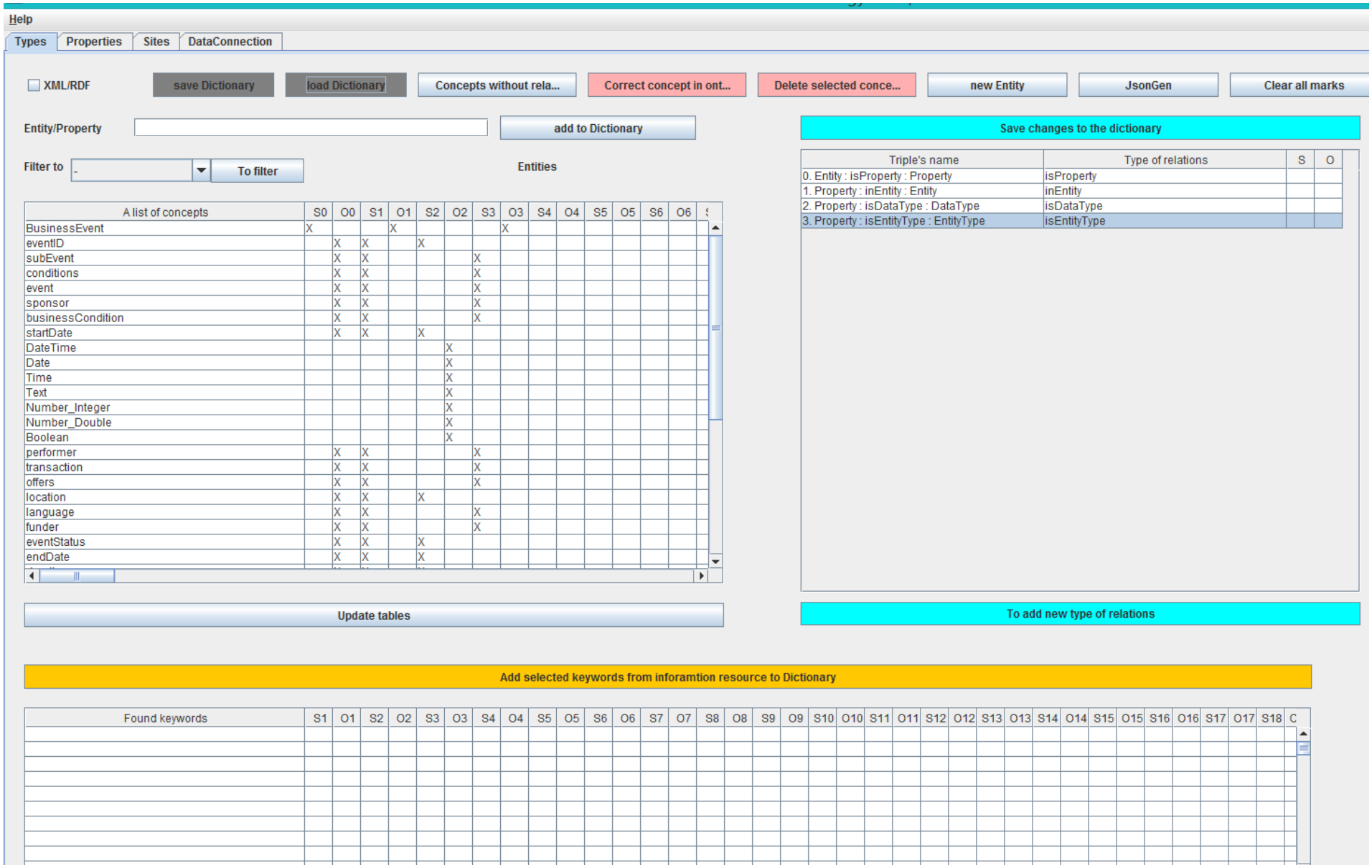
Knowledgebase tools for working with Ontology

description

Main panel. Sight



On sl.2 we can see the dictionary panel for working with concepts of our ontology.

From your right we have the list of concepts and their dependence for type of relations (triples).

The triples you can see from your left part where we can point the dependence for each concepts. As result we have a dictionary in the format on sl.4.

