CS6001 Formal Specification & Software Implementation

Coursework 2: Group Report

Student ID	Student Name
18011848	Lauren Spruce
17028469	Duarte Pires Lopes
18005806	Valeriu Bezer
18005154	Gaurangkumar Gupta

CONTENTS

CONTENTS OF TABLES

CONTENTS OF FIGURES

INTRODUCTION

Semantic approach come as a rescue when the talk is about data and its heterogeneity nature. The data need to be represented in the form of ontology that being the cornerstone of the Semantic Web for extracting information , data integration , and knowledge sharing. Ontology is the form of conceptualizing that knowledge and is a backbone of semantic web applications. This document is dedicated to the logic modelling of the selected microworld “My Family” and demonstrates the requirements and implementation of the steps and task and the results achieved through a group collaborative effort. The report is divided into two parts one is the course requirement comprising of four tasks performed to create a semantic model of the selected microworld and its objects and then to build a logical theory with axioms . Further the second part provides the evidence of the group efforts via meeting logs.

TASK 1

Create a semantic model of your selected microworld with at least 10 individual objects in its universe.

SEMANTIC MODEL

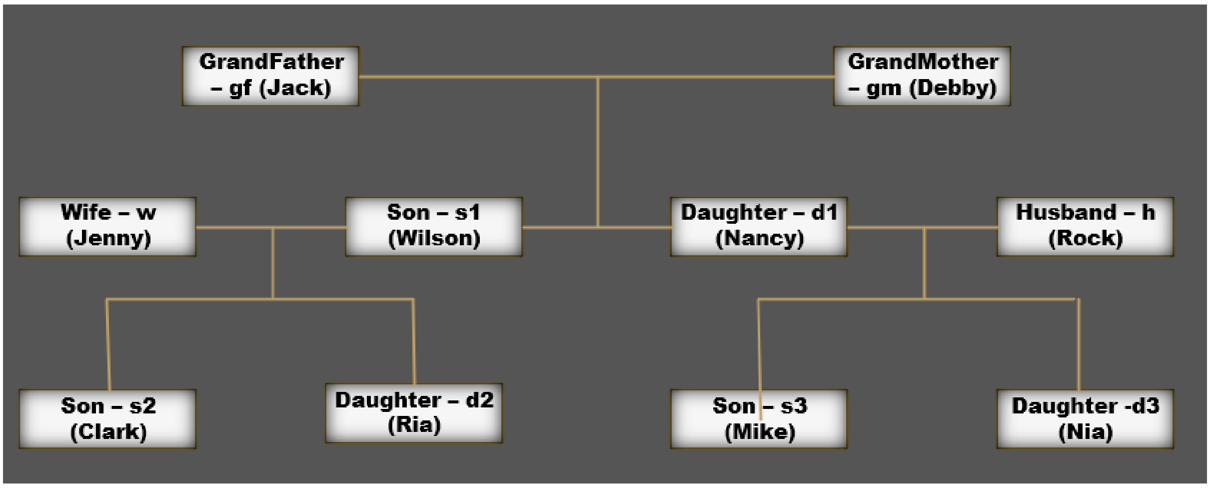

Figure 1 Semantic Model

TASK 2

Specify the vocabulary for your modelling language with at least 6 names of individual objects, 4 unary predicates, 4 binary predicates and 3 attribute functions and provide semantic interpretation for them in the model you constructed.

- **Constants** = {Jack,Debby,Jenny,Wison,Nancy,Rock,Clark,Ria,Mike,Nia}
- **Unary predicates** = {grandfather, father , son , uncle , grandmother , mother , daughter , aunt, grandson, granddaughter, child, grandchild, sibling}
- **Binary predicates** = {sister-of, brother-of, cousin-of, spouse-of, grandfather-of, grandmother-of, sibling-of}
- **Unary functions** = {father-of, mother-of, husband-of, wife-of}
- **Binary functions** = {has-wife, has-husband, has-father, has-mother}

SEMANTIC INTERPRETATION

Table 1 Semantic Interpretation

Grandfather	gf	Jack
Grandmother	gm	Debby
Wife	w	Jenny
Son	s1	Wilson
Daughter	d1	Nancy
Husband	h	Rock
Son	s2	Clark
Daughter	d2	Ria
Son	s3	Mike
Daughter	d3	Nia

