Data Integration, Querying and Reasoning over the Harmonized Survey on Households Living Standards Semantic Data Model

Marc Mfoutou Moukala
Faculty of Science and Technology
Marien NGOUABI University
Brazzaville, Congo
e-mail: moukmarc@yahoo.fr

Macaire Ngomo
Research and Innovation Department
CM IT CONSEIL
Romilly sur Seine, France
e-mail: macaire.ngomo@gmail.com

Régis Freguin Babindamana
Faculty of Science and Technology
Marien NGOUABI University
Brazzaville, Congo
e-mail: regis.babindamana@umng.cg

***Abstract*—In this paper, we build a Resource Description Framework (RDF)-based semantic data model of the Harmonized Survey on Households Living Standards (HSHLS), and propose an approach for data integration and querying and reasoning over the semantic data model built. We implement data consistency control rules, by combining RDF and Semantic Web Rule Language (SWRL). Based on SWRL rules, we then apply automated reasoning for the survey's data consistency control. We leverage the power of SPARQL to query information from the survey model and data, using Python programming language.**

Keywords-SPARQL queries; automated reasoning; semantic data model; Resource Description Framework data integration; Semantic Web Rule Language; data consistency control; Harmonized Survey on Households Living Standards.

I. INTRODUCTION

The work in [1] proposes a semantic data model of the Harmonized Survey on Households Living Standards (HSHLS) [2], which is a major household statistical survey conducted by French speaking countries, allowing public authorities to gather relevant information helping them to identify and solve daily living problems encountered by their population. The HSHLS survey consists of a set of modules or sections, each section dealing with a given theme. Among the topics addressed there are: the socio-demographic characteristics of households and their members as well as information on education, health and employment of household members. Information about each section is collected through a specific questionnaire administered using a computer application. The collected data is usually stored in a tabular structure in a relational database. After the data collection operations are executed, they are retrieved in Excel-type files for possible analyses. During data cleaning operations, discrepancies are often found between the methodology and the data actually collected, as some of the collected data do not comply with the conditions or rules defined in the methodology. Moreover, since the methodological information is not automated, data processing teams are often forced to manually consult the related documents; this sometimes causes delays in the data processing procedures. In the present work, we improve the semantic model built in [1], aimed at representing knowledge related to the survey under study in the way that facilitates the retrieval of methodological information. We also propose an approach to integrate survey data into the model built, and to query and make reasoning over the model. Doing so, the final purpose of the current work is both to document and disseminate knowledge related to this survey, and to facilitate data quality control, improving then the quality of the survey.

Our paper is structured as follows: In Section II, we define the concept of semantic data modeling and its advantages. In Section III, we present the literature review related to the semantic modeling in statistical surveys and highlight some limitations of the state of the art. Section IV presents the adopted methodology and tools. In Section V, we present our semantic data model and the implementation details. In Section VI, we propose and implement an approach to populate the built model with actual questions, and their respective conditions, constraints and dependencies. In Section VII, we prepare data consistency rules, using Semantic Web Rule Language (SWRL) [3], and apply automated reasoning over an example of data, based on those SWRL rules. In Section VIII, we present and discuss the results. Finally, we end with a conclusion along with an outline of potential future work, in Section IX.

II. SEMANTIC DATA MODELING

Data semantics is the meaning given to that data; it is its significance. It encompasses all the information that can be gathered about the data with respect to a specific objective and a particular reality. For example, regarding the data point **Age**, the following information can be inferred: Age is a property of a person, that defines their current lifespan, expressed in years, which is the mathematical difference between the year of birth and the current year, taking into account the date of the person's most recent birthday. This lifespan is an integer between 0

(minimum duration) and 120 (maximum duration). The year of birth and the current year are also properties of a person, of integer type. Semantic data modeling involves representing the data and their semantics as well as the relationships between them.

Semantic data models play a crucial role in the modeling of complex knowledge. They allow representing relationships and interactions between concepts in an accurate and structured way. These models are used in various application scenarios: representing relationships between concepts, encapsulating business knowledge, data search and analysis, integrating heterogeneous data sources, and supporting artificial intelligence and machine learning.

III. LITERATURE REVIEW

In the context of semantic modeling applied specifically to statistical survey data, the following research has strongly influenced our work.

The work in [4] addresses the challenge faced by users who need to write complex database queries to retrieve information, given their limited understanding of both the structural and semantic complexities of databases. It focuses on improving this process through the use of ontologies to facilitate better knowledge representation and interactive query generation.

In [5], the authors use an ontology-based approach to develop a semantic model for harmonizing and integrating population health data from heterogeneous sources. Following a presentation of the ontology literature, the authors of [5] identified key concepts and relationships between population health data. Then, they used this information to develop an XML schema-based semantic ontology to harmonize and integrate population health data from different sources (Excel, SQL Server and MongoDB) for early detection of COVID-19. The authors state that the model designed allows data to be inserted, updated and deleted without anomaly as the data mapping is based on schema and not on data. The authors also state that their method could be extended by creating ontologies in RDF/Turtle formats.

The work in [6] proposes an approach to improve the semantic interoperability of electronic health records using ontological management of domain ontology evolution. The researchers first developed a domain ontology representing concepts and relationships relevant to the domain of electronic health records. Then, they proposed methods to manage the evolution of this ontology over time, taking into account changes in the electronic health domain and new data requirements.

In [7], Nicholson et al. use an ontology-based approach to ensure a good level of data quality for cancer-related information in registries, in order to accurately compare indicators related to this disease on regional and national scale based on harmonized rules.

The work in [8] introduces a generic ontology designed to represent questionnaires in a machine-readable format. This ontology aims to enhance decision support systems and smart environments by facilitating automatic processing of questionnaire data, which has become more abundant and cost-effective due to mobile devices. It addresses the challenge of managing and reasoning about large volumes of collected information to gain deeper insights.

Considering the literature review, we notice that there is a great deal of similar work in semantic data modeling. However, to the best of our knowledge, there is no application of these research studies to the statistical harmonized housing surveys on household living standards. Access to methodological information is manual and the majority of data quality control is conducted manually, leading to significant delays in data processing. Therefore, our contribution lies in the application of this work in the context of documentation and popularization of knowledge relating to the HSHLS survey, and consists in proposing a semantic data model whose use would among other things, facilitate access to related knowledge and help speed up data processing.

IV. METHOD AND TOOLS

A. Methodology

The objective of this work is to design a semantic data model capable of formally and systematically representing the knowledge embedded in the HSHLS survey, with the aim of achieving the following goals: making the survey data and metadata interoperable, improving efficiency across the survey lifecycle, facilitating advanced and flexible querying, and supporting data quality control.

The HSHLS survey involves complex and multi-level data, and managing that survey effectively requires formal structures that can support automated processes, ensure conceptual consistency, and validate the integrity of collected and processed data. To meet these challenges, the literature review led us to adopt an ontology-based semantic modeling approach to represent and structure knowledge semantically. This choice is justified by the fact that:

- Ontologies enable complex logical relationships between concepts to be defined formally. Axioms and rules can be used to express logical conditions of dependency between survey questions such as IF-THEN conditions, validation constraints, hierarchical relationships, etc.
- Ontologies provide a standardized, shared data model, promoting interoperability between different systems and enabling easier data integration. This can be particularly useful in a survey context, where data needs to be collected, stored and analyzed in a consistent and standardized way.
- Ontologies enable the complexity of dependencies between survey questions to be managed in a structured way. Concepts can be organized into classes and sub-classes and properties and restrictions can be defined, making it easier to manage and understand the relationships between different questions.

- Ontologies provide a solid basis for managing the evolution and maintenance of the semantic data model. Concepts and relationships can be easily added, modified or deleted without compromising model consistency and compatibility.

B. Tools

Although there are many works available in the literature that use other representation ways, to build our semantic model, represent concepts and their relationships, we use the Resource Description Framework (RDF) [9] and RDF Schema (RDFS) [10] along with the Web Ontology Language (OWL) [11]. The reasoning part is achieved using Semantic Web Rule Language (SWRL).

RDF is a framework standardized by the World Wide Web Consortium (W3C) to represent information on the Web in a structured and interoperable way. It provides a simple data model based on subject-predicate-object assertions, also known as RDF triples. Each triple describes a relationship between two resources.

RDF Schema (RDFS) is an extension of RDF that provides a vocabulary for describing schemas and ontologies. This enables the definition of classes, properties and relationships between RDF resources.

To learn about the various concepts related to the survey under study as well as the relationships between the data, this research relies on methodological documents including household questionnaires, interviewer manuals and survey data dictionaries.