The tools can works with XML / JSON / RDF formats of files.

***JSON and RDF needs some corrections**

Dictionary XML file

```
<?xml version="1.0" encoding="UTF-8"?>
- <hierarchy-types>
  - <is-type-relation>
      <triple id="0">0. Entity : isProperty : Property|isProperty</triple>
      <triple id="1">1. Property : inEntity : Entity|inEntity</triple>
      <triple id="2">2. Property : isDataType : DataType|isDataType</triple>
      <triple id="3">3. Property : isEntityType : EntityType|isEntityType</triple>
    </is-type-relation>
  - <types>
    - <type id="0">
        <name>BusinessEvent</name>
        <s1>X</s1>
        <o3>X</o3>
        <o7>X</o7>
      </type>
    - <type id="1">
        <name>eventID</name>
        <o1>X</o1>
        <s3>X</s3>
        <s5>X</s5>
      </type>
    - <type id="2">
        <name>subEvent</name>
        <o1>X</o1>
        <s3>X</s3>
        <s7>X</s7>
      </type>
    - <type id="3">
        <name>conditions</name>
        <o1>X</o1>
        <s3>X</s3>
```

Dictionary JSON file


```
[
 {
"T":[
 "afd",
  "sdf"
   ]
 },
 {
"T":[
 "hfshf",
"fgdghjh"
   ]
 },
 {
"C":[
 "111",
  "1",
  "2"
  ]
 },
 {
"C":[
"111111",
  "1",
  "3"
  ]
 },
 {
"C":[
 "22",
  "1"
  ]
 }
]
```

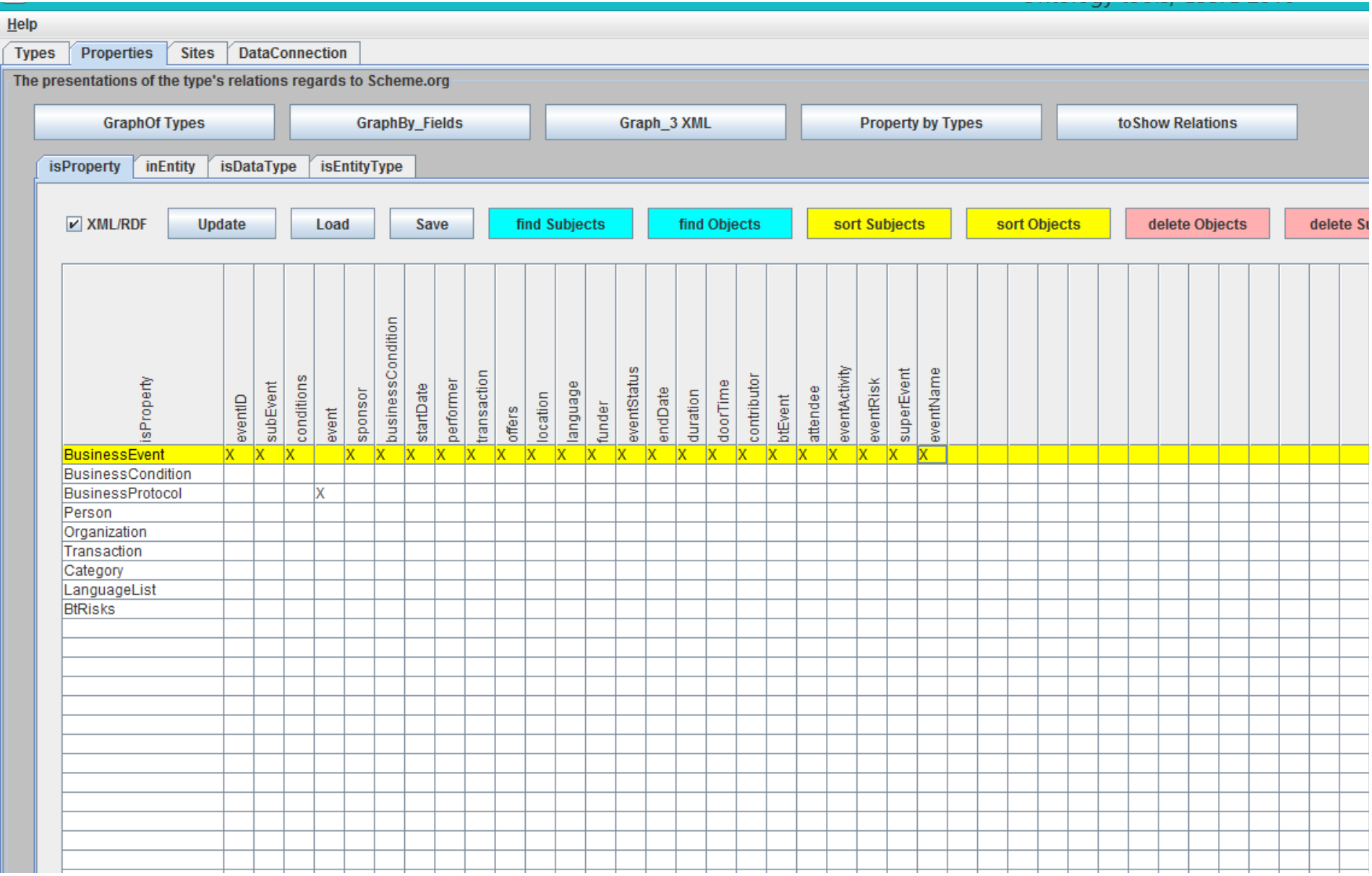
On sl.6 we can see the panel for working with semantic relations among concepts of our ontology. We have the table's presentation of semantic relations among concepts.

