On Structural Patterns of Resource Description Framework Models for One Health

Jamie Grace B. Bautista
Service Science and Software Engineering Laboratory
Department of Computer Science
University of the Philippines Diliman
jbbautista7@up.edu.ph

Angelika Juliah S. Galang
Service Science and Software Engineering Laboratory
Department of Computer Science
University of the Philippines Diliman
asgalang1@up.edu.ph

Jasmine A. Malinao
Headstart Business Solutions, Inc.
jamalinao@headstartbizsolutions.com

Jaime D.L. Caro
Department of Computer Science
University of the Philippines Diliman
jdlcaro@up.edu.ph

Michael L. Tee
University of the Philippines Manila
mltee@up.edu.ph

ABSTRACT

The One Health framework is a multidisciplinary approach to evaluate human, animal and environmental health together. It is applied to different kinds of projects by countries to solve national problems such as environmental problems and epidemiological outbreaks. Since the data from the One Health framework are heterogeneous and unstructured, Dela Rosa et al. created Resource Description Framework (RDF) models for the said framework in the Philippine setting to interpret the data into a much more comprehensive set of information. Although their models provided a means to integrate the multidisciplinary, heterogeneous data sets via transitive associations, there was still a lack of concrete, domain-consistent information that validated these associations.

This research focuses on using Dela Rosa et al.‘'s generated RDF models and convert it to a Relational Database (RDB). The advantage of converting the model into a RDB is that the data are stored and presented in a tabular format, organized in rows and columns with one record per row. This makes massive data sets much easier to manage and thus increases efficiency.

This study aims to refine the generated RDF models and data set with respect to intersecting transitive associations, convert the RDF models into a RDB, and test the predictive capabilities of the output of the mapping.

Keywords: One Health, Resource Description Framework, Relational Database, Conversion, Prediction

1. INTRODUCTION

One Health is a holistic and collaborative approach to understand and solve issues concerning the health of humans, animals, and the environment. [1] The World Health Organization (WHO) [2] defines it as the "approach to designing and implementing programmes, policies, legislation and research in which multiple sectors communicate and work together to achieve better public health outcomes."

The Resource Description Framework (RDF) is a framework for expressing information about the resources that utilizes the RDF Triple which enables a simple way of linking data together in the form of *subject*, *predicate*, and *object* [3]. RDF graphs are made of interconnected RDF Triples and RDF models are made of RDF interconnected graphs. Since One Health collects massive amounts of data, Dela Rosa et al. [4] created RDF models for the One Health framework in the Philippine setting to interpret these data into a much more comprehensive set of information. While their models provided a means to integrate multidisciplinary heterogeneous data sets, there are instances where the ***structural patterns*** in the RDF models are not verified by experts in the knowledge domain. There are components that are still needed to be refined such as the *transitive associations* that link two resources. There is also a difficulty in organizing the data sets and their relationships with the quantity of data it represents considering the size of the RDF models.

With challenges presented from Dela Rosa et al., this research aims to convert the RDF models to a Relational Database (RDB) and refine the model and the data. Additionally, this study also looks on determining the potential predictive capabilities of the RDF models that can be useful for medical insights.

1.1 RDF Models for One Health

RDF is a framework that represents different *resources* in the Web. Resources can vary from documents to people and to even other complex concepts. [3] Its core structure, as mentioned, is a triple data model that links these said resources together. It is formally defined as follows:

Definition 1. RDF Triple. [4] An RDF Triple is a triple of the form (s,p,o) where

- s = subject
- p = predicate
- o = object

s and o are the resources and p is the link that signifies the relationship in between these two resources.

Figure 1. An RDF Triple

Definition 2. RDF Triplestore. [4] An RDF triplestore is a database that can store RDF Triples.

Definition 3. RDF Graph. [4] An RDF Graph is a set of RDF Triples.

