Using SWRL and JESS to assess the risk level in an organisation risk management ontology developed in PROTEGE

Andrei Mihai Popovici, Corneliu Nicolae Zaharia

Stefan Nicolau Institute of Virology, 285 Mihai Bravu Avenue, 030304, Bucharest, Romania
e-mail: corneliu.zaharia@virology.ro, andrei_popovici@hotmail.com

ABSTRACT

Generally, risks are seen as a combination between the probability of a threat to occur and the severity of its consequence. The two separate scales lead to 5 risk classes, where RiskClass5 is considered to contain the most dangerous risks, while the RiskClass1 contains negligible risks. Depending on the risk class, specific prevention measures packages must be taken.

This study's goal is to describe the way one can represent and semantically group the risks associated with work processes within an organisation, in one of these 5 risks classes, being a well known fact that many of the work operations performed by workers carry work accidents risks of different types.

Semantic description of an organisation complex environment from work processes risks approach leads naturally to choosing an ontology as a modeling and development tool.
Within this organization ontology model developed with PROTEGE, we studied how risk level of an operation which is part of a working process, can be included in one of the risks classes, based on semantic rules build with SWRL and run using JESS engine. A separate rule is designed to decide whether an operation belongs to a certain risk class, so we ran 5 rules. After running these rules, the collection of initial instantiated operations is expected to be enriched with a boolean attribute bearing the value of TRUE for operations that verify the conditions to belong to one of the five risk classes.

Key words: *SWRL rules risk level assignment, risk management, SWRL rules for risk calculation, risk management SWRL rules*

1. INTRODUCTION

The capacity of grouping operations, activities of processes by their risk group is critical in a risk management ontology. Based on an assigned level of risk for working operations, activities or processes, certain prevention measures packages can be targeted and implemented within an organization.

In this study we aim to give a method of assigning such risk levels based on SWRL rules ran with a rule engine. In the same time, we analyse the produced rezults and discuss the limits of this method.

2. METHODS

2.1 Organization risk management model ontology

A general risk management model for an organization had to be defined first.

Model class structure
All concepts that are going to be used in the model will be implemented as classes.

An organisation can be seen as a set of processes having their finality in reaching organisation's key policy goals. Further, processes can be seen as activities collections and activities can be seen as operations collections. Thus, in the discussed ontology model, the operation is to be regarded as the atomic unit a work process can contain. The significant concepts in this brief description are: Organization, Section, Laboratory, Process, Activity and Operation.

In PROTÉGÉ significant classes were created based on these concepts, following the top-down organization structure, namely: Organization, Section and Laboratory. The model also contains classes related to working processes, namely: Process, Activity and Operation. Risk area was modeled with classes Threat, Consequence and Risk.