Unary predicates

Grandfather = {gf) ⊆ Grandparents

Grandmother = {gm} ⊆ Grandparents

Parent = { gf, gm,w,s1,d1,h } ⊆ parents

Son = {s1,s2,s3} ⊆ Sons

Daughter {d1,d2,d3} ⊆ daughters

Grandson {s2,s3} ⊆ Siblings

Granddaughter {d2,d3} ⊆ Siblings

Grandchild {s2,s3,d2,d3} ⊆ Grandchilds

Siblings = {s1,s2,s3,d1,d2,d3} ⊆ siblings

Child = {s1,s2,s3,d1,d2,d3} ⊆ Childs

Binary predicates

Spouse-of = {(gf, gm), (gm, gf), (w, s1), (s1, w), (d1, h), (h, d1) }

Brother-of = { (s1, d1), (s2, d2), (s3, d3) }

Sister-of = {(d1, s1), (d2, s2), (d3, s3)}

Cousin-of = {(s2, s3), (s2, d3), (d2, s3), (d2, d3), (s3, s2), (s3, d2), (d3, s2), (d3, d2)}

Grandmother-of {(gm, s2), (gm, d2), (gm, s3), (gm, d3) }

Grandfather-of {(gf, s2), (gf, d2), (gf, s3), (gf, d3)}

Sibling-of {(s1, d1), (d1, s1), (s2, d2), (d2, s2), (s3, d3), (d3, s3)}

Unary functions

Mother-of = {(gm, s1), (gm, d1), (w, s2), (w,d2), (d1, s3), (d1, d3)}

Father-of = {(gf, s1), (gf, d1), (s1, s2), (s1, d2), (h, s3), (h,d3)}

Husband-of = {(gf, gm),(s1, w), (h, d1)}

Wife-of = {(gm, gf), (w, s1), (d1, h)}

Binary functions

Has-husband = {(gm, gf), (w, s1), (d1,h)}

Has-wife = {(gf, gm), (s1, w), (h, d1)}

Has-father = {(s1, gf), (d1, gf), (s2, s1), (d2, s1), (s3, h), (d3, h)}

Has-mother = {(s1, gm), (d1, gm), (s2, w), (d2, w), (s3, d1), (d3,d1)}

TASK 3

8 AXIOMS WITHIN MICROWORLD

Build a logical theory of your microworld with at least 8 axioms

In this part of the coursework, we are going to build the theory using the language that was previously introduced. Axioms can also be viewed as facts that describe our semantic model. Those facts are based on the functions and predicates that have been established. For creating those axioms, we are going to use quantifiers such as the symbol ∀ means " for all " and the symbol " ∃ means "Exist".

1. **All siblings in my family there is a father.**

- ∀x sibling(x) –> ∃ z father – of(z, x)

2. **All siblings in my family there is a mother.**

- ∀x sibling(x) –> ∃ z mother – of(z, x)

3. **All siblings in my family there exist cousins.**

- ∀x, y sibling – of(x, y) –> ∃ z cousin – of(z, x) ∧ cousin – of(z, y)

4. **All grandchild's in my family there exist at least one aunt.**

- ∀x grandchild(x) –> ∃ z aunty – of(z, x)

5. **All grandchild's in my family there is exist least one uncle.**

- ∀x grandchild(x) –> ∃ z uncle – of(z, x)

6. **All sisters in my family there is a brother.**

- ∀x sister-of(x) –> ∃ z brother – of(z, x)

7. **All brothers in my family there is a sister.**

- ∀x brother – of(x) –> ∃ z sister – of(z, x)

8. **All spouses in my family there are children.**

- ∀x, y spouse – of(x, y) –> ∃ z child(z) ∧ child-of(z, x) ∧ child-of(z, y)

SATISFIABILITY

The concept of satisfiability defines that a formula in the logic Language is satisfiable if there is a Model and an interpretation. Now, we are going to see the satisfiability for 2 of the previous axioms. To do that we have to prove that the interpretation of the object is part of the model.

1. Let's first start with axiom that "for all brothers in my family there is a sister".

- ∀x **brother – of**(x) **–>** ∃ z **sister – of**(z, x)

The first step would be to retrieve each set from the model.