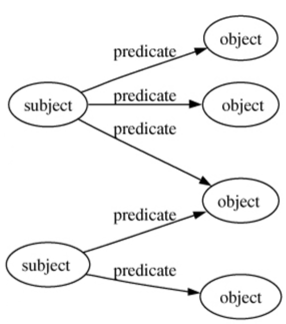


Figure 2. An RDF Graph

Dela Rosa et al. made use of publicly available forms, reports, and information profiles *(see Appendix 1)* from Philippine Integrated Disease Surveillance and Response (PIDSR) [9] [10], manually looked for as many helpful data they could, and translated it into RDF triples. Below is an illustration:

1	SUBJECT	PREDICATE	OBJECT
2	Dengue	has	Dengue ICD 10 Code
3	Dengue ICD 10 Code	is a type of	ICD 10 Code
4	Dengue ICD 10 Code	is	A90
5	Dengue ICD 10 Code	is	A91
6	Dengue	has	Dengue Description
7	Dengue Cause	is a part of	Dengue Description
8	Dengue	is synonymous tc	Dengue fever
9	Dengue hemorrhagic fever	is a type of	Dengue fever
10	Dengue	has	Dengue Cause
11	Dengue Virus	is a part of	Dengue Cause
12	Dengue Virus Type 1	is a type of	Dengue Virus
13	Dengue Virus Type 2	is a type of	Dengue Virus
14	Dengue Virus Type 3	is a type of	Dengue Virus
15	Dengue Virus Type 4	is a type of	Dengue Virus
16	Aedes Mosquito	is a type of	Mosquito
17	Female Aedes Mosquito	is a type of	Aedes Mosquito
18	Aedes Mosquito	can have	Dengue Virus
19	Female Aedes Mosquito	is a part of	Dengue Cause
20	Humans	can have	Dengue Virus

Figure 3. Screenshot of part of Dela Rosa et al.'s Excel Spreadsheet for the PIDSR Case Investigation Form of Dengue data set

They defined and constructed the predicates *(see Appendix 2)* used, for example in Fig. 3, according to different types as seen in Fig. 4. There are four (4) different types of predicates namely:

- Association Type
- Field Type
- Requirement Type
- Collection Type

and each type has sub-types of its own.

ASSOCIATION TYPES	Generalization/Specialization: *is a type of* Aggregation: *is a part of* Composition: *has* Dependency: *precedes* Equivalence: *is also known as, is synonymous to*
FIELD TYPES	Category: *is a type of* List: *is a part of* Unitary Value: *is*
REQUIREMENT TYPES	Required: *requires* Not Required: *can have*
COLLECTION TYPE	*is collected by* *is collected from* *is collected using* *is collected on* *is collected*

Figure 4. Dela Rosa et al's manually defined Predicates and its Types

A special type of predicate, *is related to*, is also defined and used for merging. To understand this predicate, the idea of *transitive association* is firstly introduced.

Definition 4. Transitive Association. [4] Two resources r_1 and r_2 are transitively associated if there is a directed path from r_1 to r_2 or from r_2 to r_1.

Definition 5. *is related to.* [4] Given p is an element of RDF1, if there is a transitive association from p to x and $x = y$, then p **is related to** y. p is obtained by selecting the vertex with the largest value from the matrix generated using Johnson's algorithm [13].

By merging, interconnecting, and finding *is related to* relationships between RDF graphs, they were able to create RDF models that showed ***structural patterns***. An example can be observed in Fig. 5.

Definition 6. Interface Nodes. [4] For every p and x in the RDF where the predicate is either of *Equivalence* (see *ASSOCIATION TYPES* in Fig. 4) or the special predicate *is related to*, we call p and x as interface nodes. Similarly, x and y are also called interface nodes.

Definition 7. ***Structural Pattern.*** A Structural Pattern is created when interface nodes form a relationship with other set of nodes.

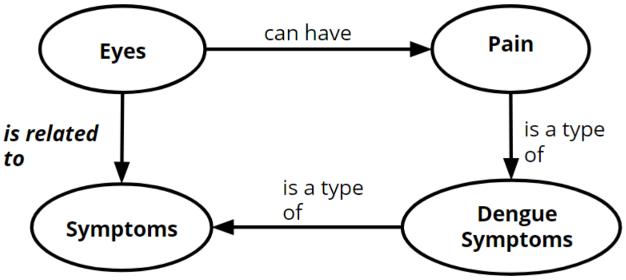


Figure 5. An example of a Structural Pattern

However, there has been a difficulty in representing these RDF Models into an RDB. While their output already provided helpful insights, it is known that there are more developed tools that can be utilized in RDBs than in RDF models [15].