A screen shot of these classes as they were defined is given in Fig. 1

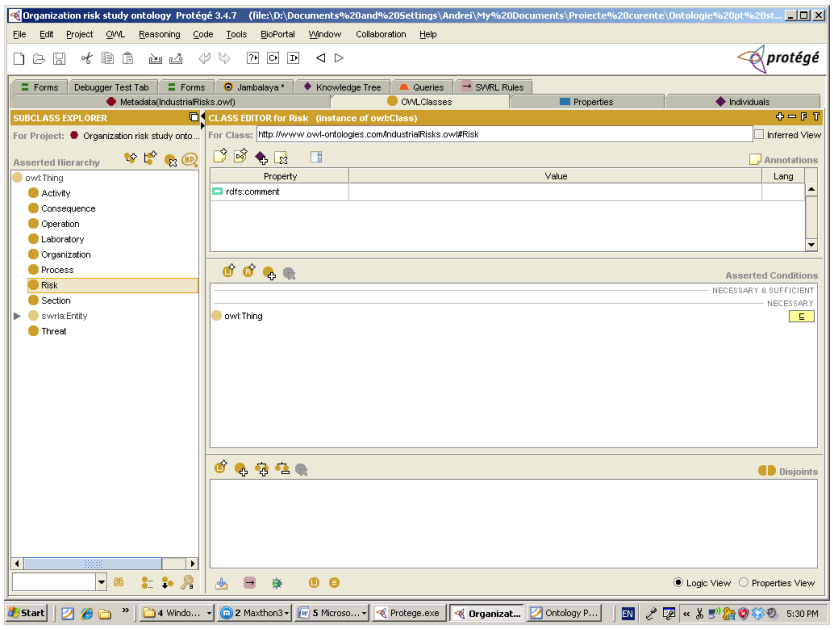

Fig.1-Organization risk management class structure

Relationships between classes

It is a well known aspect that semantics used in an ontology model is based on the relationships defined between classes. In PROTÉGÉ these relationships are called "Properties" and can be two types: Object Properties when a relationship is established between two objects (individuals, instances of classes) and Datatype Properties, when a relationship is established between an individual and a XML Schema Datatype value or an rdf literal, this being the way we assign an attribute value to an individual. The two individuals linked by an object property are called *domain*, respectively *range* for that property. Although the properties are defined between classes, the inference engine applies them between individuals belonging to those classes.

Thus, significant properties were defined to link individuals from our previously defined classes:

"Operation" | "Activity" | "Process" *has_threat* "Threat" (1)

where "|" is used as a notation to show mutual exclusion between entities.

Similarly, we defined next properties:

“Operation” | ”Activity” | ”Process” *has_risk* “Risk” (2)

“Threat” *has_consequence* “Consequence” (3)

“Organization” *has_section* “Section”

“Section” *has_laboratory* “Laboratory”

“Activity” *has_operation* “Operation”

“Process” *has_activity* “Activity”

An example of a screen shot corresponding to these relationships is given below.

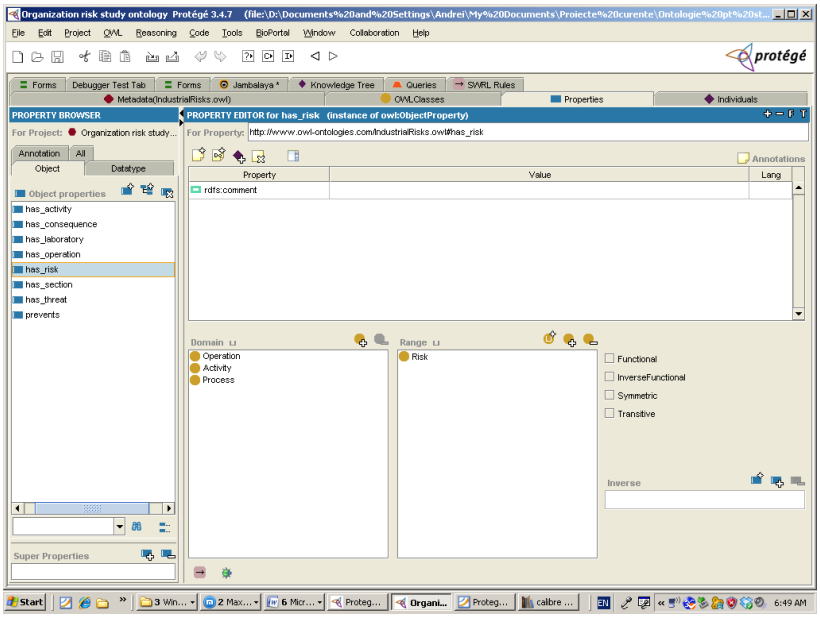

Fig.2-Relationships between classes of the organizational model

Attribute definitions

Few significant attributes had to be defined as Datatype properties.

Thus, we defined:

“Threat” *hasProbability* “String” (5)

As we can notice, “Threat” class is the *domain* and “String” is the *range*. The instances the “String” class can take may be defined as a collection of values, respectively: {„highly likely”, „likely”, „rare”, „very rare”, „almost impossible”}. This way, *hasProbability* establishes an attribute like relationship between „Threat” and the possible values it can take.

We used a different construction for datatype property notation to distinguish it from the object property type.

The screen showing this context is given below.

