**rAn experience report for the development of an ontology to be used for the automatic generation of MCQ's for teaching Java**

The procedure planned was to go through each 'Theme' in the online Moodle course, completing each one in turn. There are 8 themes in total, where each theme has a number of modules each containing glossary terms, information on concepts and practices used in Java programming

**Theme 1: Fundamentals**

*Goals:*

* Work through all 6 modules in the fundamental theme in order presented.
* Extract any concepts, keywords and add immediately into the ontology.
* Search for an appropriate definition for each entry and include that in the ontology.
* Establish links between new entrants and existing concepts, when and where possible.
* To gather metrics after completing the entry of domain knowledge from the first theme.

*Tasks accomplished:*

* Downloaded an ontology visualisation plug-in for Protégé for illustrating developed ontology.
* Completed entry of all concepts, terms and declarative domain knowledge for Fundamental theme.
* Entered definitions for each concept, where possible.
* Did not include last module 'Case study' in the ontology, as this provided a practical programming example that recapped all information explained in earlier modules.
* Sent completed ontology to Tahani for reviewing. Any improvement comments will be applied to the ontology, in a revised version.

*Duration:*

Completion of the fundamental theme took total of 10 hours 15 minutes spread over 6 days.

*Metrics:*

The metrics were gathered after completion. No metrics were gathered for each of the 6 days in which concepts from the fundamental theme were still being entered.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class count** | **Object**  **property count** | **Axiom count** | **Logical axiom count** | **Subclass of**  **logical axiom count** | **Equivalent classes**  **axiom count** | **Hidden GCI count** | **Expressivity** | **Duration** |
| 91 | 44 | 315 | 144 | 144 | 0 | 0 | ALCQ | 10hrs 15mins |

*Summary of Challenges and Lessons Learned*

* During the training session, the ontology was easier to understand when presented in a model. So I started off with downloading 'GraphViz' a visualisation software for Protégé. At the beginning, displaying the ontology in a graph helped with establishing an understanding of the ontology and keeping track of classes and relationships. However, with the increase in knowledge and classes and their relationships increased, it got very hard to maintain the same level of understanding as I had at the beginning of the development process. Even the ontology graphs did not help, especially that the diagram only showed the class hierarchy diagram and not the object properties (relationships).
* At the beginning, there was a struggle to distinguish between declarative and procedural knowledge within the course. I was tempted to include the procedural knowledge within the ontology, which was not acceptable. I had to train myself to understand the difference between procedural and declarative knowledge, which meant that I had to read through content almost twice to figure out whether procedural or declarative, which made it very time consuming to enter knowledge in ontology.
* I was aware that glossary term definitions had to be included in the ontology, however, I faced a challenge in how to include these in ontology terms. I opted for including them in the Annotation section and also representing it in terms of classes and object property if this was possible. I was unaware that any content in the annotations section will be used for question generation, until Tahani explained this. And therefore, in the first version of the fundamental theme ontology, effort was not made to make sentences comprehensible and suitable for being used within a question.
* There was some struggle in representing 1 to many relationships in the ontology. I had to make a guess on how to do this and then clarify it with Tahani. Tahani confirmed that I had chosen the correct way of doing it, which was to set the minimum cardinality to 1.
* Once the Fundamental theme was complete, I chose to go through the content and ontology again to double check that everything was understandable for me and could be used to more knowledge from the proceeding themes. Going through the ontology again, I struggled to make any sense out of what I had originally produced, and struggled to match knowledge from the ontology to information in the course. It was very confusing for me and acted as a barrier for moving onto the second theme 'Classes'. As a result of this, I created a new version of the ontology, starting from scratch. The original version was sent to Tahani for reviewing. Any comments will be added to the new revised version, which will be sent to Tahani once completed.

*Training requirements*

* How to download and use ontology visualisation plug-ins to enhance the ontology's development and comprehensibility. I believe that downloading the plug-in alone was not effective in providing an understanding of the developed ontology, as there were many aspects missing in the diagram (object property), which could have enhanced my understanding and helped me keep track of my progress, if it was available.
* How to choose appropriate title for object properties, to help eliminate redundancy.
* How to distinguish between declarative and procedural domain knowledge
* How to provide definition terms of good pedagogical quality.

*Future plans*

* To review definitions for concepts and ensure they do not include any clue for the defined concept, and are comprehensible. This will apply to all future concepts too.
* The ontology will no longer contain any Java specific syntax in the ontology. Instead, this will be used in the second half of the project, which includes developing programming based questions. The focus must remain on domain concepts and knowledge.
* To review object properties in the ontology and ensure they are not redundant and are comprehensible. For example, 'hasConcept' vs 'hasFundamentalFeature'. These meant the same thing in the ontology so one could be removed and other used instead. This will apply to all future properties.
* To review the structure of the concepts in the ontology and make sure that all child classes are added under their correct parent classes. For example, some domain concepts under the class 'OOConcept' in the ontology, did not all referred to OO-specific concepts, some of which were concepts that were used in all programming language, both OO and procedural.
* To keep copy of old and new revised versions of the ontology, by saving it under a new filename each day the ontology has been modified (ideally using date as the filename). This will be very important for keeping track of ontology progress, quality, metrics and also to easily measure and evaluate the quality of questions generated as the ontology evolves.
* To keep track of ontology metrics, by creating a spreadsheet. Progress made on each day will be entered into the spreadsheet.
* To keep note of any training requirements required to improve entry quality and time.

**Revised Theme 1: Fundamentals**

*Goals:*

* To review and improve definitions for existing concepts.
* To remove all Java-specific syntax from ontology.
* To eliminate redundancy on existing object properties, by merging similar properties together.
* To restructure the ontology so that it somewhat maps with the structure of the course.
* To create a spreadsheet for storing ontology-specific metrics.
* To save revised version under new filename.

*Tasks accomplished:*

* Created spreadsheet to hold metrics for ontology
* Restructured ontology, so that OO-specific concepts and Java-specific concepts are distinguished.
* Object properties were reviewed and redundant data was removed. Existing classes, that made use of deleted object properties, were edited to make use of existing properties.
* All definitions were edited, and appropriate, pedagogical definitions were searched in the glossary and the Sun Oracle website.
* Sent revised ontology to Tahani for reviewing and running ontology through question generator.

