Extending ontology development methodologies with foundational ontologies

Zubeida Khan
208509140

Supervisor: Dr. C. Maria Keet

November 21, 2011

Abstract

From studies, it is clear that using a foundational ontology for domain ontology development is beneficial in theory and practice. However, when it is to be used, developers don't know which one to choose and why. In order to solve this problem, a comprehensive set of criteria that influence foundational ontology selection has been compiled and a corresponding software tool has been developed to help a domain ontology developer to choose one. This report presents ONSET: a tool used for foundational ontology selection in domain ontology development. Based on an ontology developer's preferences such as ontological commitments, representation languages and other factors, ONSET selects an appropriate foundational ontology to be used for the domain ontology to be developed.

Acknowledgements

I am thankful to my supervisor, Dr. C. Maria Keet for all her support, guidance, contribution and advice. I would also like to thank the creators of DOLCE and BFO for all their feedback on the preliminary criteria lists created.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Ontology development and usage is on the increase in many diverse domains. Foundational ontologies describe general, high-level properties which are common between knowledge bases. The starting point of research was to investigate whether using a foundational ontology offers any benefit and actually improves the quality of an ontology. A controlled experiment in ontology development by Keet [1] was conducted. It was found that those ontologies that were developed using a foundational ontology were more detailed and of better quality. These findings are further reinforced in [2] where the role of foundational ontologies in industry is analysed and the advantages that they exhibit are provided. One of the questions posed in [2] - "What is a foundational ontology good for in industry" is discussed in great detail, further supporting the notion that foundational ontologies are of great importance.

Important functions of foundational ontologies are to enhance semantic interoperability and improve overall quality of systems. Semantic interoperability requires participating machines to achieve the same inference given the same data. This is essential in the transformation of the World Wide Web to the Semantic Web, which enables the automation of many tasks. Foundational ontologies can be used as a starting point in ontology development. One avoids reinventing the wheel. In addition, foundational ontologies aid in the understanding of complex systems.

Although there are a variety of foundational ontologies available, with related documentation, and foundational ontologies have been proven to be beneficial in development, existing methodologies such as METHONTOLOGY, NeOn and UPON do not include them. METHONTOLOGY [3] is used to build ontologies from scratch by identifying a set of activities to be carried out during development. NeOn [4] identifies 9 scenarios to be used for ontology development. Each scenario is broken down into processes and activities. The figure below illustrates these scenarios. It is apparent that foundational ontologies aren't considered.

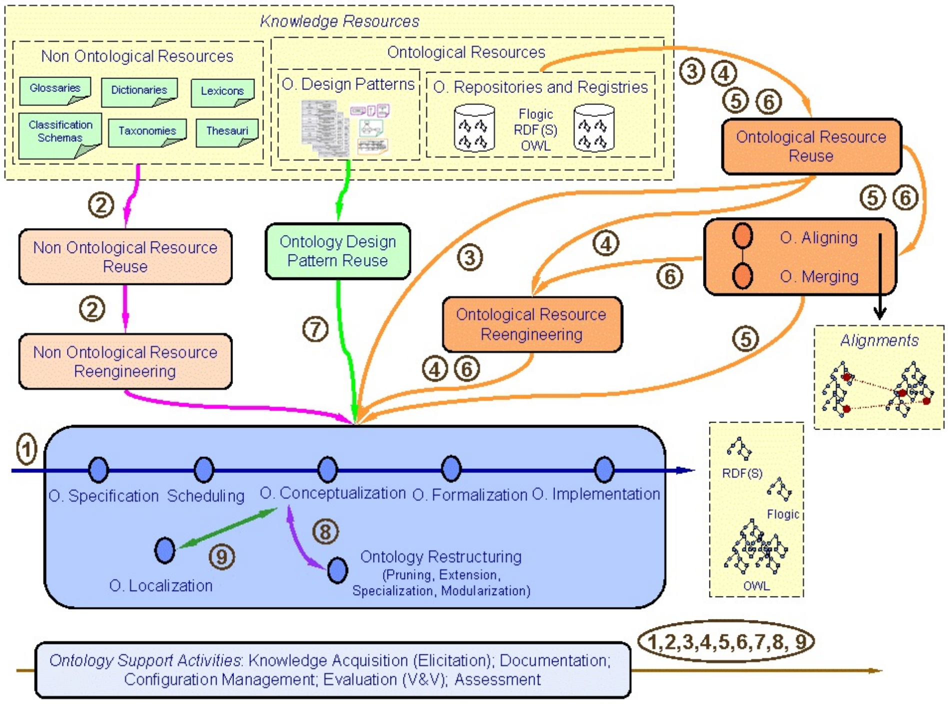

Figure 1.1: NeOn Methodology Scenarios from [4]

1.1 Motivation

Generally ontology developers don't know how to use these foundational ontologies [1]. An ontology development project conducted by my fellow classmates exhibited little usage of foundational ontologies for it was thought to be too difficult to properly implement. Foundational ontologies are often criticized as being too expressive and time consuming to implement [1]. However there are cases in in which one requires multiple classes and properties for high expressivity.

Seeing that foundational ontologies are beneficial and sometimes necessary in domain ontology development and that there aren't any methodologies or tools which consider foundational ontologies at present, there is a need for assistance in this field.

1.2 Problem Description

Foundational ontologies differ in many aspects such as philosophical distinctions, ontology languages, dimensions and extrinsic properties. In order to successfully utilise a foundational ontology, it must be analysed thoroughly to ensure that the philosophical approaches and other properties offered by it correspond to that of the project at hand.

There is no official documentation stating which foundational ontologies can be used for various scenarios. There are certain criteria associated with particular projects such as [6], [25] and [24] but this is insufficient, as ontology developers wish to explore how to use foundational ontologies across all subject domains.

1.3 Objectives

To promote the selection of foundational ontologies and fill the gap in ontology development methodologies, a tool will be developed, aimed at assisting and informing developers about the criteria and properties associated with foundational ontologies and how it relates to the domain ontology to be created.

1.4 Structure of the thesis

The rest of the document is organised as follows: A brief methodology is outlined and described in Chapter 2. This is followed by an extensive review of literature in Chapter 3. The design of ONSET is discussed in Chapter 4. Chapter 5 presents and discusses results obtained. Concluding remarks and possible future project extensions are outlined in Chapter 6. Finally, an appendix containing a user guide for ONSET, preliminary criteria lists for foundational ontologies to be implemented, comparative tables of foundational ontologies and additional references, are provided.

Chapter 2

Methodology

There are a number of tasks to be achieved to solve the problem.

- Conduct an extensive literature review on the usage of foundational ontologies.
- Carry out comparative studies of popular foundational ontologies.
- Select suitable foundational ontologies to be implemented in the tool.
- Study the selected foundational ontologies (DOLCE and BFO, at present) in great detail.
- Create an initial list of criteria, based on user requirements, on why one would use either ontology.
- Contact the creators of DOLCE/BFO to verify and contribute to the initial criteria list.
- Based on these criteria, produce an algorithm in order to assist the user in development.
- Design and implement an application in Java, based on the algorithm.
- Perform a qualitative evaluation of the software by foundational ontology usage scenarios and feedback.
- Improve and modify the tool, if required, based on evaluation results.

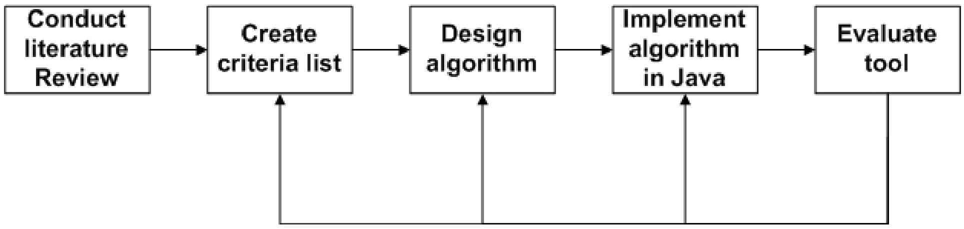

Figure 2.1: Flow of the methodology

Chapter 3

Literature Review

An estimated number of 50 scientific articles were consulted. A number of them are summarised and grouped into sections here. A list of articles that aren't referenced here are included in appendix D.

3.1 Official foundational ontology publications

The WonderWeb deliverable [5] provided much material about widely used foundational ontologies. The proposed foundational ontology library of the WonderWeb library includes 3 foundational ontologies at present: DOLCE, OCHRE and BFO. DOLCE is to be a starting point foundational ontology for comparing relationships with other foundational ontologies of the WonderWeb library. It is based on common-sense principles.

