ifi.uzh.ch/site/description/

[19] K. Kaljurand. Attempto Controlled English as a Semantic Web Language. PhD thesis, Faculty of Mathematics and Computer Science, University of Tartu, 2007.

[20] OWL 2 Web Ontology Language Document Overview (Second Edition). W3C Recommendation 11 December 2012. http://www.w3.org/TR/owl2-overview/

[21] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean: SWRL: A Semantic Web Rule Language. Combining OWL and RuleML. W3C Member Submission 21 May 2004, http://www.w3.org/Submission/SWRL/

[22] P. Kapłański, Controlled English interface for knowledge bases, "Studia Informatica" 2011, vol. 32, no. 2A (96), pp. 486-494

[23] P.Kapłański: Ontology-aided software engineering. PhD thesis. Gdańsk University of Technology, 2012

[24] B. Glimm, I. Horrocks, B. Motik, G. Stoilos:. Optimising Ontology Classification. In: P.F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J.Z. Pan, I. Horrocks, B. Glimm, editors, Proc. of the 9th Int. Semantic Web Conf. (ISWC 2010), volume 6496 of LNCS, pages 225--240, Shanghai, China, November 7-11 2010. Springer. http://www.hermit-reasoner.com/

[25] M. Horridge, S. Bechhofer. The OWL API: A Java API for OWL Ontologies. Semantic Web Journal 2(1), Special Issue on Semantic Web Tools and Systems, pp. 11-21, 2011. http://owlapi.sourceforge.net/

[26] ESMO guidelines, http://www.esmo.org/education-research/esmo-clinical-practice-guidelines.html

[27] NCCN Clinical Practice Guidelines in Oncology. Soft Tissue Sarcoma. Version 2.2012