Here, clicking the mouse, we can easy make relation "X" between subject and object. The name of table presentation is equal to the type of relations. So, when we see on sl.2 we will have 4 types of relations: "isProperty", "inEntity", "isDataType", "isEntityType".

On. sl.6 we can see the relations of properties with entity "BusinessEvent". We add other entities, but will make relations when they will have their. After, we save file we will have it on sl.7.

We execute this procedure for all type of relations (sl.8-11)

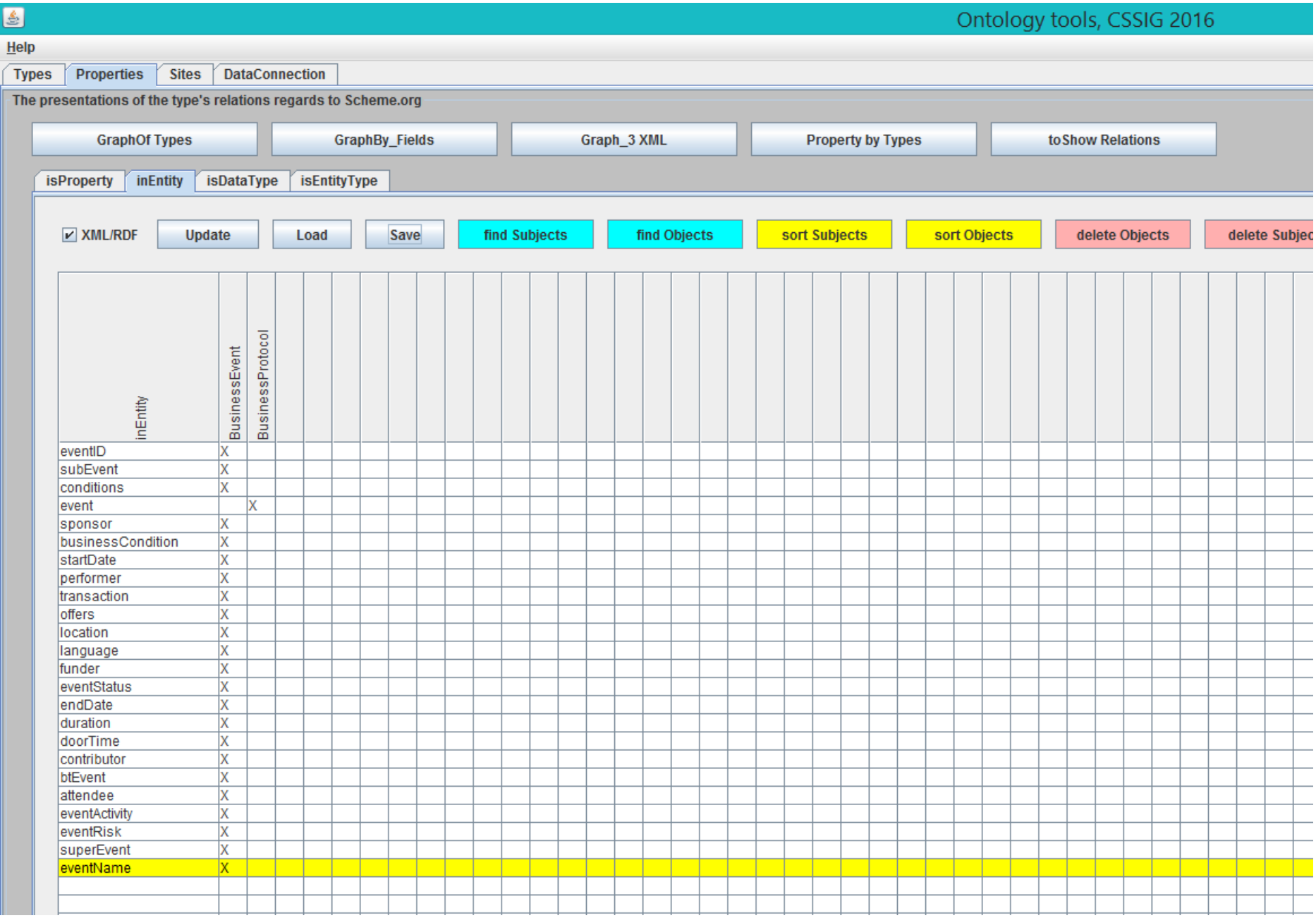
Type of relations



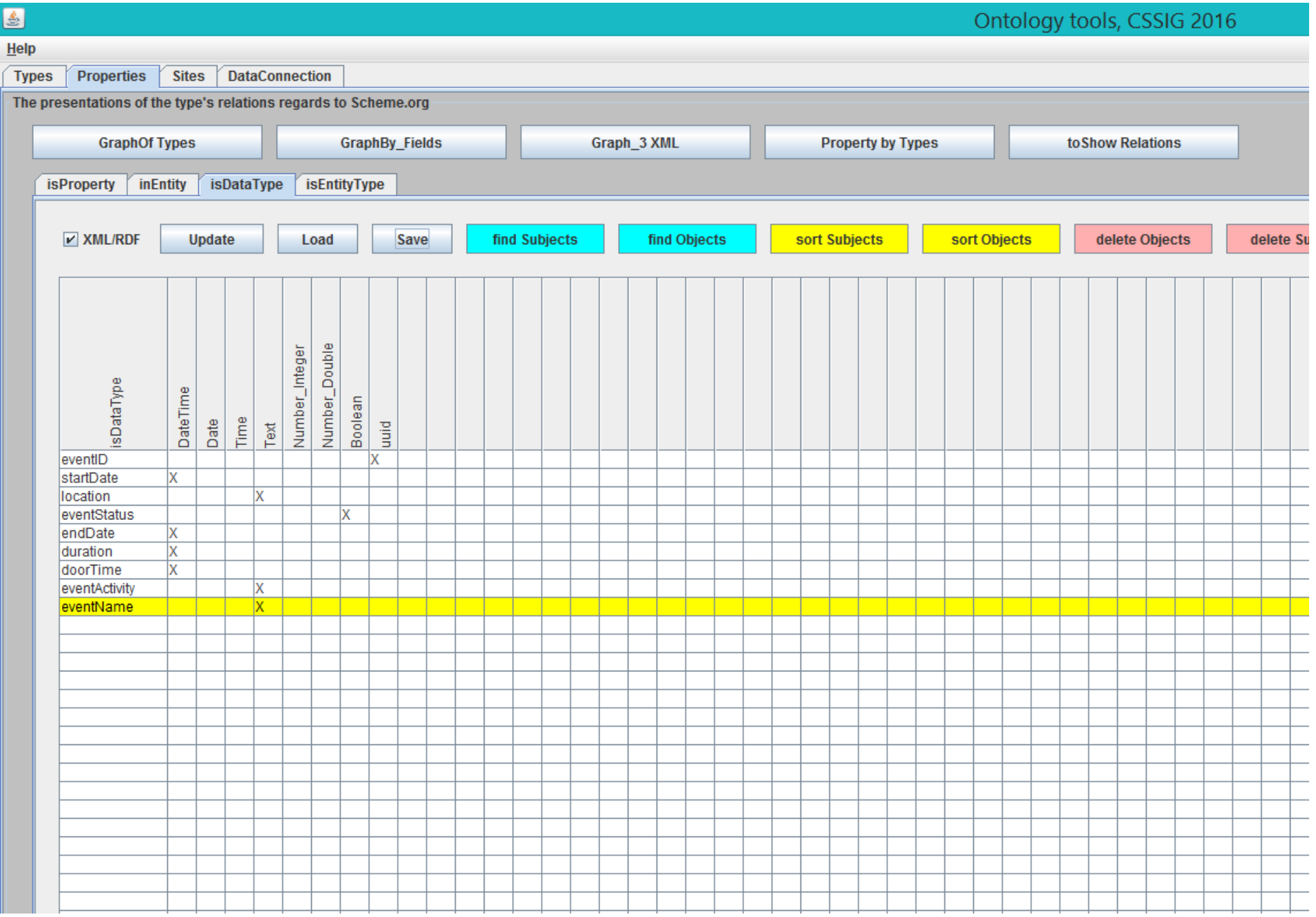
isProperty XML file

D:\Freelancer\AI_ML_NLP\Soft\Ontology\isProperty.xml