The taxonomy of DOLCE is displayed in the following figure:

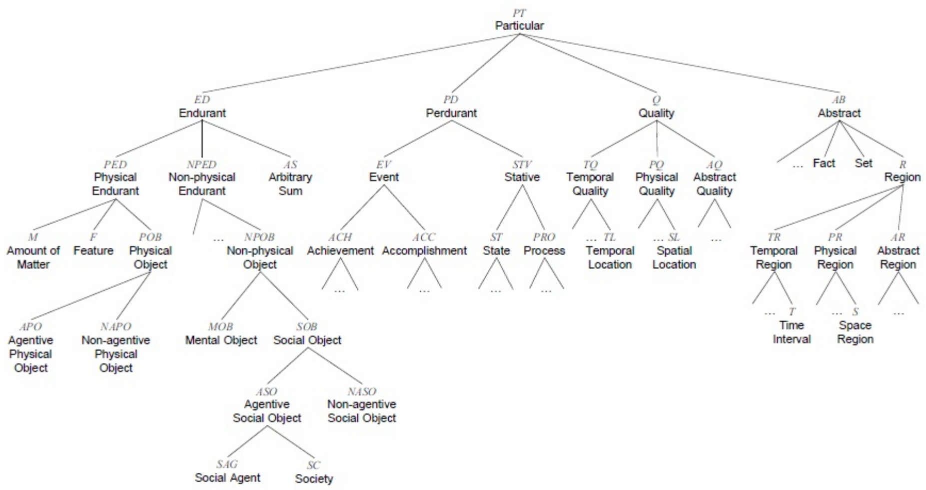

Figure 3.1: DOLCE Taxonomy from [5]

The second ontology of the WonderWeb library is OCHRE. It differs from DOLCE in a sense as it is takes on revisionary view of the world whereas DOLCE takes on a descriptive view of the world. BFO, the third module of the WonderWeb library, is a relatively small taxonomy commonly used for scientific research and data integration purposes.

The taxonomy BFO is displayed in the following figure:

Figure 3.2: BFO Taxonomy from [7]

SUMO [9] is a descriptive ontology of universals and particulars. The philosophical distinctions of SUMO make it desirable in many cases. It is descriptive in nature and offers a distinguishment between abstract and concrete entities. It may be used in a number of applications such as the Semantic Web [12], ontology-based search [10] and ontologies for natural language processes [11]. Nevertheless, it is quite a massive ontology with thousands of terms resulting in it being time consuming to understand and adapt to applications. GFO [8] is an ontology of universals, concepts and symbols, provides a model for space and time, and is used mainly in the health-care/medical field.

3.2 Comparative studies of popular foundational ontologies

A number of existing works such as [24], [13], [14], [15], and [17] critically analyse existing foundational ontologies. Information such as technical aspects, available languages and the building blocks of the compared foundational ontologies are discussed. DOLCE, BFO, GFO and SUMO are all actively used and maintained. DOLCE may be represented in OWL DL, OWL 2 DL and KIF languages. It is made up of 100 categories and 100 axioms. The simplicity of BFO allows it to be represented in many ontology languages which is a desirable feature. The dimensions of BFO differ according to the language it is represented in. GFO may be represented in OWL DL, OWL 2 DL and KIF. The full version of GFO is made up of 79 classes, 97 subclasses and 67 object properties. SUMO, which may be represented in OWL DL and SUO-KIF is made up of 1000 terms, 4000 axioms and 750 rules.

According to [16], modularity is required in an ontology when one needs to hide knowledge which is unnecessary to the task at hand. DOLCE, BFO, GFO and SUMO are all modular ontologies. DOLCE, GFO and SUMO offer lighter and more expressive versions of the ontologies. DOLCE and BFO provide the distinguishment between endurants and perdurants.

The philosophies behind these ontologies are compared. DOLCE, GFO and SUMO are descriptive in nature, meaning ontological categories underlying natural language as well as common sense are captured. BFO, on the other hand is realist in a sense as it captures the world exactly as is.

3.3 The use of foundational ontologies in applications and subject domain ontologies

On the official websites of both DOLCE and BFO, there is a helpful feature for ontology developers: a list of users and projects which make use of the foundational ontology.

Scientific ontologies such as those used in the biomedical [6], [20], [32], environment [21] and life science [18], [19], [31] domains mainly use BFO and GFO. There is an increase in BFO usage due to the fact that the OBO foundry [27] has recommended that ontologies registered on the OBO foundry use BFO. However, some granularity issues are encountered when aligning life sciences ontologies with BFO [33]. DOLCE and SUMO have been applied to a variety of subject domains including engineering [15], [25], biomedical [23], government and military [17], landscape [22] and more. According to [26], foundational ontologies are required for applications such as ontologies for natural language processes, ontology based data access and the Semantic Web.

3.4 Selected foundational ontologies for ONSET

The chosen ontologies for ONSET are BFO and DOLCE. BFO and DOLCE are both popular and up-to-date ontologies. The size and dimensions of these ontologies appear suitable to enable a thorough understanding of them. Ontological criteria of each foundational ontology such as philosophical choice,representation languages etc. differ, thereby satisfying the different requirements of ontology developers. In the near future, ONSET will be extended to include other foundational ontologies such as GFO and SUMO.

Chapter 4

Design

The design of ONSET is outlined and described here. A systematic and rigorous approach is employed, in order to ensure proper functionality of ONSET.

4.1 General Categories

By critically analysing comparative studies, documentation of foundational ontologies and related works, an initial criteria list on why one would use a particular foundational ontology but not others, is to be created. The criteria list is to be based on the following categories:

- Ontological Commitments: Philosophical choices taken by foundational ontologies; e.g. ontology of particulars vs. ontology of universals. Particulars are entities that cannot be instantiated while universals are entities that can be instantiated.

- Representation Language: Languages used to represent a domain ontology e.g. KIF, OBO, OWL DL and more. Knowledge Interchange Format (KIF) is a language designed for use in the interchange of knowledge among different computer systems. OBO began from the Gene Ontology. The Web Ontology Language (OWL) [28] is a W3C recommendation. OWL is a markup language used for publishing and sharing ontologies on the World Wide Web.

- Software engineering properties: General properties associated with foundational ontologies e.g. licensing, modularity etc.

- Subject Domain: Existing domains expressed using foundational ontologies e.g. the biomedical domain.

- Applications: Application scenarios of domain ontologies e.g. the Sematic Web, ontology driven information systems etc. The Semantic

Web is an extension of the World Wide Web and able to describe things in a way that machines may understand them. Ontology driven information systems involves integrating heterogenous information systems.

4.2 Final Criteria Lists

A preliminary criteria list, based on the above general categories for each foundational ontology was created. The creators of DOLCE and BFO were contacted and have edited and contributed to the respective initial criteria list that was created. The final lists for each foundational ontology, on which ONSET is based on is displayed here. For each category, the criteria terms associated with it is provided alongside a brief explanation of the term if necessary.

BFO

Ontological commitments

- Ontology of universals- the entities of the ontology can be instantiated.
- Realist(prescriptive)- BFO captures the world exactly as is.
- Reductionism- each space-time location contains at most one object.
- Endurantism and Perdurantism- BFO contains entities that are wholly-present and those that occur in time.
- Actualism- everything that exists is actual.
- Eternalist stance- the past, present and future all exist.
- Concrete entities only- objects that exist in space and time are considered.
- Mereology theory undertaken is not specified- mereology is the study of system decomposition and parts, wholes and boundaries.
- Temporal aspects are not provided- temporal aspects are ways in which the foundational ontology deals with time eg. by adding quantification over time.
- Snap ontologies are sensitive to the level of granularity at which their components are revealed- granularity deals with the many levels of detail that data can be broken up into.

Representation Language

- OWL 2 QL
- OWL 2 EL
- OWL 2 DL
- OWL DL
- KIF
- OBO
- FOL Computer processable

Software engineering properties

- Dimensions- differs according to representation language and other factors
  - BFO in OWL - 39 universals.
  - BFO in OBO - 23 terms and 12 typedefs.
  - BFO + RO in OWL - 33 universals and 34 object properties.
  - BFO in Isabelle (First- Order based)-18 theories.
- Licensing- freely available.
- Part of the proposed WonderWeb library of formal ontologies.
- Computer-processable representation.
- Modularity- modularisation of BFO where snap is separated from the span entities and object properties.
- Being actively maintained.
- The OBO foundry has recommended that ontologies registered on the OBO Foundry should use BFO.