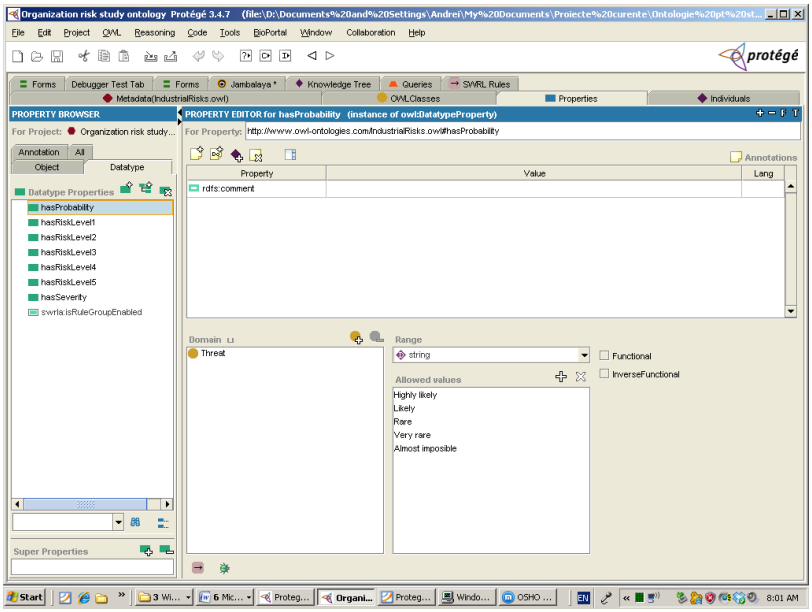

Fig.3-Datatype properties defined in organization risk management model

Similarly, were defined the following datatype properties:

"Consequence" *hasSeverity* "String" (6)

where"String" takes values in the collection of {„deadly", „very severe", „severe", „negligible"},

"Operation" | "Activity" | "Process" | "Risk" *hasRiskLevel1* "Boolean", (7)

where "Boolean" taking values in the collection of {"true", "false"} and continuing,

"Operation" | "Activity" | "Process" | "Risk" *hasRiskLevel2* "Boolean" (8)

"Operation" | "Activity" | "Process" | "Risk" *hasRiskLevel3* "Boolean" (9)

"Operation" | "Activity" | "Process" | "Risk" *hasRiskLevel4* "Boolean" (10)

"Operation" | "Activity" | "Process" | "Risk" *hasRiskLevel5* "Boolean" (11)

This way, a risk level was semantically defined for each of the classes in the *domain* space.

Instance definitions

Instances of a class are called "Individuals" in PROTÉGÉ. A class may have as many instances as are needed. In our model, in order to keep a general approach, we simply named the instances using the default naming engine given by PROTÉGÉ.

Thus, we created a single instance for the "Organization" class, 3 instances for the "Section" class, 8 instances for the "Consequence" class and so forth.

An example with instances created for "Operation" class in given below.

