

A Mobile Web-based Ontology Editor for Smartphone

Mingzhe Zhang

Department of Computer Science, University of Leicester, Leicester LE1 7RH, the United Kingdom
mzhe0329@163.com

Keywords: smartphone; ontology editor; web-based application.

Abstract. Nowadays, smartphones have become indispensable to modern human life. One of benefits of smartphones is that most places on the earth are widely covered by the signal of mobile phones, which means people can connect to the Internet wherever he wants via a mobile phone. It also means a mobile web-based application can be used almost everywhere on the earth. The aim of this research is to develop an ontology editor for smartphones, the editor will be purely mobile web-based, installation free, and cross-platform. This editor will allow the users to create, modify and query ontologies wherever they want with a handheld device and wireless network no matter it is an iPhone or Android device and so on. Users can use the editor to modify the local existed owl files, create new owl files and save them on a local disk. Through the editor, users can generate owl classes, and create sub classes. Users can also use the editor to modify the properties and instances of all the owl classes.

1. Introduction

Since mobile phones have become extraordinarily popular in the whole world. As a major part of mobile phones, smartphones have gotten many users these days. Almost everyone owns one or more than one smartphones. Since the convenience and extensiveness of smartphones, mobile web-based application becomes more popular. The benefits of mobile web-based applications are also very impressive. First, users can access to a mobile website via a browser on any handheld devices (iPhone, iPad, Android, Windows Phone, Blackberry, etc.) immediately. On the contrary, applications need to be downloaded and installed first before it can be used by the users. Second, a mobile web-based application has better compatibility, because of the standard of mobile web browsers, a mobile web-based application can be accessed by any kinds of handheld devices, and whereas people need to create different versions to make an application can be installed on different types of devices. Third, a mobile web-based is more dynamic than a native application, so it can be updated easier [1]. Forth, a mobile web-based application can be shared between users and developers easily. People can easily share the mobile web-based application via a simple link [2].

Usually, there are some shortages of editors on a PC. For instance, most of the editors must be installed to be used on a PC, or at least it need to be downloaded. This shortage decreases the immediacy of the editor, and makes it more complicated to be used. Second, most editors on PC are not cross-platform, this means that different versions must be created to make it accessed to different devices (Windows, Linux and Mac etc.). This shortage does not just increase the difficulty of the usage of the editor for the users, but also costs much money and time of the developers.

A mobile web-based editor can offset all of the shortage we mentioned above. In this paper, a mobile web-based ontology editor will be developed for smartphones. Since the editor is web-based, so it will be installation free, and cross-platform. As a normal ontology editor, this editor will allow users to create, modify and query ontologies. Through the editor, Users can use the editor to modify the local existed owl files, create new owl files and save them on a local disk. Just like using an ontology editor on a PC, the user can use this editor to generate owl classes, set the sub classes, create and modify the properties and instances of the classes. After modified the owl file, users can decide to save it or not, so a wrong modification can be prevented. JQuery mobile is chosen to develop the user interface of this editor, as a framework for generating mobile web applications, jQuery mobile is suitable for all usual handheld devices. It offers different widgets and themes to make the appearance

fits different operating systems. So using jQuery mobile will make the user interface of this editor more user friendly?

2. Background of Ontology Editor

The widely acceptable definition of ontology is “an ontology is a specification of conceptualization”. [3] According to this, ontologies are created to make the information and knowledge can be shared between different agents. When the relevant vocabularies do not exist or the developers do not satisfied with the existing vocabularies, the developers can use an ontology editor to create their own ontologies. The definition of an ontology editor is an application which is programmed to generate and operate ontologies [4]. Since the wide use of Semantic web these days, there are many different kinds of ontology editor available online. Some of them are famous and pluggable, like Protégé and NeOn ToolKit, some of them are easy and small, like SWOOP, and some of them can create an original ontology by using the data and architecture of spreadsheet, like Anzo for Excel. Since the backend of the project is built on the Protégé-OWL API, the study and development of this project is based on the Protégé-OWL ontology editor, which is created at the Stanford Centre for Biomedical Informatics Research. The project implements the basic functionalities of an ontology editor, and makes the user interface and the structure of the editor simpler. For instance, the project ontology editor can generate and modify the owl classes, instances and properties, create and delete property values, build relationships between instances, and so on. Unlike the Protégé-OWL ontology editor, since the project is mobile web-based, it will be necessary to login before getting other functionalities. User information are encrypted before being sent to be inspected. Another characteristic of this mobile web-based ontology editor is the terse user interface. It is not complicated for the user to get used to it. Since the advantages of web applications and smartphones, without a doubt, there are many other benefits of the mobile web-based ontology editor. To sum up, this project makes a basic ontology editor which is also easy to use.

3. Core Functionalities

Some core functionalities of this mobile web-based ontology editor are demonstrated below.

