Ontology Generation Pipeline Manual

|  |  |  |
| --- | --- | --- |
| Document Details | | |
| **Version** | 3.0 |
| **Issue Date** | 19 November 2021 |
| **Review Date** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Document Control | | | |
|  | **Name** | **Role** | **Date** | |
| **Document Owner** | Diego Musmarra |  |  | |
| **Document Approver** | Andrea Tesseri | Director, Copernicus Operations |  | |
| **Document Verifier** | Gaia Cipolletta | System Engineer, Copernicus Operations |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| Version History | | |  |
| **Version** | **Date** | **Reason for release/version update** | |
| Draft 1.0 | 26/11/2021 | Initial draft in progress | |
| Draft 2.0 | 03/12/2021 | Integration of OWLGrEd and updating installation requirements | |
|  |  |  | |
|  |  |  | |

Contents

1 Introduction 4

1.1 Applicable Documents 4

1.2 Reference Documents 4

1.3 Acronyms 4

2 Overview 5

2.1 Widoco 5

2.2 Protégé 5

2.3 WebOWL 5

2.4 OWLGrEd 5

3 System Design 6

3.1 System pre-requirements 6

3.1.1 Ontology Generation Environment 6

3.1.2 Ontology publication environment 7

3.1.3 Configuration Control environment 7

3.2 Deployment model 7

3.2.1 Dev Environment 8

3.2.2 Ops Environment 9

3.2.3 SSL VPN 9

4 Installation Procedure 9

4.1.1 SW installation procedures 9

4.1.2 Web Server VM SW installation procedures 11

5 Getting started 11

5.1 Documentation Publication 22

6 Maintenance Procedures 22

6.1 Backup/Restore 22

6.2 Configuration Control 22

Figures

[Figure 1: System design 6](#_Toc90393755)

[Figure 2: Deployment configuration on ONDA Public Cloud infrastructure. 8](#_Toc90393756)

[Figure 3: Private network, security group rules. 8](#_Toc90393757)

[Figure 4: Public network, security group rules. 9](#_Toc90393758)

[Figure 5: Widoco download item. 9](#_Toc90393759)

[Figure 6: Protégé download item. 10](#_Toc90393760)

[Figure 7: Webvowl download item. 10](#_Toc90393761)

[Figure 8: Widoco getting started, select a template window. 11](#_Toc90393762)

[Figure 9: Widoco getting started, select configuration file. 12](#_Toc90393763)

[Figure 10: Widoco getting started, customization. 13](#_Toc90393764)

[Figure 11: Widoco getting started, finish. 13](#_Toc90393765)

[Figure 12: Protégé getting started, create new OWL. 14](#_Toc90393766)

[Figure 13: Protégé getting started, ontology URI selection. 15](#_Toc90393767)

[Figure 14: Protégé getting started, rename ontology file. 15](#_Toc90393768)

[Figure 15: Protégé getting started, annotations button click. 16](#_Toc90393769)

[Figure 16: Protégé getting started, annotation panel. 17](#_Toc90393770)

[Figure 17: Protégé getting started, class tab overview with steps marked in red. 18](#_Toc90393771)

[Figure 18: Protégé getting started, data properties tab with steps marked in red. 19](#_Toc90393772)

[Figure 19: WebVOWL getting started, select ontology file. 20](#_Toc90393773)

[Figure 20: WebVOWL visualization example, given ontology. 21](#_Toc90393774)

Tables

**No table of figures entries found.**

# Introduction

The purpose of this document is to describe the installation ad configuration procedure of the Ontology Generation pipeline. This document has been prepared by MTL Services srl in the frame of the “Information Model definition, governance and prototyping” contract.

## Applicable Documents

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Document Title** | **Refrence** | **Version** |
|  |  |  |  |
|  |  |  |  |

## Reference Documents

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Document Title** | **Refrence** | **Version** |
|  |  |  |  |
|  |  |  |  |

## Acronyms

|  |  |
| --- | --- |
| ESA | European Space Agency |
| VM | Virtual Machine |

# Overview

An Information Model is an abstract, formal representation of entity types including their properties, relationships and the operations that can be performed on them.

It is required to provide the declarative representation of entities, being representative of the operations and being directly operable.

The realization is typically organised along the following stages.

* + Definition, providing a conceptual representation of the ontology and relative contents identified.
  + Declarative representation, resulted by the deployment of the ontology, a more extensive specification of the contents identified in the definition process in terms of classes, objects and data properties.
  + Programmatic representation, resulting in a library of APIs feeding the Policy Framework.

Governance aspects of the Information Model regulate the associated management applications along the stages outlined above, as well as the configuration control of the deployed objects.

In terms of concepts, the information model shall be representative of all the entities establishing the administrative control on a given system, exposing properties in the form of an ontology

The ontology wraps all the descriptive and functional attributes, being functional to the representation of the data, and conceptualising a schema across its properties.

For the activity the following 3rd party tools have been used:

* Widoco
* Protegé
* WebOWL
* OWL Gred

A Github repository has been set up for the purpose of the open source management configuration control.

## Widoco

WIDOCO helps you to publish and create an enriched and customized documentation of your ontology automatically, by following a series of steps in a GUI.

Visit the official GitHub at the following link:

<https://github.com/dgarijo/Widoco>

## Protégé

Protégé is a free, open-source ontology editor and framework for building intelligent systems.

Visit the official website at the following link:

<https://protege.stanford.edu>

## WebOWL

WebVOWL is a web application for the interactive visualization of ontologies. It implements the Visual Notation for OWL Ontologies (VOWL) by providing graphical depictions for elements of the Web Ontology Language (OWL) that are combined to a force-directed graph layout representing the ontology.

Visit the official website at the following link:

<http://vowl.visualdataweb.org/webvowl.html>

## OWLGrEd

OWLGrEd is another Ontology Visualization online tool.

Visit the official website at the following link:

<http://owlgred.lumii.lv>

# System Design

The Information Model System can be divided into the following conceptual areas.

* The Ontology Generation Environment, hosting the tools for creating and updating the information model
* The Ontology Publication Environment, hosting the virtual resources for the documentation publication
* The Open-source configuration control environment, for documentation lifecycle



Figure 1: System design

The installation procedures provided in this document are referred to the development activities conducted on the development environment set up within the OVH public cloud infrastructure.

## System pre-requirements

The pipeline has been put in place on two VM within the ONDA Public Cloud infrastructure for development activities.

1. A back-end VM, included into a private security group.
2. A Web Server VM, used to expose the web documentation to the internet, within the same back-end network.
3. An SSL VPN to enable access to the development environment

Both VMs should be able to reach the internet (ports 80, 443) and being reachable on port 22 via the key.