Subject Domain

- Biomedical
- Environment
- Life sciences
- Geographical

Applications

- Scientific research
- To formally represent scientific theory
- Ontology driven information systems
- Database integration
- Ontologies for natural language processing

DOLCE

Ontological Commitments

- Ontology of particulars- the entities of the ontology cannot be instantiated.
- Descriptive- captures the ontological categories underlying natural language and human common sense.
- Multiplicative - allowing different entities to be co-localized in the same space-time.
- Endurantism and Perdurantism- DOLCE contains entities that are wholly-present and those that occur in time.
- Possibilist - entities are allowed independently of their actual existence.
- Eternalist stance- the past, present and future all exist.
- Distinguishes between abstract and concrete entities.
- Includes quality and qualia- a quality is a basic entitiy we can perceive and a quale is it's corresponding value. eg. a quality may be a color of a leaf and it's quale may be green.
- Mereology theory undertaken is General Extensional Mereology (GEM).
- Temporal aspects are provided by the categories Temporal Quality and Temporal Region.
- Exhibits a high level of granularity.

Representation Language

- OWL 2 DL
- OWL DL
- KIF

Software engineering properties

- Dimensions- 100 categories and 100 axioms + relations, quality properties and qualia to represent attributes.
- Licensing- freely available.
- Part of the WonderWeb library of formal ontologies.

- Modularity- the availability of lighter and more extensive versions and endurants and perdurants are seperate.
- Being actively maintained.

Subject Domain

- Legal
- Agriculture
- Biomedicine
- Manufacturing
- Engineering design performance in microelectronics
- Requirements engineering
- Landscape
- Biomedicine
- Government
- Military
- Simulations
- Computer Programs
- Church administration
- Life sciences

Applications

- The Semantic Web
- Information Retrieval
- Ontology driven information systems
- For scientific research
- To formally represent scientific theory
- Ontologies for natural language processing

The preliminary criteria lists that were created for GFO and SUMO are included in appendix B.

4.3 Functional Requirements

The tool is required to perform the following:

1. Additional questions- These are the questions which correspond to both the implemented foundational ontologies and won't affect the results of ONSET at present. The user must be given a choice which is, whether to include or exclude these questions from the program run.

2. Scaling- Assigning a rating to each category, according to the user.

3. Store answers- Store a user's answers corresponding to each question.

4. Selection- Select a foundational ontology to be used.

5. Explain selection- Provide a neat summary of why the particular foundational ontology was selected.

6. Conflicting answers- If the user has requirements relating to more than one foundational ontology, the conflicting results: what is provided by the selected foundational ontology compared to what the user wants, is compared and displayed.

7. Provide ontology references- Provide a list of existing ontology references of the domain chosen by the user, if available.

4.4 Non-functional Requirements

A number of non-functional requirements to be implemented are essential to the overall quality of the tool. These include:

1. Maintainability- The tool must be designed and implemented in such a way that maintaining and modifying it is a quick and simple process.

2. Usability- Users must feel comfortable and at ease using the tool. The tool will be divided into windows and furthermore into tabs and panels. This display is easy to understand and follow.

3. Response time- The time taken in submitting answers and calculating results must be minimal.

4. Portability- The tool must be able to run on different operating systems and platforms.

4.5 Algorithm

Algorithms based on the final criteria list were created. These algorithms are displayed here.

The first algorithm decides whether additional questions are to be implemented in the tool (line 1), and assigns scaling per category (line 6), both according to the user's input.

Algorithm 1: ONSET- ONtology SElection Tool Algorithm 1

```
   DolceCount = 0;
   BFOCount = 0;
   DolceAnswers[] = null;
   BFOAnswers[] = null;
   ScalingValues[] = null;
   output: Include additional questions?
1  if input is yes then
2  |   Show additional questions
3  else
4  |   Hide additional questions
5  end
   output: Assign scaling per category?
6  if input is yes then
7  |   for i ← 0 to numOfCategories do
8  |   |   Read scaling value;
9  |   |   Store scaling value in ScalingValues[i];
10 |   end
11 else
12 |   ScalingValues[i] = 1;
13 end
   output: DOLCECount, BFOCount, DOLCEAnswers, BFOAnswers,
           ScalingValues
```

The second algorithm applies the selected scaling values to each category (lines 9,18,27 and 29). It displays questions per category (line 6), and accepts and stores the answers of the user (lines 10,19,28 and 30). Based on this, the selected foundational ontology is calculated and displayed, alongside reasons as to why it was chosen (lines 40-44 and 54-58). If present, conflicting results are displayed (lines 47-52 and 61-66). In addition, it provides a list of existing ontology references of the domain chosen by the user, if available (lines 45,46,59 and 60).

Algorithm 2: ONSET- ONtology SElection Tool Algorithm 2

```
input : DOLCECount. BFOCount, DOLCEAnswers, BFOAnswers,
        ScalingValues
DOLCEdomain[] = null;
BFOdomain[] = null;
k = 0;
for i ← 0 to numOfCategories do
    for j ← 0 to numOfQuestionsPerCategory do
        Display question;
        Display options;
        if option corresponds to DOLCE then
            DOLCECount = DOLCECount+(1* ScalingValues[i]);
            DOLCEANSWERS[j]= option text;
            if numOfCategories == 3 then
                for k ← 0 to numberofreferences do
                    DOLCEdomain[k] = Subject domain reference;
                end
            end

        else if option corresponds to BFO then
            BFOCount = BFOCount+(1* ScalingValues[i]);
            BFOANSWERS[j]= option text;
            if numOfCategories == 3 then
                for k ← 0 to numberofreferences do
                    BFOdomain[k] = Subject domain reference;
                end
            end

        else
            DOLCECount = DOLCECount+(1* ScalingValues[i]);
            DOLCEANSWERS[j]= option text;
            BFOCount = BFOCount+(1* ScalingValues[i]);
            BFOANSWERS[j]= option text;
            if numOfCategories == 3 then
                for k ← 0 to numberofreferences do
                    DOLCEdomain[k] = Subject domain reference;
                    BFOdomain[k] = Subject domain reference;
                end
            end
        end
    end
end
```

```
input  : Calculate result
if DOLCECount > BFOCount then
    output: Selected Foundational Ontology is DOLCE
    output: Reasons why DOLCE was chosen:
    for i ← 0 to DOLCEAnswers.length do
        if DOLCEAnswers[i]! = null then
            output: DOLCEAnswers[i]
        end
    end
    output: Existing ontologies with the specified subject domain
    for k ← 0 to DOLCEDomain.length do
        output: DOLCEDomain[k]
    end
    if BFOAnswers not empty then
        output: Conflicting results: The tool has detected that some
                of your criteria matches with BFO Ontology
        for i ← 0 to BFOAnswers.length do
            if BFOAnswers[i]! = null then
                output: BFOAnswers[i]
            end
        end
    end

else if BFOCount > DOLCECount then
    output: Selected Foundational Ontology is BFO
    output: Reasons why BFO was chosen:
    for i ← 0 to BFOAnswers.length do
        if BFOAnswers[i]! = null then
            output: BFOAnswers[i]
        end
    end
    output: Existing ontologies with the specified subject domain
    for k ← 0 to BFODomain.length do
        output: BFODomain[k]
    end
    if DOLCEAnswers not empty then
        output: Conflicting results: The tool has detected that some
                of your criteria matches with DOLCE Ontology
        for i ← 0 to DOLCEAnswers.length do
            if DOLCEAnswers[i]! = null then
                output: DOLCEAnswers[i]
            end
        end
    end

else
    output: :
    The tool was not able to select a foundational ontology due to
    many a contradictory responses. Restart the process if you wish.
end
```

4.6 Implementation Specification

ONSET was developed in Java using Netbeans IDE [29].

4.6.1 Hardware and software requirements

A machine with Intel and compatible processors is sufficient for ONSET to work. A minimum of 1MB of free disk space and 1 gig of physical RAM is required. The only software requirement is that the host machine must have JRE components from [30] installed.