*Duration:*

Completion of the revised version took 5.5 hours (half the time taken to complete the first time round) over 2 days.

*Metrics:*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activity** | **Class count** | **Object**  **property**  **count** | **Axiom**  **count** | **Logical axiom count** | **Subclass of**  **logical axiom count** | **Equivalent classes axiom count** | **Hidden GCI count** | **Expressivity** | **Duration** |
| Adding guidelines | 90 | 47 | 326 | 151 | 148 | 3 | 0 | ALCQ | 2hrs 30mins |
| Restructuring ontology | 91 | 38 | 331 | 154 | 149 | 5 | 5 | ALCQ | 3hrs |

*Summary of Challenges and Lessons Learned*

* Restructuring the ontology made it very easy to follow and understand. However, the structuring procedure itself was not very easy. There was some difficulty in deciding which categories each concept fell under. I could not do this alone, and using the online Java course was not of much help. Instead, making use of different Java teaching websites such as Sun Oracle, provided a lot of help in categorising important OOP and Java concepts.

*Training requirements*

* How to categorise Java programming language concepts, constructs and features. It would be easier if there was a link to useful teaching or pedagogical material, that could be followed as a guideline, to help reduce time and improve ontology quality and comprehensibility.

*Future plans*

* The procedure mentioned at the beginning, will no longer be followed (i.e going through each theme in turn). Instead, the glossary terms along with their definitions will be included first in the ontology. Following the 5 steps for developing an ontology by Tatiana Gavrilova, which begins with 'Glossary development'. This helps to include all main concepts and objects of the domain as early as possible in the ontology development process.
* To continue to follow the same guidelines as above for all remaining themes and glossary.
* Send email to Tahani to request a link or copy of question generator tool, to begin generating questions and evaluating quality.

**Glossary:**

*Goals:*

* To split entry of terms over several days, to allow for progress in other areas of project.
* Every concept entered must have an accompanying definition in the ontology.
* To link new concepts with existing concepts in the ontology.

*Tasks accomplished:*

* With a total of 20 alphabets, terms under 5 alphabets were added each day, giving a total of 4 days for completing entry of glossary terms.
* Accompanying definitions from the glossary were added to the ontology, with some modifications to make it suitable for use in question generation.
* Saved revised version under new filename.

*Duration:*

Completion of glossary term entry took a total of 10 hours over 4 days.

*Metrics:*

After completing entry of all glossary terms:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class count** | **Object**  **property count** | **Axiom Count** | **Logical axiom** | **Subclass of**  **Logical axiom count** | **Equivalent classes**  **axiom count** | **Hidden GCI count** | **Expressivity** | **Duration** |
| 183 | 46 | 619 | 235 | 225 | 9 | 8 | ALCHQ | 10hrs |

*Summary of Challenges and Lessons Learned*

* During the addition of new concepts, some difficulty was faced for deciding exactly where they should go in the ontology. Some terms like 'Abstract class' would have thought to go under a class, because it is a class. However, an abstract class does not inherit all properties of its parent 'Class', for example, a class can have multiple instances but an abstract class cannot. Due to these minor uncertainties, the entry of terms was a very time consuming task and only terms under A and B were added on the first day.
* To accelerate entry time, all terms were added under 'Thing' in the ontology, and will be rearranged under the appropriate parent class or linked to other existing classes after reading through the theme in which it was mentioned in.
* Adding appropriate definition for each term entered was another challenge, especially when the definition in the glossary were very vague and hard to understand. So I opted for using the Sun Oracle website to find appropriate definitions to add. As a result of this, entering terms was a very time consuming task.

* As a result of the above difficulties, on the first day, only A and B terms were added. Second day included entry for C, D and E. Third day involved entering F, G, I, J, L, M, N, O and on final day all remaining terms.

*Future plans*

* To start entering domain knowledge information from the second theme: Classes.
* To continue to follow guidelines used for implementing the first theme: Fundamentals.
* To continue to gather training requirements needed for developing a teaching ontology

**Theme 2: Classes**

*Goals:*

* To work through all 7 modules in the Classes theme in order presented.
* Extract any concepts, keywords and add immediately into the ontology.
* Search for an appropriate definition for each entry and include that in the ontology.
* Establish links between new entrants and existing concepts, when and where possible.
* To gather metrics after completing the entry of domain knowledge from the theme.

*Tasks accomplished:*

* Completed entry of all concepts, terms and declarative domain knowledge for Classes theme.
* Entered definitions for each concept that had not already been included in the Glossary of terms. Definitions were extracted from the Sun Oracle Website.
* 4 modules were not included in the ontology, two of which contained pure procedural knowledge and examples. One of the other two was a 'Case study' module, which provided a practical programming example that recapped all information explained in earlier modules and remaining module explained best practice in including comments in a Java program. For safety, a quick read was made to confirm that that no conceptual knowledge had been overlooked.
* All new entrants were linked to their parent classes where possible, or related to other existing members using relevant object properties.

*Duration:*

Completion of the Classes theme took a total of 6 hours within 1 day.

*Metrics:*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class count** | **Object**  **property**  **count** | **Axiom**  **count** | **Logical axiom count** | **Subclass of**  **logical axiom count** | **Equivalent classes**  **axiom count** | **Hidden GCI count** | **Expressivity** | **Duration** |
| 202 | 60 | 718 | 290 | 279 | 10 | 9 | ALCHQ | 6hrs |

*Summary of Challenges and Lessons Learned*

* Due to the tremendous amount of procedural knowledge in this theme, extracting declarative knowledge was a bit of a struggle. I have come to notice that this is always the case when there procedural knowledge is presented alongside declarative knowledge.
* I have also come to notice that the struggle in entering and categorising concepts in the ontology for this theme, was due to the wrong order in which the themes have been presented in the course. The theme following the Classes theme - Nuts and Bolts - should have come prior to Classes, as there were many concepts added from the Classes theme that were greatly dependant in the knowledge in the Nuts and Bolts theme.
* As a result of the above disorder, or maybe my lack of understanding and awareness of how the course should have been approached, many concepts were not categorised under their correct parent and instead ended up with most concepts being under the highest-level class 'Thing'.