Further resources to be set up for the pre-operationalization phase.

### Ontology Generation Environment

The Dev VM has been used to test the pipeline for the generation of the information model. It is a remote desktop non-core VM with the following specifications.

* OS: Ubuntu V.4.4.0-154-generic
* RAM: 30GB
* CPU: 8
* Disk: 200 GB
* Image: dias\_template-ubuntu-16-RemoteDesktop\_02
* Other: remote desktop

The following software must be installed on the VM

* WIDOCO, which requires Java Version 1.8 or higher
* PROTÉGÉ
* WEBVOWL, which requires Java Version 1.8 or higher
* OWLGRED

### Ontology publication environment

The WebServer VM has been used to output the web documentation to the internet. The VM shows the following specifications.

* OS: Debian 10
* RAM: 7GB
* CPU: 2
* Disk: 50 GB
* Image: Debian 10
* Other
  + core
  + DNS registered: webontology.onda-dias.eu

The following software must be pre-installed on the VM

* Apache Web Server

### Configuration Control environment

Pre-requirements for the configuration control are

* A GitHub service account, referenced on the VM to conduct the development activities
* A Github repository (blank), properly configured with the read/write access by the service account

## Deployment model

The following configuration reflects the development activities configuration and shall be put in place also for operationalization.

Diagram

Description automatically generated

Figure 2: Deployment configuration on ONDA Public Cloud infrastructure.

### Dev Environment

The development environment is composed by the virtual resources gave in Section 3.1.

A security group has been added and linked to the virtual resources within the Dev environment. This shows the characteristics in Figure below.

Graphical user interface, text, application, email

Description automatically generated

Figure 3: Private network, security group rules.

The ingress specifications ensure that communication is established with the Web Server VM on the Public network.

### Ops Environment

The ops/pre-ops environment is characterised by the same virtual resources of the development environment, but with a set-up which could support operations in this context. This means that whether the system administrator would be able to access the ops environment via VPN, users could access the Web Server on the Internet, on a dedicated domain. Separately, virtual resources within the ops environment are inserted into security groups to ensure a minimum set of security practises.

A security group has been added and linked to the Web Server VM. This shows the characteristics reported below.

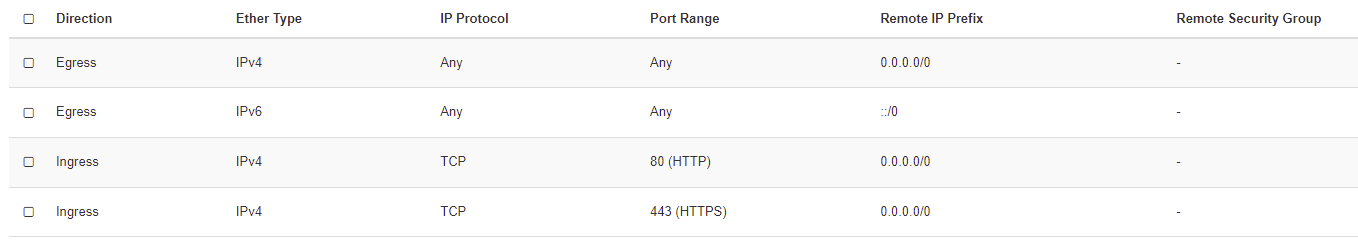


Figure 4: Public network, security group rules.

### SSL VPN

TBW

# Installation Procedure

### SW installation procedures

The required SW (Widoco, Protégé, WebVOWL) can be installed following the steps below (root not required):

**WIDOCO**

* Download the .jar file of latest release at the following link:

<https://github.com/dgarijo/Widoco/releases>

Immagine che contiene testo

Descrizione generata automaticamente

Figure 5: Widoco download item.

**PROTÉGÉ**

* Download the Linux installer at the following link:

<https://protege.stanford.edu/download/protege/4.0/installanywhere/#top>

Immagine che contiene testo

Descrizione generata automaticamente

Figure 6: Protégé download item.

* Run the command “sh ./installerDirectory” and follow the steps of the installation wizard.

**WEBVOWL**

* Download WebVOWL and the OWL2VOWL converter at the following link:

<http://vowl.visualdataweb.org/webvowl.html>

Immagine che contiene testo

Descrizione generata automaticamente

Figure 7: Webvowl download item.

* Unzip the .zip files.

### Web Server VM SW installation procedures

* Install Apache Web Server:
  1. Begin by updating the local package index to reflect the latest upstream changes: “sudo apt update”
  2. Then, install the apache2 package: “sudo apt install apache2“
  3. To check your server status use the following command: “sudo systemctl status apache2“

If the status is “Active” then you can check the default Debian 9 Apache web page: “http://your\_server\_ip“