Our semantic model is built using RDF and RDF Schema. We propose to use metamodeling techniques to implement the models needed to manipulate the elements of the Harmonized Survey on Households Living Standards questionnaire and the survey data consistency control. The following section presents our semantic model.

V. MODELIZATION AND IMPLEMENTATION

The original concepts of the survey under study are in French, since this survey concerns French-speaking countries. However, for illustration purposes, in this work, we provide an English version of those concepts to allow a large audience to easily understand our work.

A. Definitions of Key Concepts

1) Harmonized Survey on Households Living Standards(HSHLS): HSHLS is the main harmonized statistical survey conducted by French-speaking countries in West and Central Africa to capture household living conditions.

2) Section: The HSHLS survey is composed of sections. A section is named according to the topic addressed: Socio-demographic characteristics, Education, Health, etc.

3) Questionnaire: Every section has a single questionnaire. A questionnaire captures the main information of surveyed households related to a given section.

4) Question: A questionnaire is composed of questions.

5) Household: This is the main statistical unit on which information are gathered. We distinguish two kinds of households: ordinary households and collective households. An ordinary household is a group of people, whether related or not, who usually live in the same dwelling, pool their resources, share their meals, and recognize the authority of the same person as the head of the household. Households that are not ordinary are referred to as collective households. These are people or groups of people living in collective housing (such as military barracks, boarding schools, hospitals, etc.). This survey only concerns ordinary households. An ordinary household consists either of a single person (for example, a student who rents a room alone) or several people. In the latter case, the household usually consists of a spouse, their husband/wife, and their children, with or without other dependents (family members, friends, etc.). An ordinary household can also be made up of people who live together and have no familial ties. In all the rest, the concept of household refers to an ordinary household.

6) Household member: A person belonging to a particular household. A household member is a person who usually resides in the household. An individual usually resides in the household in two situations: he/she has lived in the household for at least 6 months or has arrived in the household less than 6 months ago but with the intention of staying for at least 6 months.

B. Presentation of the HSHLS Semantic Model

Our model consists of a resource class HSHLS representing HSHLS surveys, an instance of the rdfs:Class class of the RDF model. A survey is made up of sections. A section contains a questionnaire. A questionnaire is of a certain type, depending on the information collected. This may be information characterizing household members or common household characteristics. These characteristics are variables represented by questions. Each question is an instance of the rdfs:Class class in our model, and concerns a household member or a household as a whole. A household member belongs to a unique household, and has answers to the survey questions. A household member can have answers to all or some questions, depending on their characteristics and the eligibility criteria (for example, a minimum age threshold for a surveyed to answer to questions on Education). An answer is related (refers) to both a household member and a question, and is of a certain type (integer, float, string) depending on the nature of the expected response. Each question is linked to a questionnaire. A question may depend on other questions and may have constraints, which are acceptable values of its answers.

The general model includes the class declaration model (Figure 1) and the property declaration model (Figure 2). More details of the semantic model are given in Appendix A.

In the class declaration, the declaration C rdf:type C' means that the class C is an instance of the class C'. For example, hshls:HSHLS rdf:type rdfs:Class states that the HSHLS is an instance of rdfs:Class. In the property declaration, a triple of the form: P rdfs:domain C declares that P is an instance of the rdf:Property class, that C is an instance of the rdfs:Class class, and that the resources indicated by the subjects of triplets whose predicate is P are instances of the C class. This implies that hshls:hasQuestion rdfs:domain hshls:Questionnaire states that hasQuestion is an instance of rdf:Property class, Questionnaire is an instance of the rdfs:Class class, and that the resources indicated by the subjects whose predicate is hasQuestion are instances of the class Questionnaire. The triple P rdfs:range C means that P is an instance of the rdf:Property class, that C is an instance of the rdfs:Class class, and that the resources indicated by the objects in the triple whose predicate is P are instances of the C class. In this case, hshls:hasQuestion rdfs:range hshls:Question means that hasQuestion is an instance of the rdf:Property class, Question is an instance of the rdfs:Class class, and that the resources indicated by the objects in the triple whose predicate is hasQuestion are instances of the Question class. The following code illustrates the class declaration model, serialized in Turtle format, and its (simplified) graph representation (Figure 1).

```
hshls:HSHLS rdf:type rdfs:Class ;
    rdfs:label "Harmonized Survey on
Households Living Standards" .
hshls:Section rdf:type rdfs:Class ;
            rdfs:label "A section or
module of the survey" .
hshls:Questionnaire rdf:type
rdfs:Class .
hshls:Question rdf:type rdfs:Class ;
    rdfs:label "A question of the
survey" .
hshls:Constraint rdf:type rdfs:Class ;
          rdfs:label "Constraint on
the question" .
hshls:Household rdf:type rdfs:Class
;
    rdfs:label "A household surveyed"
.
hshls:Household_Member rdf:type
rdfs:Class ;
    rdfs:label "A household member
surveyed" .
hshls:Answer rdf:type rdfs:Class ;
    rdfs:label "A household member's
answer related to a question" .
```

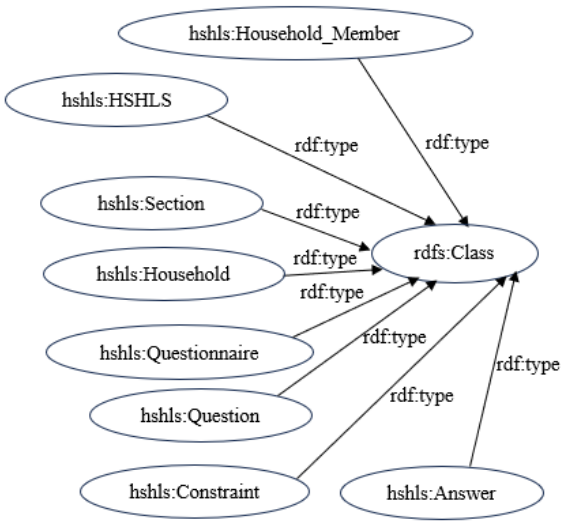

Figure 1: HSHLS RDFS model graph with class declaration.

Figure 1 represents 8 class declarations, all being instances of rdfs:Class.

Figure 2 illustrates the property declaration model, with the properties written in bold. For space-saving purposes, not all properties are represented in the figure.

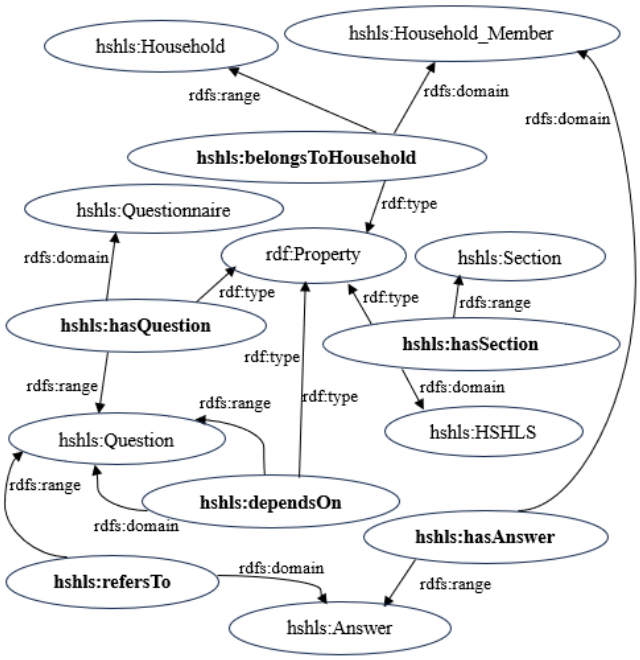

Figure 2: HSHLS RDFS model graph with property declaration

C. HSHLS Model Specialization

To serve as a proof of concept, we propose a specialization of our model, considering personalized methodological information used during the first edition of this survey in Congo. The overall survey has 21 sections. In this article, we do consider a particular section: the Education section. The specialization is as follows:

1) HSHLS ontology with actual questions: Figure 3 illustrates the map of HSHLS ontology with actual questions, for the section which captures education's information, codified S02. The education information concerns household members of three (3) years old or above. So, the entry in this questionnaire depends on the response to the question on the age of the corresponding household member surveyed, here codified S02Q_Age. The question S02Q1a captures whether the surveyed household member can read a little text written in French. S02Q1a depends on the question S02Q_Age: if the value of the answer to the question S02QAge by the concerned household member is less than a given minimum (9, in this case), the question S02Q1a must not be asked, so the value of the answer related to that question must be empty. The question S02Q03, which also depends on the question S02Q_Age, captures whether the surveyed household member is currently attending or have attended a formal school. The question S02Q04 captures the reason why the surveyed household member has never attended a formal school. S02Q04 depends on the response to the question S02Q03 which can be "Yes" or "No, never attended" meaning that the concerned surveyed has never attended a formal school. S02Q04 is asked only if the response to S02Q03 takes the second valid value. A formal school (public or private) is a place of learning where the programs offered are organized and structured (primary school, high school, university, etc.), and formal learning typically leads to the validation and awarding of a diploma.