4.6.2 General flow of ONSET

The flow of ONSET is summarized in the following activity diagram:

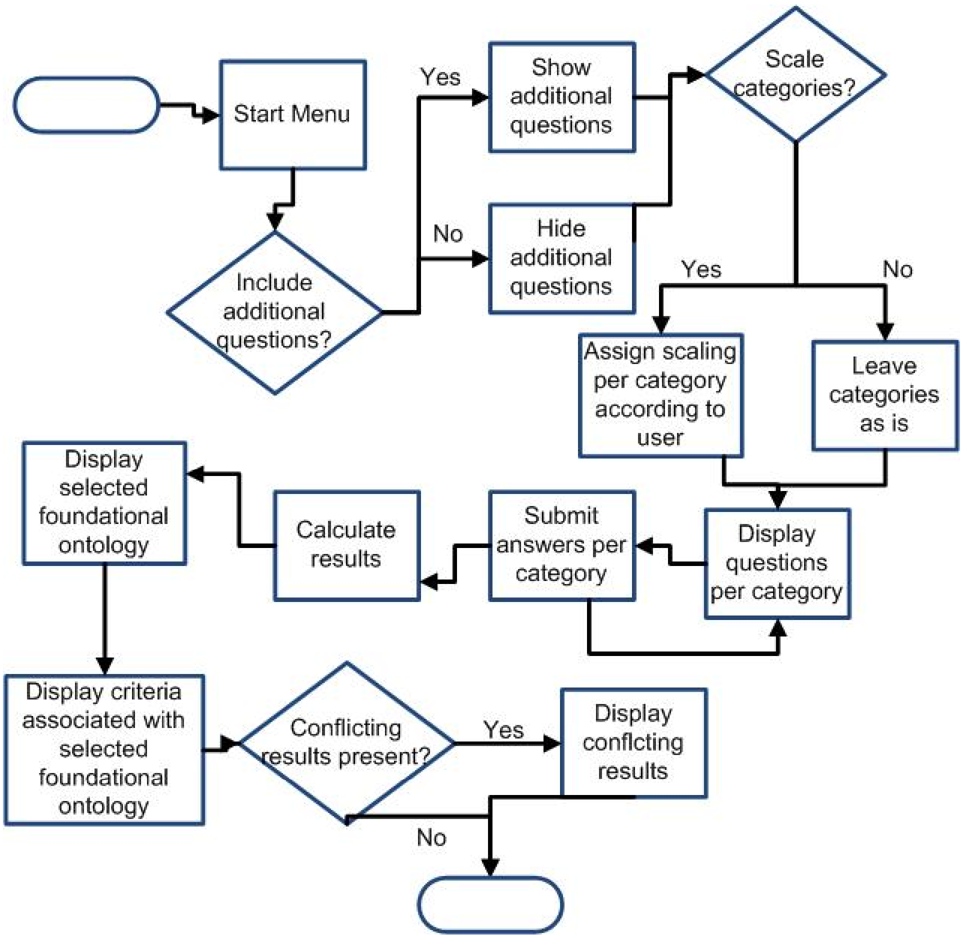

Figure 4.1: Activity Diagram of ONSET

4.6.3 Meeting functional and non-functional requirements

This section describes how all the functional and non-functional requirements described earlier are met by ONSET.

Functional requirements

1. Additional questions- A simple if-else statment in the programming language of choice facilitates this.
2. Scaling- For each category, a scaling value between 0-5 where 0 represents omit, 1 represents least important and 5 represents most important is captured. For each question answered, the scaling value of the category it belongs to is applied to the answer.
3. Store answers- This is met by capturing a user's answers and inserting it into an array.
4. Selection- ONSET meets this by performing calculations which correspond to the user's responses to the questions and numerical scaling assigned by the user per category.
5. Explain selection- Answers are stored in an array corresponding only to the foundational ontology that the answer corresponds to. After selection, the array of the selected foundational ontology is neatly displayed to the user.
6. Conflicting answers- If the array of the foundational ontology that wasn't selected is not empty, conflicting answers are found. These are displayed alongside what is offered by the selected foundational ontology, offering a comparison to the user.
7. Provide ontology references- When a user chooses a subject domain, all it's references are added to an array of subject domains and displayed.

Non-functional requirements

1. Maintainability- This is met by including useful comments throughout the code and generating java docs.
2. Usability- The tool tip text support implemented in ONSET for explaining complicated terms promotes this. A help menu with support and creator details is available also.
3. Response time- The simplistic yet uniform and neat design of the tool promotes a minimal response time.
4. Portability- The .jar file produced by Netbeans IDE allows ONSET to be used throughout different operating systems and platforms.

Chapter 5

Results and Discussion

The evaluation of ONSET is performed by testing it against existing ontologies and simulated ontologies. User-feedback is required to critically assess and improve ONSET.

5.1 Evaluation

The results of applying ONSET to existing and simulated ontologies are illustrated here.

5.1.1 Scenario 1

Firstly, the tool was tested according to the requirements of [24] which is an application of the semantic web. Ontological choices of the test case include: descriptiveness, a multiplicative approach, possibilism, perdurantism, modularity (the existence of lightweight versions) and an executable language.

ONSET has chosen DOLCE as a foundational ontology. The output of ONSET for this case study is displayed in Fig 5.1

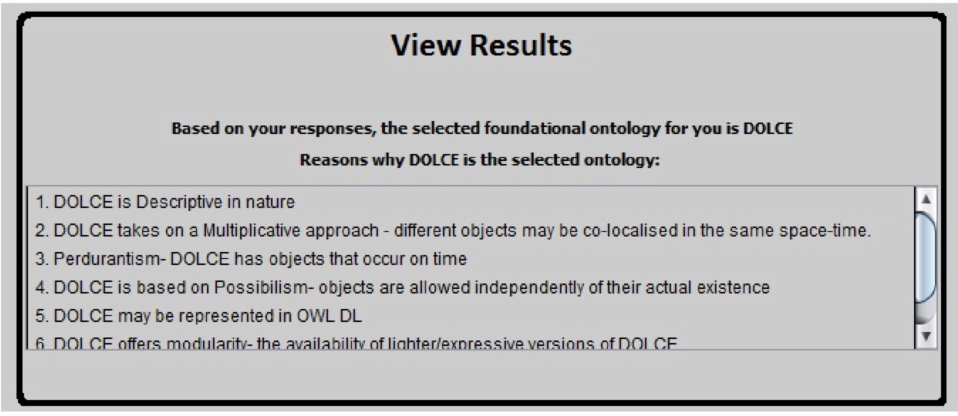

Figure 5.1: Output of ONSET: Scenario 1

5.1.2 Scenario 2

Secondly, ONSET was tested according to the requirements of [31], which describes a microbial loop model ontology. For this ontology, the requirements were: a realist approach, different levels of granularity, endurantism and perdurantism, temporal aspects and the usage of quality and qualia.

ONSET chooses DOLCE as the selected foundational ontology. However, there is a conflicting answer in this scenario: DOLCE is descriptive in nature whereas the user wants a realist ontology. The conflicting results present for this scenario is illustrated in the following screenshot.

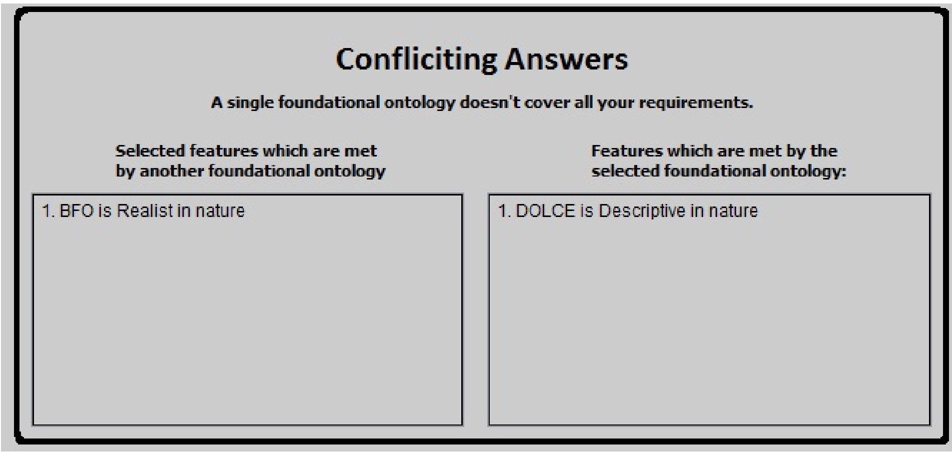

Figure 5.2: Output of ONSET: Scenario 2

5.1.3 Scenario 3

An ontology of mental disease is presented in [32]. It is required to be a realist ontology with a clear distinguishment between continuants and occurents (endurants and perdurants). Furthermore, it is to be registered with the OBO foundry.