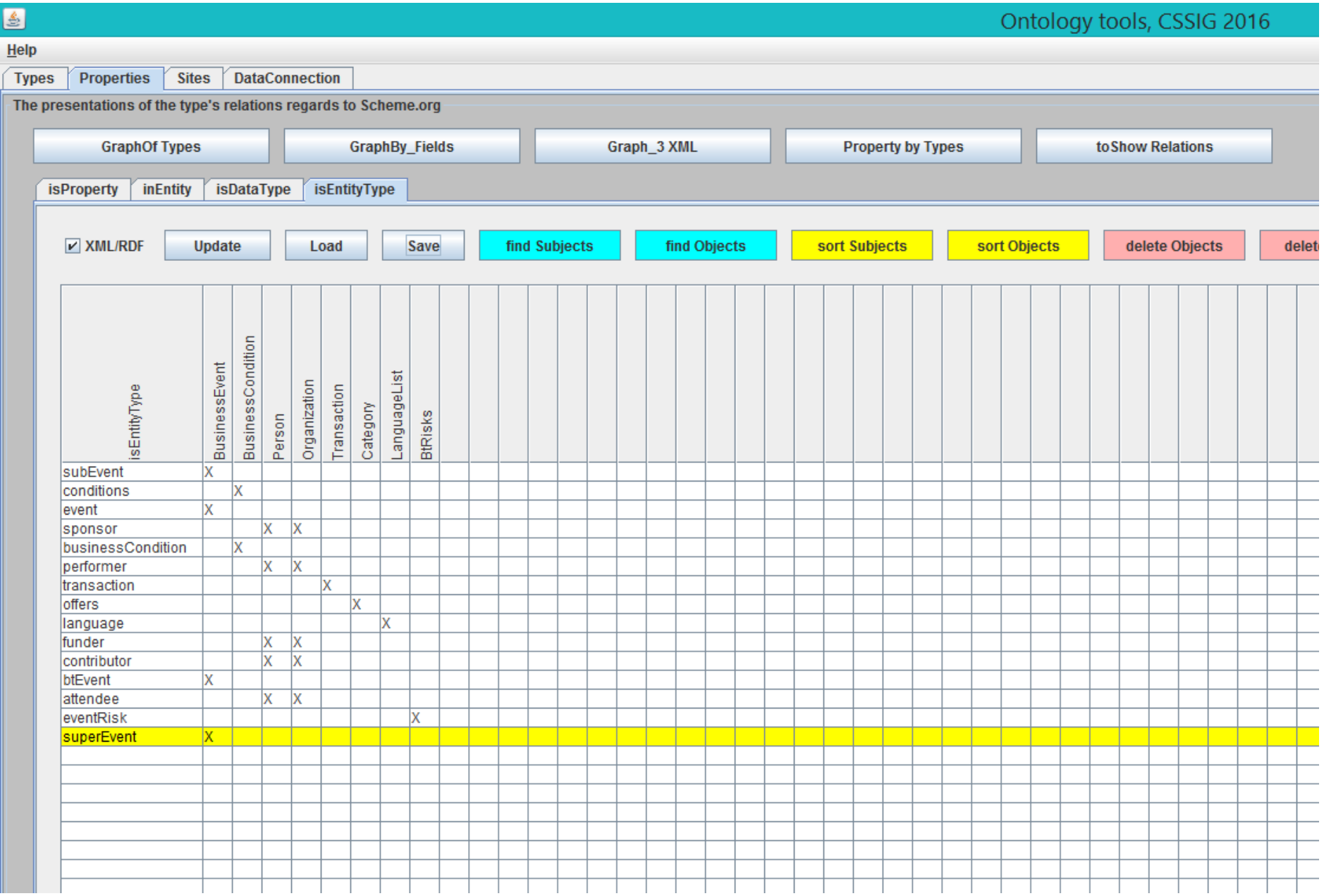
```
- <subject id="0">
    <subject_name>BusinessEvent</subject_name>
    <object_name>eventID</object_name>
    <object_name>subEvent</object_name>
    <object_name>conditions</object_name>
    <object_name>sponsor</object_name>
    <object_name>businessCondition</object_name>
    <object_name>startDate</object_name>
    <object_name>performer</object_name>
    <object_name>transaction</object_name>
    <object_name>offers</object_name>
    <object_name>location</object_name>
    <object_name>language</object_name>
    <object_name>funder</object_name>
    <object_name>eventStatus</object_name>
    <object_name>endDate</object_name>
    <object_name>duration</object_name>
    <object_name>doorTime</object_name>
    <object_name>contributor</object_name>
    <object_name>btEvent</object_name>
    <object_name>attendee</object_name>
    <object_name>eventActivity</object_name>
    <object_name>eventRisk</object_name>
    <object_name>superEvent</object_name>
    <object_name>eventName</object_name>
  </subject>
- <subject id="1">
    <subject_name>BusinessCondition</subject_name>
  </subject>
- <subject id="2">
    <subject_name>BusinessProtocol</subject_name>
    <object_name>event</object_name>
  </subject>
- <subject id="3">
    <subject_name>Person</subject_name>
  </subject>
- <subject id="4">
    <subject_name>Organization</subject_name>
```