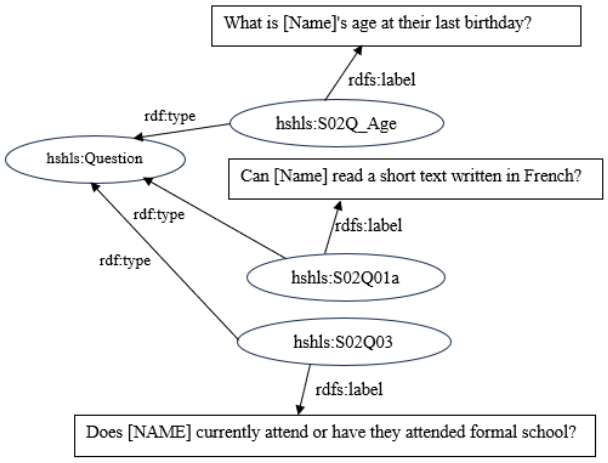

Figure 3: The HSHLS RDFS Ontology with actual questions.

In Figure 3, triples X rdf:type hshls:Question (where X is one of hshls:S02Q_Age, hshls:S02Q01a and hshls:S02Q03) mean that X is an instance of hshls:Question, in other words, X is a question. Triples X rdfs:label y mean that the question X has as label y.

2) HSHLS constraints and dependency specification: Figure 4 gives an illustration of actual questions and their constraints and dependencies. A constraint in our model represents valid values of a question. In Figure 4, the triple hshls:ConstraintS02Q01a rdf:type hshls:Constraint declares that hshls:ConstraintS02Q01a is a constraint. The property hshls:hasConstraint links a question to its constraint. Both a question and a constraint being instances of resources of rdfs:Class, the property hshls:hasConstraint is an object property. The data property hshls:validValues specifies the valid (possible) values of the related constraint. In the figure, the triples hshls:ConstraintS02Q01a hshls:validValues "Yes" and hshls:ConstraintS02Q01a hshls:validValues "No" ("Yes" and "No" being literal values) specify that the valid values of the constraint hshls:ConstraintS02Q01a are "Yes" and "No". That means that the response to the related question (S02Q01a) must be "Yes" or "No". In the figure, the question S02Q01a depends on the question S02Q_Age and the dependency condition is that the value of the answer to S02Q_Age (which precedes S02Q01a) must be greater or equal to 9. That means, to ask the question S02Q01a, the concerned surveyed must be 9 years old or above. If this condition is not satisfied, then the answer to S02Q01a must be empty for the concerned surveyed.

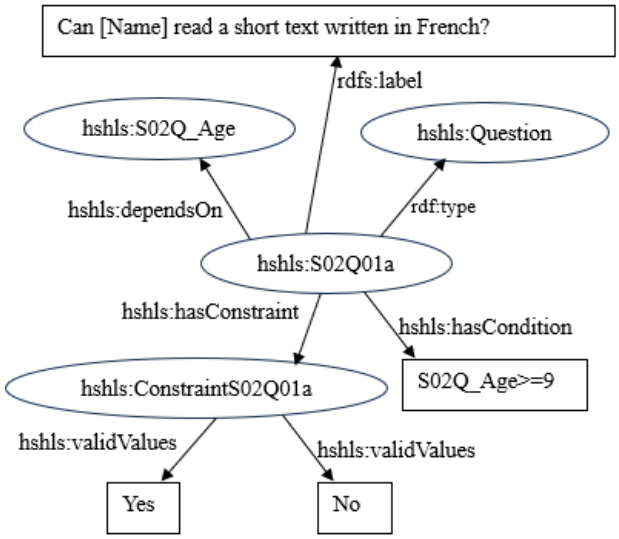

Figure 4: HSHLS ontology constraints and dependency specification.

To enhance the clarity of constraint meanings and enable seamless navigation within the model, we chose using a clear and intuitive coding scheme for each

constraint, as outlined hereafter: ConstraintQuestionName where QuestionName is the name of the related question. For example, hshls:ConstraintS02Q01a in Figure 4 denotes the constraint related to the question S02Q01a.

In the above, we built a semantic data model to represent the Harmonized Survey on Households Living Standards (HSHLS) methodological information in a human and computer readable format. The proposed model is built using the Python RDFLib package [12], in a Jupyter notebook environment. The model is saved in Turtle (ttl) format and can be exploited as an RDF graph. To make it possible to get access to the model in a persistent way, we defined an International Resource Identifier (IRI) for the model. We also developed an HTML ontology documentation file using PyLODE [13]. We created a public Github repository [14] and saved the rdf graph and its HTML documentation in it.

In the following sections, we propose and implement a way to populate the model and to make reasoning based on the model. In this work, the reasoning system is intended for data consistency control, in order to solve the problem of discrepancies which are often found (during data cleaning operations) between the methodology and the actual data collected. We propose the reasoning system to make it possible to regularly detect the inconsistencies in data during data collection operations, improving then the efficiency of the survey operations.

The rest of this paper is structured as follows: in Section VI, we propose and implement an approach to automatedly populate the semantic data model with actual questions, and their respective conditions, constraints and dependencies. Section VII presents the automated reasoning mechanism proposed for the survey's data consistency control, through SWRL rules. This section ends with a presentation of some SPARQL queries over the data model built. In Section VIII, we present and discuss the results. Finally, we end with a conclusion along with an outline of potential future work, in Section IX.

VI. HSHLS RDFS ONTOLOGY AND DATA INTEGRATION

In this section, we propose and implement an approach to integrate model's ontology and data in an automated way. The aim of the ontology integration is to map different instances of the model (questions, labels, constraints, dependencies, conditions) from different survey versions into a formal RDF structure that conforms to our existing model. The data integration part proposes a way to integrate actual data from the survey into the RDFS model.

1) HSHLS RDFS ontology integration: In our approach, we prepare the instances of the model in an Excel-type file, and we integrate those instances into the model, using Python.

Figure 5 illustrates the implemented process. We first load the Turtle RDF model and the Excel file containing questions and their description (Question ID, label, constraint, dependency, condition). Next, URIs are generated for every element of each iterated row, and triples are added to the RDF graph based on generated URI. Finaly, the updated RDF graph is saved.

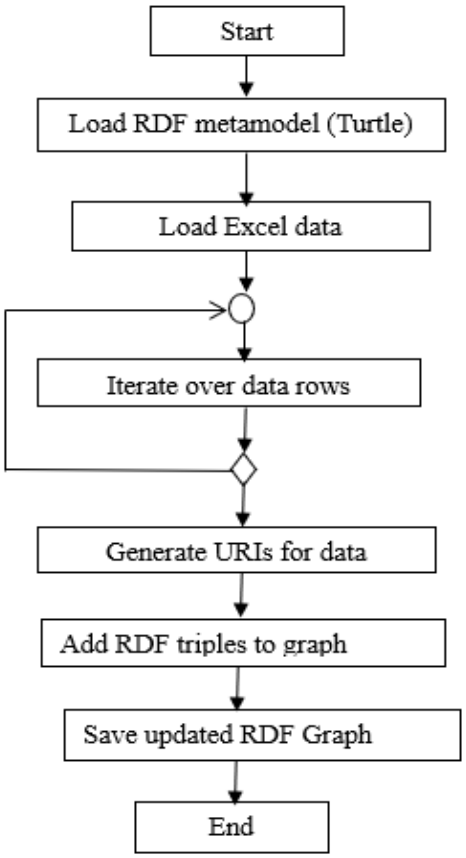

Figure 5: Diagram for populating the HSHLS RDF model.

A pseudo-code representation of the presented process is detailed as follows:

```
Begin
    Initialize an empty graph ;
    Parse the RDF Turtle file into the
graph ;
    Load the Excel file containing the
data (questions, constraints,
conditions) ;
    For each row in the DataFrame:
        Generate URIs for question and
constraint;
        Construct the URI for the
question using the "Question ID"
column ;
        Construct the URI for the
constraint using the "Constraint"
column ;
```

```
        Add a triple : (question URI,
RDF.type, Question class) ;
        Add a triple: (question URI,
hasConstraint, constraint URI) ;
Add a triple : (question URI,
dependsOn, value from "Dependency"
column) ;
        Add a triple : (question URI,
hasCondition, value from "Condition"
column) ;
        Add a triple : (constraint
URI, RDF.type, Constraint class) ;
        Add a triple: (constraint URI,
validValues, values from "Constraint"
column) ;
    Serialize the updated graph to a
new Turtle file and save it.
End
```

Table I and Table II (which actually constitute the same table but separated for space-saving purposes) illustrate the shape of the Excel file with questions to insert automatedly in the survey semantic data model.