Brother – of = { (s1, d1), (s2, d2), (s3, d3) }

Sister – of = { (d1, s1), (d2, s2), (d3, s3) }

Now we can assign the semantic value of the object **s1** to **x** from the set brother-of and find where it provides interpretation of the Binary predicate **sister-of**. We can see there is an instance where a pair includes the semantic object **s1** in the interpretation of **sister-of** making the axiom satisfiable.

- ∀x **brother – of**(s1) **–>** ∃ z **sister – of**(d1, s1)

Now, in order to prove the validity of the axiom we must see all interpretations of the semantic set **brother-o**f and use it as input like we did before.

- ∀x **brother – of**(s2) **–>** ∃ z **sister – of**(d2, s2)
- ∀x **brother – of**(s3) **–>** ∃ z **sister – of**(d3, s3)

As shown above there is a pair for every element in the interpretation of **sister-of** therefore the axiom is valid.

2. The second axiom is that "for all siblings in my family there is a father" :

- ∀x **sibling**(x) **–>** ∃ z **father – of**(z, x)

The first step would be to retrieve each set from the model.

sibling = { s1, s2, s3, d1, d2, d3 }

father-of = { (gf, s1), (gf, d1), (s1, s2), (s1, d2), (h, s3), (h,d3) }

Now we can assign the semantic value of the object **s1** to **x** from the set **sibling** and find where it provides interpretation of the unary function **father-of**. We can see there is an instance where a pair includes the semantic object **s1** in the interpretation of **father-of** making the axiom satisfiable.

- ∀x **sibling**(s1) **–>** ∃ z **father – of**(gf, s1)

As shown above there is a pair for every element in the interpretation of **father-of** therefore the axiom is valid.

Now, to prove the validity of the axiom we must see all interpretations of the semantic set **sibling**. and use it as input like we did before.

- ∀x **sibling**(s2) **–>** ∃ z **father – of**(s1, s2)
- ∀x **sibling**(s3) **–>** ∃ z **father – of**(h, s3)
- ∀x **sibling**(d1) **–>** ∃ z **father – of**(gf, d1)
- ∀x **sibling**(d2) **–>** ∃ z **father – of**(s1, d2)
- ∀x **sibling**(d3) **–>** ∃ z **father – of**(h, d3)

INFERENCE RULES

Inference rules are functions that take expressions and produce other expressions. If the rule of inference takes an expression with a true value, the result of applying the rule will produce true if the rule is truth preserving. Truth preserving rules are also called deductive rules. These rules are based on premises from our theory and provide conclusions. If the example below we are going to show through different inference rules that :

All sons have father. Some father has a sister who is an auntie.

therefore :

some son has an auntie.

1.	For all sons there is father.	∀x son(x) –> ∃ z father–of(z, x)	Premise. (given)
2.	From our model we can establish that Clark is a son.	son(clark) –> ∃ z father–of(z, clark)	Universal Instantiation from (1).
3.	We can retrieve the premise.	Son(clark)	Premise.
4.	If clark is a son then there is some father of clark and if we know that clark is a son it's true, then we can say there is some father of clark.	∃ z father–of(z, clark)	Modus Ponens from (2) and (3).
5.	From our model we can establish the father as Wilson.	father–of(Wilson, clark)	Existential Instantiation from (4).
6.	For some father there are sister who are auntie.	∃x father-of(x,y) –> ∃z Sister-of(z, x) –> Auntie-of(z,y)	Premise (given)

7.	From our model we can establish the father as Wilson the sister and auntie as Nancy and the son as clark.	father-of(Wilson,clark) –> Sister-of(Nancy, Wilson) –> Auntie-of(Nancy,clark)	Existential Instantiation from (6).
8.	We can retrieve the premise.	father–of(Wilson, clark)	Premise
9.	If Wilson is the father of clark then there is Nancy the sister of Wilson and the auntie of clark. If we know that Wilson is the father of Clark is true, then we can say that Nancy is the sister of Wilson and the Auntie of clark.	Sister-of(Nancy, Wilson) ^ Auntie-of(Nancy,clark)	Modus Ponens from (7) and (8)
10.	If Nancy is both a sister and an auntie, then we can say that Nancy is a Sister.	Sister-of(Nancy, Wilson)	Simplification from (9).
11.	If Nancy is both a sister and an auntie, then we can say that Nancy is an Auntie.	Auntie-of(Nancy,clark)	Simplification from (9).
12.	Clark is a son and Nancy are an Auntie.	Son(clark) ∧ Auntie-of(Nancy,clark)	Conjunction from (3) and (9).
13.	There we have shown that if clark is a son then there is some auntie.	∃x Son(x) –>∃z Auntie-of(x,z)	Existential Generalisation from (12).

