An OWL Ontology for the Generic Statistical Information Model (GSIM): Design and Implementation

Antoine Dreyer[1], Guillaume Duffes[1], Daniel Gillman[2], Monica Scannapieco[3], Laura Tosco[3]

[1] INSEE – National Institute of Statistics and Economic Studies
(antoine.dreyer, guillaume.duffes)@insee.fr
[2] BLS - U.S. Bureau of Labor Statistics
Gillman.Daniel@bls.gov
[3] ISTAT – Italian National Institute of Statistics
(monica.scannapieco,laura.tosco)@istat.it

Abstract. In this paper, we propose an OWL ontology for GSIM – the Generic Statistical Information Model, which is a framework for statistical metadata promoted by the UNECE and internationally endorsed by statistical organizations. After illustrating some use cases on possible different uses of the GSIM ontology, we detail the design and development process followed to generate it.

1 Introduction

GSIM – Generic Statistical Information Model – is an internationally endorsed reference framework for statistical information, which enables generic descriptions of the definition, management, and use of data and metadata throughout the statistical production process. It has been increasingly adopted by national statistical institutes over the past years as their reference conceptual model.

Linked Data and Semantic Web standards are also being progressively adopted by statistical organizations. For instance, in 2016 UNECE (United Nations Economic Commission for Europe) established a project [1] that prescribes the use of Linked Data for modeling statistical metadata as one of one of its principal aims.

In this paper, we propose an OWL ontology for GSIM. There are multiple benefits to this effort, including having a formal and machine-actionable representation of a statistical model, allowing consistency checks among different models, and promoting interoperability across statistical conceptual models.

The rest of the paper is organized as follows: Section 2 introduces GSIM basics; Section 3 illustrates some use cases of the GSIM ontology; Section 4 provides the detail of how the GSIM ontology was generated; and, finally, Section 5 provides a conclusion and a description of future work.

2 Background: What is GSIM?

The text and figures in this section are adapted from the GSIM Communication Paper on the Generic Statistical Information Model (GSIM) website [2] under the UNECE (United Nations Economic Commission for Europe) [3].

GSIM provides a set of standardized, consistently described classes, which are the inputs and outputs in the design and production of statistics. The design and production processes for which GSIM defined objects are the inputs and outputs are themselves defined in the Generic Statistical Business Process Model (GSBPM) [4]. As such, GSIM and GSBPM are dual models of statistical production and data.

In general, information and process models describing the same domain are duals of each other in the sense that the classes in one model are the relationships in the other. Though GSIM and GSBPM are not designed to directly convey this connectedness, as GSBPM is more of an outline than a model, they nevertheless imply it. More details about how GSIM and GSBPM are inter-connected follow below.

GSIM does not include classes related to business functions within an organization such as human resources, finance, or legal functions, except to the extent that this information is used directly in statistical production.

At the highest level, GSIM is designed and was developed in four sections. These four top-level groups are described below:

- The **Business** group is used to capture the designs and plans of statistical programs, and the processes undertaken to deliver those programs. This includes the identification of a Statistical Need, the Business Processes that compose the Statistical Program, and the evaluations of them.
- The **Exchange** group is used to catalogue the information that comes in and out of a statistical organization via Exchange Channels. It includes classes that describe the collection and dissemination of information.
- The **Concepts** group is used to define the meaning of data, providing an understanding of what the data are measuring.
- The **Structures** group is used to describe and define the terms used in relation to structures for organizing data.

Figure 1 gives an example of GSIM classes that tell a story about some of the information that is important in a statistical organization. In particular:

- "A statistical organization initiates a *Statistical Program*. The *Statistical Program* corresponds to an ongoing activity such as a survey or an output series and has a *Statistical Program Cycle* (for example it repeats quarterly or annually).
- The *Statistical Program Cycle* will include a set of *Business Processes.* The *Business Processes* consist of a number of *Process Steps* which are specified by a *Process Design.* These *Process Designs* have *Process Input Specifications* and *Process Output Specifications.*
- The specifications will often be pieces of information that refer to Concepts and Structures (for example, *Statistical Classification, Variable, Population, Data Structure,* and *Data Set*). If, for example, the *Business Process* is related to the col-

lection of data, there will be an *Information Provider* who agrees to provide the statistical organisation with data (via a *Provision Agreement*). This *Provision Agreement* specifies an agreed *Data Structure* and governs the *Exchange Channel* used for the incoming information. The *Exchange Channel* could be a *Questionnaire* or an *Administrative Register*. It will receive the information via a particular mechanism (*Protocol*) such as an interview or a data file exchange.