*Future plans*

* Once entry for concepts in the Nuts and Bolts theme has been completed, a review of the ontology will be made, to ensure that all concepts from both Classes and Nuts and Bolts themes have been categorised correctly, and as many concepts as possible with relevant links to other members, where possible.
* To start entering domain knowledge information from the third theme: Nuts and Bolts.
* To continue to follow guidelines used for implementing the previous themes.
* To continue to gather training requirements needed for developing a teaching ontology.
* To continue to gather ontology metrics.

**Theme 3: Nuts and Bolts**

*Goals:*

* To work through all 6 modules in the Nuts and Bolts theme in order presented.
* Extract any concepts, keywords and add immediately into the ontology.
* Search for an appropriate definition for each entry and include that in the ontology.
* Establish links between new entrants and existing concepts, when and where possible.
* To gather metrics after completing the entry of domain knowledge from the theme.
* To rearrange uncategorised concepts from previous theme 'Classes', and aim to list as many as possible under their parent classes, or provide links to other relevant members of the ontology.

*Tasks accomplished:*

* Completed entry of all concepts and terms from the Nuts and Bolts theme.
* Entered definitions for each new entrant that had not already been included from the Glossary of terms. Definitions were extracted from the Sun Oracle Website.
* Similar to prior themes, modules including 'Case Study', 'Best Practice' and general introduction to the theme were omitted from the ontology, after reviewing their contents for possible conceptual domain knowledge. One other module omitted in this theme was 'Differences from C', that included information regarding procedural differences between Java and C programming language.
* The overall theme contained information that was building up on knowledge presented in previous themes, so the main tasks carried out was developing and improving the details and properties of existing concepts, and building up their relationships with other concepts in the ontology.

*Duration:*

Completion of the Nuts and Bolts theme took a total of 5 hours within 1 day.

*Metrics:*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class count** | **Object**  **property**  **count** | **Axiom**  **count** | **Logical axiom count** | **Subclass of**  **Logical axiom count** | **Equivalent classes**  **axiom count** | **Hidden GCI count** | **Expressivity** | **Duration** |
| 233 | 69 | 823 | 344 | 333 | 10 | 9 | ALCHQ | 6hrs |

*Summary of Challenges and Lessons Learned:*

* With more and more concepts being added, the struggle of how to categorise concepts cropped up once again. Many concepts were simply listed under Thing, and remained isolated, with no parent-child relationship. This was mainly due to the fact that many of the existing concepts ended up with multiple possible parent classes, and choosing from these options was a major challenge. I assume this will make the ontology less effective in producing valuable and valid assessment questions later on. For example, an 'Array' could be listed as a child under 'Collections' or 'Reference Data Type', both of which listed under 'Java Language Fundamental Constructs', yet they have no direct object properties linking them together.
* As the new concepts were introduced, finding an appropriate existing object property to use for linking these with existing classes was a bit of a struggle. There was no other option than to keep adding new object properties, with appropriate titles, that would link each class with another in a comprehensible manner. This led to a redundant object property list, where several properties had a similar meaning but worded differently.

*Training requirements:*

* How is it possible to include a child class under more than one parent class, if it shares the same properties? and what happens if at least one property does not apply to the child element. Providing training on how to choose the correct parent class very early in the development process would eliminate the problem of significant amount of classes that still remain under the 'Thing' concept.

*Future plans*

* Completing entry for the fourth theme: Inheritance.
* To continue to gather training requirements needed for developing a teaching ontology.
* To continue to gather ontology metrics.
* To continue to follow guidelines used for implementing the previous themes.

**Theme 4: Inheritance**

*Goals:*

* To review the ontology and clean up any redundant object property entries.
* To eliminate isolated entries in the ontology, by listing as many concepts as possible, when and where possible, under appropriate existing or new parent classes.
* To work through all 5 modules in the Inheritance theme.
* Extract new concepts, keywords and add immediately into the ontology.
* Search for an appropriate definition for each new entry and include that in the ontology.
* Establish links between new entrants and existing concepts, when and where possible.
* To gather metrics after completing the entry of domain knowledge from the theme.

*Tasks accomplished:*

* All new concepts from Inheritance theme have been added to the ontology.
* Entered definitions for each new entrant that had not already been included from the Glossary of terms. Definitions were extracted from the Sun Oracle Website.
* Similar to prior themes, modules including 'Case Study', 'Best Practice' and general introduction to the theme were omitted from the ontology, after reviewing their contents for possible conceptual domain knowledge.
* Reviewed existing object properties and class categorisation. A total of 5 redundant properties have been eliminated.

*Duration:*

Completion of the Inheritance theme took a total of 8 hours within 1 day.

*Metrics:*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class count** | **Object**  **property**  **count** | **Axiom**  **count** | **Logical axiom count** | **Subclass of**  **Logical axiom count** | **Equivalent classes**  **axiom count** | **Hidden GCI count** | **Expressivity** | **Duration** |
| 237 | 75 | 861 | 372 | 360 | 10 | 9 | ALCHQ | 8hrs |

*Summary of Challenges and Lessons Learned*

* Due to the complexity in the way the information has been presented for this theme in the course, extracting the conceptual knowledge from the content was challenging. Although the theme contained two fundamental modules, they both presented vast amounts of procedural knowledge and examples, making the extraction process very tough. The wording of the information was not very comprehensible. Substantial amount of effort and time was spent going very the information in hope to find some possible way to add this knowledge in the ontology.
* Came across a number of concepts that could have been listed under more than one parent, however, at the same time the concept did not share all properties of the parent class. To save time and effort, these were listed in isolation, under 'Thing'. This has been a persistent problem from the beginning of the ontology development, and with more concepts to come this problem will remain until the conceptualisation task has been completed.
* With half of the themes conceptualised, the ontology was reviewed once again, and once again, there was some struggle in making sense out of the ontology, without looking back at the online and following it for guidance. No changes were made, however, a note has been made to make a final review and modification of the ontology if required, once all 8 themes have been conceptualised.