1.2 Related Works on RDF to RDB Conversion

- *A Path-based Relational RDF Database*
  Matono et al. proposed a path-based relational RDF database. Their proposition not only converted RDF models to RDB, but it also significantly increased efficiency of queries time complexity wise. They stored RDF data based on path expressions in a database and this led to a greatly reduced number of join operations. [6] Their study can also make distinctions between schema information and instance data which translates to the level of the model and level of the data respectively in our study. Their research can be a basis on the conversion of the RDF models from Dela Rosa et al.'s work to a relational database. The only downside of Matono et al.'s path-based RDB is its performance is not yet tested on a larger amount of RDF data.

- *Tools For Conversion*
  RDF2RDB is a tool created Michael Brunnbauer to convert RDF data in several formats to RDB using Python and MySQL. [17] RDF2X is another tool for converting big RDF datasets to RDB created by Merck using Java and PostgreSQL. [18] The tool can also convert to CSV, JSON and ElasticSearch. While these tools presented challenged such as naming of tables, mapping of components and preservation of relationships, they can be a starting point and basis of creating programs converting RDF to RDB.

- *Extracting Entity Relationship Diagram (ERD) from English Sentences*
  An ERD is a graphical representation of any set of information system with different relationships from different data. Al-Btoush et al. created guidelines for extracting entity relationship diagrams from English sentences. This provided conventions on defining the mappings of the words to an ERD. [7] Since the predicate relationships of the structural patterns are similar to English sentences, this study can be a basis for setting the guidelines of the conversion.

2. METHODOLOGY

The data used in this research is focused on health related to vector-borne diseases commonly present in the Philippines: Dengue and Malaria. Vector-borne diseases are infections transmitted by the bite of infected arthropod species, such as mosquitoes, ticks, triatomine bugs, sandflies, and blackflies. [19] These diseases are very much widespread and in fact, the Western Pacific Region reported more than 375,000 suspected cases of dengue in 2016, of which the Philippines reported 176,411. [16]

The aforementioned PIDSR collects and reports health cases on these diseases through publicly available forms and information profiles. There are also dengue sample scenarios from CHED-PCARI illustrated through flowcharts and these are mainly used for the ongoing development of the One Health (OH) application.

The following are the data sets used in this study:

- *Publicly Available Forms and Documents*
    1. PIDSR Case Investigation Form for Malaria [9]
    2. PIDSR Case Report Form for Dengue [9]
    3. Information on Malaria [10]
    4. Information on Dengue [10]
- *Private Documents (Available upon request)*
    1. Triple Stores and RDF Graphs from Dela Rosa et al.
    2. CHED-PCARI Dengue Sample Scenarios / Flowchart for OH App

The manually defined predicates by Dela Rosa et al., as shown in Fig. 4, is also part of the input of this research.

2.1 Data Refinement

Dela Rosa et al. translated data from the available forms and documents *(see Appendix 1)* into RDF triples and listed them all down inside an Excel Spreadsheet. This was done using keywords from the datasets as subject or objects and matching them with the appropriate predicates. The OH App Flowchart containing data related to Dengue cases is converted to RDF Triples according to Dela Rosa et al.'s paper. The converted RDF triples are then appended to the Excel Spreadsheet containing the previous RDF triples from the Dengue accounts. The dataset containing the RDF triples were validated. The purpose of refining the data is to remove or change triples that are not medically accurate for a finer result. Refining the data can also mean adding new triples to accommodate the exiting triples from the dataset.

2.2 Model Refinement

The first step in refining the model is to recreate the structural patterns for the refined data. To do this, Dela Rosa et al.'s previous method is used and modified. (*Steps in getting interface nodes* [4])

- **Steps in creating structural patterns**
  * *Steps that are italized are the newly added ones*
    1. Input 2 RDF graphs $RDF1$ and $RDF2$
    2. Exact Matching: Get the list of common terms between the two RDF graphs
    3. *Store each subject/object and its corresponding predicate in a matrix*
    4. Generate distance matrix for $RDF1$ using Johnson's algorithm
    5. *Modify the predicate matrix following the distance matrix of $RDF1$*
    6. For each common term x:
        (a) Get list of terms ('dlist') with the largest value connected to x from the distance matrix
        (b) If dlist is not empty:
            – For each p in dlist, create the triple
            * **Subject:** p
            * **Predicate:** *is related to*
            * **Object:** x
            and export to a new file in CSV format
        (c) *Export each structural pattern in every is related to relationship generated to a text file*
    7. New file now contains triples with interface nodes of $RDF1$
    8. *Text file now contains structural patterns of $RDF1$*
    9. Repeat steps (3) to (6) using $RDF2$

After recreating the structural patterns, the relationships between interface nodes are then again validated. Upon validation, *One Health (OH) Patterns* can now be generated via user input.