By using our theory and proving that Clark is a son, Wilson is a father and Nancy is a sister and Auntie we were able to generalize and say that in our model some son has an auntie.

2. All spouses in my family there are children :

1.∀x, y **spouse – of**(x, y) **–>** ∃ z **child**(z)

∧ **child-of**(z, x) ∧ **child-of**(z, y) - Premise

2. spouse – of(Jenny, Wilson) **–>** ∃ z **child**(z)

∧ **child-of**(z, Jenny) ∧ **child-of**(z, Wilson) - 1, Universal Instantiation

spouse – of(Jenny, Wilson) - premise

3. ∃ z **child**(z) ∧ **child-of**(z, Jenny) ∧ **child-of**(z, Wilson) - 2,3 Modes Ponens

(@child,Jenny) ∧ **child-of**(@child, Wilson) - 4, Existential Instantiation

TASK 4

HORN CLAUSE THEORY

Convert your theory into Horn-clause theory

1. All siblings in my family there is a father

∀x sibling(x) –> ∃ z father – of(z, x)

Eliminate implication

∀x ¬sibling(x) V ∃ z father – of(z, x)

Eliminate dependent existential quantifiers

∀x ¬sibling(x) V father – of(@f(x), x)

Drop Universal quantifiers

¬sibling(x) V father – of(@f(x), x)

CNF

¬sibling(x) V father – of(@f(x), x)

Clausal form

{ { ¬sibling(x) }, {father – of(@f(x), x) }}

2. All siblings in my family there is a mother

∀x sibling(x) –> ∃ z mother – of(z, x)

Eliminate implication

∀x¬sibling(x) V mother– of(z, x)

Eliminate dependent existential quantifiers

∀x ¬sibling(x) V mother– of(@f(x), x)

Drop Universal quantifiers

¬sibling(x) V mother– of(@f(x), x)

CNF

¬sibling(x) V mother– of(@f(x), x)

Clausal Form

{ { ¬sibling(x) }, {mother – of(@f(x), x) }}

3. All siblings in my family there exist cousins

∀x, y sibling – of(x, y) –> ∃ z cousin – of(z, x) ∧ cousin – of(z, y)

Eliminate implication

∀x, y ¬sibling – of(x, y) V ∃ z cousin – of(z, x) ∧ cousin – of(z, y)

Eliminate dependent existential quantifiers

∀x, y ¬sibling – of(x, y) V cousin – of(@f(x, y), x) ∧ cousin – of(@f(x, y), y)

Drop Universal quantifiers

¬sibling-of(x, y) V cousin – of(@f(x, y), x) ∧ cousin – of(@f(x, y), y)

Distribute Disjunction over the conjunction

¬sibling-of(x, y) V cousin – of(@f(x, y), x) ∧ ¬sibling – of(x, y) V cousin – of(@f(x, y), y)

CNF

¬sibling-of(x, y) V cousin – of(@f(x, y), x) ∧ ¬sibling – of(x, y) V cousin – of(@f(x, y), y)

Clausal Form

{ {¬sibling-of(x, y) }, { cousin – of(@f(x, y), x) }, { cousin – of(@f(x, y), y }}

4. All grandchild's in my family there exist at least one aunt

∀x grandchild(x) –> ∃ z aunty – of(z, x)

Eliminate implication

∀x ¬grandchild (x) V ∃ z aunty – of (z, x)

Eliminate dependent existential quantifiers

∀x ¬grandchild (x) V aunty – of (@f(x), x)

Drop quantifiers

¬grandchild (x) V aunty – of(@f(x), x)

CNF

¬grandchild (x) V aunty – of(@f(x), x)

Clausal Form

{{¬grandchild (x) }, { aunty – of(@f(x), x) }

5. All grandchild’s in my family there is exist least one uncle

∀x grandchild(x) –> ∃ z uncle – of(z, x)

Eliminate implication

∀x ¬grandchild (x) V ∃ z uncle – of(z, x)