*Future plans:*

* To start entering domain knowledge information from the fifth theme: Exceptions.
* To continue to follow guidelines used for implementing the previous themes.
* To continue to record any training requirements needed for developing a teaching ontology.
* To continue to gather ontology metrics.

**Theme 5: Exceptions**

*Goals:*

* To continue to maintain a non-redundant listing of object properties.
* To read through and extract conceptual knowledge and keywords from all 5 modules in the Exceptions theme.
* Search for an appropriate definition for each new entry and include that in the ontology.
* Establish links between new entrants and existing concepts, when and where possible.
* To gather metrics after completing the entry of domain knowledge from the theme.

*Tasks accomplished:*

* All new concepts from Exceptions theme have been added to the ontology.
* Entered definitions for each new entrant that had not already been included from the Glossary of terms. Definitions were extracted from the Sun Oracle Website.
* Similar to prior themes, modules including 'Case Study', 'Best Practice' and general introduction to the theme were omitted from the ontology, after reviewing their contents for possible conceptual domain knowledge.

*Duration:*

Completion of the Exceptions theme took a total of 1 hour within 1 day.

*Metrics:*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class count** | **Object**  **property**  **count** | **Axiom**  **count** | **Logical axiom count** | **Subclass of**  **Logical axiom count** | **Equivalent classes**  **axiom count** | **Hidden GCI count** | **Expressivity** | **Duration** |
| 247 | 86 | 910 | 396 | 384 | 10 | 9 | ALCHQ | 1hr |

*Future plans*

* To start entering domain knowledge information from the sixth theme: Collections.
* To continue to follow guidelines used for implementing the previous themes.
* To continue to record any training requirements needed for developing a teaching ontology
* To continue to record ontology metrics.

**Theme 6: Collections**

*Goals:*

* To continue to maintain a non-redundant listing of object properties.
* To read through and extract conceptual knowledge and keywords from all 7 modules in the Collections theme.
* Search for an appropriate definition for each new entry and include that in the ontology.
* Establish links between new entrants and existing concepts, when and where possible.
* To gather metrics after completing the entry of domain knowledge from the theme.

*Tasks accomplished:*

* All new concepts from Collections theme have been added to the ontology.
* Entered definitions for each new entrant that had not already been included from the Glossary of terms. Definitions were extracted from the Sun Oracle Website.
* Ensured that all new entries have been listed under appropriate parent classes, when and where possible, and not left isolated.

*Duration:*

Completion of the Collections theme took a total of 6 hour within 1 day.

*Metrics:*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class count** | **Object**  **property**  **count** | **Axiom**  **count** | **Logical axiom count** | **Subclass of**  **Logical axiom count** | **Equivalent classes**  **axiom count** | **Hidden GCI count** | **Expressivity** | **Duration** |
| 265 | 95 | 989 | 435 | 423 | 10 | 9 | ALCHQ | 6hr |

*Future plans*

* To start entering domain knowledge information from the seventh and eighth themes: Graphics and GUI, and Streams and I/O.
* To continue to follow guidelines used for implementing the previous themes.
* To continue to record any training requirements needed for developing a teaching ontology
* To continue to record ontology metrics.
* To review all the ontology and ensure all definitions have been entered appropriately, no redundant object properties that could be eliminated, classes categorised appropriately and as much as possible listed under parent classes, and not 'Thing'.

**Theme 7&8: Graphics and GUI & Streams and I/O**

*Goals:*

* To continue to maintain a non-redundant listing of object properties.
* To read through and extract conceptual knowledge and keywords from all modules in last two themes 7 and 8.
* Search for an appropriate definition for each new entry and include that in the ontology.
* Establish links between new entrants and existing concepts, when and where possible.
* To clean-up the ontology and make sure links between concepts are understood and are valid.
* To categorised and group concepts under parent classes, when possible.
* To gather metrics after completing the entry of domain knowledge from the course.
* To submit finalised ontology to Tahani for review and generation of sample questions for reviewing.

*Tasks accomplished:*

* All new concepts from both themes have been entered in order presented.
* Entered definitions for each new entrant that had not already been included from the Glossary of terms. Definitions were extracted from the Sun Oracle Website.
* Ensured that all new entries have been listed under appropriate parent classes, when and where possible, and not left isolated.
* Linked new entrants to existing classes whenever possible. If not, these were left under Thing for review by Tahani.
* Content in last two themes, was easily understood, hence no difficulty found in extracting declarative knowledge from content.
* Object properties was reviewed for final time before submission. No redundant listings were found, as it was hard to use one particular property to match a number of relationships, so all object property listings remained in the ontology.

*Duration:*

Completion of the both Graphics and GUI, and Streams and I/O theme took a total of 6 hour within 1 day.

*Metrics:*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class count** | **Object**  **property**  **count** | **Axiom**  **count** | **Logical axiom count** | **Subclass of**  **Logical axiom count** | **Equivalent classes**  **axiom count** | **Hidden GCI count** | **Expressivity** | **Duration** |
| 319 | 107 | 1151 | 513 | 501 | 10 | 9 | ALCHQ | 6hr |

**Revised Ontology - Version 2**

*Goals:*

* To review the first batch of generated questions from the ontology.
* To extract and clearly document any problems or notes for improvements, with respect to the questions generated by the question generator or the developed ontology, when and where possible.
* To discuss the noted outcomes, problems and improvements (gathered by all three reviewers) with Tahani.
* To implement the agreed improvements and amendments in the first revision phase of the ontology.
* To keep note of any problems and struggles faced during the second revision phase, to discuss with Tahani during the second ontology revision phase.
* To finally run the reasoner for detecting any possible logical errors, and correct this.
* To gather metrics after completing the second revision phase of the ontology.

*Tasks accomplished:*

* A total of 65 questions were reviewed. Notes on possible improvements or errors detected were made in a spreadsheet, indicating the level of difficulty, usefulness and adherence to MCQ rules.
* Discussed improvements and errors with developer (Tahani) before starting the amendment process.
* Rearranged some classes to make clear the logical relationship amongst concepts in the ontology.
* Added definitions to classes which were missing definitions, and improved on the existing definition of classes to ensure they were grammatically correct.
* Finalised the object property list by removing redundant properties and renaming properties that were seen as grammatically infeasible for being used in a question.Existing classes, that made use of deleted object properties, were edited to make use of existing properties
* Executed the reasoner to ensure no logical errors existed in the ontology.
* Sent the second revised version of the ontology to Tahani for reviewing. Any improvement comments will be applied to the ontology in the third revision phase.