TABLE I. HSHLS SURVEY QUESTIONS AND THEIR LABELS AND CONSTRAINTS

Question ID	label	Constraint
S02Q01a	Can [Name] read a short text written in French?	Yes;No
S02Q01b	Can [Name] read a short text written in Lingala?	Yes;No

TABLE II. HSHLS SURVEY QUESTIONS AND THEIR CONDITIONS AND DEPENDENCIES

Question ID	Dependency	Condition
S02Q01a	S02Q_Age	S02Q_Age>=9
S02Q01b	S02Q_Age	S02Q_Age>=9

In Table I and Table II, **Question ID** is the codification of the question. The column **label** represents the way to ask the given question to the respondent or the surveyed. The column **Constraint** represents valid values of the related questions. The column **Dependency** specifies the question(s) on which the given question depends, and the dependency condition is given in the column **Condition**. In other words, the given question may remain unanswered, depending on the answers to the question(s) on which it is contingent. In the case illustrated in the table, the decision to ask question S02Q01a depends on the response given to question S02Q_Age, and the dependency condition is that S02Q_Age must be greater or equal to 9.

The semantic data model with concepts inserted automatedly following the described process for questions presented in Table I and Table II is presented as follows:

```
  hshls:ConstraintS02Q01a
  a hshls:Constraint ;
      hshls:validValues
"No"^^xsd:string,
          "Yes"^^xsd:string .
  hshls:ConstraintS02Q01b
  a hshls:Constraint ;
      hshls:validValues
"No"^^xsd:string,
          "Yes"^^xsd:string .
  hshls:S02Q01a a hshls:Question ;
      rdfs:label " Can [Name] read a
short   text   written   in   French?
"^^xsd:string ;
      hshls:dependsOn "S02Q_Age" ;
      hshls:hasCondition  "S02Q_Age>=9"
;
      hshls:hasConstraint
hshls:ConstraintS02Q01a .
  hshls:S02Q01b a hshls:Question ;
      rdfs:label " Can [Name] read a
short   text   written   in   Lingala
?"^^xsd:string ;
      hshls:dependsOn "S02Q_Age" ;
      hshls:hasCondition  "S02Q_Age>=9"
;
      hshls:hasConstraint
hshls:ConstraintS02Q01b .
```

2) HSHLS RDFS data integration: In this part, we propose a way to retrieve actual data from an input data file, and insert them into the model built. Here's an explanation of each step involved in this workflow:

- Initialize RDF Graph: This step involves creating an empty RDF graph, which will store all the triples (subject-predicate-object relationships) for the data.
- Load existing RDF model: Load the RDF survey model to ensure that the new data is added to the pre-defined structure.
- Load survey data from Excel file: The Pandas library is used to read the survey data from an Excel file into a DataFrame. The survey data, which includes household details, household members information, and answers to various questions, is stored in a structured tabular form (rows and columns). This makes it easier to iterate over and extract specific information for RDF generation.
- Iterate through each row in the DataFrame: The program iterates through each row in the

DataFrame, processing the data for one household member at a time. Each row corresponds to a specific household member and its associated answers.

- Extract household information: For each row, key details related to the household are extracted, such as: department number, which is the geographic region of the household; the cluster number; the household number (a unique identifier for the household within the cluster), and the wave number which indicates the survey wave or time period the data is from.
- Create and add household URI to RDF: Using the extracted household information, a unique URI is generated for each household, following a naming pattern such as: household_departementNumber_clusterNumber_householdNumber_waveNumber. This URI is used as the subject for all triples related to the household. this makes it possible to consistently reference and link the household's data (members, answers) within the RDF graph.
- Create and add household member information: A household member is given an order number inside the concerned household. But that number is not unique across the entire survey. So, to uniquely identify a household member across the entire survey and ensure proper linking in the RDF model, we propose a combination of the household identifier and the household member order number. Then, for each member of the household, the program generates a unique URI. The program adds RDF triples for each member, including they details. These triples represent key attributes of the household member in the RDF graph. A triple is also added to link the household member to their corresponding household using the predicate *belongsToHousehold*. This relationship ties each member to the household, creating a clear connection between the two entities in the RDF graph.
- Add answers for survey questions: The program iterates over the columns of the DataFrame that represent survey questions (S02Q_Age, S02Q01a, etc.). For each question, the program checks if there's a non-null value. This ensures that only non-empty answers are processed, excluding any empty or missing data. For each valid answer, a unique URI for the answer is created, incorporating details of the household member identifier and the related question. RDF triples are added to the graph, associating the answer URI with the answer's value (a numeric or textual response). The answer is also linked to the relevant question URI, and the member is linked to the answer using the predicate *hasAnswer*. This ensures that each survey response is recorded as a separate RDF entity, linked to both the question and the household member who provided the answer.
- Serialize and save the updated RDF Graph: After all rows are processed and the RDF graph is populated with all the data, the program serializes the graph into a Turtle (.ttl) file. This allows the RDF graph to be saved in a standard format that can be shared, queried, or used by other applications. At the end, the survey turtle file will contain the RDF representation of the entire survey.

Table III and Table IV illustrate an example of survey data from households in the 11th department, from three distinct clusters (the names of some columns are shortened and some columns are not represented for space-saving purposes).

TABLE III. EXAMPLE OF HSHLS SURVEY DATA

departement	cluster	householdNumber	waveNumber
11	516	4	2
11	516	4	2
11	516	4	2
11	320	2	2
11	324	4	2

TABLE IV. EXAMPLE OF HSHLS SURVEY DATA (CONTINUE)

OrderNumber	Name	Sex	S02Q_Age
1	Mouk	M	51
2	Marc	F	32
3	Annan	F	2
1	Lotté	M	56
2	Bruno	M	8

In Table III and Table IV, each row represents the responses of a respondent or a surveyed to the questions for a given section. Columns **departement**, **cluster**, **householdNumber**, **wavenumber** and **OrderNumber** compose identification information of the surveyed. An illustration of the result of data integrated is given as follows:

```
hshls:householdmember_11_320_2_2_1
a hshls:Household_Member ;
    hshls:Name "Lotté"^^xsd:string ;
    hshls:Sex "M"^^xsd:string ;
    hshls:belongsToHousehold
hshls:household_11_320_2_2 ;
```

```
    hshls:hasAnswer
hshls:answer_11_320_2_2_1_S02Q01a,
hshls:answer_11_320_2_2_1_S02Q03,
hshls:answer_11_320_2_2_1_S02Q_Age .
hshls:answer_11_320_2_2_1_S02Q01a
a hshls:Answer ;
    hshls:hasValue "Yes" ;
    hshls:refersTo hshls:S02Q01a .
hshls:answer_11_320_2_2_1_S02Q03
a hshls:Answer ;
    hshls:hasValue "Yes" ;
    hshls:refersTo hshls:S02Q03 .
hshls:answer_11_320_2_2_1_S02Q_Age
a hshls:Answer ;
    hshls:hasValue 56 ;
    hshls:refersTo hshls:S02Q_Age .
hshls:household_11_320_2_2
a hshls:Household ;
    hshls:clusternumber
"320"^^xsd:string ;
    hshls:departementNumber
"11"^^xsd:string ;
    hshls:householdNumber
"2"^^xsd:string ;
    hshls:waveNumber "2"^^xsd:string .
```

VII. AUTOMATED REASONING

Automated reasoning is a branch of artificial intelligence that focuses on the development of algorithms and software tools that allow machines to deduce new knowledge or validate logical arguments without human intervention. In our work, we apply automated reasoning for data consistency verification. We first build a set of data consistency rules, using SWRL, next, we import actual data into Protégé software and apply those SWRL rules against actual data. The result of the reasoning is saved into an RDF-OWL Turtle file for querying purposes. In the following, we first give an overview of SWRL mechanisms and the software (Protégé) used in our work for reasoning, before presenting SWRL rules defined on an example of survey data, and the inferred axioms from the described reasoning software.