Eliminate dependent existential quantifiers

∀x ¬grandchild (x) V uncle – of(@f(x), x)

Drop quantifiers

¬grandchild (x) V uncle – of(@f(x), x)

CNF

¬grandchild (x) V uncle – of(@f(x), x)

Clausal Form

{{¬grandchild (x) }, { uncle – of(@f(x), x) }}

6. All sisters in my family there is a brother

∀x sister-of(x) –> ∃ z brother – of(z, x)

Eliminate implication

∀x ¬sister – of(x) V ∃ z brother – of(z, x)

Eliminate dependent existential quantifiers

∀x ¬sister – of(x) V brother – of(@f(x), x)

Drop quantifiers

¬sister – of(x) V brother – of(@f(x), x)

CNF

¬sister – of(x) V brother – of(@f(x), x)

Clausal Form

{{¬sister – of(x) }, { brother – of(@f(x), x) }}

7. All brothers in my family there is a sister

∀x brother – of(x) –> ∃ z sister – of(z, x)

Eliminate implication

∀x ¬brother – of(x) V ∃ z sister – of(z, x)

Eliminate dependent existential quantifiers

∀x ¬brother – of(x) V sister – of(@f(x), x)

Drop quantifiers

¬brother – of(x) V sister – of(@f(x), x)

CNF

¬brother – of(x) V sister – of(@f(x), x)

Clausal Form

{{¬brother – of(x)}, {sister – of(@f(x), x)}}

8. All spouses in my family there are children

∀x, y spouse – of(x, y) –> ∃ z child(z) ∧ child-of(z, x) ∧ child-of(z, y)

Eliminate implication

∀x, y ¬spouse – of(x, y) V ∃ z child(z) ∧ child – of(z, x) ∧ child – of(z, y)

Eliminate dependent existential quantifiers

∀x, y ¬spouse – of(x, y) V child(@f(x, y)) ∧ child – of(@f(x), x) ∧ child – of(@f(y), y)

Drop quantifiers

¬spouse – of(x, y) V child(@f(x, y)) ∧ child – of(@f(x), x) ∧ child – of(@f(y), y)

Distribute Disjunction over the conjunction

¬spouse – of(x, y) V child(@f(x, y)) ∧ ¬spouse – of(x, y) V child – of(@f(x), x) ∧

¬spouse – of(x, y) V child – of(@f(y), y)

{{¬sister – of(x) }, { brother – of(@f(x), x) }}

GROUP WORK AND MEETING LOGS

MEETING LOGS

During our time completing our task, we continued to communicate we had already completed for the previous coursework 1. This was to communicate and discuss points of interest and meeting planning. Throughout our coursework we would regularly converse and update one another on specific parts within the task each of us had completed or refined.

Our report was stored within google docs, this was to allow everyone within the group a chance to edit and add relevant information, all in real time. Within meetings, a team member would explain what section they had added whilst sharing the google doc. This was to ensure all team members had a clear understanding on work completed and which sections were left to be completed.

After all relevant sections were completed and discussed, the report was finalised within a word document and formatted correctly. The final report draft was sent within the WhatsApp group chat; therefore, all team members were kept up to date and feedback could be given regarding overall quality and structure.