ONSET chooses BFO as a selected ontology for this study.

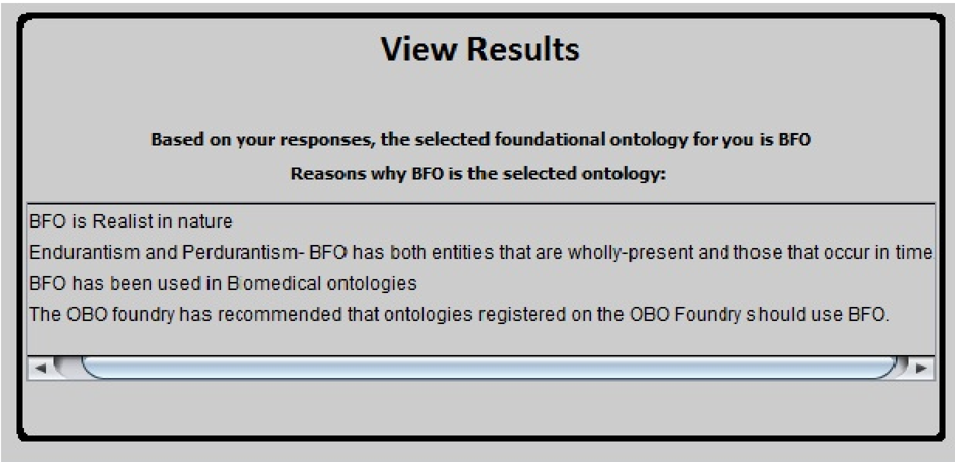

Figure 5.3: Output of ONSET: Scenario 3

For each case study, ONSET accurately calculates the selected foundational ontology based on criteria provided. The results produced by ONSET for these 3 scenarios correspond to the foundational ontologies used in each case study.

5.1.4 Scenario 4

By simulating ontological commitments, and other criteria, it is possible to test whether the scaling feature of ONSET works. It may be assumed that there is an ontology to be created with the following requirements: an ontology of universals, realist in nature, to be represented in OWL DL, modularity (lighter/expressive versions and endurants and perdurants separate), applying it to formally represent scientific theory and a domain of life sciences.

Without scaling, ONSET chooses BFO as the selected foundational ontology as can be seen below in Fig 5.4.

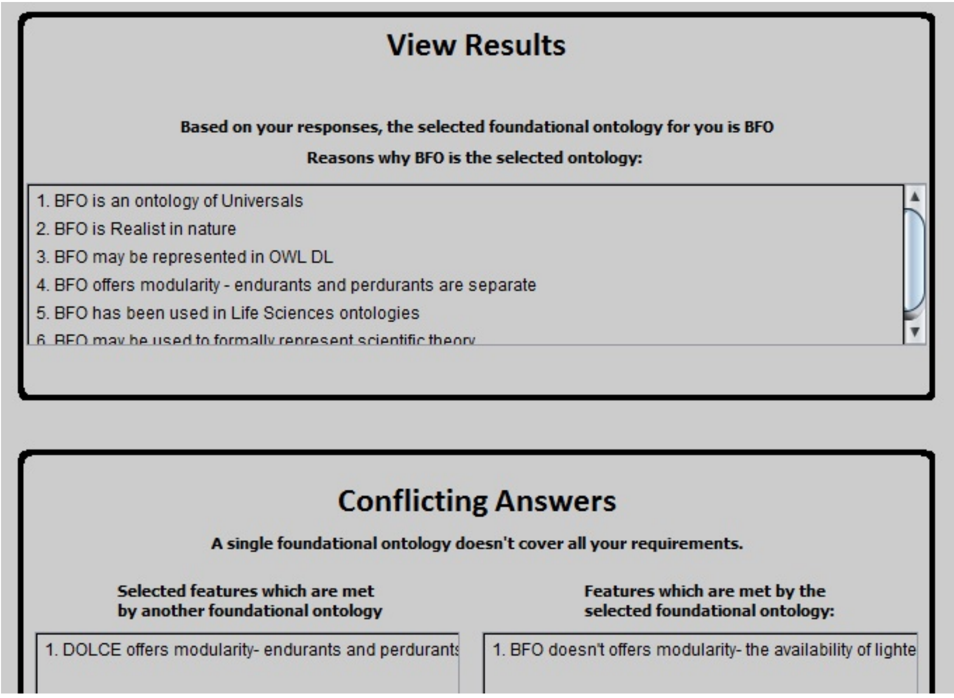

Figure 5.4: Output of ONSET: Scenario 4 (without scaling)

Scaling is then applied to ONSET. The categories are scaled in the following manner: ontological commitments is assigned a value of 1, representation languages is assigned a value of 5, software engineering properties is assigned a value of 3, subject domain is assigned a value of 5 and applications is assigned a value of 4.

Now ONSET chooses DOLCE as the selected foundational ontology.

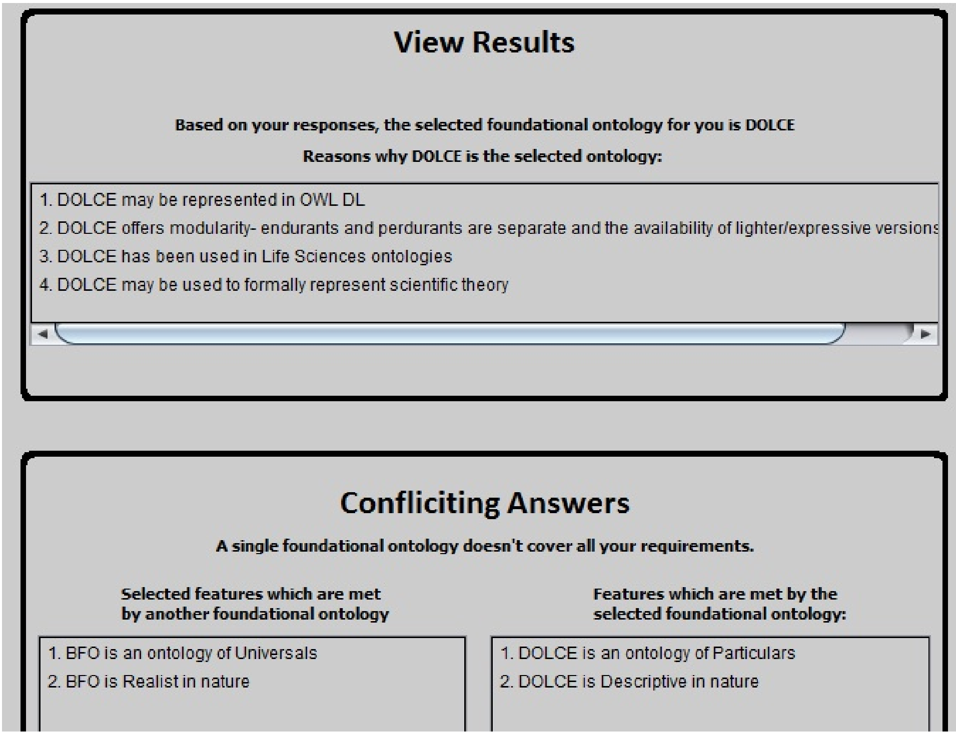

Figure 5.5: Output of ONSET: Scenario 4 (with scaling)

The results of ONSET changed for the same scenario when scaling was implemented, demonstrating that the scaling feature of ONSET works properly.

5.2 User feedback

ONSET was initially presented at Masters Artificial Intelligence Spring School 2011 (MAIS '11). Positive feedback and keen interest was exhibited by the participants of the seminar and suggestions were made to further improve the user-friendliness of the tool. A suggestion, to implement a tooltip to explain complicated ontological terms used in ONSET was taken into consideration and implemented. A lecturer at University of South Africa (UNISA), Ken Halland has shown interest in the project and will be using ONSET in his ontology engineering honours course next year. Stefano Borgo, one of the creators of DOLCE is pleased with ONSET.

Chapter 6

Conclusions and future work

6.1 Conclusions

The problem at hand, ontology developers' severe difficulties in selecting which foundational ontology to use for domain ontology development and why, has been successfully solved with ONSET. ONSET does this by calculating a selected foundational ontology to be used based on ontology requirements. The criteria lists that were compiled during the study and thereafter verified by the creators of the foundational ontologies, are an important contribution to ontology development using foundational ontologies. The developed tool, ONSET assists and informs developers about the criteria and properties associated with foundational ontologies and how they relate to the domain ontology to be created.