# Security

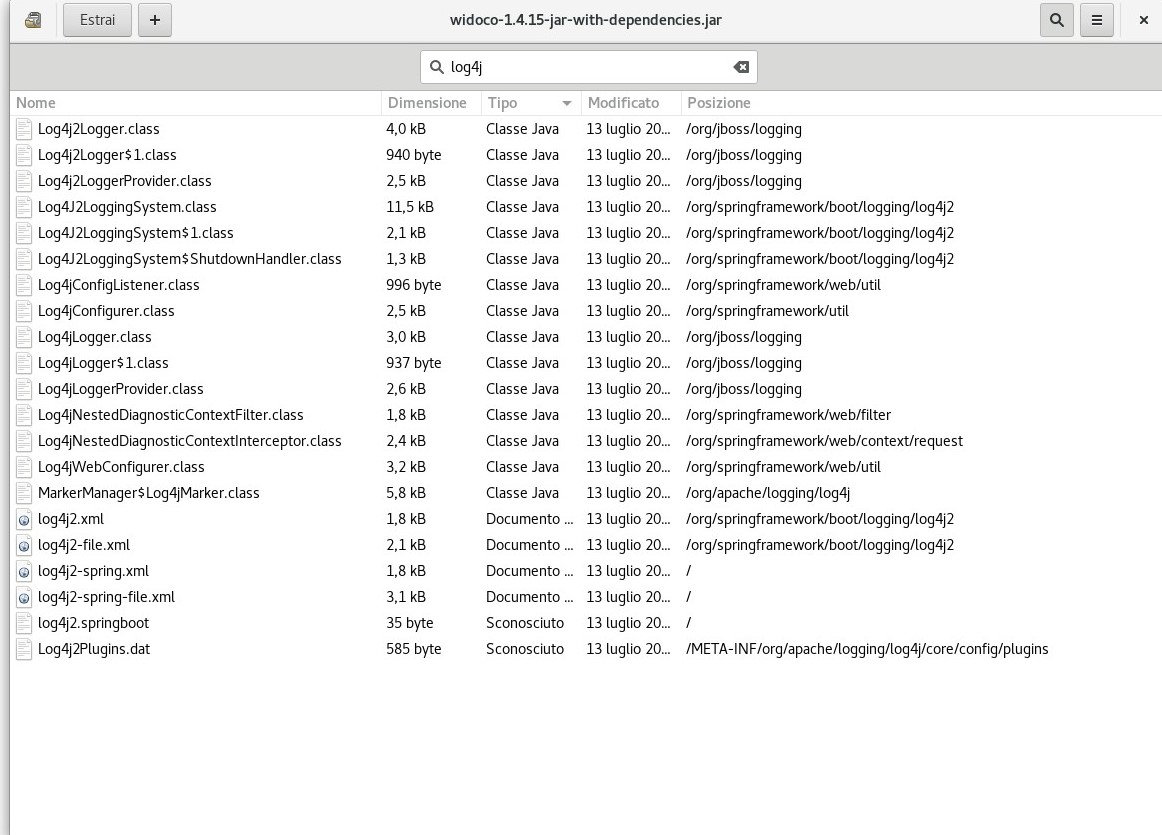
### Log4j Vulnerability

* **Description:** Apache Log4j2 versions 2.0-alpha1 through 2.16.0 did not protect from uncontrolled recursion from self-referential lookups. When the logging configuration uses a non-default Pattern Layout with a Context Lookup (for example, $${ctx:loginId}), attackers with control over Thread Context Map (MDC) input data can craft malicious input data that contains a recursive lookup, resulting in a StackOverflowError that will terminate the process. This is also known as a DOS (Denial of Service) attack.

(Source: <https://logging.apache.org/log4j/2.x/security.html>)

* **Solution**: There are some Log4j files in the Widoco .jar file and, even if they are not used by the tool, it’s safer to delete them.

You can open the .jar file as an archive and search for “Log4j”, you should find something like this:

****

Remove all these files and test the functionality of the Widoco tool, it should work without any problem.

# Getting started

**WIDOCO**

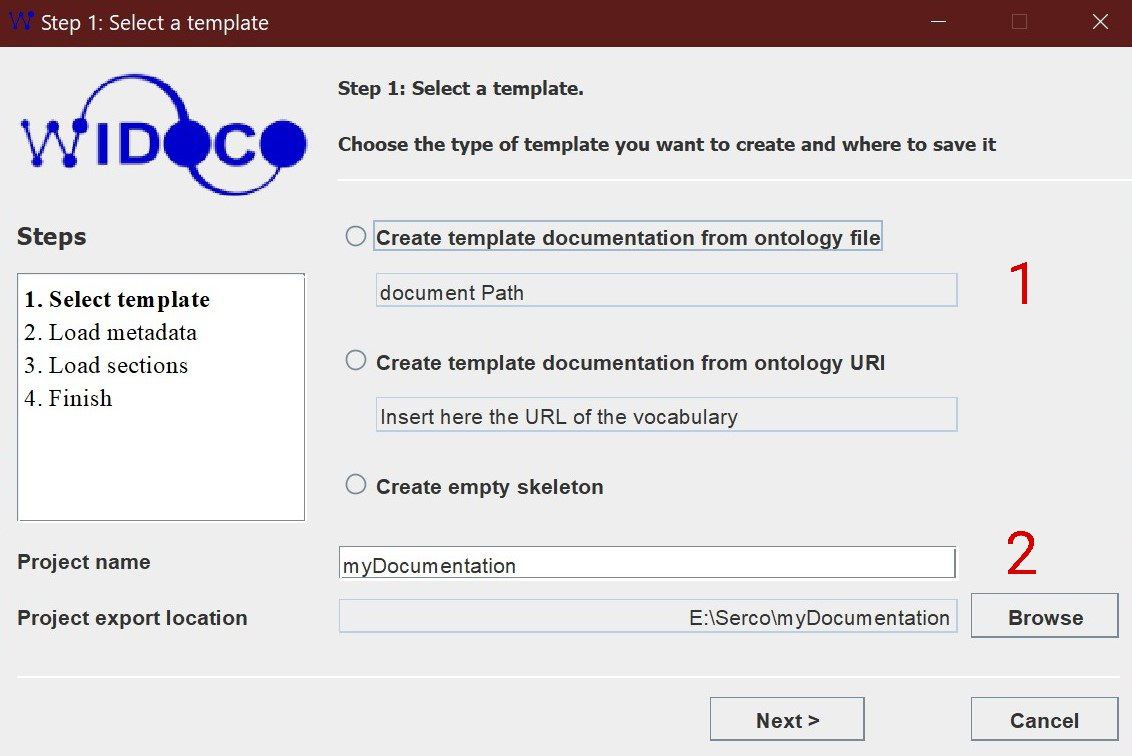


Figure 8: Widoco getting started, select a template window.

* **Select template**

1. Select the Ontology file;
2. Select the path of your documentation;

Immagine che contiene testo, tavolo

Descrizione generata automaticamente

Figure 9: Widoco getting started, select configuration file.

* **Load metadata**
* Download a sample config file [**here**](http://dgarijo.github.io/Widoco/doc/tutorial/config.properties)and modify it as you wish.
* Load your config file.