A. Overview of Semantic Web Rule Language

Semantic Web Rule Language (SWRL) is a rule-based framework that extends Web Ontology Language (OWL) and RDF with logic-based rules, enabling the Semantic Web to infer new knowledge from existing data. Combining OWL and Horn Logic, SWRL allows for the definition of logical rules that can infer new knowledge from existing data, facilitating dynamic and intelligent web-based systems. SWRL's integration with RDF enables it to operate over structured data, making it a critical component for knowledge representation, inference, and automated reasoning. SWRL is essentially an intersection of first-order logic and description logic. It allows for a more dynamic and complex set of reasoning capabilities over ontologies. By utilizing rules, SWRL supports the reasoning of facts and the automatic generation of inferences based on the provided input. SWRL's syntax is based on Horn Logic, where each rule is an implication that takes the form of an "if-then" statement. A typical rule has an antecedent (the "if" part), which represents conditions or facts that must hold, and a consequent (the "then" part), which defines the new fact that will be inferred if the conditions are true. The general structure of a SWRL rule is: **Antecedent -> Consequent**, for example:

```
Person(?p)^hasAge(?p,?a)^
swrlb:greaterThan(?a, 18) -> Adult(?p).
```

This rule states that if a Person has an age greater than 18, they should be classified as an Adult. SWRL rules can also be expressed in the form of **body** and **head**, where body consists of a set of conditions or premises that must be satisfied for the rule to apply (equivalent to Antecedent), and head contains the conclusion that follows when the conditions in the body are met (equivalent to Consequent). SWRL rules can involve complex logic, including data type comparisons, existential quantification, and object property relations. Once SWRL rules are defined, they can be processed by reasoning engines such as Pellet [15], HermiT [16], or FaCT++ [17]. These reasoning engines are capable of interpreting the rules and performing inference over the knowledge base represented by the OWL ontology. When the reasoning engine processes an ontology with SWRL rules, it applies the rules to infer new facts or detect contradictions. This process makes SWRL especially valuable for tasks such as data validation, automated decision-making, and semantic search.

B. Overview of Protégé Software

In our work, we use Protégé (software, version 5) to implement the rules and apply reasoning over the RDFS-OWL data model built. Protégé ([18], [19]) is a free, open-source platform (W3C standards compliant) that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. Protégé is offered in two formats: Protégé Desktop and WebProtégé. Protégé Desktop (used in our work) is a feature rich ontology editing environment with full support for the OWL 2 Web Ontology Language, and direct in-memory connections to description logic reasoners like HermiT and Pellet. Protégé Desktop supports creation and editing of one or more ontologies in a single workspace via a completely customizable user interface. Visualization tools allow for interactive navigation of ontology relationships. Advanced explanation support aids in tracking down inconsistencies. Refactor operations available including ontology merging, moving axioms between ontologies, rename of multiple entities, and more. WebProtégé is an ontology development environment for the Web that makes it easy to create, upload, modify, and share ontologies for collaborative viewing and editing.

C. Automated Reasoning over the Harmonized Survey on Households Living Standards Data Model

In the survey under study, there are some dependencies between questions, and those dependencies specify some eligibility conditions for surveyed household members in respect to some questions. In others words, not all questions are responded by all surveyed, depending on the age range, sex, and others characteristics. For example, a surveyed of less than three years old cannot attend a school. For that constraint, an eligibility condition is defined such that: if S02Q_Age < 3, then skip all questions related to the education of the surveyed. In the same way, if a surveyed answered "Yes" to the question S02Q03 asking if he/she is attending or has ever attended a formal school, it should be inconsistent to ask the question S02Q04 to know the reason why the surveyed didn't attend a formal school. If the answer to S02Q03 is "No, never attended", we need to make sure that the reason of non-attendance is captured, and we need to skip all school attendance related questions such as asking the highest school level achieved by the surveyed and others. So, in regard to all the precedent, we chose to translate those constraints into rules aiming to facilitate reasoning. The code below illustrates those rules, built using SWRL and executed using Protégé:

```
hshls:Household_Member(?HMember)^
hshls:hasAnswer(?HMember,
?answer)^hshls:refersTo(?answer,
hshls:S02Q_Age)^hshls:hasValue(?answer,
?value)^swrlb:lessThan(?value,3)^
hshls:hasAnswer(?HMember,
?answer2)^hshls:refersTo(?answer2,
hshls:S02Q03)->
hshls:Inconsistent(?HMember)^
hshls:HasInconsistency(?HMember,
"Inconsistent data. The age of this
individual is less than 3, therefore he
or she must not answer to questions on
Education. Please double check and
correct the information accordingly.")
```

The rest of the rules can be found in Appendix B.

After setting rules and executing them in Protégé, we get inferred axioms with potential inconsistencies pinpointed.

D. Querying on the Harmonized Survey on Households Living Standards Semantic Data Model

In this subsection, we present the feasibility of doing some queries for information retrieval from the survey ontology and data. We use SPARQL for that purpose. We first give a brief overview of SPARQL concepts and mechanisms before presenting some queries as a proof of concept.

1) An overview of SPARQL: SPARQL [20] is a query language designed specifically for querying RDF data. RDF structures data in triples: subject-predicate-object, where the subject represents the entity, the predicate represents the property, and the object represents the value of that property.

Below are some types of queries included in SPARQL:

- SELECT: The fundamental query in SPARQL is the SELECT query, which retrieves specific data from RDF stores based on pattern matching. A typical SPARQL query looks like this:

```
SELECT ?name WHERE {
?person rdf:type foaf:Person .
?person foaf:name ?name .}
```

In this example, the query selects the names of all individuals classified as foaf:Person.

- CONSTRUCT: Generates a new RDF graph based on the results of a query.
- ASK: Returns a boolean answer indicating whether a given query pattern exists in the dataset.
- DESCRIBE: Provides an RDF graph that describes the resources identified by the query.
- FILTER expressions: Allow for complex logical and arithmetic operations to restrict query results.

2) SPARQL queries for HSHLS survey methodological information retrieval: Following are some queries for illustration:

- List questions and their dependencies: Here is a query to list questions and their dependencies:

```
from rdflib.plugins.sparql import
prepareQuery
query_onto_depends = prepareQuery('''
PREFIX
hshls: <http://w3id.org/HshlsOnto/>
PREFIX
owl: <http://www.w3.org/2002/07/owl#>
PREFIX
rdf:  <http://www.w3.org/1999/02/22-
rdf-syntax-ns#>
PREFIX
rdfs:  <http://www.w3.org/2000/01/rdf-
schema#>
PREFIX
xsd:
<http://www.w3.org/2001/XMLSchema#>
SELECT ?question ?dependency
WHERE {
?question rdf:type hshls:Question .
?question  hshls:dependsOn  ?dependency
.
}
 ''')
```

Execute the query and check the results:

```
results_onto_depends=
graph.query(query_onto_depends)
for row in results_onto_depends:
    print(row)
```

The result is as follows:

```
(rdflib.term.URIRef('http://w3id.org/H
shlsOnto/S02Q01a'),
rdflib.term.URIRef('http://w3id.org/Hs
hlsOnto/S02Q_Age'))
(rdflib.term.URIRef('http://w3id.org/H
shlsOnto/S02Q03'),
rdflib.term.URIRef('http://w3id.org/Hs
hlsOnto/S02Q_Age'))
(rdflib.term.URIRef('http://w3id.org/H
shlsOnto/S02Q04'),
rdflib.term.URIRef('http://w3id.org/Hs
hlsOnto/S02Q03'))
```

The interpretation of this result is the following: the question S02Q01a depends on S02Q_Age, S02Q03 depends on S02Q_Age, and S02Q04 depends on S02Q03.

The result can be further formated in a more human-readable way.

- List questions with their dependency conditions:

```
query_onto_condition = prepareQuery('
'
 PREFIX hshls: <http://w3id.org/HshlsO
nto/>
 PREFIX owl: <http://www.w3.org/2002/0
7/owl#>
PREFIX rdf: <http://www.w3.org/1999/02
/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/0
1/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XM
LSchema#>
SELECT ?question ?dependency_condition
WHERE {
        ?question rdf:type hshls:Quest
ion .
        ?question hshls:hasCondition ?
dependency_condition .
    }
''')
```

Execute the query and check the results:

```
results_onto_condition = graph.query(q
uery_onto_condition)
for row in results_onto_condition:
    print(row)
```

The result is as follows:

```
(rdflib.term.URIRef('http://w3id.org/H
shlsOnto/S02Q01a'), rdflib.term.Litera
l('( S02Q_Age >= 9)'))
(rdflib.term.URIRef('http://w3id.org/H
shlsOnto/S02Q03'),   rdflib.term.Liter
al('( S02Q_Age >= 3)'))
(rdflib.term.URIRef('http://w3id.org/H
shlsOnto/S02Q04'),   rdflib.term.Liter
al("(S02Q03 = No, never attended)"))
```

3) SPARQL queries for data consistency information retrieval: During the reasoning process, we built a class for inconsistent data, and a property dedicated to pinpointing the inconsistency. The Turtle RDF below is an example of a household member information.

```
hshls:householdmember_11_516_4_2_3
a hshls:Household_Member ;
    hshls:Name "Annan"^^xsd:string ;
    hshls:Sex "F"^^xsd:string ;
    hshls:belongsToHousehold
hshls:household_11_516_4_2 ;
    hshls:hasAnswer
hshls:answer_11_516_4_2_3_S02Q01a,
hshls:answer_11_516_4_2_3_S02Q_Age .
```

Below is one of SPARQL requests prepared and tested:

```
# Select individuals who are household members and are
marked with the inconsistency class
SELECT ?member ?label  WHERE {?member
rdf:type hshls:Household_Member ;
rdf:type hshls:Inconsistent ;
hshls:HasInconsistency ?label .}
```

In this request, we retrieve information about household members whose some information contains inconsistencies, by displaying the name of the concerned household member and their label (inconsistency details through the property hshls:HasInconsistency).