Definition 8. One Health Pattern. A co-occurence of a set of nodes in the unified/integrated RDF models derived from the models of transitive associations.

- Steps in generating OH patterns via user input
    1. User inputs number of terms to generate OH patterns with
    2. User chooses which terms from the list given
    3. Program looks for all the OH patterns in the structural patterns text file in which chosen term/s can be observed
    4. Program outputs all OH patterns in which chosen term/s can be observed

The program is written in Python and works via command prompt/terminal as for now.

2.3 RDF to RDB Conversion

For the conversion of the RDF to RDB, each predicate from the model is converted to an Entity Relationship Diagram (ERD). The ERD can then be constructed and converted to a Relational Schema with the online tool *ERDPlus*. The purpose of converting the predicates individually is to have a convention when translating whole sets of structural patterns to an RDB. It is also a way to preserve the relationship of each triple in a pattern.

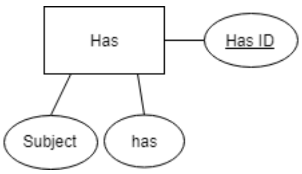
The tools available for us to use, RDF2RDB and RDF2X, presented challenges in our first testing for the conversion. RDF2RDB did not preserve the names of the tables in which the RDF were converted from. RDF2X, while able to convert the RDF with the naming of the tables and columns, did not preserve the relationships from the original RDF form. The tool converted the RDF without taking into account the meaning of the relationships to each other.

Since the tools upon testing presented challenges mentioned before, the predicates are manually converted to an Entity Relationship

Diagram by defining their components and how they will be mapped into the diagram.

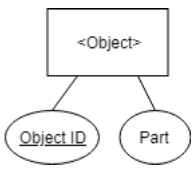
The following are the converted predicates with their equivalent tables.

- Has



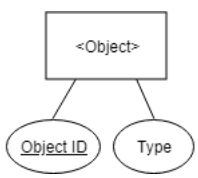
HAS		
Has_ID	has	subject

- Is a Part of



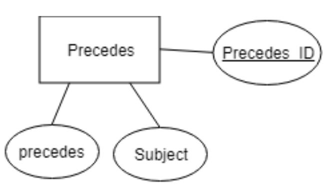
OBJECT (is a part of)	
Object_ID	Part

- Is a Type of



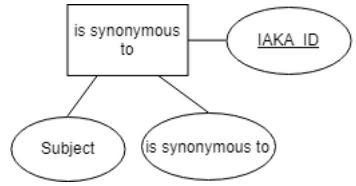
OBJECT (is a type of)	
Object_ID	Type

- Precedes



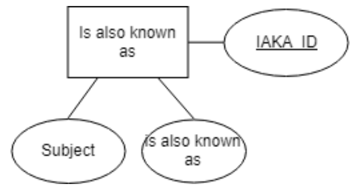
PRECEDES		
Precedes_ID	subject	precedes

- Is Synonymous To



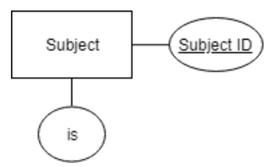
IS SYNONYMOUS TO		
IST_ID	subject	is synonymous to

- Is Also Known As



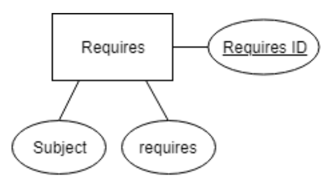
IS ALSO KNOWN AS		
IAKA_ID	subject	is also known as