- The *Data Set* produced by the *Exchange Channel* will be stored in a *Data Resource* and structured by a *Data Structure*."

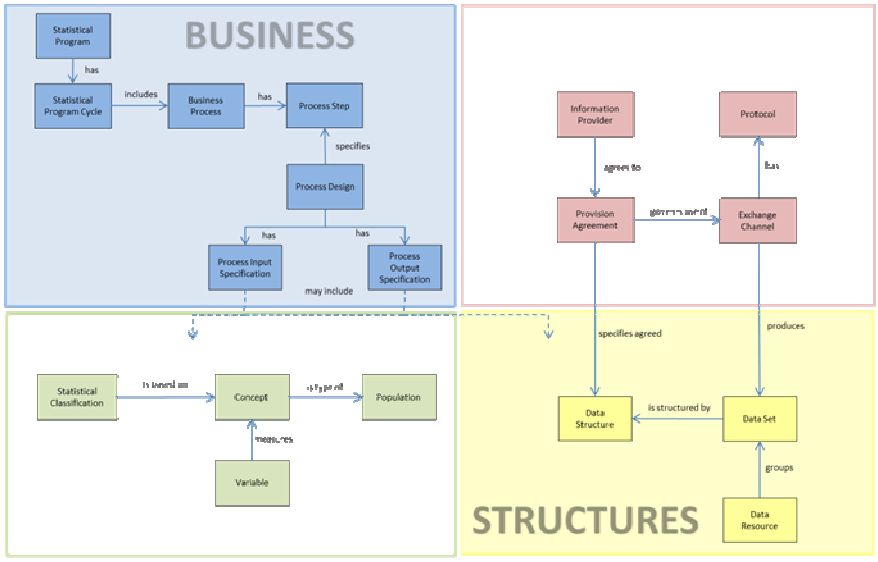

Figure 1: GSIM information objects

As described above, GSIM and GSBPM are dual models for the production and management of statistical data. GSBPM models the statistical production process and identifies the activities undertaken by producers of official statistics that result in information outputs. These activities are broken down into sub-processes, such as "Impute" and "Calculate aggregates". As shown in **Figure 2**, GSIM helps describe GSBPM sub-processes by defining the objects that flow between them, that are created in them, and that are used by them to produce official statistics.

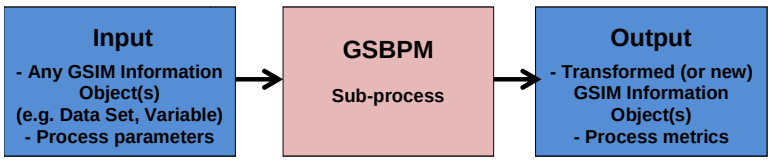

Figure 2: GSIM and GSBPM

Greater value will be obtained from GSIM if it is applied in conjunction with GSBPM. Likewise, greater value will be obtained from GSBPM if it is applied in conjunction with GSIM. Nevertheless, it is possible (although not ideal) to apply one without the other. In the same way that individual statistical business processes do not use all of the sub-processes described within GSBPM, it is very unlikely that all classes in the GSIM will be needed in any specific statistical business process.

Good metadata management is essential for the efficient operation of statistical business processes. Metadata are present in every phase of GSBPM, either created, updated, or carried forward unchanged from a previous phase. In the context of GSBPM, the emphasis of the over-arching process of metadata management is on the creation, updating, use, and reuse of metadata. Metadata management strategies and systems are therefore vital to the operation of GSBPM, and are facilitated by GSIM.

Applying GSIM together with GSBPM (or an organization-specific equivalent) can facilitate the building of efficient metadata driven collection, processing, and dissemination systems, and help harmonize statistical computing infrastructures.

GSIM supports a consistent approach to metadata, facilitating the primary role for metadata envisaged in Part A of the Common Metadata Framework "Statistical Metadata in a Corporate Context" [5], that is, metadata should uniquely and formally define the content and links between objects and processes in the information system.