VIII. RESULTS AND DISCUSSION

A. Results

Figure 6 illustrates the graph of actual data integrated, considering one household. We can see that the household member is linked to its household he belongs to, and has answers. This makes it possible to easily query information about a particular household, since the naming is simplified. In this illustration, we demonstrate how we can input the data of a household member in a way that provides ease of navigability through survey data.

Knowing the codification structure of household member identification information allows for straightforward formulation of data queries. In the case given in this figure (Figure 6), we have a household member whose name is “Annan” and sex “F” (for Female). That household member is identified by householdmember_11_516_4_2_3. We can easily navigate through that household member’s information using his identifier.

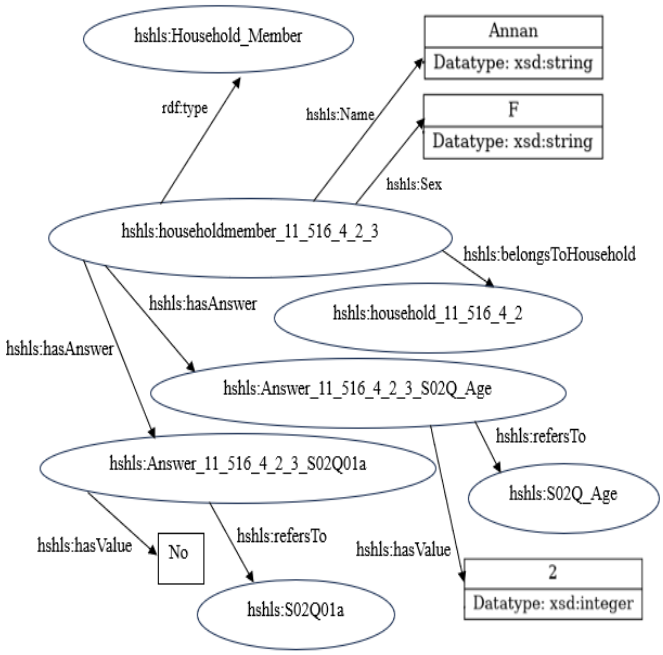

Figure 6: HSHLS graph visualization of actual data.

The following code gives an illustration of the result of the reasoning over the example of data given before. The result is saved in OWL in Turtle format (# denotes comments):

```
###
http://w3id.org/HshlsOnto/householdmem
ber_11_516_4_2_3
hshls:householdmember_11_516_4_2_3
rdf:type      owl:NamedIndividual      ,
hshls:Household_Member                 ,
hshls:Inconsistent                     ;
hshls:belongsToHousehold
hshls:household_11_516_4_2             ;
hshls:hasAnswer
hshls:answer_11_516_4_2_3_S02Q01a      ,
hshls:answer_11_516_4_2_3_S02Q_Age     ;
hshls:HasInconsistency    "Inconsistent
data. The age of this individual is
less than 3, therefore he or she must
not answer to questions on Education.
Please double check and correct the
information       accordingly."        ;
hshls:Name           "Annan"           ;
hshls:Sex "F" .
```

We can clearly see that this household member has been flagged as inconsistent, and this is in accordance with the rules we have defined. We now use the reasoning resulted graph to query information where there are inconsistencies.

For experimental purposes, we tested the performance of the reasoning engine on a small dataset of 60 household members (surveyed), described by 10 variables, including one quantitative variable (capturing the age of the surveyed) and nine qualitative variables. The data integration process took 0.5393 seconds.

Table V details the information (indicators and their respective values) sent to the reasoning engine as input.

TABLE V. STATISTICS OF INFORMATION SENT TO THE REASONING ENGINE

Indicator	Value
Number of SWRL rules exported to rule engine	9
Number of OWL class declarations exported to rule engine	9
Number of OWL individual declarations exported to rule engine	252
Number of OWL object property declarations exported to rule engine	6
Number of OWL data property declarations exported to rule engine	8
Total number of OWL axioms exported to rule engine	1321

The RDF input data model has 9 OWL class declarations, 252 OWL individual declarations, 6 OWL object property declarations, 8 OWL data property declarations, for a total number of 1321 OWL axioms. The transfer of all OWL axioms to the rule engine took 125 millisecond(s). The reasoning process took 517 millisecond(s).

The experimental dataset used contained 14 inconsistent values, and all inconsistencies were identified, resulting in a 100% success rate for the reasoner.

B. Discussion

This model makes it possible to store the methodological information of the Harmonized Survey on Households Living Standards in such a way that it can be understood by the computer and retrieved automatically. By specifying the semantics of the questions addressed, this model helps to better understand the meaning of the data manipulated in this survey as well as the semantic relationships that exist between these data. A data integration approach is also implemented. This allows inserting actual data in the model to make it possible to perform queries and reasoning on them. A reasoning process is proposed, through SWRL rules, that allows data consistency control during data collection and/or data processing. This serves as a fundamental tool for data quality control. Since the model is saved in a persistent repository, one can easily get access and perform some retrievals and analysis requests using SPARQL or any appropriate data analysis tool. Also, a large audience can get access and learn related knowledge. Therefore, the proposed solution will not only help improving the efficiency during the survey data collection and

processing activities, but also contributes to the dissemination of the survey knowledge.

This model highlights semantic information derived from the methodology of the Harmonized Survey on Households Living Standards. The reasoning system for data quality control is based on a set of SWRL rules. This allows detecting potential inconsistencies in actual data, in respect to the rules defined. However, since there are lot of rules for the entire survey, implementing all those rules would be a time-consuming task. Therefore, we propose to complement our system with a non-deterministic one: a machine learning system. Doing so, the rule-based system will not only be used for data inconsistency verification, but it will also involve in the validation process of the quality of the machine learning system to be built for the purpose of actual data quality control.

IX. CONCLUSION AND FUTURE WORK

In this work we propose a semantic data model of the Harmonized Survey on Households Living Standards, along with a way to integrate data and to query and reason over the built model. The model built makes it possible to store the semantic information contained in the methodology of this survey so that it can be consulted automatically. The results of this work can be exploited as part of the automatic retrieval of methodological information. Generally speaking, this work completes the state of the art and serves as a proof of concept to demonstrate the feasibility of documenting the knowledge contained in a statistical survey questionnaire through ontology-based semantic modeling. The work illustrates also the feasibility of data integration in an RDF-based model. The results of this work can also be used for data consistency control to improve the survey data quality. In the future, we will complement the built rule-based reasoning system with a machine learning approach, to discover hidden anomalies in actual data. As part of future work, we also propose the creation of a graphical user interface that will enable end-users to remotely query information from the model via SPARQL. This approach aims to provide an intuitive, user-friendly platform for exploring and querying ontological data without requiring users to manually interact with raw RDF files or directly write SPARQL queries.

For the model implementation, we will adopt an approach based on a knowledge base, enabling inference through an inference engine. This will be carried out using the Prolog programming language.

A program is a set of axioms, logical statements, that express the knowledge and hypotheses of the problem, and a calculation is a constructive proof of a goal based on the statements of the problem [21]. Ideally, the programmer expresses the logic of the problem to be solved, while the control is embedded or included in the interpreter of the language as such.

A logic program is a finite set of facts and rules, also called defined program clauses. A logic program is executed by assigning it a goal. Unification is the uniform mechanism for parameter passing, selection, and data construction.

Logic programming is based on the use of formal tools such as model theory and resolution theory to capture its semantics in a simple and efficient way [22]. Model theory is used to characterize the declarative semantics of the language, while resolution theory forms the basis of its operational semantics.