- Is



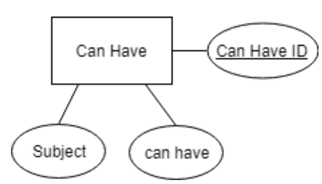
SUBJECT		
Subject_ID	subject	is

- Requires



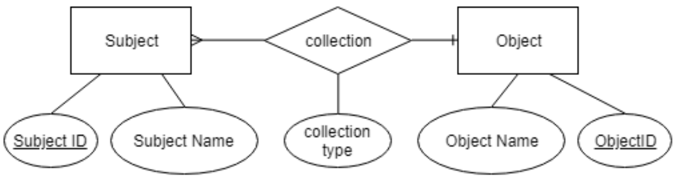
REQUIRES		
Requires_ID	subject	requires

- Can Have



CAN HAVE		
Can_Have_ID	subject	can have

- Collection Types



SUBJECT	
Subject_ID	subject_name

COLLECTION		
Subject_ID	Object_ID	Collection Type

OBJECT	
Object_ID	object_name

2.4 Prediction of One Health Patterns

The generated OH patterns via user input already provide a basis for the possible predictive capabilities of the model. Improving on this could lead to significant changes on how to assess and read Dengue and Malaria cases; moreover, on other diseases as well.

The conversion of OH patterns to tables can be used for prediction analysis. It is done by filling the tables with sample data that can simulate the prediction of the patterns towards a certain query.

3. RESULTS AND DISCUSSIONS

3.1 Refined Data

After consulting with medical experts about the RDF Triples, the data is validated and refined. Sample unrefined data for the Dengue Account can be seen in Fig. 6. The new text as seen in Fig. *7*, on the other hand, shows medically validated triples.

1	SUBJECT	PREDICATE	OBJECT
121	Kidneys	can have	Kidney impairment
122	Liver impairment	is a type of	Severe organ impairment
123	CNS impairment	is a type of	Severe organ impairment
124	Heart impairment	is a type of	Severe organ impairment
125	Kidney impairment	is a type of	Severe organ impairment
126	Liver impairment	is	AST greater than or equal to 1000
127	Liver impairment	is	ALT greater than or equal to 1000
128	Seizures	is a type of	CNS impairment
129	Impaired consciousness	is a type of	CNS impairment
130	Myocarditis	is a type of	Heart impairment
131	Renal failure	is a type of	Kidney impairment
132	Dengue	has	Laboratory Confirmation methods
133	Isolation	is a type of	Laboratory Confirmation methods
134	Serum	is a part of	Blood
135	Plasma	is a part of	Blood

Figure 6. Dela Rosa et al.'s Triples for the Dengue Account

1	SUBJECT	PREDICATE	OBJECT
121	Kidneys	can have	Kidney impairment
122	Liver Failure	is a type of	Severe organ impairment
123	Encephalopathy	is a type of	Severe organ impairment
124	Heart Failure	is a type of	Severe organ impairment
125	Congestion	is a type of	Severe organ impairment
126	Myocarditis	is a type of	Severe organ impairment
127	Cardiomyopathy	is a type of	Severe organ impairment
128	Liver Failure	is a type of	Organ failure
129	Encephalopathy	is a type of	Organ failure
130	Heart Failure	is a type of	Organ failure
131	Congestion	is a type of	Organ failure
132	Myocarditis	is a type of	Organ failure
133	Cardiomyopathy	is a type of	Organ failure
134	Liver impairment	is	AST twice the normal
135	Liver impairment	is	ALT twice the normal
136	Seizures	is a type of	CNS impairment
137	Impaired consciousness	is a type of	CNS impairment