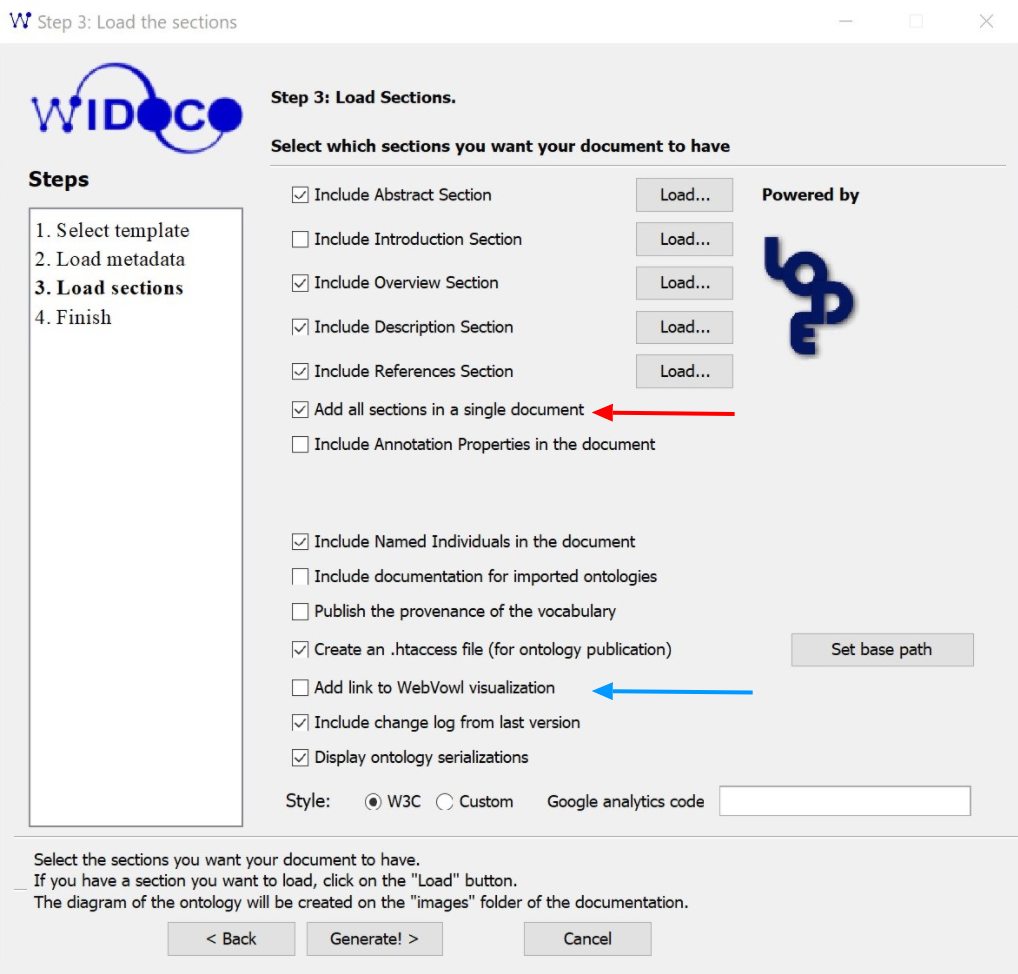
****

Figure 10: Widoco getting started, customization.

* **Load sections** 
  1. Select red options;
  2. Be sure to select the blue option, that will generate a WebVOWL graph in your documentation. (N.B. That will only work if the documentation runs on a Web Server)

**Immagine che contiene testo

Descrizione generata automaticamente**

Figure 11: Widoco getting started, finish.

* **Finish**
* Click on “View the ontology documentation in your browser” or open the index.html file in the documentation directory to visualize the result.

**For more information visit the following links:**

* GUI Tutorial: <http://dgarijo.github.io/Widoco/doc/tutorial/>
* GitHUB:<https://github.com/dgarijo/Widoco>

**PROTÉGÉ**

* Run the command “./run.sh” in the “protege” folder created by the installation wizard.

Immagine che contiene testo

Descrizione generata automaticamente

Figure 12: Protégé getting started, create new OWL.

* Create a new ontology by clicking “Create new OWL ontology”.

Immagine che contiene testo

Descrizione generata automaticamente

Figure 13: Protégé getting started, ontology URI selection.

* Rename your ontology here.

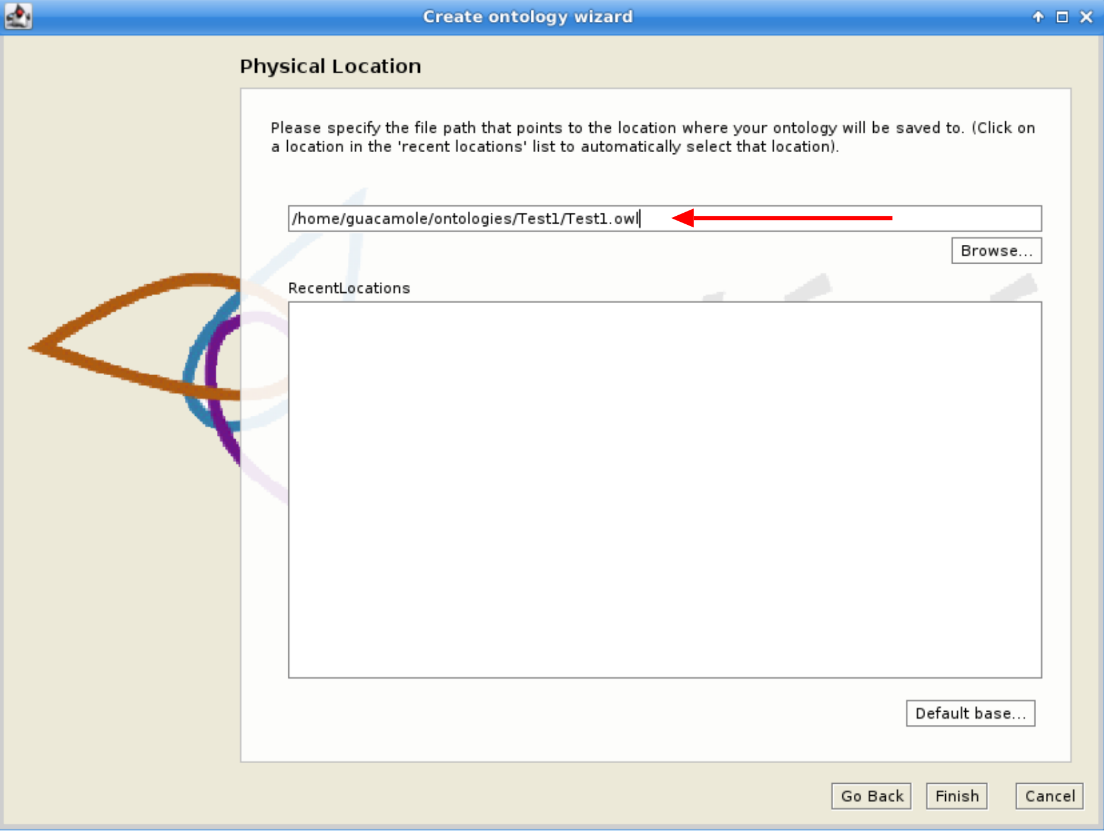


Figure 14: Protégé getting started, rename ontology file.

* Select the path where to save your ontology.