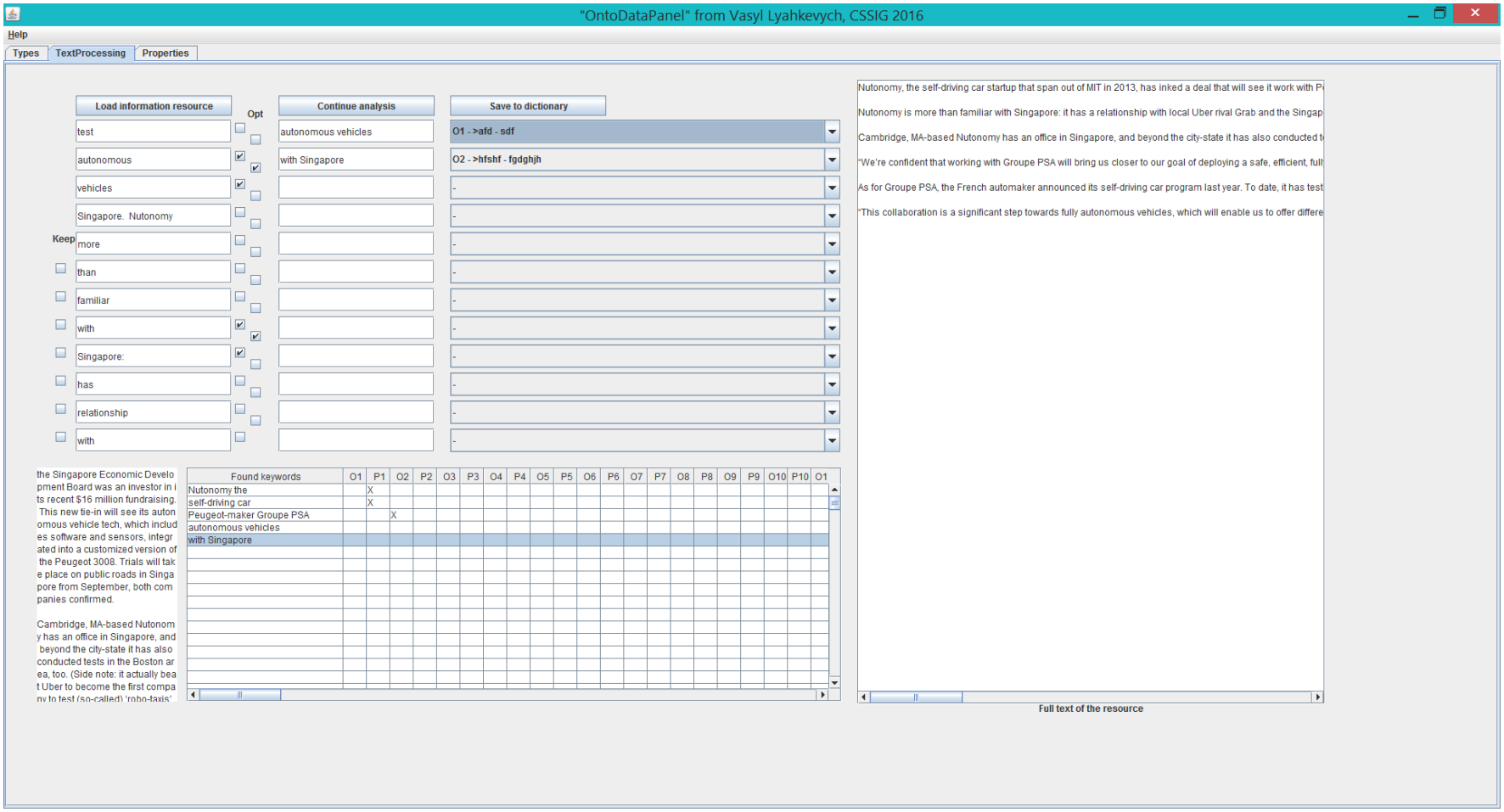
inEntity	BusinessEvent	BusinessProtocol
eventID	X	
subEvent	X	
conditions	X	
event		X
sponsor	X	
businessCondition	X	
startDate	X	
performer	X	
transaction	X	
offers	X	
location	X	
language	X	
funder	X	
eventStatus	X	
endDate	X	
duration	X	
doorTime	X	
contributor	X	
btEvent	X	
attendee	X	
eventActivity	X	
eventRisk	X	
superEvent	X	
eventName	X	



isDataType	DateTime	Date	Time	Text	Number_Integer	Number_Double	Boolean	uuid
eventID								X
startDate	X							
location				X				
eventStatus							X	
endDate	X							
duration	X							
doorTime	X							
eventActivity				X				
eventName				X				