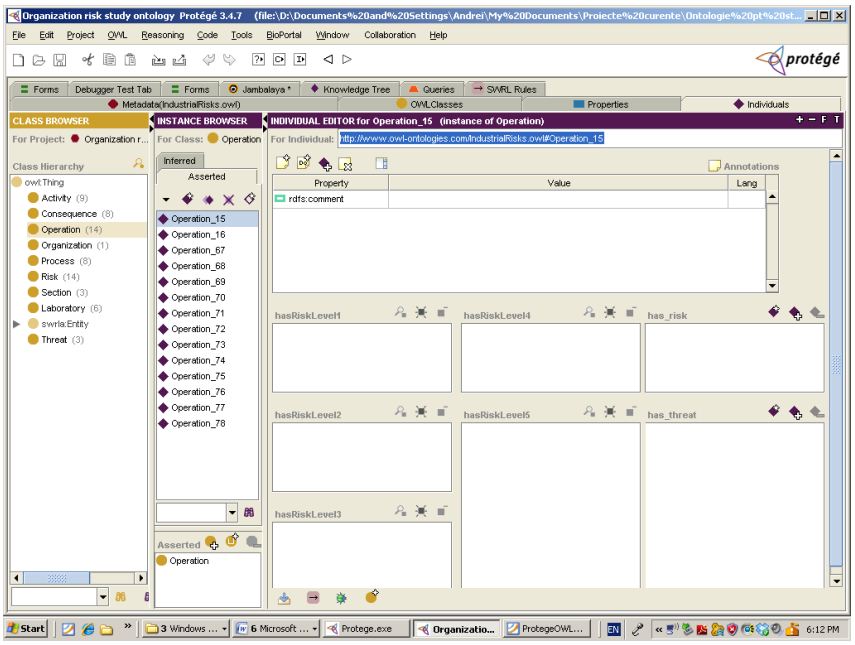

Fig.4-Individuals in organization risk management model

2.2 Using SWRL rules in the Risk Management Model Ontology

The problem

Up to this stage, we created a simplified model for risk management where we are to decide using SWRL rules, what risk level can be assigned to a “Risk” and “Operation” individual based on the particular values given to the “String” individuals involved in the datatype properties previously defined, as is illustrated below:

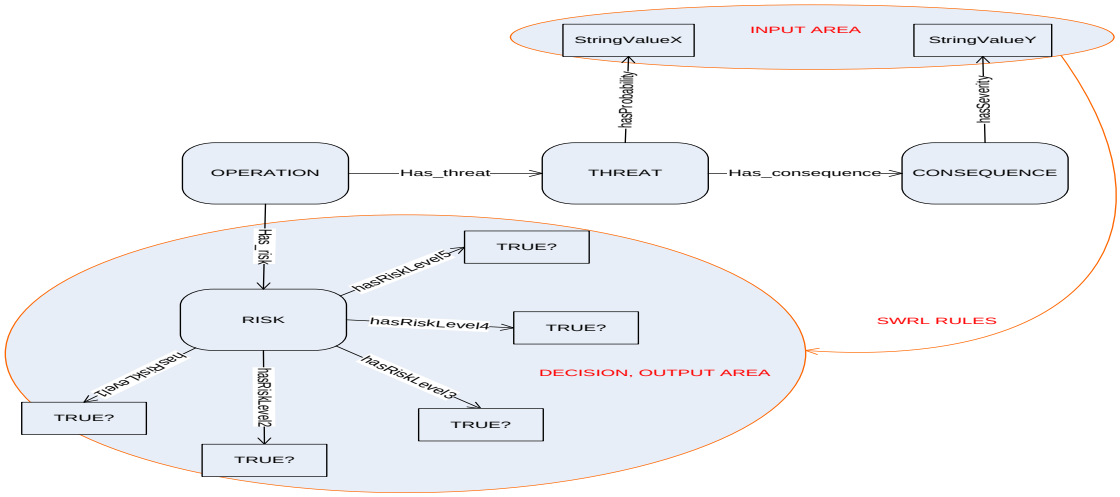

Fig.5-The problem of deciding the risk level based on threat-consequence combination

INPUT and OUTPUT areas

We'll consider as INPUT the particular values given in the datatype properties previously defined for "Threat" and "Consequence" individuals, meaning:

"Threat" *hasProbability* "String"

where "String" = StringValueX from {„highly likely", „likely", „rare", „very rare", „Almost impossible"}

"Consequence" *hasSeverity* "String"

where "String" = StringValueY from {„deadly", „very severe", „severe", „negligible"}

SWRL rules and decision on the risk level

The SWRL rules we used take into account the values given to StringValueX and StringValueY and based on these values is decided that one of the previously datatype properties (7) to (11) is TRUE.

The rules we created are listed below:

RiskClass5:

Operation(?o) ∧ Risk(?r) ∧ has_threat(?o, ?t) ∧ hasProbability(?t, ?P) ∧ swrlb:equal(?P, "Highly likely") ∧ has_consequence(?t, ?c) ∧ hasSeverity(?c, ?S) ∧ swrlb:equal(?S, "Deadly") ∧ has_risk(?o, ?r) → hasRiskLevel5(?o, true) ∧ hasRiskLevel5(?r, true)

RiskClass4:

Operation(?o) ∧ Risk(?r) ∧ has_threat(?o, ?t) ∧ hasProbability(?t, ?P) ∧ swrlb:equal(?P, "Likely") ∧ has_consequence(?t, ?c) ∧ hasSeverity(?c, ?S) ∧ swrlb:equal(?S, "Very severe") ∧ has_risk(?o, ?r) → hasRiskLevel4(?o, true) ∧ hasRiskLevel4(?r, true)