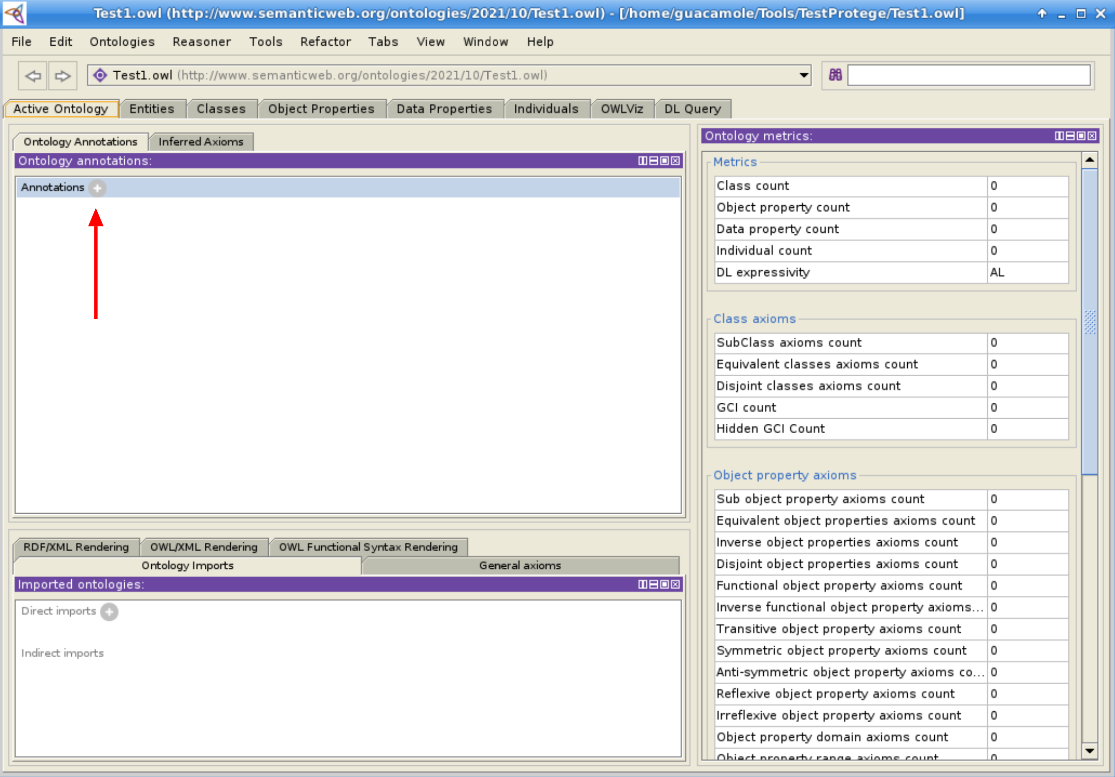


Figure 15: Protégé getting started, annotations button click.

* Click on “Annotations” to add a short description of your ontology.

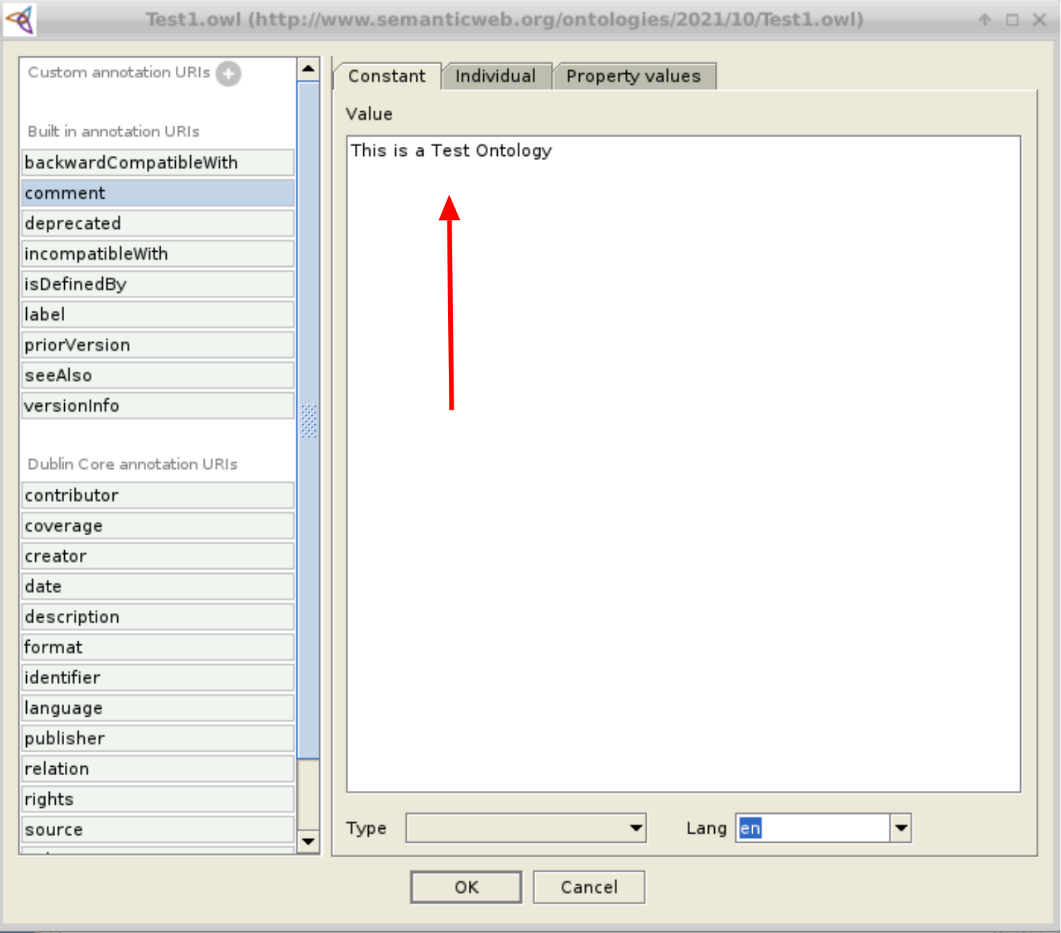


Figure 16: Protégé getting started, annotation panel.

* Here you can write your description and select the language of the comment (e.g., English).

Immagine che contiene testo

Descrizione generata automaticamente

Figure 17: Protégé getting started, class tab overview with steps marked in red.