Prolog, as an implementation of logic programming is a powerful and simple language with a well-defined semantics, based on predicate calculus, offering several advantages such as unification and backtracking. Prolog language allows modeling knowledge using clauses (facts and rules) and inferring from this knowledge.

We will then compare the results of the two approaches, particularly in terms of performance and simplicity.

REFERENCES

[1] M. Mfoutou Moukala, M. Ngomo, and R. F. Babindamana, A Semantic Data Model of Harmonized Survey on Households Living Standards, SEMAPRO 2024 : The Eighteenth International Conference on Advances in Semantic Processing, 2024, pp. 1-7.

[2] Harmonized Survey on Households Living Standards 2018-2019, Food and Agriculture Organization of the United Nations, 2022, https://microdata.fao.org/index.php/catalog/2355/study-description (consulted on February 3, 2025).

[3] Semantic Web Rule Language. https://www.w3.org/submissions/SWRL/ (consulted on February 10, 2025).

[4] K. Munir and M. Sheraz Anjum, The use of ontologies for effective knowledge modelling and information retrieval, Applied Computing and Informatics, Vol. 14, N°2, pp. 116–126, 2018.

[5] R. Thirumahal, G. Sudha Sadasivam, and P. Shruti, Semantic Integration of Heterogeneous Data Sources Using Ontology Based Domain Knowledge Modeling for Early Detection of COVID-19, SN Computer Science, Vol. 3, No. 428, 2022, https://doi.org/10.1007/s42979-022-01298-4.

[6] I. Berges, J. Bermúdez, and A. Illarramendi, Towards Semantic Interoperability for Electronic Health Records, IEEE Trans Inf Technol Biomed, Vol. 16, N°3, pp. 424-431, 2012, doi: 10.1109/TITB.2011.2180917.

[7] N. C. Nicholson et al., An ontology-based approach for developing a harmonised data-validation tool for European cancer registration, Journal of Biomedicam semantics, Vol. 12, N°1, pp. 1-15, 2021.

[8] A. V. Borodin and Y. V. Zavyalova, An ontology-based semantic design of the survey questionnaires, 19th Conference of Open Innovations Association (FRUCT), 2016, Jyvaskyla, Finland, pp. 10-15.

[9] RDF 1.1 Primer, W3C Working Group Note 24 June 2014 https://www.w3.org/TR/rdf11-primer/, (consulted on February 3, 2025).

[10] RDF Schema 1.1, W3C Recommendation 25 February 2014, (Document updated on December, 1th 2023), https://www.w3.org/TR/rdf-schema/, (consulted on February 3, 2025)

[11] OWL2 Web Ontology Language, W3C Recommendation 11 December 2012, Document Overview (Second Edition),

https://www.w3.org/TR/owl2-overview/, (consulted on February 3, 2025).

[12] RDFLib. https://pypi.org/project/rdflib/, (consulted on February 3, 2025).

[13] PyLODE. https://pypi.org/project/pylode/, (consulted on February 10, 2025).

[14] M. Mfoutou Moukala, HSHLS survey ontology Web Page, https://moukmarc.github.io/.

[15] E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz, Pellet: A Practical OWL-DL Reasoner, Journal of Web Semantics, Vol. 5, N°2, pp. 51-53, 2007.

[16] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, HermiT: An OWL 2 Reasoner, Journal of Automated Reasoning, Vol. 53, pp. 245-269, 2014.

[17] D. Tsarkov and I. Horrocks, FaCT++ Description Logic Reasoner:System Description, 3rd International Joint conference on Automated Reasoning (IJCAR-2006), 2006, pp. 292-297.

[18] Protégé. https://protege.stanford.edu/software.php (consulted on February 3, 2025).

[19] Protégé 5 Documentation. http://protegeproject.github.io/protege/ (consulted on February 3, 2025).

[20] SPARQL 1.1 Query Language, W3C Recommendation 21 March 2013, https://www.w3.org/TR/sparql11-query/, (consulted on February 3, 2025).

[21] L. Sterling and E. Shapiro, L'Art de Prolog, Masson 1990.

[22] J. Jaffar, J.-L. Lassez, and M.J. Maher, A logic Programming Language Schemes, Logic Programming (DeGroot et Lindstrom), 1986, pp. 441-467.