RiskClass3:

Operation(?o) ∧ Risk(?r) ∧ has_threat(?o, ?t) ∧ hasProbability(?t, ?P) ∧ swrlb:equal(?P, "Rare") ∧ has_consequence(?t, ?c) ∧ hasSeverity(?c, ?S) ∧ swrlb:equal(?S, "Severe") ∧ has_risk(?o, ?r) → hasRiskLevel3(?o, true) ∧ hasRiskLevel3(?r, true)

Similarly, can be constructed rules for RiskClass2 and RiskClass1 and the rest of rules for deducting the risk levels included in the risk levels matrix, given below:

Risk levels matrix:

Severity Probability	Deadly	Very severe	Severe	Negligible
Highly likely	**5**	**5**	**4**	**4**
Likely	**5**	**4**	**4**	**3**
Rare	**4**	**4**	**3**	**3**
Very rare	**4**	**3**	**3**	**2**

Almost impossible	**3**	**3**	**2**	**1**

INPUT values

We gave values to few individuals defined within the ontology, following the diagram given in Fig.5 and grouped by the RiskClass rule that should apply:

- Group1-RiskClass5

Threat_3 *hasProbability* „Highly likely”

Threat_3 *has_consequence* Consequence_12

Consequence_12 *hasSeverity* „Deadly”

Operation_15 *hasThreat* Threat_3

Operation_15 *hasRisk* Risk_24

- Group2-RiskClass4

Threat_4 *hasProbability* „Likely”

Threat_4 *has_consequence* Consequence_13

Consequence_13 *hasSeverity* „Very severe”

Operation_16 *hasThreat* Threat_4

Operation_16 *hasRisk* Risk_25

- Group3-RiskClass3

Threat_5 *hasProbability* „Rare”

Threat_5 *has_consequence* Consequence_14

Consequence_14 *hasSeverity* „Severe”

Operation_67 *hasThreat* Threat_5

Operation_67 *hasRisk* Risk_26

The screen showing the values given to Operation_15 is given below:

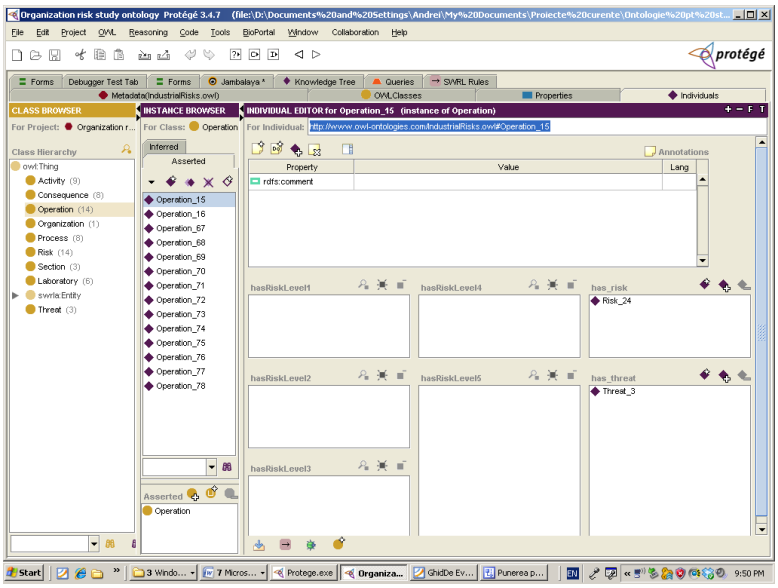

Fig.6-INPUT values for risk and threat assigned to an operation individual

3. RESULTS

In order to run the rules we used JESS rules engine that was installed to work with PROTEGE

Expected results

All selected rules will be run in JESS at once, producing their specific results.