* **Class tab:**
  1. Add a Sub-Class
  2. Add a Class on the same level of the selected Class
  3. Delete Class
  4. Add a description for your Class

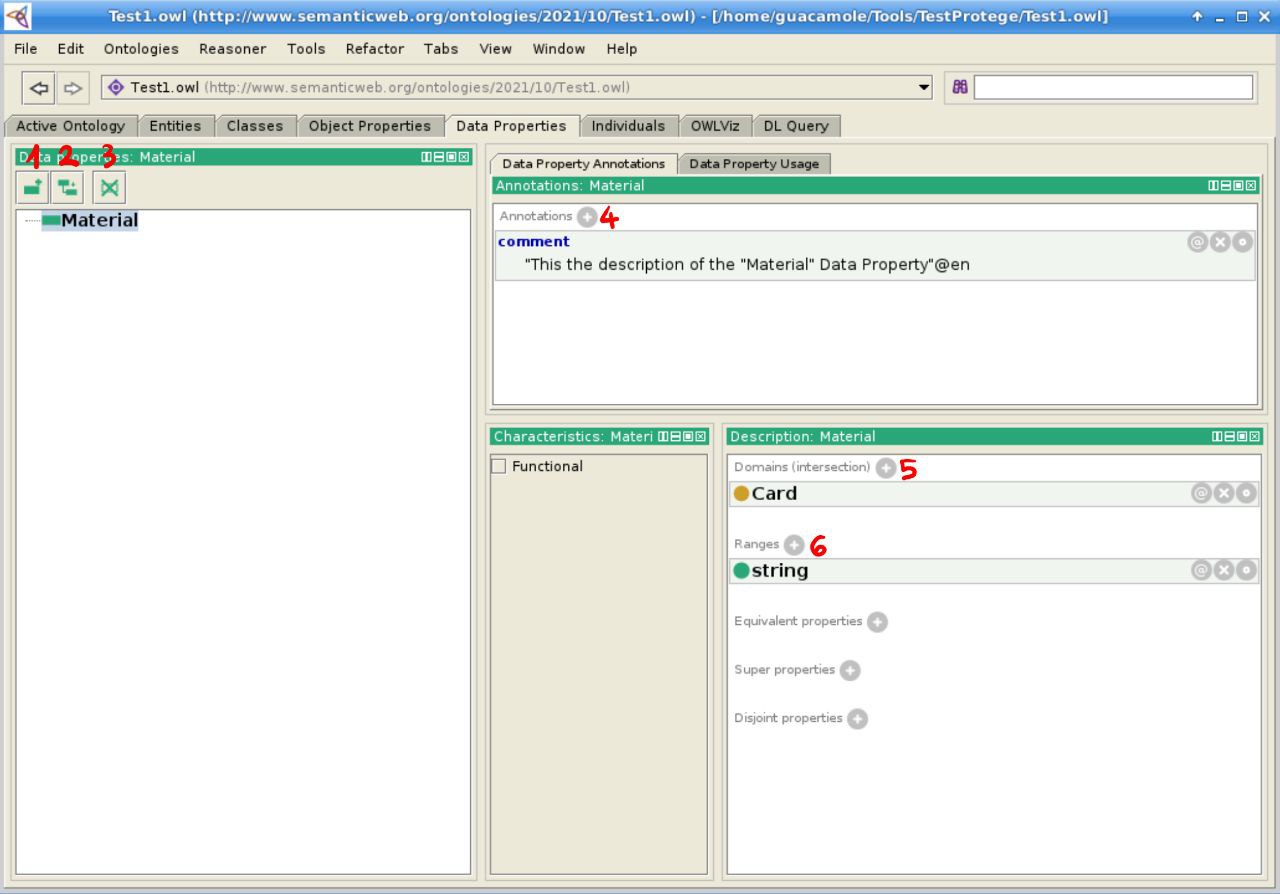
****

Figure 18: Protégé getting started, data properties tab with steps marked in red.

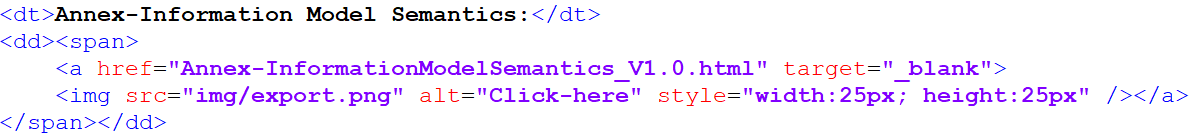
* **Data Properties tab:**
  1. Add a new Data Property
  2. Add a Sub-Property
  3. Delete Property
  4. Add a description for your Data Property
  5. Select the Class linked to the Data Property
  6. Select the type of the Data Property

**For more information visit the following links:**

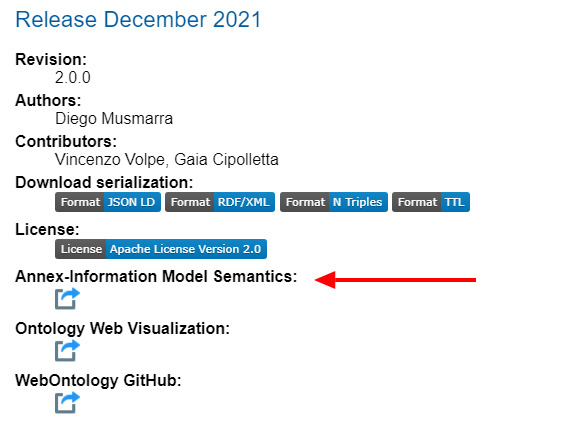
* Protégé Documentation: <http://protegeproject.github.io/protege/>

**ANNEX-INFORMATION MODEL SEMANTICS**

You can also incorporate a .txt file in you documentation, we did an .html file (that you can find in the “Documentation” folder of the [GitHub Repository](https://github.com/SercoSPA/WebOntology)) containing the .txt file information, and then we integrated it into the documentation modifying the “index-en.html:



And the result:



## Documentation Publication

WebServer getting started:

* Commands to control your Web Server:
  + To stop your web server, type: “sudo systemctl stop apache2”
  + To start the web server when it is stopped, type: “sudo systemctl start apache2”
  + To stop and then start the service again, type: “sudo systemctl restart apache2“
  + If you are simply making configuration changes, Apache can often reload without dropping connections. To do this, use this command: “sudo systemctl reload apache2”
  + By default, Apache is configured to start automatically when the server boots. If this is not what you want, disable this behavior by typing: “sudo systemctl disable apache2”
  + To re-enable the service to start up at boot, type: “sudo systemctl enable apache2”
* Put your documentation folder generated with Widoco (see “**WIDOCO**” under “[**Getting Started**](#_Getting_started)”) in the following directory with root permissions: “/var/www/html”
* Give to the Documentation Folder the following permissions: \_ “sudo chmod o+rx DocumentationFolder”
* Now you could reach your documentation from this url: “http://your\_server\_ip/DocumentationFolder/index-en.html"

# Maintenance Procedures

## Backup/Restore

* Backup procedures (applications)

## Configuration Control

**Pre-Requirements**:

* GitHub account, configured on the VM
* GitHub repository

Git instance link: <https://github.com/SercoSPA/WebOntology>

**Git instructions**:

1. Clone repository:
   * “git clone <https://github.com/SercoSPA/WebOntology>”
   * Insert your GitHub username.
   * Insert your Access Token that you can generate [here](https://github.com/settings/tokens).
2. Modify your repository:
   * Put your new files in the clone directory.
   * Verify your pending files: “git status”
   * Add all your pending files: “git add -A”
   * Create a commit: “git commit -a -m CommitDescription”
   * Push your commit: “git push”
3. Update your clone:
   * Go in your clone directory and use the following command: “git pull”
   * Insert your GitHub username.
   * Insert your Access Token that you can generate [here](https://github.com/settings/tokens).