Figure 7. Refined Triples for the Dengue Account

3.2 Refined Model

Fig. 10 shows an example of one outputted structural pattern. A sample of generating OH patterns via user input can be seen in Fig. 8 and Fig. 9.

```
Enter filename of RDF without extension [ex. Dengue]: Dengue
[1]: Dengue
[2]: Dengue ICD 10 Code
[3]: ICD 10 Code
[4]: Dengue Importance of Surveillance
[5]: Importance of Surveillance
[6]: Dengue hemorrhagic fever
[7]: Dengue fever
[8]: Dengue fever incubation period
[9]: Incubation period
[10]: Plant axils
[11]: Water storage
[12]: Stagnant Water
[13]: Mosquito Breeding Sites
[14]: Mosquito
[15]: Cisterns
[16]: Flower vases
[17]: Tanks
[18]: Backyard litter
[19]: Myalgia
[20]: Pain
[21]: Dengue Symptoms
[22]: Symptoms
[23]: Petecheal
[24]: Rash
[25]: Probable Dengue Case
[26]: Suspected Dengue Case
[27]: Suspected Dengue Case Symptoms
[28]: Eyes
[29]: Retro orbital pain
[30]: Arthralgia
[31]: Herman's sign
[32]: Muscle Pain
[33]: Joint Pain
[34]: Body malaise
[35]: Headache
Enter number of nodes to be inputted [1,2,3...35]: 2
```

Figure 8. Generating OH Patterns via User Input

- Summary of example given
  - User inputs number of terms to be chosen: 2
  - User chooses terms: 20 (**Pain**), 21 (**Dengue Symptoms**)
  - Program outputs all patterns in which **Pain** and **Dengue Symptoms** can be observed

```
Enter number of nodes to be inputted [1,2,3...35]: 2
Choose an interface node to get patterns of (via number [#]): 20
Choose an interface node to get patterns of (via number [#]): 21
Pattern #1:
Myalgia is a type of Pain -> Pain is a type of Dengue Symptoms -> Dengue Symptoms is a type of Symptoms
Pattern #2:
Eyes can have Pain -> Pain is a type of Dengue Symptoms -> Dengue Symptoms is a type of Symptoms
Pattern #3:
Retro orbital pain is a type of Pain -> Pain is a type of Dengue Symptoms -> Dengue Symptoms is a type of Symptoms
Pattern #4:
Arthralgia is a type of Pain -> Pain is a type of Dengue Symptoms -> Dengue Symptoms is a type of Symptoms
Pattern #5:
Muscle Pain is a type of Pain -> Pain is a type of Dengue Symptoms -> Dengue Symptoms is a type of Symptoms
Pattern #6:
Joint Pain is a type of Pain -> Pain is a type of Dengue Symptoms -> Dengue Symptoms is a type of Symptoms
Pattern #7:
Body malaise is a type of Pain -> Pain is a type of Dengue Symptoms -> Dengue Symptoms is a type of Symptoms
Pattern #8:
Headache is a type of Pain -> Pain is a type of Dengue Symptoms -> Dengue Symptoms is a type of Symptoms
[PATTERNS GENERATED]: 8
Generate another pattern? [y/n]:
```

Figure 9. Generating OH Patterns via User Input

Fig. 10 is a part of the multiple patterns of the output of the program. These provide numerous significant insights about Dengue and Malaria that can benefit the current status of Philippine health on these vector-borne diseases. Any user with any data can utilize and observe different possible instances on the interplay of humans, animals, and the environment relating to Dengue and Malaria.

3.3 Converted RDF Models to RDB

Using the output of the refined data and model, the OH patterns that are generated can be converted to an ERD.

Eyes	can-have	Pain	is-a-type-of	Dengue-Symptoms
	Dengue-Symptoms		is-a-type-of	Symptoms

Figure 10. A pattern generated from the refined data and model

The path generated shown by Fig. 10 is an example of an *is related to* predicate and the relationships and predicates in between the first node and last node.

The ERD is then constructed using the online tool ERDPlus.

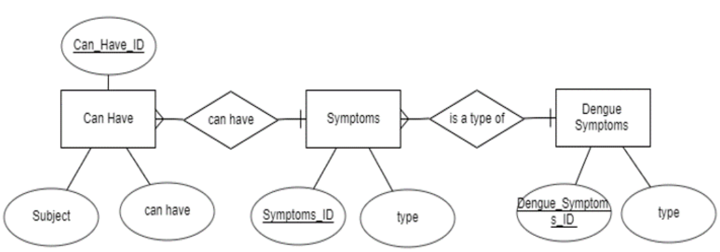


Figure 11. Entity Relationship Diagram of Figure 10

The RDB Schema is also converted using ERDPlus.