In our case, we expect JESS to produce the following results, based on running the rules RiskClass3, RiskClass4 and RiskClass5:

Operation_15 *hasRiskLevel5* =TRUE Risk_24 *hasRiskLevel5* = TRUE

Operation_16 *hasRiskLevel4* =TRUE Risk_25 *hasRiskLevel4* = TRUE

Operation_67 *hasRiskLevel3* =TRUE Risk_26 *hasRiskLevel3* = TRUE

Running the rules with JESS

Activating the JESS rule engine can be done by clicking on the "J" icon from the icon group at the right side of the tab "SWRL rules" in PROTÉGÉ and the result is an additional form with buttons and tabs that manages interaction with JESS. Running the rules with JESS is done in 3 steps, as it follows:

- Step1-Loading the rules in Jess

Click on the button "OWL + SWRL---> Jess"

The screen below shows the result of Step1.

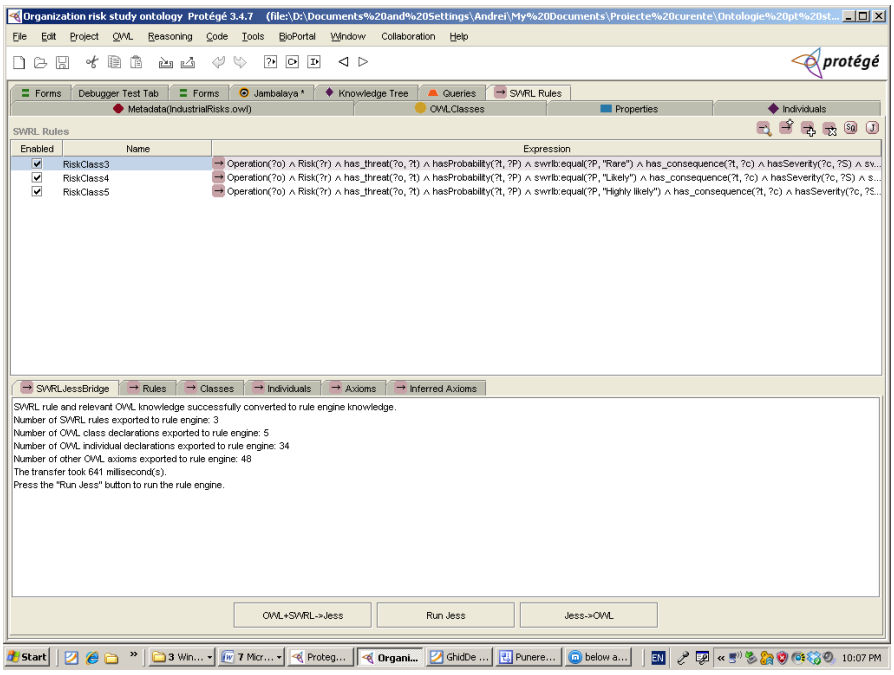

Fig.7-Step 1:Loading the rules in JESS

- Step2-Inferred axioms

We run the selected rules in JESS, by clicking on “Run Jess” button.

Intermediary results of the process of running the rules can be inspected with tabs “Rules”, “Classes”, “Individuals”, “Axioms” and “Inferred Axioms”. Each of these tabs will show respectively the selected rules run with Jess, the selected classes included, the individuals selected and the whole number of axioms processed before providing the final results of inferred axioms. These are the results we expect and in the screen below can be seen they are the same with the ones predicted previously:

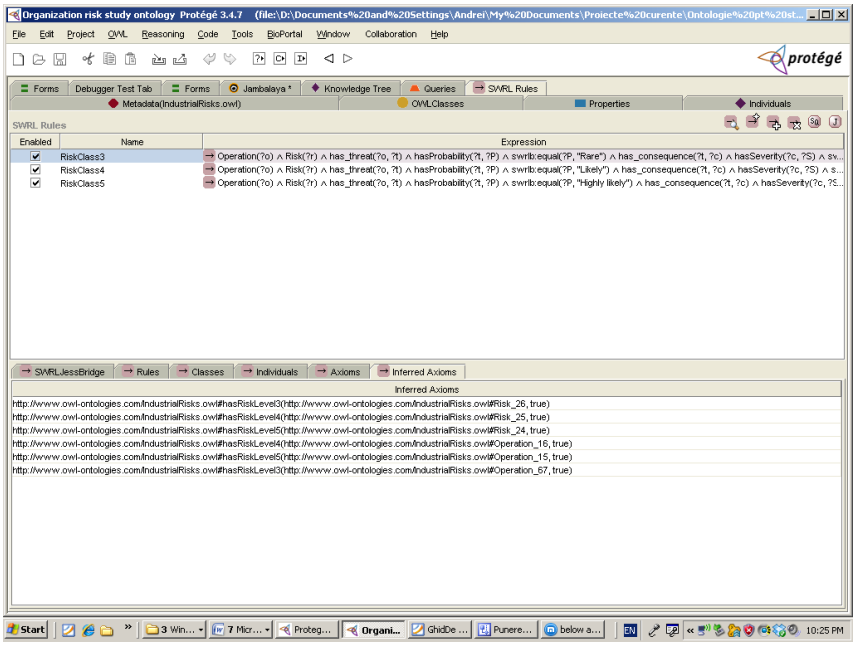

Fig.8-Step 2:Inferred axioms in running the rules with JESS

As an example, we can study one of the records:

http://www.owl-ontologies.com/IndustrialRisks.owl#hasRiskLevel5(http://www.owl-ontologies.com/IndustrialRisks.owl#Risk_24, true)

From this record, it can be noticed that a new fact was inferred, namely:
Risk_24 *hasRiskLevel5* = TRUE, which is one the expected results stated previously.

- Step 3- Loading the new facts back in the ontology

This is a very important step, the final one, by which the new facts are loaded back in the ontology. In this way, the ontology gets enriched with new facts coming from inferred axioms obtained with JESS running initially designed rules. Loading the ontology with the inferred axioms is done by clicking on the third button "Jess--->OWL". The screen showing that new 6 axioms have been loaded back in the ontology is given below.

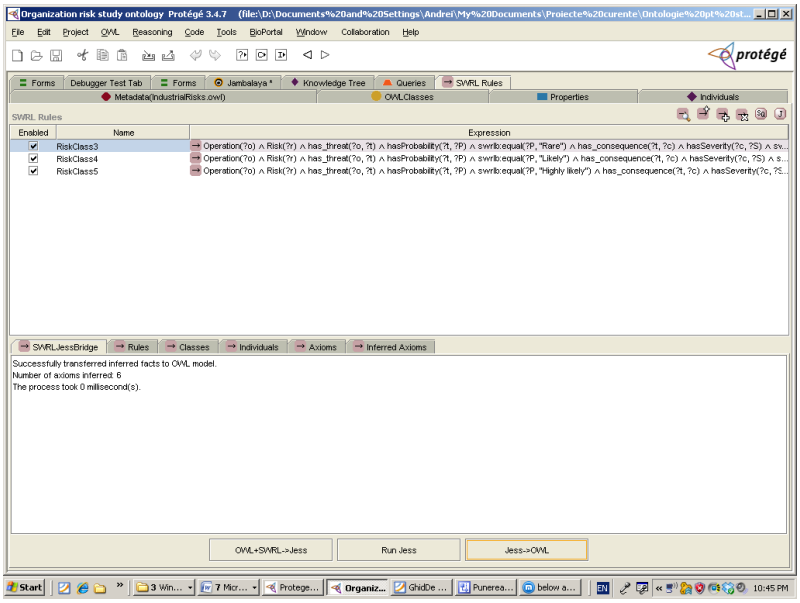

Fig.9-Step 3:Inferred axioms loaded back in the ontology by JESS

Inspecting the new facts loaded back in ontology by JESS

We expect to find the following values assigned to the involved "Risk" and "Operation" individuals according the previously predicted results.

By inspecting the individuals Risk_24 we can notice that *hasRiskLevel5* field received a value of TRUE and similarly, Risk_25 with *hasRiskLevel4* field received a value of TRUE and Risk_26 with *hasRiskLevel3* field received a value of TRUE.

We have the same situation concerning the "Operation" individuals, where Operation_15 has in the field *hasRiskLevel5* a value of TRUE, Operation_16 has in the field *hasRiskLevel4* a value of TRUE and Operation _67 has in the field *hasRiskLevel3* a value of TRUE.