Webpage doc link [here](https://imms.onda-dias.eu/InformationModel/Documentation/index-en.html).

# Annex: Alternative Visualization

**WEBVOWL**

If WIDOCO has been able to generate your WebVOWL graph you don’t have to follow this guide.

* WebVOWL needs the Ontology .owl file that can be generated with Protégé (see **“PROTÉGÉ”** under **“**[**Getting Started**](#_Getting_started)**”)**
* **Converting the .owl file into .json file:** In the converter folder you'll find a .jar file, run the following command to convert your .owl file into a .json file:

java -jar pathOfTheJarFile -file pathOfTheOwlOntologyFile

* **Generating the graph:**
* Open the index.html file that you can find in the WebVOWL folder.

Immagine che contiene testo

Descrizione generata automaticamente

Figure 19: WebVOWL getting started, select ontology file.

* Select the “Ontology” tab and click on “Select ontology file” to import your .json ontology file.

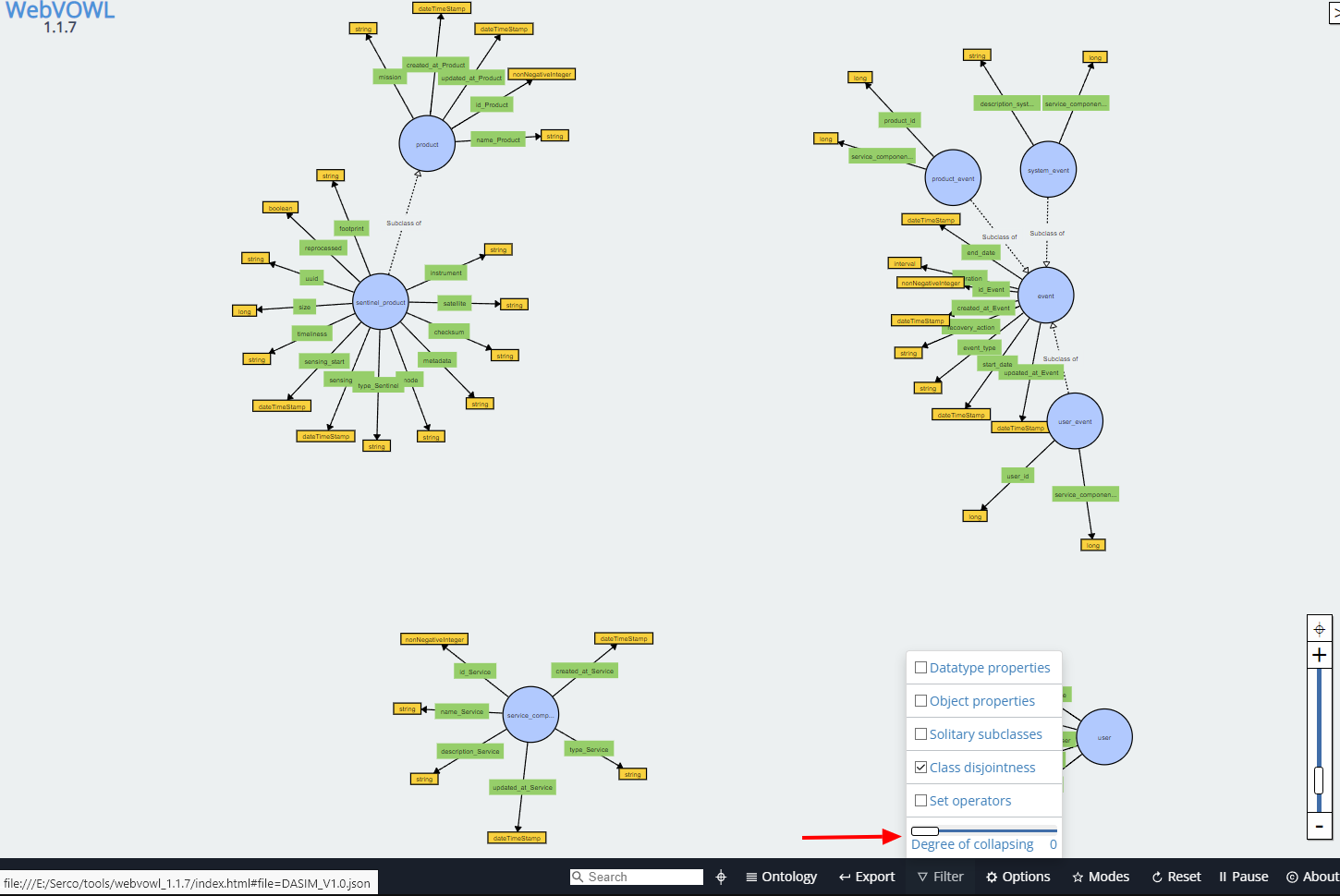
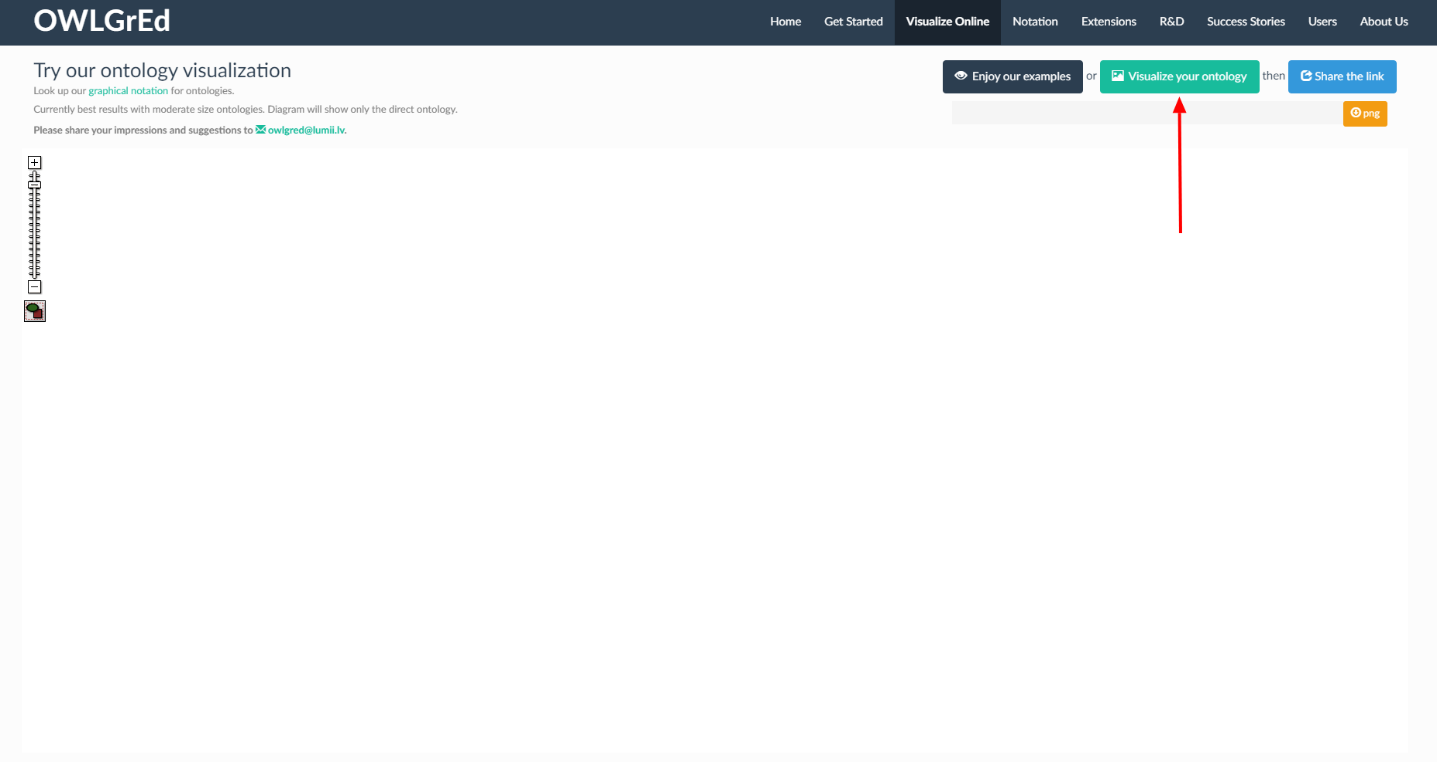