*Duration:*

The Duration for reviewing the first round of generated questions (65 questions) took a total of 2hours 30 minutes.

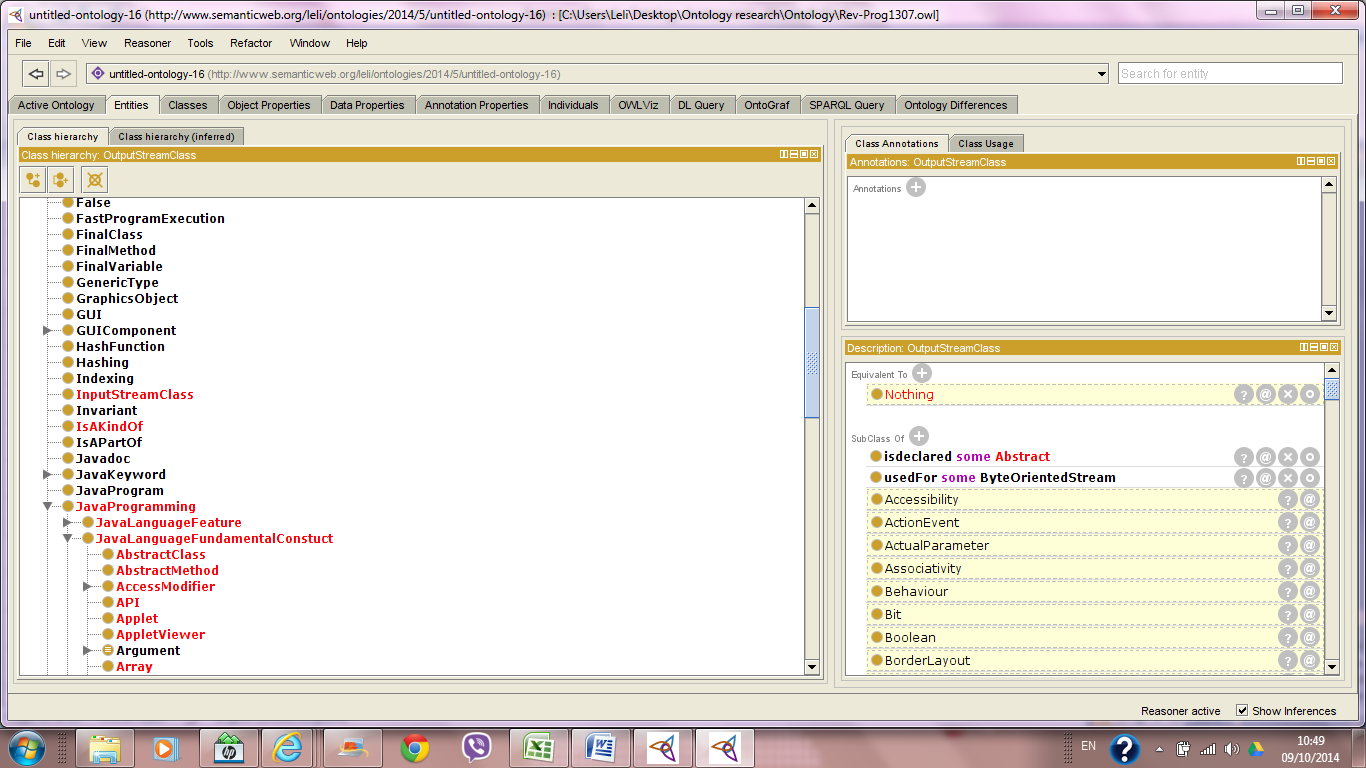
The first revision phase of the ontology was divided into three sections: Implementing the improvements/amendments from reviewing the questions, implementing the guidelines from reviewing the ontology with the developer (Tahani), restructuring the ontology. Completion of the first revision phase tool a total of 38 hours and 30 minutes over a total of 6 days (approximately two thirds of the time taken to complete the ontology first time round).