This is the first paper-based and software-assisted approach in foundational ontology selection.

6.2 Future Work

Future works of the project include allowing users to map their existing ontologies to a selected foundational ontology. Integrating other foundational ontologies such as GFO and SUMO in ONSET is also possible. The preliminary criteria lists of these foundational ontologies are presented in the appendix B. The design of ONSET, with the additional questions feature caters for this. In these cases, perhaps the additional questions will be a necessity. When more foundational ontologies are implemented, these questions will affect the output of ONSET. Integrating ONSET in existing ontology methodologies is another future direction of the project. Performing a broader user evaluation of ONSET, by allowing prospective ontology developers to use it may further improve ONSET.

Bibliography

[1] Keet C.M., The use of foundational ontologies in ontology development: an empirical assessment. In: Proceedings of the 8th Extended Semantic Web Conference (ESWC'11). G. Antoniou et al (Eds.), Springer-Verlag, Lecture Notes in Computer Science LNCS vol. 6643, 321-335

[2] Borgo S. and Lesmo L., The Attractiveness of Foundational Ontologies in Industry, in Formal ontologies meet industry, S. Borgo and L. Lesmo, Editors, IOS: Amsterdam, 2008.

[3] Fernandez, M., Gomez-Perez, A. and Juristo, N., METHONTOLOGY: From Ontological Art Towards Ontological Engineering, AAAI-97 Spring Symposium on Ontological Engineering, Stanford University, March 24-26th.

[4] Surez-Figueroa, M.C. NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse. PhD Thesis. Universidad Politcnica de Madrid. (2010)

[5] Masolo, C., Borgo, S., Gangemi, A., Guarino, N. and Oltramari, A.,WonderWeb Deliverable D18: The WonderWeb Library of Foundational Ontologies, http://wonderweb.semanticweb.org (2003).

[6] Grenon P, Smith B, Goldberg L. Biodynamic ontology: Applying BFO in the Biomedical Domain, in Pisanelli DM (ed). Ontologies in Medicine. Proceedings of the Workshop on Medical Ontologies, Rome October 2003. IOS Press, Studies in Health Technology and Informatics, vol 102, 2004. p. 20-38.

[7] Cocos, C.,Brochhausen, M., Bonsma E. and Martin, L.,Design Principles of the ACGT Master Ontology: Examples and Discussion

[8] Heinrich Herre, Barbara Heller, Patryk Burek, Robert Hoehndorf, Frank Loebe, and Hannes Michalek. General Formal Ontology (GFO) A foundational ontology integrating objects and processes, Part I: Basic Principles. Onto-Med Report 8, Research Group Ontologies in Medicine, Institute of Medical Informatics, Statistics and Epidemiology, University of Leipzig, Leipzig, 2006.

[9] Niles, I., and Pease, A. 2001. Towards a Standard Upper Ontology. In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Chris Welty and Barry Smith, eds, Ogunquit, Maine, October 17-19, 2001.

[10] Eid, M., Liscano, R. and El Saddik, A., 2007. A Universal Ontology for Sensor Networks Data. 2007 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, (June), p.59-62.

[11] Niles, I. and Pease, A., 2003. Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In H. R. Arabnia, ed. Time. Citeseer, pp. 412-416.

[12] Farrar, S. and Langendoen, T., 2003. A linguistic ontology for the semantic web. Glot International, 7(3), p.97-100.

[13] Mascardi, V., Cord, V., Rosso, P.: A Comparison of Upper Ontologies;Technical Report DISI-TR-06-2, Genova, Italy (2007)

[14] Grenon, P. BFO in a Nutshell: A Bi-categorial Axiomatization of BFO and Comparison with DOLCE. IFOMIS Technical Report Series, 6/03 (2003).

[15] Jureta, I.J., Mylopoulos, J., and Faulkner, S., A Core Ontology for Requirements. Applied Ontology, 4(3-4):169244, 2009.

[16] Parent C., Spaccapietra S. (2009) An Overview of Modularity, Modular Ontologies. pp. 5-23.

[17] Semy, S.K., Pulvermacher, M.K., and Obrst, L.J., Toward the use of an upper ontology for U.S. government and U.S. military domains: An evaluation. Technical Report MTR 04B0000063, The MITRE Corporation, 2004.

[18] E. Beisswanger, S. Schulz, H. Stenzhorn, U. Hahn, BioTop: An Upper Domain Ontology for the Life Sciences. A Description of its Current Structure, Contents, and Interfaces to OBO Ontologies. Forthcoming in: Applied Ontologies.

[19] Keet, C.M. Bottom-up ontology development reusing semi-structured life sciences diagrams. AFRICON'11 – Special Session on Robotics and Artificial Intelligence in Africa, Livingstone, Zambia 13-15 September, 2011. IEEE (in print).

[20] Ceusters W, Smith B: Malaria Diagnosis and the Plasmodium Life Cycle: the BFO Perspective. In Book Malaria Diagnosis and the Plasmodium Life Cycle: the BFO Perspective (Editor ed.). City; 2009.

[21] Environment ontology available at, http://environmentontology.org/

[22] Sinha, G. and Mark, D., 2007. Toward A Foundational Ontology of the Landscape.

[23] Ahmad F. and Lindgren H., Selection of foundational ontology for collaborative knowledge modeling in healthcare domain, In: AIMSA'10 Proceedings of the 14th international conference on Arti

cial intelligence: methodology, systems, and applications, Springer-Verlag Berlin, Heidelberg 2010

[24] Oberle, D. Semantic Management of Middleware. In: The Semantic Web and Beyond. vol. I, Springer,New York (2006)

[25] Ermolayev, V., Keberle, N., Matzke, W.-E.: An Upper-Level Ontological Model for Engineering Design Performance Domain. In: Li, Q., Spaccapietra, S., Yu, E. (eds.) ER 2008. LNCS, vol. 5231, pp. 98–113. Springer, Heidelberg (2008)

[26] Keet, C.M.: Dependencies between ontology design parameters. International Journal of Metadata, Semantics and Ontologies 5(4) (2010) 265-284

[27] Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L., Eilbeck, K., Lewis, S.: The

obo foundry: Coordinated evolution of ontologies to support biomedical data integration. Nature Biotechnology 25(11) (2007) 12511255

[28] Mike Dean, Dan Connolly, Frank van Harmelen, James Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. OWL web ontology language reference. W3C Working Draft, 31 March 2003. Available at http://www.w3.org/TR/2003/WD-owl-ref-20030331.

[29] Netbeans IDE available at, http://netbeans.org/

[30] Java JRE available at ,http://www.oracle.com/technetwork/java/javase/downloads/jre-6u25-download-346243.html

[31] Keet, C.M. Factors affecting ontology development in ecology In: Ludascher, B. and Raschid, L. (Eds.): Data Integration in the Life Sciences 2005 (DILS2005), Vol. 3615 of LNBI, 20-22 July, Springer-Verlag, San Diego, USA, pp.46-62

[32] Ceusters, W., Smith, B.: Foundations for a realist ontology of mental disease. Journal of Biomedical Semantics 1(1), 10 (2010)

[33] Stefan Schulz, Martin Boeker, and Holger Stenzhorn. How granularity issues concern biomedical ontology integration. In MIE08 Medical Informatics Europe 2008. Proceedings of the 21th International Congress of the European Federation for Medical Informatic (to appear), 2008. Accepted for publication.

Appendix A

User Manual

There is no installation required for ONSET. Simply opening the file "ONSET.jar" begins the program run. This brings the user to the start screen.