Figure 20: WebVOWL visualization example, given ontology.

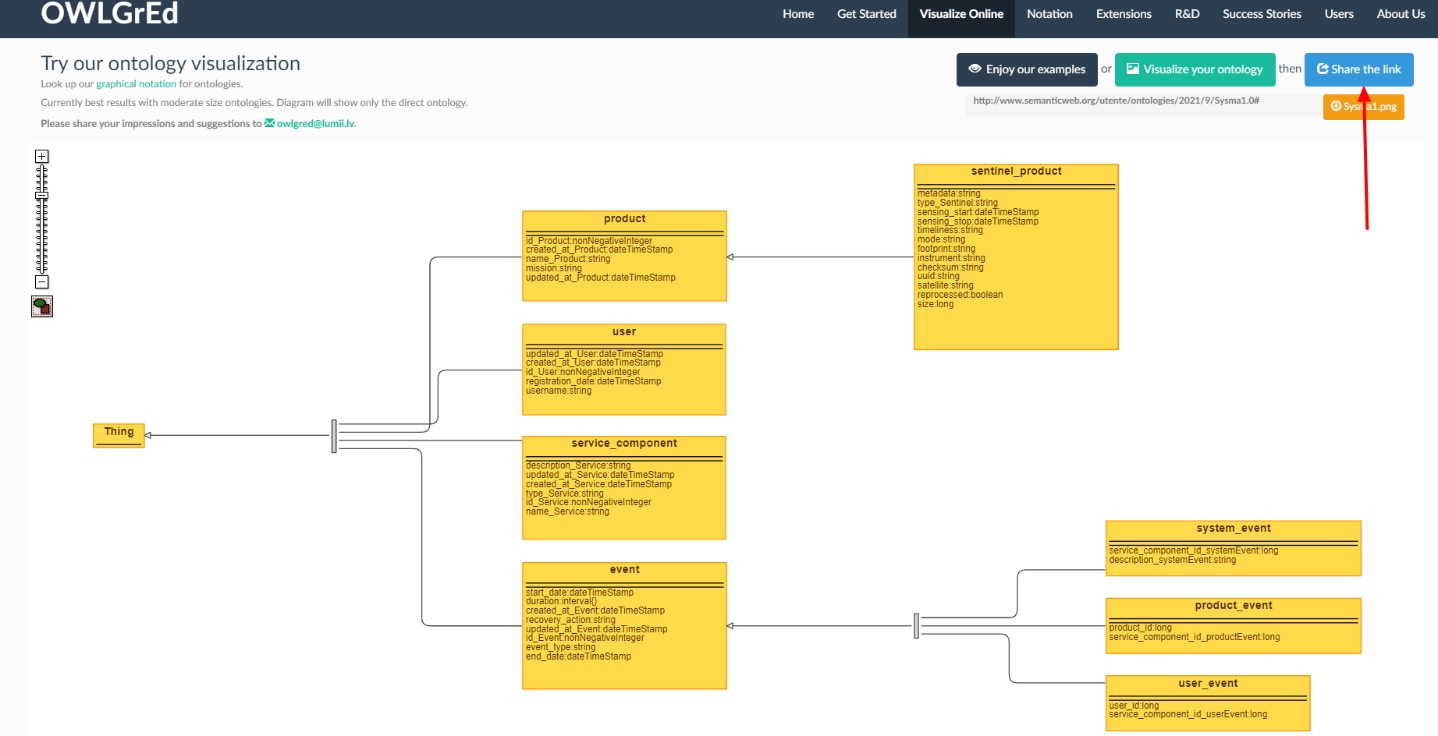
* Select the “Filter” tab and bring the “Degree of collapsing” to 0.

**OWLGRED**

* Go to <http://owlgred.lumii.lv/online_visualization>



* Click on “Visualize your ontology” to import your .owl file



* Click on “Share the link” to copy your Ontology Visualization link.   
  You can now insert the link in your “index-en.html” file to incorporate it in your documentation.
* Here’s an example of how you can incorporate the OWLGrEd link in your documentation:

**Immagine che contiene testo

Descrizione generata automaticamente**

* And here’s the result:

Immagine che contiene testo

Descrizione generata automaticamente