A screen showing one of these results is given below.

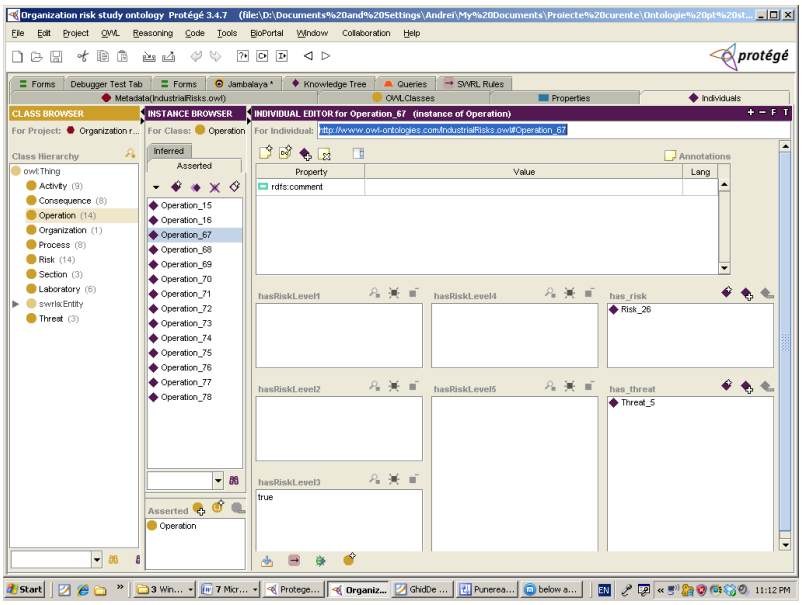

Fig.10 New facts loaded back in ontology by JESS

With this, we can conclude that our purpose of inferring a risk level for Operation and Risk individuals based on SWRL rules run with JESS was reached.

4. CONCLUSIONS AND DISCUSSIONS

4.1 We can define in a risk management ontology risk levels assigned to Risk and Operations individuals, based on SWRL rules that use the probability of a threat and the severity of its consequence. Thus, this can be a method of implementing the risk levels matrix in a risk management ontology.

4.2 At this stage we only managed to model the risk levels matrix by using boolean values attached to specifically designed properties. Would be interesting to define rules that assign integer values to a RiskLevel datatype property and based on implemented measures of preventions to be able to modify this values. At the moment this seems a dificult task to be achieved, due to the monotonicity of SWRL rules, that needs to be overcome.

5. REFERENCES

1. COPPÉE, G., **Occupational Health Services and Practice**, in J. Mager Stellman (Ed.), Encyclopaedia of Occupational Health and Safety 1998, Geneva, PA: International Labour Office.
2. METHODWARE, **ISO 31000: Risk Management Standard**, 2009, www.methodware.com/iso-31000-risk-management-standard-published [visited, May 14, 2010]
3. KLEMEN, M., WEIPPL, E., EKELHART A., FENZ, S., **Security ontology: Simulating threats to corporate assets**, Proc. 2nd Intl. Conf. on Information Systems Security (ICISS), Springer, 2006
4. O'CONNOR, M., KNUBLAUCH, H., TU, S., GROSOF, B., DEAN, M., GROSSO, W., and MUSEN, M. (2005). **Supporting Rule System Interoperability on the Semantic Web with SWRL**. In The Semantic Web – ISWC 2005, Y. Gil, E. Motta, V. Benjamins, and M. Musen, eds. (Springer Berlin / Heidelberg), pp. 974–986.

5. ALEXANDRU, A., FILIP, F., GALATESCU, A., JITARU, E., **Using Ontologies in eHealth and Biomedicine**, in book A. Shukla and R. Tiwari (Eds) "Intelligent Medical Technologies and Biomedical Engineering: Tools and Applications", IGP Global, May, 2010
6. GALATESCU, A., ALEXANDRU, A., ZAHARIA, C., KOVACS, S. **Ontology-based Modeling and Inference for Risk Prevention**, Proc. Intl. Conf. on Advances in Semantic Processing, SEMAPRO, Florence, Italy, October, 2010