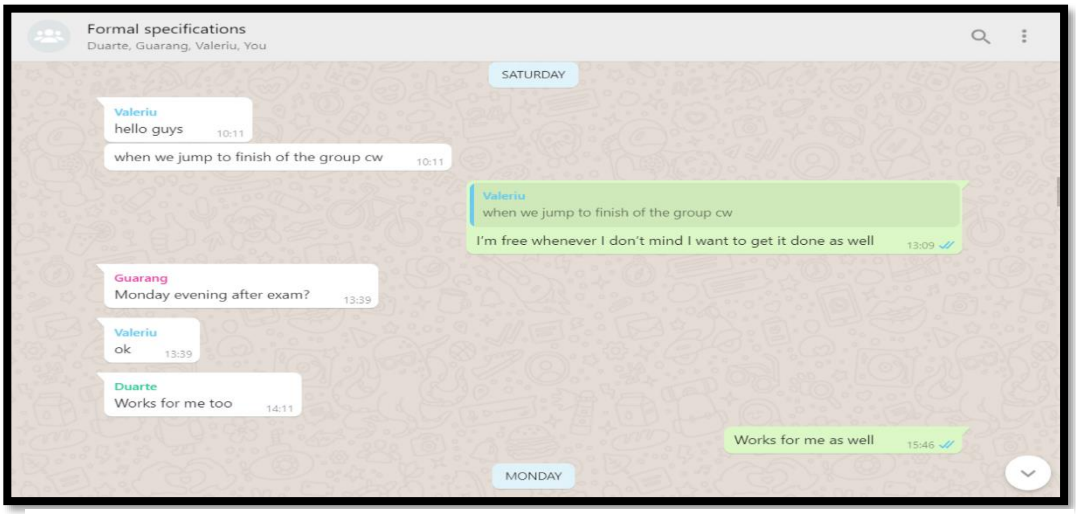

Figure 2 Meeting Log 1

This screenshot shows the confirmed time to join the meeting to discuss the group assignment. All group members would agree to a specific time, if there was anyone unable to attend, we would reschedule the meeting to the next most suitable time.

Figure 3 Meeting Log 2

This screenshot is also to show team members confirming their attendance for a meeting, also planning out when everyone was free. Meetings were planned around consistently making sure everyone's availability, to ensure everyone had time for other responsibilities.

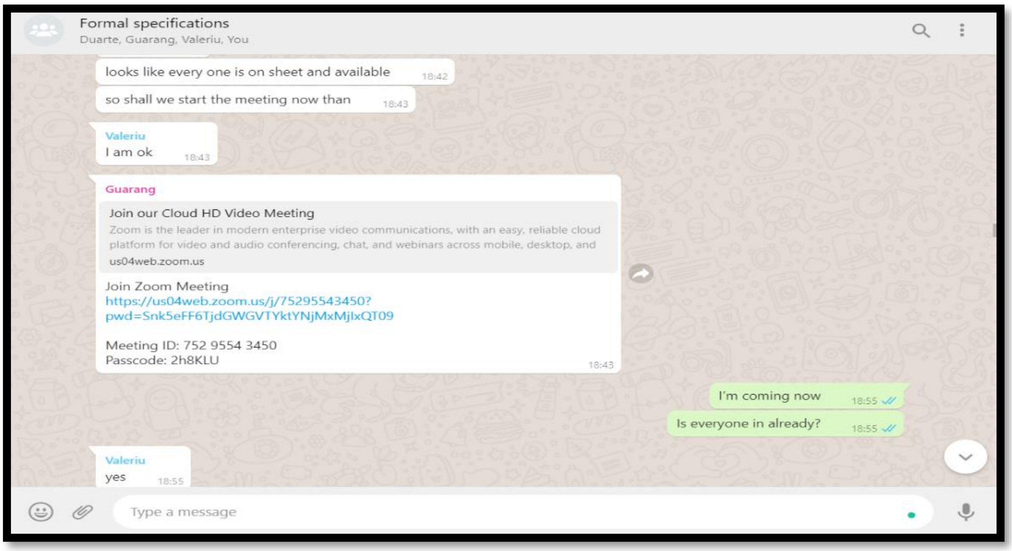

Figure 4 Meeting Log 3

This screenshot is to show a meeting taking place within the WhatsApp group chat. A team member would create a zoom meeting, where everyone joined and discussed the next steps within the coursework's progression.

ZOOM MEETINGS

Within this section, we have included screenshots within our meetings. All meetings were held on zoom, where we would share our screens among each other and discuss all current work produced.

