Web-based ontology analysis and partitioning tool

Prepared by:
Ahmad Alzeitoun
Iulia Buga
Imuwahen Osazuwa
Dr. Gökhan Coskun
Irlan Grangel

Software Requirements Specification

Document

Version: (1) **Date: (06/05/2015)**

Revision History

Date	Description	Author	Comments
6.05.2015	First version	Ahmad Alzeitoun Iulia Buga Imuwahen Osazuwa	Initial version

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Dr. Gökhan Coskun		
	Irlan Grangel		

Table of Contents

1. Introduction

Ontologies, i.e. semantic structures encoding concepts, relations and axioms, providing a model of a given domain, are the backbone of the Semantic Web, a semantic-aware version of the World Wide Web. Ontologies allow web resources to be semantically enriched. Ontology building is a task that pertains to ontology engineers, an emerging expert profile that requires the expertise of knowledge engineers (KE) and domain experts (DE). Even though automatic ontology learning methods significantly support ontology engineers by speeding up their task, there is still the need of significant manual effort, in the completiong, consolidation and validation of the automatically generated ontology [1]

Ontology reuse attracts the interest of the Semantic Web community and its current pragmatic version the Linked Data community where ontologies are considered as shared knowledge and interlinked. Even though ontology reuse is part of various ontology engineering methodologies, there are no best practice solutions which describe how existing ontologies should be analyzed for their resusability. [2] The reuse and collaborative development of vocabularies and ontologies require tool support to understand the content and the change. To accomplish this, the **"Web-based analysis and partitioning tool"** is developed as a scalable, open platform web application. Its design allows for quickly uploading large ontologies and partitioning the ontologies into new ones.

1.1 Purpose

This document defines the software requirements involved in the project. Furthermore, it describes the objectives and scope of the project.This document is to be used in the software development process by stakeholders, software developers, testers and project managers.

1.2 Scope

The scope of the **WAPT** project is defined in what follows :

- analyze aspects like structure and semantics of vocabularies or ontologies
- understand the content and evolution of vocabularies or ontologies
- support the partitioning through existing modularization techniques
- employ the MEAN stack with a NodeJS backend as its software development base
- export ontologies
- work cross platform

1.3 Definitions, Acronyms, and Abbreviations.

WAPT - “Web-based analysis and partitioning tool”
KE – Knowledge engineers
DE – domain experts
MEAN – MongoDB, Express, AngularJS and NodeJS
FR – Functional requirement
NFR – Non-functional requirement

1.4 References

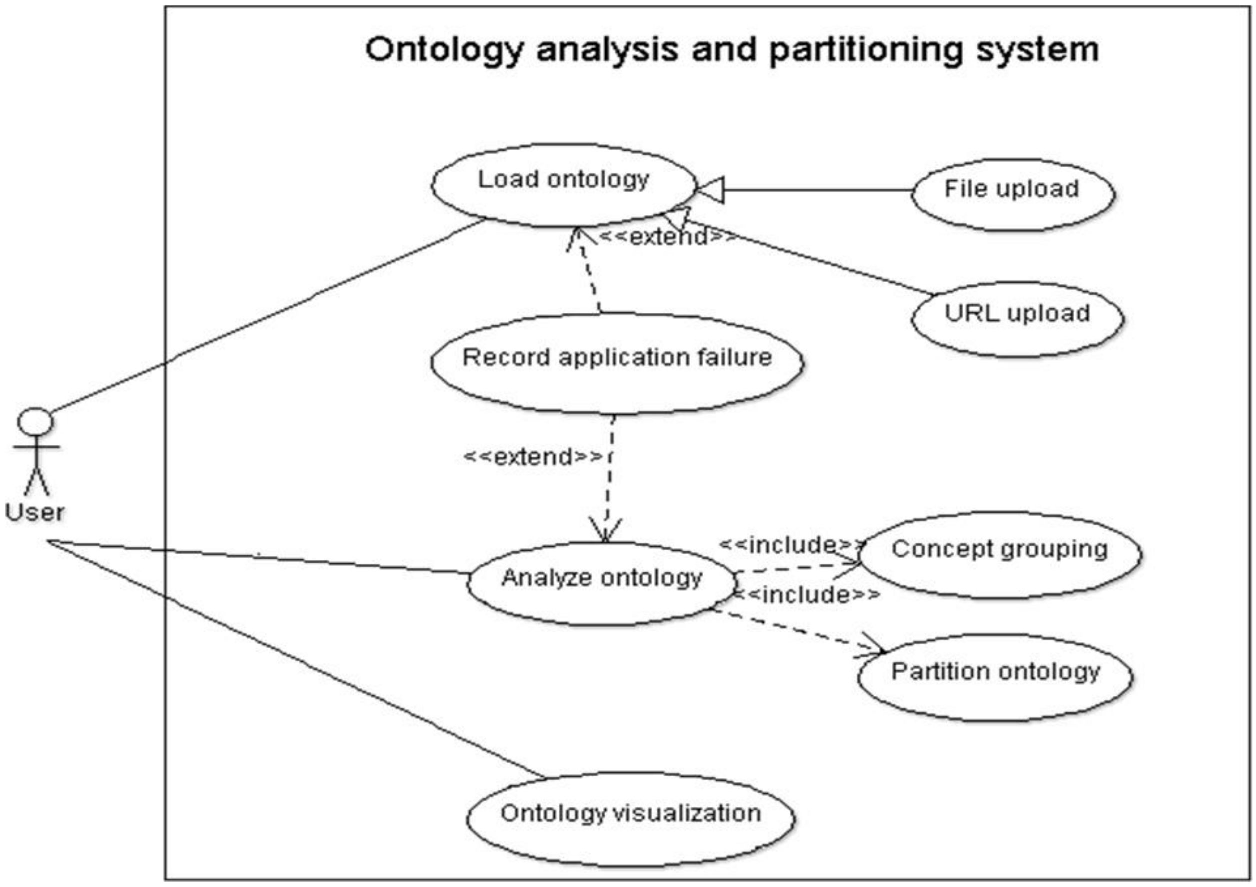
[1] A software engineering approach to ontology building - AntonioDeNicola , MicheleMissikoff, RobertoNavigli
[2] Ontology Content “At A Glance” – Gökhan Coskun, Mario Rothe, Adrian Paschke