3 Use Cases

This section contains descriptions of some possible use cases for implementing the GSIM OWL ontology within National Statistical Offices (NSOs). Linked Data standards permit the expression of data and metadata according to machine-actionable, formal, and interoperable models. NSOs on the other hand need to share common metadata models, as witnessed by proposals such as GSIM. Improving information systems supporting statistical production is made possible by representing such models with Linked Data standards, as shown by the following: the use case described in Section 3.1 shows the usage of the GSIM ontology to improve a central system for metadata representation in Istat (the Italian National Institute of Statistics); the use case described in Section 3.2 shows how the GSIM ontology enables interoperability between statistical models.

3.1 Using GSIM OWL ontology to query data and metadata within NSOs

The GSIM representations provided so far, such as the GSIM UML representation, have been mainly used as a source of documentation on a common and shared model. In addition, the GSIM ontology can be used as a model for data and metadata representation, thus enabling direct queries on data and metadata.

Istat currently has a unitary metadata system that centralizes metadata used within its statistical production processes. This system has an underlying relational database that has been designed according to GSIM. This means that there are relational tables for representing classifications, code lists, statistical variables etc. This system could

be integrated with the GSIM OWL ontology by mapping the ontology to those data that are currently in that relational database. The following specific benefits can be identified as a result of such an integration:

1. Decoupling the conceptual representation of GSIM from the logical data storage implemented by the unitary metadata system, and in this way be able to (i) make queries on data directly specified in the GSIM standard and (ii) easily align data if modifications to the GSIM standard occur overtime.
2. Ability to easily make metadata queries, e.g. *Which are the "gsim:process steps" that a given "gsim:business process" X "gsim:has"?*

3.2 Interoperability between GSIM and GSBPM ontologies

An additional use of the GSIM ontology is to map from one statistical model to another, establishing interoperability across models. As mentioned in Section 2, GSBPM is the model adopted for representing the phases of Official Statistics production. A GSBPM ontology has been already proposed in [6].

As will be described in Section 4, there are some concepts of the GSIM ontology that have been identified as "equivalent to" some concepts of the GSBPM ontology.

As an example, "gsim:business process" is "equivalent to" "gsbpm:statistical production activity". But notice instead, "gsim:process step" is NOT "equivalent to" "gsbpm:phase". Indeed, GSBPM does not imply a sequencing of the "gsbpm:phases", instead the sequence is implied by "gsim:process step".

Now let us suppose that there is a database (e.g. triple-store) with data described according to the GSIM ontology and to the GSBPM ontology. A user could in this case pose the following query exploiting both ontologies:

For a given "gsbpm:statistical production activity" Y , which is the set of "gsim:business service" "gsim:used by" "gsim:businessprocess"X (equivalent to Y)?

This use case illustrates how two statistical models proposed independently, GSIM for statistical information and GSBPM for statistical processes, can be integrated. The main benefits of such an integration are: (i) the ability for joint use of the models when developing systems based on them and (ii) the possibility of checking and monitoring the consistency of modifications that are proposed to the standards over time.

4 Building the GSIM Ontology

In the GSIM official documentation, UML (Unified Modeling Language) is used to visually represent the information model. UML is based on the object-oriented paradigm, thus it employs classes, relationships, and attributes for representing a model.

Starting from the UML for GSIM, we defined a GSIM ontology by expressing it in OWL. OWL is a knowledge representation language with the following basic notions:

(i) *Axioms* – the basic assumptions that an OWL ontology expresses; (ii) *Entities* – elements used to refer to real-world objects; and (iii) *Expressions* – combinations of entities that form complex descriptions from basic ones. Entities are all atomic constituents of statements: objects, categories, or relations. In OWL objects are denoted as *individuals*, categories as *classes*, and relations as *properties*. Properties are further subdivided into *Object properties* that relate objects to objects, and *Datatype properties* that assign data values to objects. Finally, *Annotation properties* encode information about the ontology itself instead of the domain of interest.

In the creation process of the GSIM ontology, the UML classes have been mapped into OWL Classes, relations into Object properties, and attributes into Datatype properties. The subclasses of UML schemas have been modeled with the OWL object property called *subClassOf* that relates the main class with its subclasses.