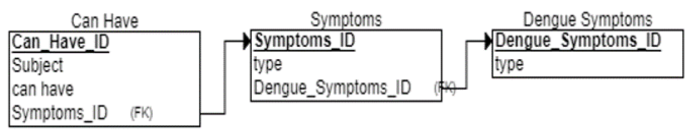


Figure 12. RDB Schema of Figure 11

Using the RDB schema in Fig. 12, it can now be easily converted to a table.

3.4 Tested Conversion and Visualization of Tables

The application of the conversion results to tables with relations. For example, the tables below came from the structural pattern from Fig. 5 with additional data placed in its rows.

CAN HAVE			
Can_Have_ID	**Subject**	**can have**	**Symptoms_ID**
CH1	Eyes	Symptoms	S1
CH2	HH Member	Symptoms	S2

Table 1: Can Have.

SYMPTOMS			
Symptoms_ID	**Type**	**Dengue_Symptoms_ID**	**Malaria_Symptoms_ID**
S1	Dengue Symptoms	DS1	0
S2	Malaria Symptoms	0	MS1

Table 2: Is a type of.

DENGUE SYMPTOMS			
Dengue_Symptoms_ID	**Type**	**Pain_ID**	**Rash_ID**
DS1	Pain	P4	0
DS2	Rash	0	R2
DS3	Heart Rate	0	0

Table 3: Is a type of.

MALARIA SYMPTOMS		
Malaria_Symptoms_ID	**Type**	**Pain_ID**
MS1	Pain	P2
MS2	Vomiting	0

Table 4: Is a type of.

PAIN	
Pain_ID	**Type**
P1	Body Pain
P2	Joint Pain
P3	Headache Pain
P4	Retro Orbital Pain
P5	Back Pain

Table 5: Is a type of.

RASH	
Pain_ID	**Type**
R1	Petecheal
R2	Herman's Sign

Table 6: Is a type of.

The values placed in the rows can be changed according to the type of data that is more fit to the field. For example, "Can_Have_ID" can be an integer instead of a string. The "0" value is placed in a row to represent a null value or something that does not exist in that row of data. In Table II, the first row that has Type "Dengue Symptoms" does not have any values for "Malaria_Symptoms_ID".

Simple SQL queries, such as select statements or join statements, will be able to filter data that a user wants to see.

4. CONCLUSION

The data and the RDF models are refined according to what is medically approved by an expert. Additionally, the models are improved by modifying the algorithm to taking the whole structural pattern from Dela Rosa et al.'s RDF graphs instead of only the *is related to* one. The modification is of storing the corresponding predicates of each subject or object and saving all structural patterns whenever an *is related to* relationship is established.

Users can now also look on helpful insights with any Dengue/-Malaria related data they have by generating One Health (OH) patterns through the algorithm provided.

Upon refinement of the triple stores and RDF models, the said models are converted into a relational database. Each OH pattern is manually converted to an Entity Relationship Diagram (ERD) basing on the predicate of each RDF triple in the pattern. Using the online tool *ERDPlus*, each ERD is then finally converted to a relational schema. This gave an advantage of having a structured data unlike the initial unstructured and heterogeneous one. The downside with the presented model is that it will store numerous tables, worst-case scenario, whenever the longest OH patterns are generated. However, having RDB as the model makes way for an easier management of massive data sets; thus, an increase in efficiency.

Moreover, a starting point was set in this study to establish and test the predictive capabilities of the RDB model using the OH patterns it represents.

5. RECOMMENDATIONS

5.1 Automated Conversion of Patterns

Currently, the conversion of the patterns are done manually by constructing the RDF predicates in the patterns and converting them to RDB schema using ERDPlus. Although, this provides a better translation of the RDF to RDB, this method would be considered as time consuming and tedious in an event that new patterns are introduced to the framework and database.

An algorithm that can automate the conversion of these patterns would enable a much more improved efficiency.

5.2 Prediction Analysis

With the tables from the converted RDB, filling the database with data is a starting point to be able to predict certain types of results. For example, filling in a patient's symptoms together with the environment they are living in will be able to determine results such as the type of illness they have or if the illness is a false alarm. This will help in having predictive analytics when using the database and in determining results based on the inputted data. This will be useful for people in the medical field and even to other One Health workers.

Acknowledgment

We extend our gratitude to Dr. Richelle Ann Juayong, Mr. Herbie Villafranca, and Ms. Jean Malolos for comments on this paper.