2. The Overall Description

2.1 Product Perspective

Ontologies are the backbone of the Semantic Web, a semantic-aware version of the World Wide Web. The availability of large-scale high quality domain ontologies depends on effective and usable methodologies aimed at supporting the crucial process of ontology building. Ontology building exhibits a structural and logical complexity that is comparable to the production of software artefacts. [1] In the context of existing Ontology Modularization Frameworks, there exists a Java based application that can perform this task. However, due to emerging new technologies, a version of a product that delivers a quick, scalabale and fast interface is highly needed. For this purpose, the **WAPT** is being developed.

The WAPT use case diagram is presented in what follows.



2.1.1 Interfaces

WAPT product is based on the MEAN stack and employing technologies which are managed as standalone projects with dedicated organization. The backend is NodeJS and the frontend is AngularJS. Additionally, the Bootstrap package is used in the scope of constructing a fluid and intuitive user interface.

The coding language for the new development is mostly influenced by the coding language of the used modules. Licence compliance review is done at the selected stages of the project lifecycle. The goal is to verify the open source packages.

2.1.2 Software Interfaces

The minimal requirements for running the application are:

- NodeJS v0.12.2 (release date may 2015)
- AngularJS v1.3.15
- Bootstrap v3.3.4

2.3 User Characteristics

The user shall be an ontology engineer with extensive knowledge on Semantic Web, ontologies, meaning and usage. The user’s knowledge may be backed up by theoretical experience or by practical experience.

2.5 Assumptions and Dependencies

It is assumed that the user has Javascript activated in the browser. This is a precondition for the application to run. Otherwise, the application will not run. Only browsers that support Javascript are targeted.

3. Specific Requirements

The analysis of Software Requirements leads to discussion with Customer that are formalized and tracked. The project will be built in the scope of the initial requirements agreement.

3.1 External Interfaces

WAPT allows the input of OWL or RDF/XML file. The output will be an ontology in either OWL or RDF/XML type. In the case of the unsuccessful upload of an ontology, the user will be prompted with an error message and the processing will stop.

3.2 Functions

The following requirements describe the functions of the system. A priority notation has been used. This notation is a marker for requirements importance. A low number, such as 1, describes a priority of high importance. A higher number describes a priority of lower importance. The system functions will be implemented in the order of priority. Requirements with priority 2 may be skipped from implementation, if time constraints do not allow it.

ID	NAME	PRIORITY
FR_1	**Load Ontology from file/ URL** • The user shall be able to upload an Ontology from a file stored in the user's PC. • The file shall be uploaded using browse control. • The file extension can be RDF/XML, Turtle, JSON-LD and text file. • The user shall be able to provide a URL. • The URL should be linked to an Ontology web file.	1
FR_2	**Ontology visualization** • The system shall be able to visualize an ontology in a form of a graph/tree. • The elements of the ontology (concepts, relations, instances) are represented by nodes in the graph.	1
FR_3	**Analyse Ontology** • The system shall be able to show ontology metrics. • The system shall be able to show structural	1

	properties • The user shall be able to analyze the graph on different levels(RDF,RDFS and so on). • The user shall be able to find the depth of hierarchy trees.	
FR_4	**Partition Ontology** • The system shall be able to partition Ontology using existing algorithms. • The system shall be able to group all the concepts, relations and instances in an Ontology. • The user shall be able to select the algorithm for Ontology partitioning. • The user shall have the possibility to apply more than one algorithms on an ontology.	1
FR_5	**Export generated Ontology** • The system shall be able to export an Ontology.	1
FR_6	**Record application failure** • The system should report error messages. • The error messages are related to the syntax of the Ontology	2

3.3 Performance Requirements

ID	NAME	PRIORITY
NFR_1	The response time shall be less than one minute	1
NFR_2	The number of simultaneous users supported will be…	2

3.4 Availability

ID	NAME	PRIORITY
NFR_3	The web application will have 99% uptime.	2

3.5 Constraints

ID	NAME	PRIORITY
NFR_4	WAPT shall be developed by a team of 3 developers	1
NFR_5	WAPT shall have well defined time constraints and gates for completing.	2

3.6. Security

ID	NAME	PRIORITY
NFR_6	WAPT shall check the uploaded ontology file for malicious damage.	2

3.7. User interface

ID	NAME	PRIORITY
NFR_7	• The user shall click the "Upload" button to upload the Ontology. • The user interface shall contain a text-box to enable the user to write/paste the URL. • The user can click the "Load" button to save the file from the provided URL.	2