Given the differences between the two languages described above, there is not always a direct correspondence between their respective elements. As an example, an abstract class in UML does not have a direct correspondent with any OWL class because the notion of what belongs to a class in OWL is more fluid. Additional statements on Classes, ObjectProperties, and DataProperties (e.g. *objectIntersectionOf*, *disjointClasses*) allow representing the domain of interest more fully.

In the following sections we describe the two different approaches we used to build the GSIM ontology. In the first approach, we started from the official specification document of the GSIM model [7] rendered in UML, and we modeled the classes and the properties defined therein in OWL; in the second approach, we started from the UML models and exported them into XMI (XML Metadata Interchange[1]) files. The results of the two processes were compared against each other in order to reason about inconsistencies. Then, the effort was made to resolve those inconsistencies.

4.1 First approach: starting from the Generic Statistical Information Model Specification Document

We started by studying the specification document of the GSIM model. From this we deduced which concepts are to be represented in the ontology, which properties link the concepts to each other, and what the data properties of the concepts are.

As summarized in Section 2, GSIM concepts are grouped in five main topics areas: (i) Base, (ii) Business, (iii) Concepts, (iv) Exchange, and (v) Structures. In the ontology, we maintained this structure by defining five Classes with those names (see **Figure 3**); in each Class we defined sub-classes for each concept contained in the topic.

[1] http://www.omg.org/spec/XMI/

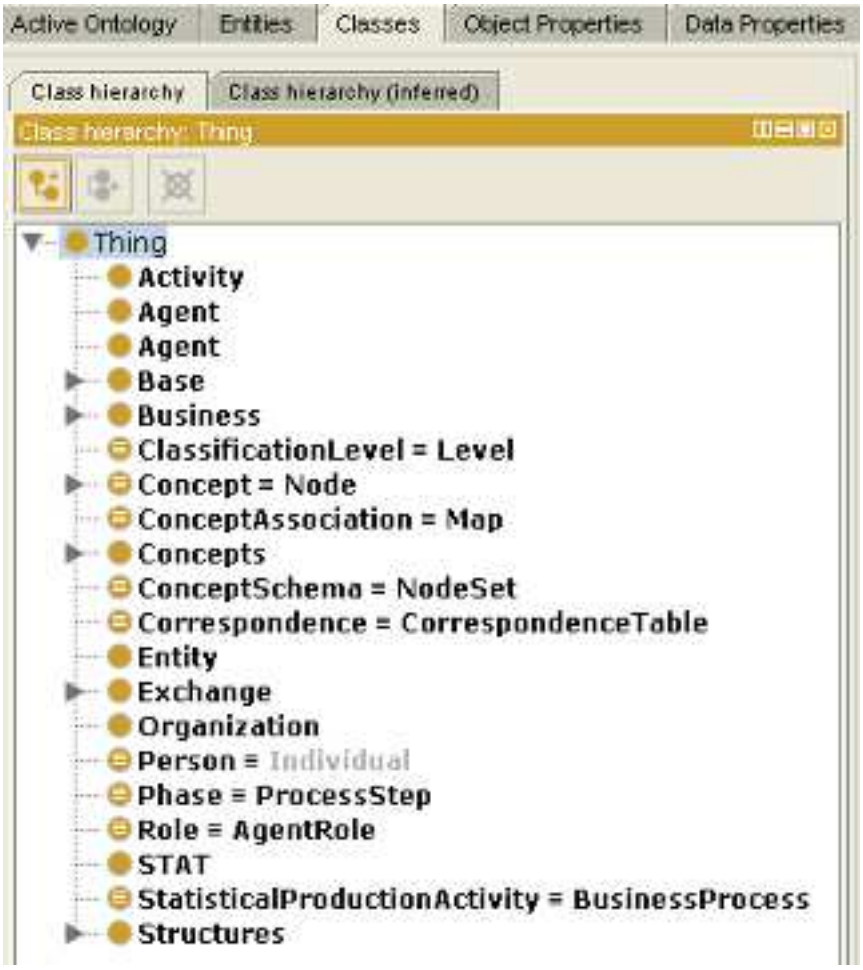

Figure 3: Class Definition shown with Protégé

In the GSIM specification document, the relationships between concepts often have the same name as the attributes of the concepts. As an example in **Figure 4**, two different concepts, Information Provider and Information Consumer, use the same relation "agrees to" in their relationships with the concept Provision Agreement. In the ontology representation, the concepts Information Provider and Information Consumer are the "domains" of the relations "agrees to" and the concept Provision Agreement is the common range. However "agrees to" must become two different properties in the ontology otherwise, it is implicitly assumed an intersection of the domains, which is not true in general.