We would also like to thank Ms. Michelle Louise B. Dela Rosa and Mr. Nathaniel V. Tesoro, authors of the study where ours branched out, for willingly giving us necessary and helpful files such as the complete Triple Stores and RDF Graphs from their research.

Appendices

1. Publicly-available Data Sets used in Profiling [4]:
   `http://bit.ly/ProfilingDataSets`
2. Model for Predicate Construction [4]:
   `http://bit.ly/Predicates`
3. ERDPlus:
   `https://erdplus.com/#/`

6. REFERENCES

[1] American Veterinary Medical Association. (July 2008). *One Health: A New Professional Imperative*, One Health Initiative Task Force: Final Report [Online]. Retrieved from `https://www.avma.org/KB/Resources/Reports/Documents/onehealt_final.pdf`.

[2] World Health Organization. (October 2017). *One Health*. Retrieved from `https://www.who.int/features/qa/one-health/en/`.

[3] W. D. Cyganiak, R. & M. Lanthaler. (February 2014). *RDF 1.1 Concepts and Abstract Syntax*. W3C Working Group Note, Retrieved from `https://www.w3.org/TR/rdf11-concepts/`.

[4] Dela Rosa et al. (2018). *Building RDF Models of Multidisciplinary Data Sets under the One Health Framework in the Philippine Setting*. Retrieved from `https://drive.google.com/file/d/1mLnBaziMl4n1uFtZFR2QqbG05wXjyoAL/view`.

[5] Cormen, T. (2001). *Johnson's algorithm for sparse graphs*. Introduction to Algorithms, MIT Press and McGraw-Hill, ISBN 978-0-262-03293-3. Section 25.3, pp. 636–640.

[6] Matono et al. (2005). *A Path-based Relational RDF Database*. Retrieved from `https://dl.acm.org/citation.cfm?id=1082233`.

[7] Al-Btoush, A.A. (2015). *Extracting Entity Relationship Diagram (ERD) from English Sentences*. Intl J Database Theory and Application, 8(2), 235–244.

[8] Jukic, N., Vrbsky, S., & Nestorov, S. (2017). *Database systems: Introduction to databases and data warehouses*. Burlington, VA: Prospect Press.

[9] National Epidemiology Center. *PIDSR Forms*. Retrieved from `http://ro6.doh.gov.ph/index.php/health-statistics/disease-surveillance/190-pidsr-forms`.

[10] National Epidemiology Center. *Section 10: Guidelines for Diseases, Syndromes and Health Events under Surveillance*. Department of Health, Philippines. Retrieved from `https://www.doh.gov.ph/sites/default/files/publications/PIDSRMOP3EDVOL2pp1-542014.pdf`.

[11] Samanta, J. & Sharma, V. (2015). *Dengue and its effects on liver*. World journal of clinical cases, 3, pp. 195–31.

[12] Anwar, N. & Pustulka, E. (2009). *Francisella tularensis novicida proteomic and transcriptomic data integration and annotation based on semantic web technologies*. BMC bioinformatics. 10 Suppl 10. S3. 10.1186/1471-2105-10-S10-S3.

[13] *Johnson's Algorithm for All-Pairs Shortest Paths*, (n.d.). Lecture presented in Columbia University. Retrieved from `http://www.columbia.edu/~cs2035/courses/ieor6614.S16/johnson.pdf`.

[14] Hert et al. (2011). *A Comparison of RDB-to-RDF Mapping Languages*. In: Proceedings of the 7th International Conference on Semantic Systems (I-Semantics), Graz, Austria.

[15] *Relational Databases on the Semantic Web*. (1998). Retrieved from `https://www.w3.org/DesignIssues/RDB-RDF.html`.

[16] World Health Organization. (April 2019) *Dengue and severe dengue*. Retrieved from `https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue`.

[17] M. Brunnbauer. (2012). *RDF2RDB-convert RDF data to relational databases*. `https://www.netestate.de/en/software-development/rdf2rdb/`.

[18] Merck. *RDF2X*. `https://github.com/Merck/rdf2x`.

[19] *Vector-borne diseases* (January 2019). Retrieved from `https://ecdc.europa.eu/en/climate-change/climate-change-europe/vector-borne-diseases`.