Chapter 14

Findings for Ontology in IS and Discussion

Ahlam F. Sawsaa
University of Huddersfield, UK & Benghazi University, Libya

Joan Lu
University of Huddersfield, UK

ABSTRACT

Ontology development is meaningful and useful for both users and IR; therefore, it needs to be evaluated. In this chapter, we are going to test and evaluate the results produced in the research, which is the development of the OIS ontology life cycle. It describes the testing and validation which was applied to the whole model from the initial implementation to ensure consistency of modelled knowledge. The evaluation objective was to collect feedback on OIS ontology by using our evaluation system. The Ontocop system is a platform that has been implemented to get feedback from the IS community. The feedback is assessing and eliciting further details that support the ontology development. The evaluation and discussion will be at two levels based on Gòmez-Pérez’s view.

1. EVALUATION OIS ONTOLOGY

1.1. Ontology Validation

The validation of the OIS ontology is conducted from two points to measure in which way the ontology has been written, and that the ontology syntax does not contain any errors and anomalies. Thus, we make certain of richness and complexity of syntactic issues of the ontology, not just correctness.

On the one hand, testing the modelled knowledge coherence by the FaCT++ reasoner which is an owl-Dl. in OWL semantic languages the OWL statements are constructed on formal logic to provide high expressive and automated reasoning. The reasoning aims to check the consistency of the ontology entities, relationships, and restrictions.

Significantly, the reasoner checks whether or not the statements and class definitions are consistent. Furthermore, FaCT++ was applied during the developing process of the ontology. With respect to DOI: 10.4018/978-1-5225-2058-0.ch014
Findings for Ontology in IS and Discussion

consistency checking of the OIS, the reasoner was used. It achieved this by using the FaCT++ plug-in that combines with Protégé 4.0.2.

This tool infers classification and class hierarchy in the ontology, which helps to correct any errors and inconsistence classes in ontology classification. In fact checking the consistency is necessary to find out if there are any contradictions; to ensure the modelling constructs are being used correctly, and avoid reaching any incorrect inferences.

In Protégé there are two structures of taxonomy; the computed method is called inferred hierarchy and the manual way is called asserted hierarchy. The main evidence of automatic computation of the ontology checking is revealed through appearance of the root of hierarchy (nothing) in red colour in the pane of the inferred hierarchy.

The FaCT++ reasoner shows errors in the classes that had been classified in a red colour. The changing of the OIS ontology model was driven by the discovery of errors during the implementation stage. The process of improving it considered its inadequate performance and improvement of the domain knowledge. The early tests around the reasoner highlighted many errors, some of which arose from adding more information to the model without revising the existing axioms. These errors have been eliminated. However, in practice the first round revealed some errors as shown in Table 1.

The table reveals that these classes were classified under different meta-classes, such as that Analytics is a sub-class of Actors while it should be a subclass of Quantitative class under Methods. Also, the classes ArchitecturLibrary, GovernmentLibrary and SpecialLibrary are classified under the different meta–classes Actors, Domain, and Space whereas they should classified under Libraries Class.

Figure 1 illustrates that some classes have circularity in the OIS after running the reasoner second time. These classes are: DataPrivacy, InformationPrivacy, CopyRight, IntellectualProperty, Computer-Crime, InternetCrime, FreeSpeech, FreedomExpression, IdentificationCode, AccessCode.

Figure 2 illustrates that the asserted and inferred hierarchies after running the FaCT++ reasoner are decreased. It can be seen that there is inconsistency in the class GovernmentLibrary which appears in red colour under Domains class; this means it should be under Mediator as sub-class of Libraries. Otherwise, after that the reasoner was run many times to ensure there is no difference between the inferred and asserted taxonomies and nothing appeared that indicates tasks to be completed and semantically validated. This is also to ensure there are no confounding and contradictory concepts. Also, ensuring terms have consistency of meaning with clarity. Ontology should provide mapping according to the meaning

<table>
<thead>
<tr>
<th>First Round of Running Fact++ Reasoned</th>
<th>Second Round</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Inconsistence Class</td>
</tr>
<tr>
<td>Actors</td>
<td>Analytics, ArchitectureLibrary, Dissemination</td>
</tr>
<tr>
<td>Domain</td>
<td>ElectronicDocumentDelivery, GovernmentLibrary, InformationDiffusion</td>
</tr>
<tr>
<td>Practice</td>
<td>ReallSimpleSyndication</td>
</tr>
<tr>
<td>Resource</td>
<td>SelectiveDisseminationOfInformation</td>
</tr>
<tr>
<td>Space</td>
<td>SpecialLibrary</td>
</tr>
</tbody>
</table>