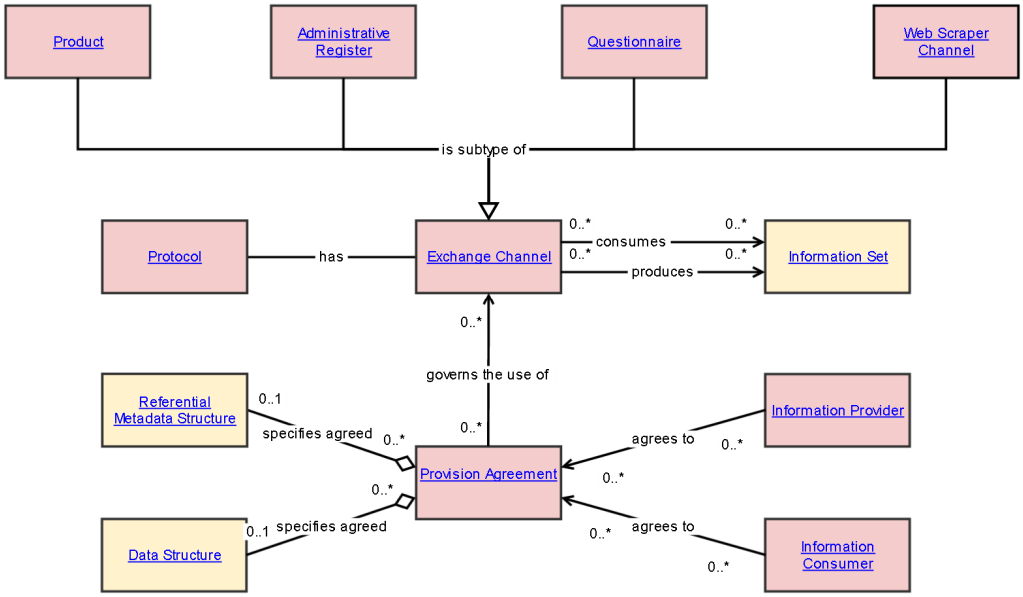

Figure 4: GSIM UML Schema

To distinguish the properties we adopted the following notation (see also **Figure 5**):

Property Name || first letters of Domain || first letters of Range

The notation is valid for Object Properties, while in the case of Data Properties the "first letters of Range" are missing.

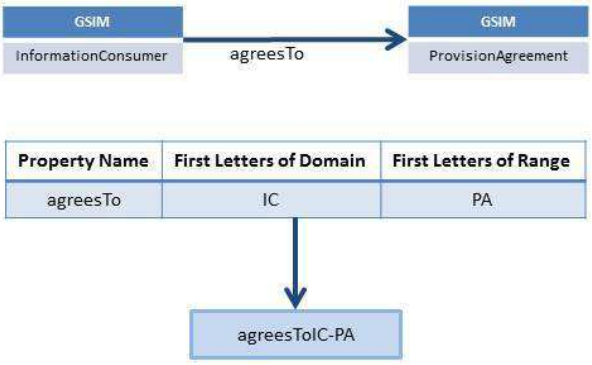

Figure 5: Example of the notation

Table 1 summarizes the number of classes, properties, and axioms we defined. We also defined equivalence with concepts of other ontologies as shown in **Table 2**.

Concept	Number
Classes	134
Object Properties	203
Data Properties	383
subClassOf Axioms	117
equivalentClass Axioms	14
ObjectPropertyDomain Axioms	204
ObjectPropertyRange Axiom	240
FunctionalDataProperty Axioms	378
DatapropertyDomain Axioms	378
DataPropertyDomain Axioms	382

Table 1. Number of classes, properties, and axioms defined in the GSIM ontology

GSIM Concept	Other Ontology Concept	
	Ontology	Concept
Level	xkos	ClassificationLevel
Node	skos	Concept
Map	xkos	ConceptAssociation
NodeSet	skos	ConceptSchema
CorrespondenceTable	xkos	Correspondence
Individual	foaf	Person
AgentRole	prov	Role
BusinessProcess	gsbpm	StatisticalProductionActivity

Table 2. Equivalence between GSIM concepts and concepts of other ontologies

The resulting ontology has a DL expressivity *ALCIRQ(D)* ([8]) according to Protégé.