After creating a new owl file or loading the local files, the editor will allow the user to check the last modified date of this file, OWL classes in the project, properties, and instances. And user can also decide which part of the project he wants to query and modify. The user can check the OWL classes, properties and Instances. In the OWL class's part, all the classes are separated by their hierarchical relationships, so the first page will show the direct subclasses of owl: Thing class, which is the superclass of all classes in this file. User can decide to create a subclass of owl: Thing, but since owl: Thing cannot be deleted, so no delete button is shown on the first hierarchy page. The first hierarchy page is shown in Figure 1.

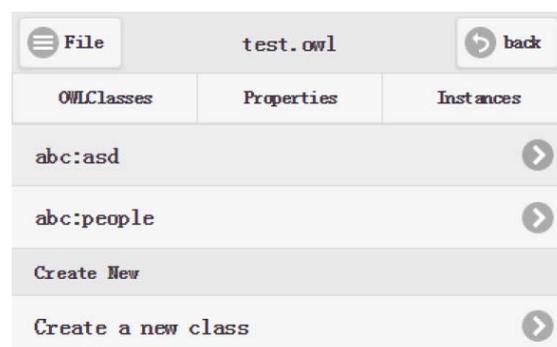


Fig.1 The first hierarchy page of OWLclasses

In this page, both abc: asd and abc:people are the direct subclasses of owl:Thing. User can click “Create a new class” button to simply generate a new subclass under owl: Thing. The user only need to define the name of the new class.

Then, user can choose the class name on the list to check the direct subclass of that class. For example, here the subclasses of abc: people are abc: woman and abc:man. Here the user can decide whether he wants to check the subclass of these classes, create a new class or delete the abc: people class. The second hierarchy page is shown in Figure 2.

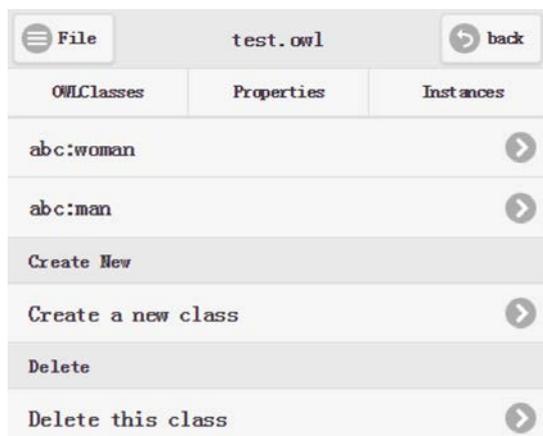


Fig. 2 The second hierarchy page of OWLclasses

Also, the user can also check the properties of this project by simply choose the properties button, in the first hierarchy page of properties, all the top level properties will be shown. Just as the OWL classes page, the user can decide to check each property and create new properties. There are two types of properties of can be created by this editor, the user can decide whether he wants to create an object property or a datatype property. In the second hierarchy page of properties, the user can check the subproperty of the property which he just selected, unlike the first hierarchy page, in the second page, user can choose to delete the property, or he can do other operations. The user can check the subproperties by selecting the properties on the list as well. The second hierarchy page of object property is shown in Figure 3.



Fig.3 The second hierarchy page of object property

In this page, user can also set the maximum cardinality, minimum cardinality, and exact cardinality for the property which he just selected. The user need to type in an integer value when he is setting the cardinalities. User can also set the domain and range of this property. The set buttons are put under the list of the properties. In this page, the user can add the property value and remove the property value as well. In the select domain and select range pages, the user can select the existing

domain or range of this property and delete it. Or the user can set a new domain or range of it. In the existing domain or range form, the user can check the existing domain or range, if there is one, and decide to delete it. In the set domain or range form, all the classes in the project will be shown in the list, the user can select one of them and set it as the new domain or range.

4. Conclusion

Because of the convenience and extensiveness of smartphones nowadays, the wild use of wireless network, and the benefits of mobile web-based applications, which means the user can access to a mobile website via a browser on any handheld devices immediately. This paper points out the shortage of the normal applications especially the editors on a PC, and develops a mobile web-based editor for smartphone. The core functionalities and the ways of using this web-based ontology editor is described in detail, like creating and operating owl classes, properties and individuals, set property values, and so on.

References

- [1]. Thomas R. Gruber. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199-220.
- [2]. García-Alcalde F, García-López F, Dopazo J, et al. Paintomics: a web-based tool for the joint visualization of transcriptomics and metabolomics data.[J]. *Bioinformatics*, 2011, 27(1): 137–139.
- [3]. Maeche A, Staab S. Ontology Learning for the Semantic Web[J]. *IEEE Intelligent Systems*, 2001, 16(2):72-79.
- [4]. Gosling James, Joy Bill, Steele Guy, Bracha Gilad, Buckley Alex, (2014). *The Java® Language Specification*. Oracle America.