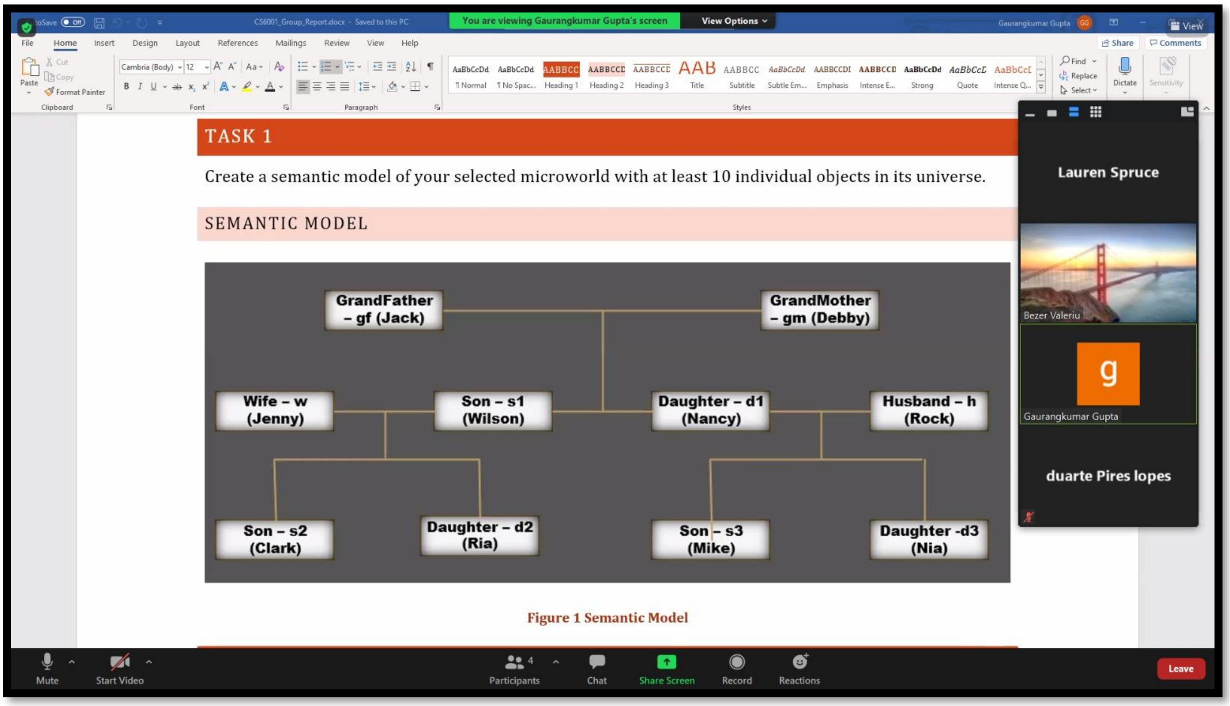

Figure 5 Meeting Log 4

This screenshot shows use discussing our semantic model. Within this meeting all team members discussed every section of the report giving feedback and improvements to one another.