4.2 Second approach: starting from UML schemas

Introduction

Through the use of UML modeling with Enterprise Architect (EA), GSIM is already fully specified in machine-actionable format. Since the approach described previously showed that GSIM could be expressed in OWL, the next step was to translate directly and automatically from EA to RDF. We tried to create a fully automated procedure, so that it could accommodate future GSIM modifications.

The method used here to transform UML to RDF is an *ad hoc* XSL transformation, which is direct, efficient, and simple, but depends on the way UML is used. We could do it because the GSIM specification is coherent and well-written. Other approaches could have been used, for example Model Driven Architecture based solutions [9] or the Ontology Definition Metamodel[2], a standard defined by the Object Management

[2] http://www.omg.org/spec/ODM/1.1/.

Group (OMG) and implemented in EA[3], which extends UML with additional modeling notations to allow representation in OWL. Since GSIM uses only simple UML constructs, we felt that those approaches were too complex in this specific case.

From UML to XML (XMI) to simplified XML

The first step was to export the UML description from EA (version 10) to a file in the XMI 2.1 format, which is the standard created by the OMG for expressing UML in XML. We then used an existing XSL transformation[4] (XSLT) provided by the UNECE to extract the relevant information in a simpler and more convenient XML format. From there, we wrote the XSL transformation to produce RDF/XML.

From XML to RDF for classes and packages

The UML and RDF concepts for classes are very close, so no adaptation was needed. Basically, we did a one-to-one mapping. However, since the UML model was divided into packages corresponding to the different GSIM groups described above, we adopted the same method as in the previous approach and made all classes of a package sub-classes of a class representing the package (Structure, Business, etc.).

From XML to RDF for properties: first part

UML attributes and relationships are both represented by, and transformed into, RDF properties. RDF distinguishes between annotation, data, and object properties, depending on the type of their range (we did not use annotation properties).

Most parts of UML attributes and relationships are easy to transform into RDF. Only the nature of the relationships between classes (associations, compositions, etc.) was not used. The approach described in 4.1 proved that this was not necessary, and we felt the complexity added in taking them into account was not worth it.

The domain of a property (the class described by the attribute or relationship) is always known, by construction of the UML. Cardinality restrictions are specified in UML the same way as in OWL, even if zero-minimum and n-maximum cardinality restrictions need not be specified in OWL. The range of a relationship (the class it points to) is found by a one-to-one mapping, but the range of an attribute cannot always be kept as is. If the original attribute range cannot be mapped to a known class or type (binary to xs:boolean for instance), we transformed it to a xs:string.

Among the UML attributes are also three types of comments (Definition, Explanatory text, and Synonyms), which we transformed into corresponding RDF properties.

From XML to RDF for properties: difficulty to find a name

The name part of UML attributes and relationships is much harder to transform into RDF, since in UML the name is a tag, whereas in RDF it must uniquely identify the property. The conversion between UML and RDF is not straightforward, because UML attributes are parts of one class and UML relationships exist only to connect classes, whereas RDF properties are first-level objects by themselves. That is why we had to build a clear algorithm to construct unique names for properties.

[3] http://www.sparxsystems.com/enterprise_architect_user_guide/9.2/domain_based_models/mdg_technology_for_odm.html

[4] https://github.com/FranckCo/GSIM-SPAP/blob/master/transformations/read-xmi.xsl

A simple choice would have been to create one property for each attribute and each relationship, as in the previous approach, but this raises a problem by creating redundant properties. A good example is the "name" attribute that many UML classes have: all of them link the property to a character string, and most of them have the same cardinality restriction (at most one name is possible). Creating only one property for those cases is more desirable, because someone wanting to know the label of a class would only have to query its "name", but there is a risk of merging relationships or attributes where it is not appropriate. Those accidents produce weird property domains or ranges that are easily spotted in the resulting ontology. They can thus be reported back to the GSIM designers in order to be fixed directly in the UML model.