*Metrics:*

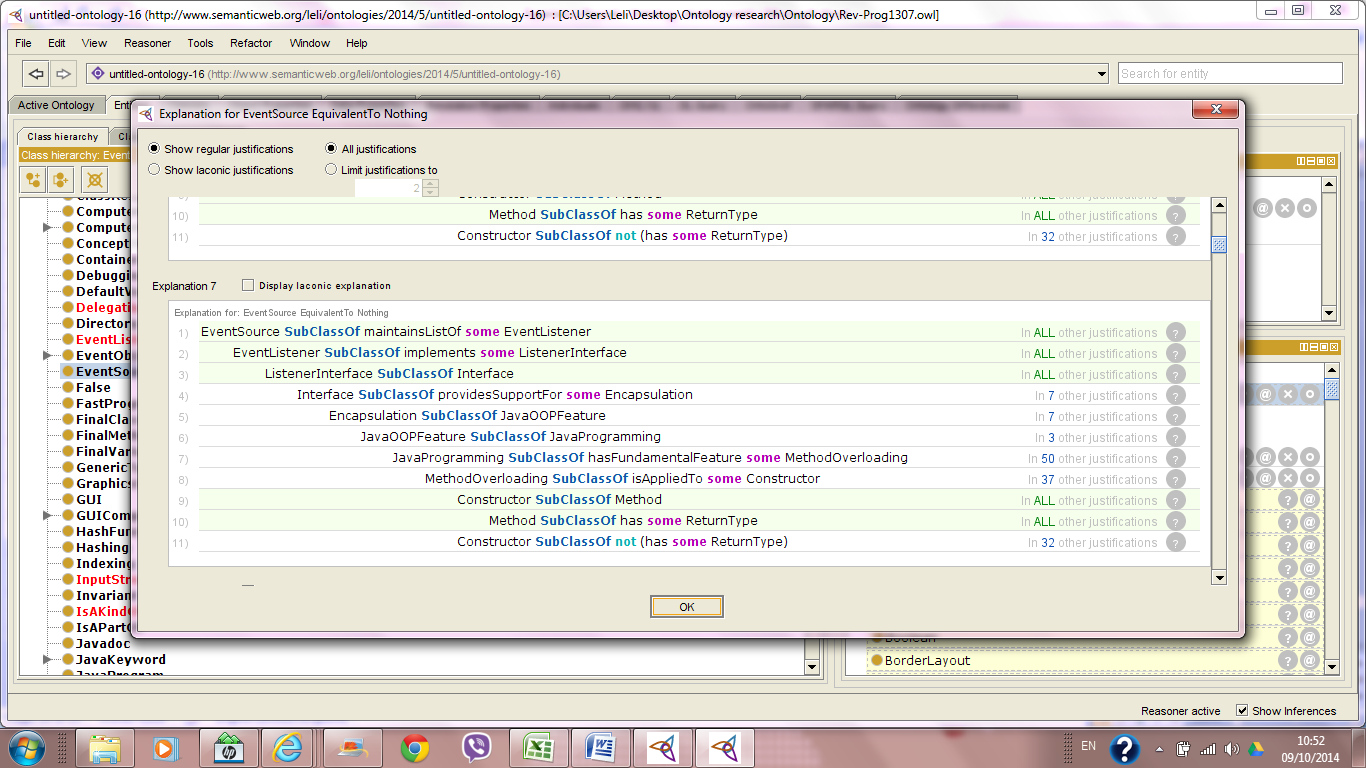
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class count** | **Object**  **property**  **count** | **Axiom**  **count** | **Logical axiom count** | **Subclass of**  **Logical axiom count** | **Equivalent classes**  **axiom count** | **Hidden GCI count** | **Expressivity** | **Duration** |
| 317 | 73 | 1213 | 578 | 567 | 9 | 15 | ALCHQ | 38hrs 30mins |

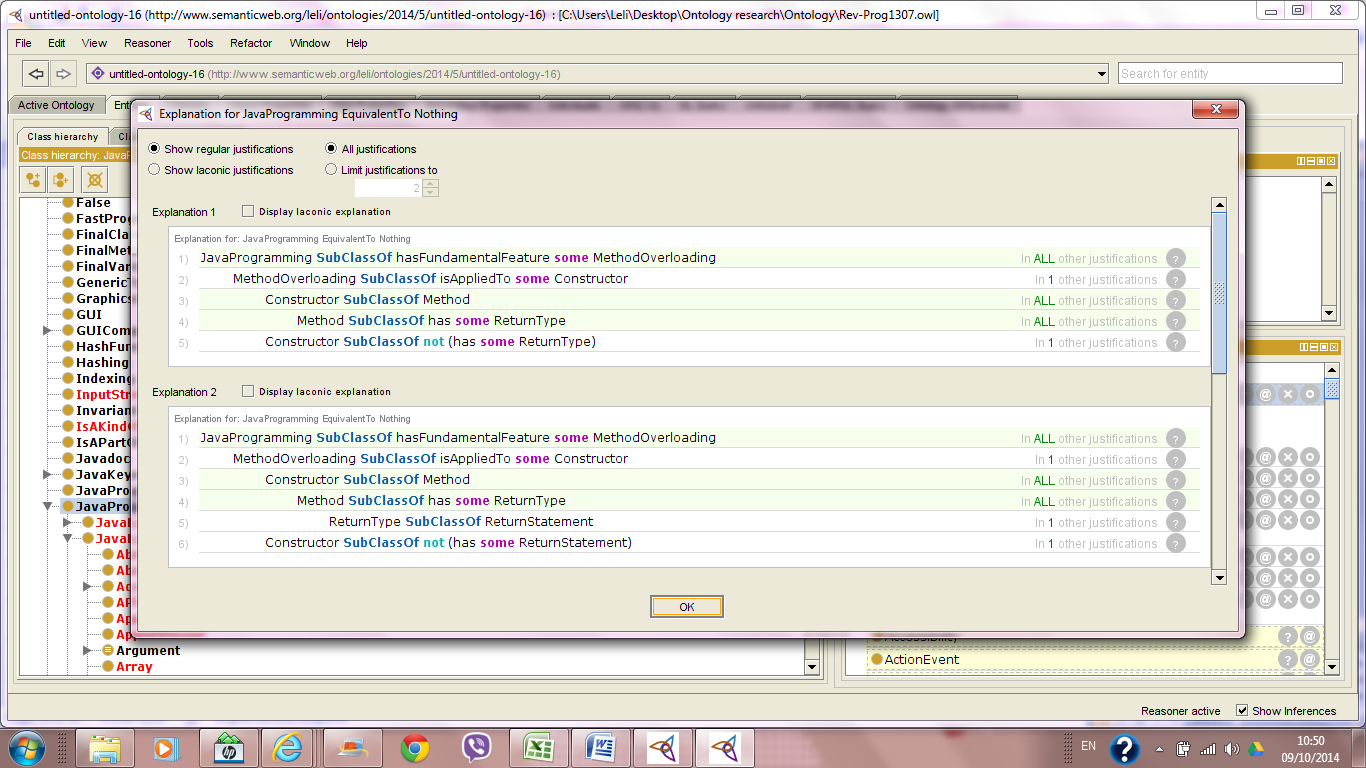
*Summary of Challenges and Lessons Learned*

* After running the reasoner, a major problem was identified. The class 'Constructor' was initially listed as a child to its parent class 'Method', however, some of the class properties assigned to 'Constructor' were contradictive to the properties of its parent class 'Method'. A total of 15 classes, along with their children, were affected by this logical error (as can be seen highlighted in red in the following screenshot).