APPENDIX A

Here is a part of the semantic model of the Harmonized Survey on Households Living Standards:

```
@prefix hshls: <http://w3id.org/HshlsOnto/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd:<http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix vann: <http://purl.org/vocab/vann/> .
hshls: a owl:Ontology;
   rdfs:seeAlso "https://github.com/moukmarc/HshlsOnto" ;
   dcterms:creator "Marc Mfoutou Moukala" ;
   dcterms:title "Harmonized survey on household living standards Ontology (HshlsOnto)" ;
   vann:preferredNamespacePrefix "hshls" .
# Core classes declaration
hshls:HSHLS a rdfs:Class ;
   rdfs:label " Harmonized Survey on Households Living Standards" .
hshls:Section rdf:type rdfs:Class ;
        rdfs:label " A section or module of the survey " .
hshls:Questionnaire rdf:type rdfs:Class .
hshls:Household a rdfs:Class ;
   rdfs:label " A household " .
hshls:Household_Member rdf:type rdfs:Class ;
   rdfs:label " A household member surveyed " .
hshls:Constraint rdf:type rdfs:Class ;
        rdfs:label " Constraint on the question " .
hshls:Question a rdfs:Class .
hshls:Answer a rdfs:Class .
# Properties declaration
hshls:Name a rdf:Property ;
   rdfs:domain hshls:Household_Member ;
   rdfs:range xsd:string .
hshls:Sex a rdf:Property ;
   rdfs:domain hshls:Household_Member ;
   rdfs:range xsd:string .
hshls:belongsToHousehold a rdf:Property ;
   rdfs:domain hshls:Household_Member ;
   rdfs:range hshls:Household .
hshls:clusternumber a rdf:Property ;
   rdfs:domain hshls:Household ;
   rdfs:range xsd:string .
hshls:departementNumber a rdf:Property ;
   rdfs:domain hshls:Household ;
   rdfs:range xsd:string .
hshls:hasAnswer a rdf:Property ;
   rdfs:domain hshls:Household_Member ;
   rdfs:range hshls:Answer .
hshls:hasConstraint a rdf:Property ;
   rdfs:domain hshls:Question ;
   rdfs:range hshls:Constraint .
hshls:hasQuestion a rdf:Property ;
   rdfs:domain hshls:Section ;
   rdfs:range hshls:Question .
hshls:hasSection a rdf:Property ;
   rdfs:domain hshls:HSHLS ;
   rdfs:range hshls:Section .
hshls:hasValue a rdf:Property ;
   rdfs:domain hshls:Answer ;
   rdfs:range rdfs:Literal .
hshls:householdNumber a rdf:Property ;
   rdfs:domain hshls:Household ;
   rdfs:range xsd:string .
hshls:refersTo a rdf:Property ;
   rdfs:domain hshls:Answer ;
   rdfs:range hshls:Question .
hshls:waveNumber a rdf:Property ;
   rdfs:domain hshls:Household ;
   rdfs:range xsd:string .
# Model specialization
hshls:HSHLS-C1 rdf:type hshls:HSHLS ;
        hshls:hasSection hshls:S00, hshls:S01, hshls:S02, hshls:S03, hshls:S04, hshls:S05, hshls:S06, hshls:S07,
        hshls:S08, hshls:S09, hshls:S10, hshls:S11, hshls:S12, hshls:S13, hshls:S14, hshls:S15, hshls:S16, hshls:S17,
        hshls:S18, hshls:S19, hshls:S20, hshls:S21 ;
        rdfs:comment "The different sections of HSHLS survey for the first edition in Congo named here HSHLS-C1" .
# Questions in section S02
hshls:S02 a hshls:Section ;
hshls:hasQuestion hshls:S02Q_Age, hshls:S02Q01a, hshls:S02Q01b, hshls:S02Q01c, hshls:S02Q01d,
```

```
hshls:S02Q02a, hshls:S02Q02b, hshls:S02Q02c,
hshls:S02Q02d, hshls:S02Q03a, hshls:S02Q03b,
hshls:S02Q03c, hshls:S02Q03d, hshls:S02Q04,
hshls:S02Q05, hshls:S02Q06, hshls:S02Q07,
hshls:S02Q08, hshls:S02Q09, hshls:S02Q10,
hshls:S02Q11, hshls:S02Q12, hshls:S02Q13,
hshls:S02Q14, hshls:S02Q15, hshls:S02Q16,
hshls:S02Q17, hshls:S02Q18, hshls:S02Q19,
hshls:S02Q20, hshls:S02Q21 ;
rdfs:comment "This section captures household member's education information for members of 3 years old and more" .
# Constraint for valid responses for the question S02Q01a
hshls:ConstraintS02Q01a a hshls:Constraint ;
hshls:validValues "Yes","No" .
# Constraint for valid responses for the question S02Q03
hshls:ConstraintS02Q03 a hshls:Constraint ;
hshls:validValues "Yes","No, never attended" .
# Example of survey data integrated
hshls:householdmember_11_320_2_2_1 a
hshls:Household_Member ;
   hshls:Name "Lotté"^^xsd:string ;
   hshls:Sex "M"^^xsd:string ;
   hshls:belongsToHousehold
hshls:household_11_320_2_2 ;
   hshls:hasAnswer
hshls:answer_11_320_2_2_1_S02Q01a,
      hshls:answer_11_320_2_2_1_S02Q03,
      hshls:answer_11_320_2_2_1_S02Q_Age .
hshls:householdmember_11_324_4_2_2 a
hshls:Household_Member ;
   hshls:Name "Bruno"^^xsd:string ;
   hshls:Sex "M"^^xsd:string ;
   hshls:belongsToHousehold
hshls:household_11_324_4_2 ;
   hshls:hasAnswer
hshls:answer_11_324_4_2_2_S02Q01a,
      hshls:answer_11_324_4_2_2_S02Q03,
      hshls:answer_11_324_4_2_2_S02Q_Age .
hshls:householdmember_11_516_4_2_1 a
hshls:Household_Member ;
   hshls:Name "Mouk"^^xsd:string ;
   hshls:Sex "M"^^xsd:string ;
   hshls:belongsToHousehold
hshls:household_11_516_4_2 ;
   hshls:hasAnswer
hshls:answer_11_516_4_2_1_S02Q01a,
      hshls:answer_11_516_4_2_1_S02Q_Age .
hshls:householdmember_11_516_4_2_2 a
hshls:Household_Member ;
   hshls:Name "Marc"^^xsd:string ;
   hshls:Sex "F"^^xsd:string ;
   hshls:belongsToHousehold
hshls:household_11_516_4_2 ;
   hshls:hasAnswer
hshls:answer_11_516_4_2_2_S02Q01a,
      hshls:answer_11_516_4_2_2_S02Q_Age .
hshls:householdmember_11_516_4_2_3 a
hshls:Household_Member ;
   hshls:Name "Annan"^^xsd:string ;
   hshls:Sex "F"^^xsd:string ;
   hshls:belongsToHousehold
hshls:household_11_516_4_2 ;
   hshls:hasAnswer
hshls:answer_11_516_4_2_3_S02Q01a,
      hshls:answer_11_516_4_2_3_S02Q_Age .
hshls:refersTo a rdf:Property ;
   rdfs:domain hshls:Answer ;
   rdfs:range hshls:Question .
hshls:waveNumber a rdf:Property ;
   rdfs:domain hshls:Household ;
   rdfs:range xsd:string .
hshls:answer_11_320_2_2_1_S02Q01a a hshls:Answer ;
   hshls:hasValue "Yes" ;
   hshls:refersTo hshls:S02Q01a .
hshls:answer_11_320_2_2_1_S02Q03 a hshls:Answer ;
   hshls:hasValue "Yes" ;
   hshls:refersTo hshls:S02Q03 .
hshls:answer_11_320_2_2_1_S02Q_Age a hshls:Answer ;
   hshls:hasValue 56 ;
   hshls:refersTo hshls:S02Q_Age .
hshls:answer_11_324_4_2_2_S02Q01a a hshls:Answer ;
   hshls:hasValue "No" ;
   hshls:refersTo hshls:S02Q01a .
hshls:answer_11_324_4_2_2_S02Q03 a hshls:Answer ;
   hshls:hasValue "No, never attended" ;
   hshls:refersTo hshls:S02Q03 .
hshls:answer_11_324_4_2_2_S02Q_Age a hshls:Answer ;
   hshls:hasValue 8 ;
   hshls:refersTo hshls:S02Q_Age .
hshls:answer_11_516_4_2_1_S02Q01a a hshls:Answer ;
   hshls:hasValue "No" ;
   hshls:refersTo hshls:S02Q01a .
hshls:answer_11_516_4_2_1_S02Q_Age a hshls:Answer ;
   hshls:hasValue 51 ;
   hshls:refersTo hshls:S02Q_Age .
hshls:answer_11_516_4_2_2_S02Q01a a hshls:Answer ;
   hshls:hasValue "No" ;
   hshls:refersTo hshls:S02Q01a .
hshls:answer_11_516_4_2_2_S02Q_Age a hshls:Answer ;
   hshls:hasValue 32 ;
   hshls:refersTo hshls:S02Q_Age .
hshls:answer_11_516_4_2_3_S02Q01a a hshls:Answer ;
   hshls:hasValue "No" ;
   hshls:refersTo hshls:S02Q01a .
hshls:answer_11_516_4_2_3_S02Q_Age a hshls:Answer ;
   hshls:hasValue 2 ;
   hshls:refersTo hshls:S02Q_Age .
hshls:household_11_320_2_2 a hshls:Household ;
   hshls:clusternumber "320"^^xsd:string ;
   hshls:departementNumber "11"^^xsd:string ;
   hshls:householdNumber "2"^^xsd:string ;
   hshls:waveNumber "2"^^xsd:string .
hshls:household_11_324_4_2 a hshls:Household ;
   hshls:clusternumber "324"^^xsd:string ;
   hshls:departementNumber "11"^^xsd:string ;
   hshls:householdNumber "4"^^xsd:string ;
   hshls:waveNumber "2"^^xsd:string .
```

```
hshls:household_11_516_4_2 a hshls:Household ;
   hshls:clusternumber "516"^^xsd:string ;
   hshls:departementNumber "11"^^xsd:string ;
   hshls:householdNumber "4"^^xsd:string ;
   hshls:waveNumber "2"^^xsd:string .
```

APPENDIX B

Here are some Semantic Web Rule Language rules implemented over the semantic model of the Harmonized Survey on Households Living Standards:

```
hshls:Household_Member(?HMember) ^ hshls:hasAnswer(?HMember, ?answer) ^ hshls:refersTo(?answer, hshls:S02Q_Age) ^ hshls:hasValue(?answer, ?value) ^ swrlb:lessThan(?value, 3) ^ hshls:hasAnswer(?HMember, ?answer2) ^ hshls:refersTo(?answer2, hshls:S02Q01a) -> hshls:Inconsistent(?HMember) ^ hshls:HasInconsistency(?HMember, "Inconsistent data. The age of this individual is less than 3, therefore he or she must not answer to questions on Education. Please double check and correct the information accordingly.")

hshls:Household_Member(?HMember) ^ hshls:hasAnswer(?HMember, ?answer) ^ hshls:refersTo(?answer, hshls:S02Q_Age) ^ hshls:hasValue(?answer, ?value) ^ swrlb:lessThan(?value, 3) ^ hshls:hasAnswer(?HMember, ?answer2) ^ hshls:refersTo(?answer2, hshls:S02Q01b) -> hshls:Inconsistent(?HMember) ^ hshls:HasInconsistency(?HMember, "Inconsistent data. The age of this individual is less than 3, therefore he or she must not answer to questions on Education. Please double check and correct the information accordingly.")

hshls:Household_Member(?HMember) ^ hshls:hasAnswer(?HMember, ?answer) ^ hshls:refersTo(?answer, hshls:S02Q_Age) ^ hshls:hasValue(?answer, ?value) ^ swrlb:lessThan(?value, 3) ^ hshls:hasAnswer(?HMember, ?answer2) ^ hshls:refersTo(?answer2, hshls:S02Q01c) -> hshls:Inconsistent(?HMember) ^ hshls:HasInconsistency(?HMember, "Inconsistent data. The age of this individual is less than 3, therefore he or she must not answer to questions on Education. Please double check and correct the information accordingly.")

hshls:Household_Member(?HMember) ^ hshls:hasAnswer(?HMember, ?answer) ^ hshls:refersTo(?answer, hshls:S02Q_Age) ^ hshls:hasValue(?answer, ?value) ^ swrlb:lessThan(?value, 9) ^ hshls:hasAnswer(?HMember, ?answer2) ^ hshls:refersTo(?answer2, hshls:S02Q01a) -> hshls:Inconsistent(?HMember) ^ hshls:HasInconsistency(?HMember, "Inconsistent data. The age of this individual is less than 9, therefore he or she must not answer to questions on Languages. Please double check and correct the information accordingly.")
```