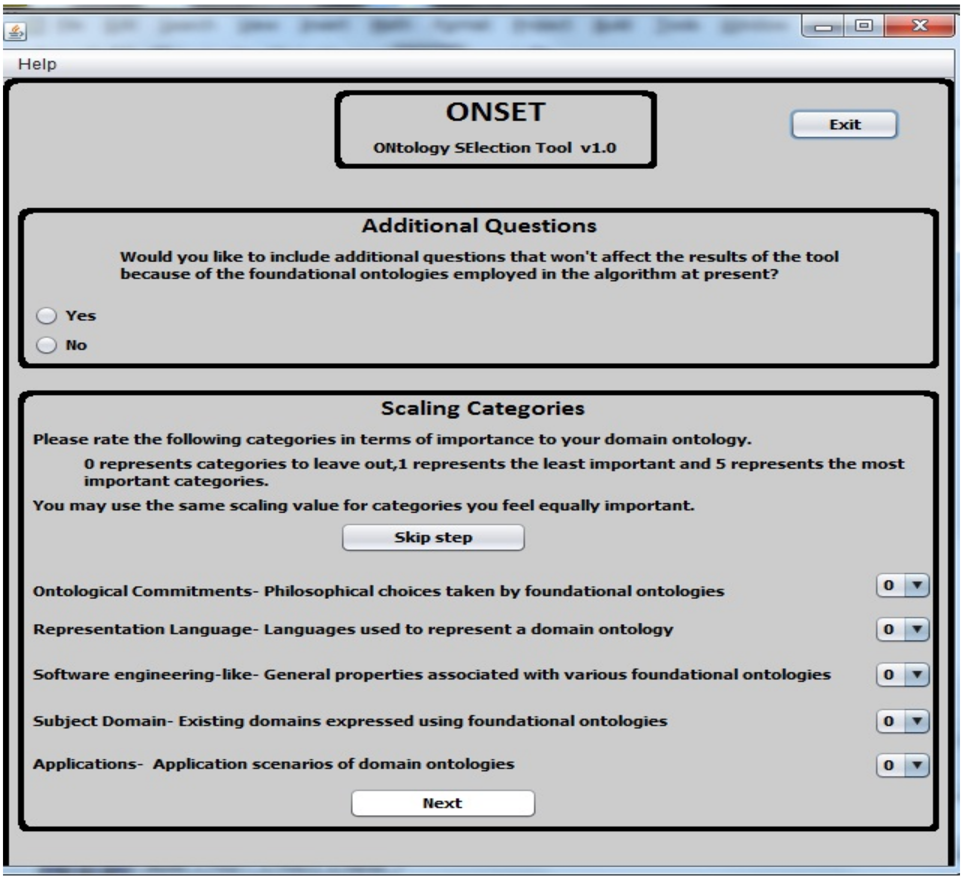

Figure A.1: Start screen of ONSET

After deciding on additional questions and scaling, the user is brought to a tabbed pane with the questions per category in a tab.

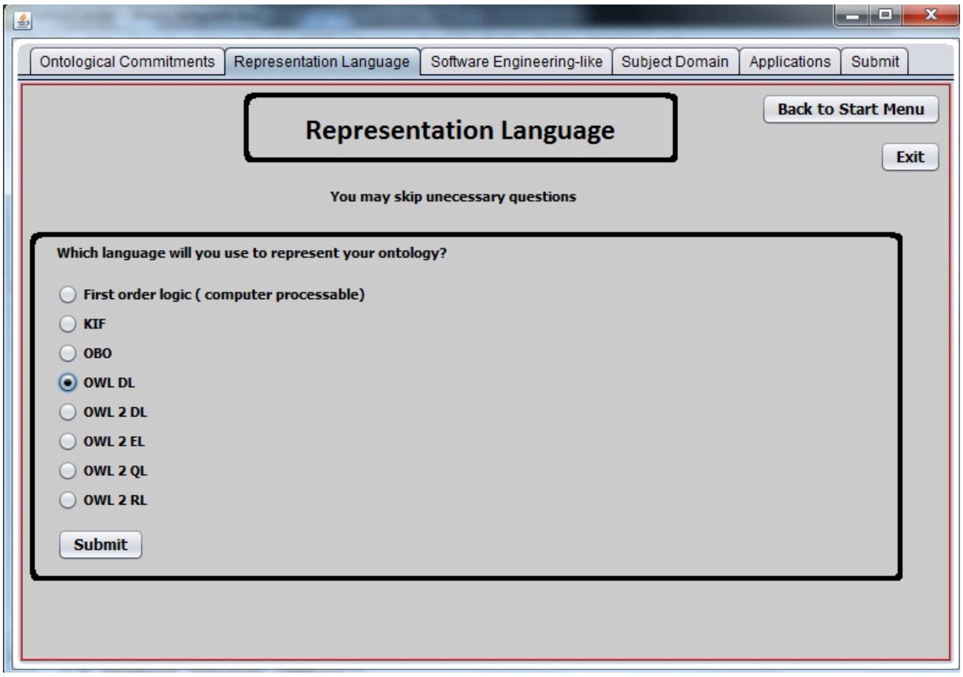

Figure A.2: Tabbed pane of categories in ONSET

The user goes to the submit tab and calculates the results by clicking the “Calculate result” button.

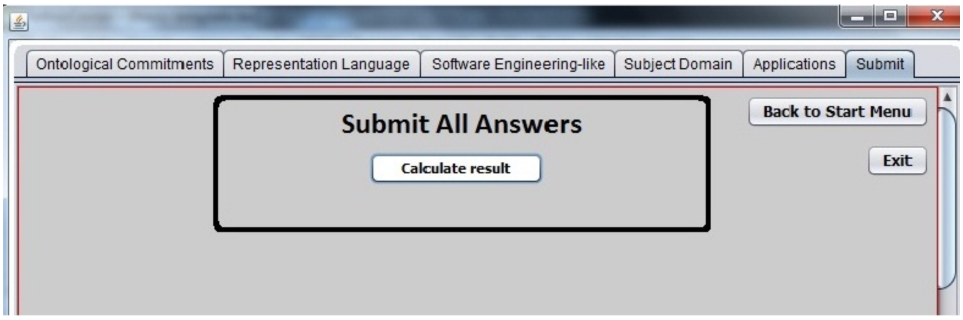

Figure A.3: Submit tab in ONSET

Once the button is pressed, all results are displayed and grouped neatly.

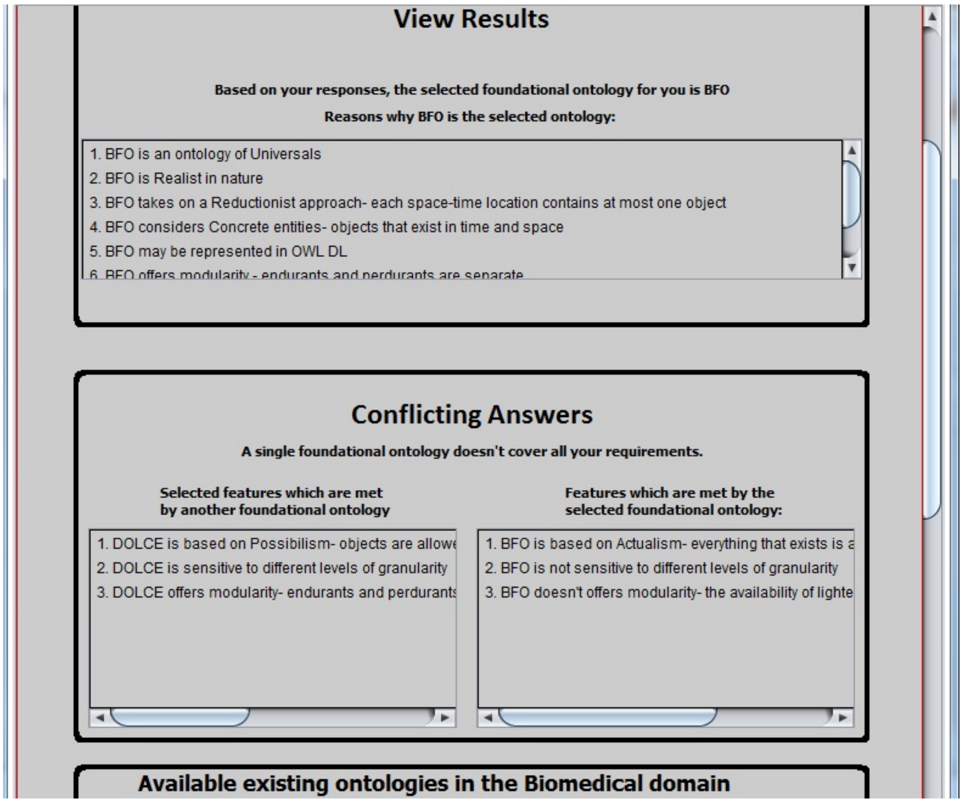

Figure A.4: Results screen of ONSET

Appendix B

Preliminary Criteria Lists for GFO and SUMO

B.0.1 GFO

Ontological Commitments

- Provides a model for space and time.
- Ontology of universals , concepts and symbols.
- Descriptive- captures the ontological categories underlying natural language and human common sense.
- Endurantism and Perdurantism - GFO contains entities that are wholly-present and those that occur in time.
- Mereology theory undertaken is not specified- mereology is the study of system decomposition and parts, wholes and boundaries.
- Temporal aspects provided by GFO are in the form basic entities of time called chronoids which are described as being temporal intervals with boundaries.