Detailed process to obtain RDF properties

In this section, we describe the algorithm used to decide whether two or more attributes or relationships of the same name can be grouped into one RDF property. Apart from the name, the decision criteria are range, domain, range cardinality restrictions, and domain cardinality restrictions.

When several attributes have to be merged into one property, different non-empty comments are merged by concatenating the domain of the attribute and the comment.

To merge properties together, the following decision tree is used:

Is original attribute/link name unique?

- [Yes] one property
- [No] is the original name with source name and destination name unique?
  - [Yes] one property
  - [No] is the combination original name / range name unique?
    - [Yes] one property
    - [No] is the cardinality restriction on range unique?
      - [Yes] one property with a union of domains
      - [No] since in the original file, the triple property range domain is unique, we build for each triple a property name including the names of property range domain.

The implementation in XSLT is too long to be included here but the interested reader can find the complete transformation on GitHub [9].

5 Conclusions

In this paper, we presented a preliminary effort to represent the Generic Statistical Information Model as an RDF ontology. This ontology resulted from a two-pronged approach: (i) a human-based approach that coded the ontology manually and that was basically used as a benchmark to evaluate (ii) an automated approach that transformed the XML format (XMI) exported from the existing UML model into an OWL ontology. While the automation of the ontology creation in the second approach allows significant efficiency gains, its reusability in other contexts is still to be assessed.

The resulting ontology also illustrated how the UML version of GSIM contains some incoherencies or incompatibilities with the rules defined by OWL. The type of some attributes and relationship names were two such examples. Additionally, some modeling choices had to be made, and they we will be submitted to the GSIM man-

agement team for discussion and potential update. Nevertheless, we are convinced that OWL enhances the semantic coherence of the model and should continue to be used for representing the next GSIM versions as well as the other statistical models.

The work ahead must be put in the perspective of the international collaborations going on. The Common Statistical Production Architecture (CSPA) is an effort to standardize production processes across all statistical offices, and work is currently carried on to produce an OWL ontology for CSPA, as well as for the GSBPM as already mentioned. In CSPA, production systems are built up from small modules that can be shared. Hence, the GSBPM will guide the development of which modules must be built, and GSIM will guide which metadata are inputs and outputs to each process.

The current SDMX and DDI specifications are indeed potential implementations of GSIM. The next model-driven DDI version will not change the status of the DDI standard: it will remain an implementation model. There is no mapping between GSIM and DDI currently, but building in parallel an OWL ontology for GSIM and a DDI information model expressed also in OWL should facilitate the mapping between both without worrying about the practical limitations of the automatic transformation from XML to RDF as described in this paper. For instance the next version of DDI, which aims to achieve a more complete coverage of GSIM could directly reference the GSIM OWL classes and properties since they are expressed in a RDF syntax.

The same approach could be applied to SDMX. The RDF Data Cube vocabulary covers only a very partial part of the SDMX information model. However, a next version of the Data Cube could possibly map some of its classes/properties to GSIM ones, and potentially to DDI by transitivity.

As these standards are formalized as linked metadata on both conceptual and implementation sides, it will be possible to move towards more integration and a more formalized semantics of statistical production within and across statistical offices.

6 References

1. Implementing ModernStats Standards Project: http://www1.unece.org/stat/platform/display/hlgbas/Implementing+Modernstats+Standards
2. GSIM website: http://www1.unece.org/stat/platform/display/gsim/Generic+Statistical+Information+Model
3. UNECE website: http://www1.unece.org
4. GSBPM website: www.unece.org/stats/gsbpm
5. Common Metadata Framework: http://www1.unece.org/stat/platform/display/metis/The+Common+Metadata+Framework
6. Cotton F., Gillman D.: “Modeling the Statistical Process with Linked Metadata”, SemStats 2015, available at http://ceur-ws.org/Vol-1551/article-06.pdf
7. GSIM specification: http://www1.unece.org/stat/platform/display/gsim/GSIM+Specification
8. Description Logic: https://en.wikipedia.org/wiki/Description_logic
9. Dragan Đurić, MDA-based Ontology Infrastructure, http://www.comsis.org/pdf.php?id=nnn-1105
10. XSLT transformation: https://github.com/FranckCo/Stamina/blob/master/src/xsl/gsim-xmi-to-owl.xsl.