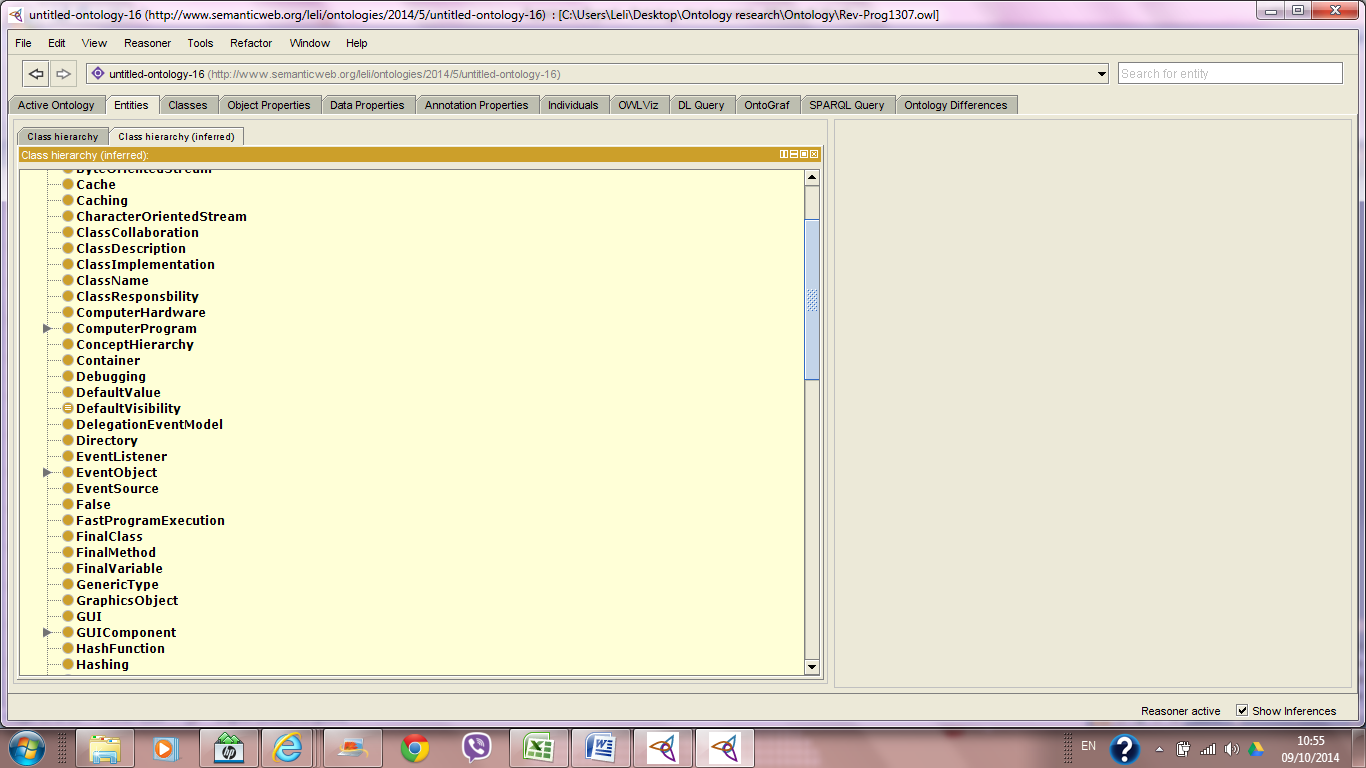
An example of some of the logical errors found can be seen in the following screenshots showing an explanation of the inference (highlighted with a red circle):





To correct this, the class 'Constructor' was moved to the next level up in the hierarchy of classes, the parent of the class 'Method' which is 'Fundamental Java Language Construct'.

To ensure that no more logical errors existed, the reasoner was executed once more. No errors were found:



* After reviewing the set of generated questions, it appeared that the current way in which the object properties have been named resulted in grammatically infeasible questions being generated, to the extent that it also had an impact on the level of difficulty predicted by the reviewers compared to that allocated by the developer. Based on this outcome, all object properties were reviewed and altered to ensure they result in grammatically acceptable questions, for example the following object properties 'executedBy', 'accessedBy' and 'alwaysDeclared' were changed to 'isExecutedBy', 'isAccessedBy' and 'isAlwaysDeclared', making the latter more 'grammatically' adequate compared to the former set of object properties, when presented within the context of an assessment question.
* A total of 23 improvements were made as a result of 3 reviewers reviewing the set of 65 generated questions. The improvements can be summarised in:
* Rephrasing or changes made to the definition of some of the classes to enhance the comprehensibility of the generated questions.
* Corrected typo errors that were found in some of the definitions used in some of the questions.
* Altered some of the object property names used to enhance the grammatical validity of the questions making use of these names. For example, Swing 'hasFundamentalFeature' Accessibility. If the phrase would be presented to a student in a question, the difficulty level experienced by the student would probably be very difficult, not because the question itself is difficult, but because the phrasing of the question is very vague, hence, effecting the comprehensibility of the question by the student. Therefore, object property names and axiom phrasing were altered where possible. The previous example is now changed to: Accessibility 'isAFundamentalFeatureOf' Swing.
* Removed redundant classes to avoid it appearing in question distractors, which could have an impact on the answer selected by the student. For example, 'Portability' and 'WriteOnceRunAnywhere' are both classes in the ontology, however, in programming terms mean the same thing. One of the reviewed questions contained both classes (Portability and WriteOnceRunAnywhere) in the choices, with 'Portability' being the stem. In real assessment, the question could have caused some confusion to students.
* During the initial development of the ontology, it was clear that some classes/concepts should have been allocated to multiple parent classes. For example, 'CatchBlock' is an exception handling block, but also a Java keyword. However, during the development of the ontology it was not yet clear to me as to whether this can be done, and so, the 'CatchBlock' was only listed under the class 'ExceptionHandlingBlock'. After reviewing the initial version of the ontology with Tahani, she explained that allocating more than one parent class to a class is possible, and doing so would make the Java ontology more complete and would avoid having redundant classes that refer to the same concept but named differently. For example, instead of creating two new classes 'Catch' and 'CatchBlock', where the former is listed under 'JavaKeyword' and latter listed under 'ExceptionHandlingBlock', we simply create only one class name 'Catch' to refer to both, and instead list both 'Java keyword' and 'ExceptionHandlingBlock' as parent classes. Based on this, all classes were reviewed to identify if they belonged to any other listed classes. If so, the parent class was added to the set of properties for the child class.
* All parent/child relationships in an ontology should no represented by adding the former 'parent' to the list of properties for the latter 'child'. Despite doing this for most of the classes, the ontology still contained object properties named 'is', 'isAn', 'canBeUsedAs', 'isTypeOf' etc, and some classes were linked to each other using these properties. For example, 'PrintStream isTypeOf some OutputStream'. This is bad practice when developing an ontology. Any kind of parent/child relationship in an ontology should never be represented by using an intermediary link that explicitly describes the kind of relationship. Therefore, I went through the ontology to make sure that these 'object properties' are removed, and instead the subclasses are directly linked to their superclasses.
* During the initial version of the ontology, there was a misunderstanding on how to represent two equivalent classes. For example, in Java a primitive data type is also known as scalar. In the ontology, this was initially represented by first creating an object property called 'alsoKnownAs', then under the *Equivalent To* section in the description of the class 'PrimitiveType' the following was added: PrimitiveType 'alsoKnownAs' some Scalar. After reviewing the ontology with Tahani, this was the wrong way of representing such relationships. Instead, the equivalent class is directly listed under the *Equivalent To* section, without having to create intermediary links. To correct this, I reviewed all classes that had equivalents in the ontology, removed the 'alsoKnownAs' object property from the ontology and directly listed classes under their equivalent classes.
* The ontology contained many acronyms of Java concepts. These were added as represented in the online Java course, with the full concept term represented as a comment in the concept description section. During the first revision phase, the full concept was added as a new class, and the acronym was linked to it using a newly created object property 'StandsFor'. For example, 'JFC' class is now linked to the newly created full term class 'JavaFoundationClasses', by adding the following statement: 'StandsFor some JavaFoundationClasses' to the list of axioms for the JFC class.
* Reviewing the ontology required, in some cases, comparing what was already on the ontology with the original content on the online course. It was apparent in some cases that a misinterpretation of the course content was made in some of the axioms. For example, the concept 'Invariant' was comprehended as a kind of 'Postcondition' or 'Precondition'. Therefore, in the ontology the following axiom was used: Invariant isKindOf some (Postcondition or Precondition). After reviewing the content again, it was clear that the relationship established between these classes was invalid as a result of misinterpretation of the course content. It was now essential to review all the ontology against the course content, which required more time and effort in return for enhanced validity, quality and accuracy.
* After the initial completion of the ontology, there was a total of 71 classes directly listed under root class 'Thing'. For example, the boolean values 'True' and 'False' were listed under Thing, which in logical terms shows that both terms/classes have no association, when in fact, they are both 'Literals' in Java and are used as boolean values in Java. Also during the review of questions generated, it was clear that keeping classes listed under 'Thing', had an impact on the plausibility of the generated distractors. Therefore, the structure of the ontology was reviewed and classes were grouped and allocated to appropriate parent classes (either new or existing) as much as possible to help eliminate the number of isolated classes. For example, the Java ontology contained multiple classes representing fundamental programming language features (portability, accessibility, reusability, platform independence) all of which were listed directly under the root 'Thing'. In addition to this, an object property called 'hasFundamentalFeature' was used to link the class 'JavaProgramming' with each of these features. To eliminate this problem, all the programming language features were relocated under the existing class 'JavaLanguageFeature' and the object property 'hasFundamentalFeature' was deleted.

*Important note I had to consider during the restructuring process:* Any relocation of classes to existing parent classes mean that axioms of the parent class must be checked before hand to ensure that the rules in the parent class axioms does not violate the axioms in the child class. For example, the classes 'Precondition', 'Postcondition' and 'Invariant' were relisted under 'Statement', as it was the most appropriate class in the ontology and the axiom rules had no violation to this relationship.

It was not always the case that an existing class could be used as an appropriate parent class, sometimes new classes were added to accommodate with the restructuring process. For example, a new class called 'CRCComponents' was created to group all the existing classes representing CRC card components (ClassResponsibility, ClassCollaborators, ClassName).

After this restructuring, a total of 26 non-parent classes remained directly under 'Thing', with 7 classes representing the full name for the Acronyms used in the ontology, therefore, leaving a total of only 19 non-parent classes in isolation under the root.

**Revised Ontology - Version 3**

*Goals:*

* To review the second ontology revision and extract any improvements or amendments.
* To finally run the reasoner for detecting any possible logical errors, and correct this.
* To gather metrics after completing the third revision phase of the ontology.

*Tasks accomplished:*

* Rearranged some classes to make clear the logical relationship amongst parent-child concepts in the ontology.
* Added definitions to classes which were missing definitions.
* Added new object property to existing classes to provide a more complete knowledge base. Existing classes, that made use of object properties that were removed or replaced, were edited to make use of existing properties.
* Executed the reasoner to ensure no logical errors existed in the ontology.
* Sent the third revised version of the ontology to Tahani for reviewing. Any improvement comments will be applied to the ontology in the fourth revision phase.

*Duration:*

Completion of the third revision took a total of 8 hours within 1 day.

*Metrics:*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class count** | **Object**  **property**  **count** | **Axiom**  **count** | **Logical axiom count** | **Subclass of**  **Logical axiom count** | **Equivalent classes**  **axiom count** | **Hidden GCI count** | **Expressivity** | **Duration** |
| 305 | 74 | 1185 | 554 | 548 | 4 | 7 | ALCHQ | 8 hrs |

*Summary of Challenges and Lessons Learned*

* During the second revision some parent-child relationships have been violated. For some classes which represented acronyms of concepts (AWT, UML etc) were set a subclass property using the object properties 'standsFor' that linked it with another class representing the full form of the concept. For example, API 'standsFor' ApplicationProgrammingInterface. API has children 'AWT' and 'Swing'. The property 'API 'standsFor' ApplicationProgrammingInterface' does not represent the children of the parent class 'API'. To eliminate this problem, and other similar problems, the object property was added as an annotation. The classes that were no longer needed (such as Application programming interface, unified modelling language etc) were removed along with the object property 'standsFor'.
* Definitions were added to classes that were not defined. For example, buffer, buffered reader, file writer etc.
* New properties were added to some classes to enhance the overall ontology and provide links between its components. For example, the class 'javac' had a new property added 'generates some javabytecode'.