Representation Language/s

- FOL
- OWL DL
- OWL 2 DL
- KIF

Software engineering properties

- Dimensions
  - GFO Full in OWL- 79 classes, 97 subclass relations, 67 object properties.
  - GFO Basic in OWL- 44 classes, 28 subclass relations, 41 object properties.
- Licensing- released under the modified BSD License.
- Modularity - the availability of a lighter version (basic).
- Being actively maintained.

Subject domain

- Biomedical domain
- Bioinformatics
- Medical informatics systems

Applications

- Domain specific semantic wikis
- Ontological foundation of conceptual modelling
- Modelling methodologies and languages to be used in software applications making them more explicit
- To formally represent scientific theory

B.0.2 SUMO

Ontological Commitments

- Ontology of universals and particulars- contains both entities that can be instantiated and those that cannot be instantiated.
- Endurantism and Perdurantism - SUMO contains entities that are wholly-present and those that occur in time.
- Descriptive-captures the ontological categories underlying natural language and human common sense.
- Distinguishes between abstract and concrete entities.
- Mereology theory undertaken is not specified- mereology is the study of system decomposition and parts, wholes and boundaries.
- Temporal aspects are not provided- temporal aspects are ways in which the foundational ontology deals with time eg. by adding quantification over time.

Representation Language/s

- SUO-KIF
- OWL DL

Software engineering properties

- Dimensions
  - 1000 terms, 4000 axioms, 750 rules.
  - 20,000 terms and 70,000 axioms when all domain ontologies are combined.
- Licensing- SUMO is owned by IEEE but basically public domain -open license.
- Modularity- divided into SUMO, MILO and other domain ontologies.

Subject domain

- Home energy management
- Military
- Simulation
- Sensors
- Management business process

Applications

- Search
- Linguistics
- Reasoning and language generation
- Ontology driven information systems
- The Semantic Web
- Ontologies for natural language processing

Appendix C

Comparative Tables of Foundational Ontologies

Data is extracted from the criteria lists to compare foundational ontologies. Representation languages and ontological commitments are compared in the following tables.

	DOLCE	BFO	GFO	SUMO
OBO		X		
FOL		X		
KIF	X	X	X	X
OWL DL	X	X	X	X
OWL 2 DL	X	X	X	
OWL 2 QL		X		
OWL 2 EL		X		

Table C.1: Comparison of representation languages

DOLCE	BFO	GFO	SUMO
Ontology of Particulars	Ontology of Universals	Ontology of Universals, concepts and symbols	Ontology of Universals and particulars
Descriptive	Realist	Descriptive	Descriptive
Endurantism and perdurantism	Endurantism and perdurantism	Endurantism and perdurantism	Endurantism and perdurantism
Possibilism	Actualism	unclear	unclear
Eternalist stance	Eternalist stance	unclear	unclear
Concrete and abstract entities	Concrete entities	Concrete and abstract entities	Concrete and abstract entities
GEM theory theory	No mereology theory	No mereology theory	No mereology theory
Temporal aspects provided	Temporal aspects not provided	Temporal aspects provided	Temporal aspects not provided
High level of granularity	Sensitive to granularity	unclear	unclear

Table C.2: Comparison of ontological commitments

Appendix D

Additional references

1. Garcia, A., ONeill, K., Garcia, L.J., Lord, P., Stevens, R., Corcho, O., Gibson, F.: Developing Ontologies within Decentralised Settings. In: Chen, H., Wang, Y., Cheung, K.-H. (eds.): Semantic e-Science, Vol. 11. Springer US (2010) 99-139

2. Spaccapietra, S., Stuckenschmidt, H. and Parent, C., 2009. Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization H. Stuckenschmidt, C. Parent, and S. Spaccapietra, eds., Springer.

3. Hoehndorf, R., Kelso, J. and Herre, H., 2009. The ontology of biological sequences. BMC Bioinformatics, 10(1), p.377.

4. Simon, J. and Smith, B., 2004. Using Philosophy to Improve the Coherence and Interoperability of Applications Ontologies : A Field Report on the Collaboration of IFOMIS and L & C. Methods.

5. Spear, A.D., 2006. Ontology for the Twenty First Century: An Introduction with Recommendations. Science, p.1-132.

6. Bruaux, S., Kassel, G. and Morel, G., 2005. An ontological approach to the construction of problem-solving models. Main, (LRR 2005-03), p.181.

7. Guarino, N., 2006. Making Basic Ontological Assumptions O. Group, ed. Applied Ontology.

8. Temal, L. et al., 2008. Towards an ontology for sharing medical images and regions of interest in neuroimaging. Journal of Biomedical Informatics, 41(5), p.766-778.

9. Gangemi, A. et al., 2002. Sweetening Ontologies with DOLCE A. Gmez-Prez and V. R. Benjamins, eds. Knowledge engineering and

knowledge management Ontologies and the semantic Web, 2473, p.166-181.

10. Gangemi, A. and Mika, P., 2003. Understanding the Semantic Web through Descriptions and Situations. In R. Meersman, Z. Tari, eds. Proceedings of International Conference on Ontologies Databases and Applications of SEmantics ODBASE 2003. Springer, pp. 689-706.

11. Porzel, R. and Warden, T., 2010. Working Simulations with a Foundational Ontology. In K. Schill, B. Scholz-Reiter, and L. Frommberger, eds. Artificial Intelligence and Logistics AiLog Workshop at ECAI 2010. pp. 67-72.

12. Gangemi, A. et al., 2003. Some ontological tools to support legal regulatory compliance, with a case study. Lecture Notes in Computer Science, 2889, p.607-620.

13. Despres, S. and Szulman, S., 2004. Construction of a Legal Ontology from a European Community Legislative Text. In T. Gordon, ed. Legal Knowledge and Information Systems Jurix 2004 The Seventeenth Annual Conference. IOS Press, pp. 79-88.

14. Garbacz P., Trypuz R., Szady B., KulickibP., Gradzki P., and Lechniak M. 2010. Towards a formal ontology for history of church administration. In Proceeding of the 2010 conference on Formal Ontology in Information Systems: Proceedings of the Sixth International Conference (FOIS 2010), Antony Galton and Riichiro Mizoguchi (Eds.). IOS Press, Amsterdam, The Netherlands, The Netherlands, 345-358

15. Fonou-Dombeu, J.V., and Huisman, M. Semantic-Driven e-Government: Application of Uschold and King Ontology Building Methodology for Semantic Ontology Models Development in International Journal of Web & Semantic Technology (IJWesT) Vol.2, No.4, October 2011

16. Gangemi, A., F. Fisseha, J. Keizer, J. Lehmann, A. Liang, I. Pettman, M. Sini, M. Taconet, 2004: A Core Ontology of Fishery and its Use in the FOS Project, in Gangemi, A., Borgo, S. (eds.): Proceedings of the EKAW*04 Workshop on Core Ontologies in Ontology Engineering.

17. Arndt, R., Troncy, R. and Hardman, L., 2009. COMM : A Core Ontology for Multimedia Annotation S. Staab and R. Studer, eds. Group, p.403-421.

18. Ai, J.; Almeida, M. B.; Rutenberg, A.; Andrade, A.Q.; Smith, B. (2011). Body Fluids Ontology: A Unified Application Ontology for Basic and Translational Science - Proceedings of the International Conference of Biomedical Ontologies, (ICBO), 2011.

19. Keet, C.M. A Formal Theory of Granularity. PhD Thesis, KRDB Research Centre, Faculty of Computer Science, Free University of Bozen-Bolzano, Italy. 2008.

20. Borgo, S. and Masolo, C., 2009. Foundational choices in DOLCE S. Staab and R. Studer, eds. Applied Ontology, 2, p.361-381.

21. D. Oberle, S. Grimm, and S. Staab. An Ontology for Software. In S. Staab and R. Studer, editors, Handbook on Ontologies, International Handbooks on Information Systems, chapter 18, pages 383402. Springer, 2nd edition edition, 2009.

22. Lando, P., Lapujade, A., Kassel, G., Furst, F.: Towards a general ontology of computer programs. In: Proceedings of the 2nd International Conference on Software and data Technologies: ICSOFT 2007. (2007) 25-27 July 2007, Barcelona, Spain.