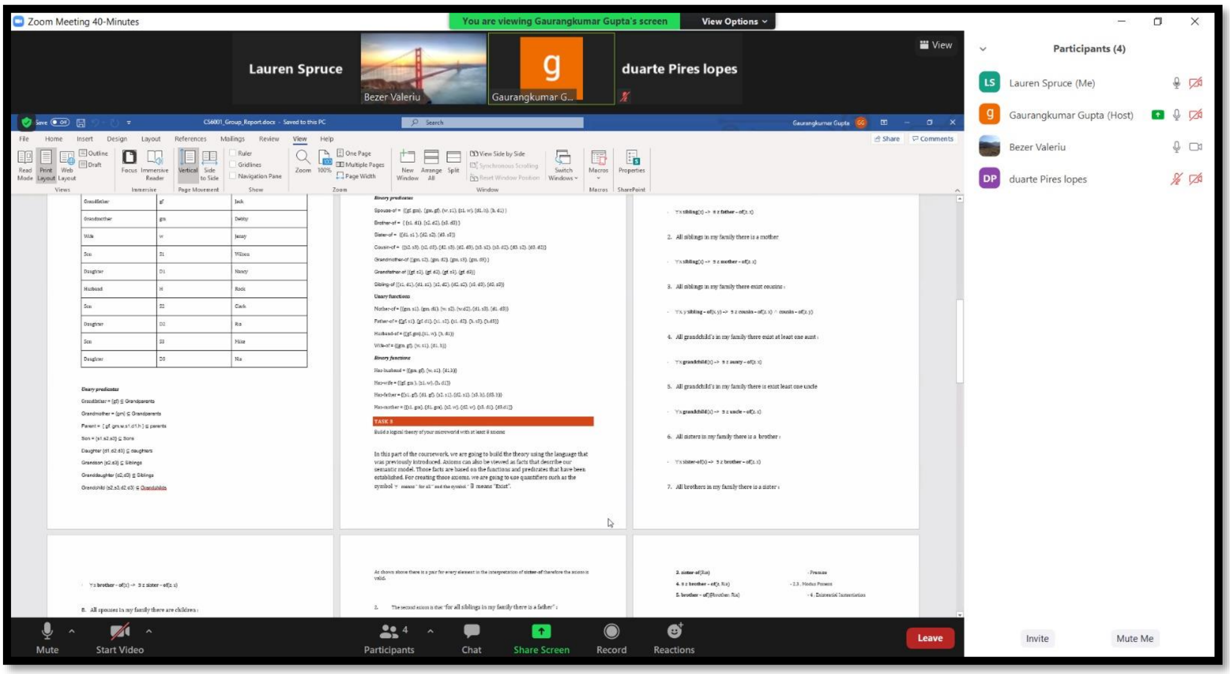

Figure 6 Meeting Log 5

Here is another meeting, within this meeting we were checking our work against the coursework brief, making sure that we covered all the deliverables and sections our group coursework required. All team members gave their opinions on what needed to be improved within our report draft on the google document.

Here is summary table, specifying the meetings taken place, the agenda of the meeting and achievements that had been made and the general discussion that took place.

Table 2 Meeting Log

Meeting	Agenda	Achievements	Discussion
1.	• Choosing a microworld • Creating semantic model with 10 objects	• Task assignment • Distributing tasks among team members • Creating WhatsApp group and Google Document • Choosing Microworld topic	• Within this meeting we decided which team members would be assigned which parts of the coursework and discuss the methodology used within achieving desired results of the coursework brief and goals.
2.	• Specifying vocabulary, 6 individual object names, 4 unary predicates and 4 binary predicates.	• Structuring draft report • Group discussion of object names and predicates	• Within our second meeting we discussed more into detail about specific parts within the coursework, names of individual objects and discussing predicates. We discussed solutions to Section A.
3.	• Creating 8 axioms within microworld • Satisfiability of 2 axioms, using 3 inference rules to derive conclusions	• Discussing 3 rules for satisfiability • Deriving conclusions from 3 rules • Continuing report structuring	• Within the third meeting we discussed the entirety of Section B, and discussed the results achieved from previous meeting and made slight improvements.
4.	• Converting theory into Horn clause theory	• Picking DL or Horn Clause Theory for conversion. • Conversion into Horn Clause Theory	• Within our fourth meeting we started progression on Section C, also revisited all other Sections to ensure they were finalised and correctly implemented.
5.	• Viva Preparation • Finalising Report • Requirements against achievements	• Looking back within all sections • Polishing up report • Discussion of final improvements.	• Finally, we revisited all sections within the coursework, and revised all of the report, converting the google doc into a final word document and all formatting was completed before submission.

CONCLUSION

This document demonstrates a good understanding of the semantic model based on the logical theory of the chosen microworld “My Family” and how to implement that in the form of ontology in protégé software. Further this exercised developed the members of the group on classes, objects and properties including OWL axioms and the relations to all individuals.

What went well was that all team members engaged consistently throughout the whole project, everybody successfully implemented their own individual contribution, the WhatsApp group improved our group activity a lot as everyone could consistently keep up to date with the current work production. The use of google docs also improved the efficiency of our workflow, as everything stayed up to date in real time. Zoom meetings helped us a lot as we were able to share content with one another and discussion important issues that needed resolving in our coursework.

A part that we struggled on was creating the inference rules for our Microworld, we overcame this by asking for advice from our lecturer and workshop tutor, therefore we created many drafts and finalised the correct rules and successfully implemented within our Section B.

Overall, the group really had a good understanding to work on the individual work task of this project. We found the individual task to be enjoyable, we as a group thank the university and the lecturer to provide this opportunity.

Bibliography

- Grigoris Antoniou et al. (2012) A Semantic Web Primer. Cambridge, MA: The MIT Press (Cooperative Information Systems). Available at: http://0-search.ebscohost.com.emu.londonmet.ac.uk/login.aspx?direct=true&db=nlebk&AN=480970&site=ehost-live (Accessed: 21 May 2021).
- Szeredi, P., Lukácsy, G., Benkő, T. and Nagy, Z., 2014. The semantic web explained. Cambridge: Cambridge University Press.
- Huth, M. and Ryan, M., 2018. Logic in computer science. Cambridge: Cambridge University Press.