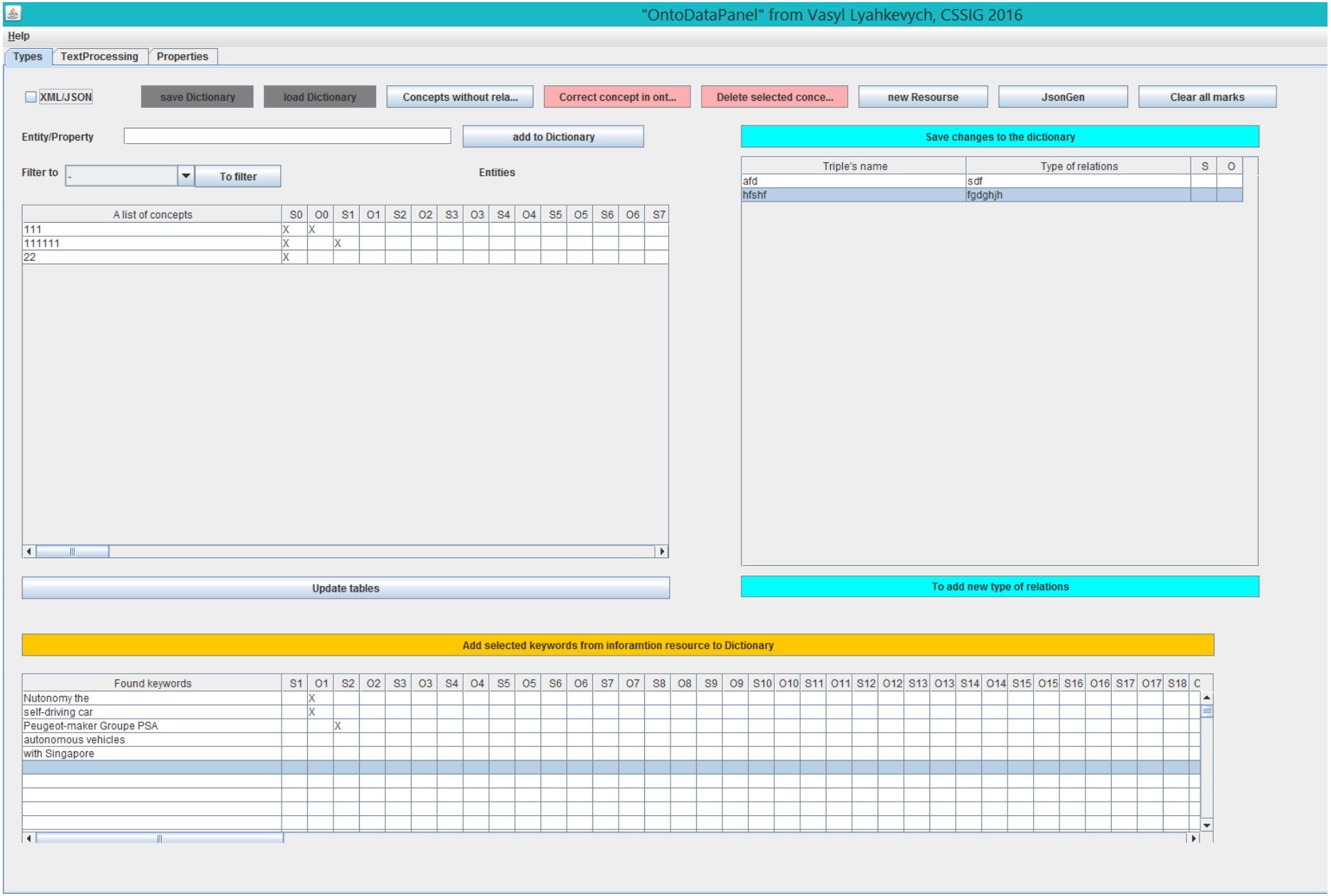

Help
Types
Properties
Sites
DataConnection
The presentations of the type's relations regards to Scheme.org
GraphOf Types
GraphBy_Fields
Graph_3 XML
Property by Types
toShow Relations
isProperty
inEntity
isDataType
isEntityType
XML/RDF
Update
Load
Save
find Subjects
find Objects
sort Subjects
sort Objects
delete Objects
isEntityType
BusinessEvent
BusinessCondition
Person
Organization
Transaction
Category
LanguageList
BtRisks
subEvent X
conditions X
event X
sponsor X X
businessCondition X
performer X X
transaction X
offers X
language X
funder X X
contributor X X
btEvent X
attendee X X
eventRisk X
superEvent X

Tokenizing of textual resources by taxonomy



Expert manually can create different n-grams from real textual resources and CREATE relations with appropriate table like: object or subject in it

Add new tags, concepts and other n-grams to the dictionary



Conclusion

When we have data we can program once taxonomy and generate it many times automatically when we need after improving or changing.

The taxonomy doesn’t requires any changes in the concepts presentation.

Thank you