

Improving Data Quality by Rules: A Numismatic Example

Karsten Tolle

Databases and Information
Systems
Johann Wolfgang Goethe-
University of Frankfurt
tolle@dbis.cs.uni-frankfurt.de

David Wigg-Wolf

Römisch-Germanische
Kommission des Deutschen
Archäologischen Instituts
David.Wigg-Wolf@dainst.de

Abstract

The archaeological data dealt with in our database solution Antike Fundmünzen in Europa (AFE), which records finds of ancient coins, is entered by humans. Based on the Linked Open Data (LOD) approach, we link our data to Nomisma.org concepts, as well as to other resources like Online Coins of the Roman Empire (OCRE). Since information such as denomination, material, etc. is recorded for each single coin, this information should be identical for coins of the same type. Unfortunately, this is not always the case, mostly due to human errors. Based on rules that we implemented, we were able to make use of this redundant information in order to detect possible errors within AFE, and were even able to correct errors in Nomimsa.org. However, the approach had the weakness that it was necessary to transform the data into an internal data model. In a second step, we therefore developed our rules within the Linked Open Data world. The rules can now be applied to datasets following the Nomisma.org modelling approach, as we demonstrated with data held by Corpus Nummorum Thracorum (CNT). We believe that the use of methods like this to increase the data quality of individual databases, as well as across different data sources and up to the higher levels of OCRE and Nomisma.org, is mandatory in order to increase trust in them.

Keywords: data quality, SWRL, uncertainty

Introduction

As is reflected in the title of this paper, our work has concentrated on the field of digital numismatics. The field already has a long history, and involves dealing with legacy systems and data that were often compiled over many decades by many different authorities. The challenge, then, is how to adopt and make use of new approaches such as Linked Open Data (LOD) without compromising existing systems. Consequently this paper also reflects our experiences in applying LOD in this context. For most of the issues discussed, solutions exist, but changing or rebuilding legacy systems to optimise them is in most cases not an option.

Data quality is often not felt to be as important as

we think it should be. For many, especially managers, the addition of 100 new datasets sounds far more impressive than eliminating 20 errors. For example, in one case that we encountered, those entering data on coin finds into a national database were aware that they had six (!) different ways of entering the name of the Roman emperor Caligula, which severely restricted the validity of search results. Yet for a long time no effort was taken to rectify this despite the fact that the proprietary software being used very much simplified the use of standard vocabularies. But the truth is that, without knowing further details, it is impossible to decide which of the two tasks, adding 100 new datasets or eliminating 20 errors, would be more difficult or take more time. However, in our experience, we can state that dealing with data qual-

ity can sometimes bring very useful, if unforeseen, benefits. This is epitomised in a statement by Thomas Carlyle (1795-1881): “Do not be embarrassed by your mistakes. Nothing can teach us better than our understanding of them. This is one of the best ways of self-education.”

Currently many archaeologists have their own datasets, and these are what they trust. But how often are these data checked for mistakes or inconsistencies? How can it be done? How is it possible to ensure that no dirty data gets into a system? With new approaches like Linked Open Data (LOD) (Berners-Lee, 2006), data are becoming publicly available, and the question arises as to whether these data are always consistent and how this can be checked? Our approach is to define logical rules and implement them in a corresponding system in order to allow automated testing. These tests could function as a gatekeeper before allowing data to enter a system, or as a regular check performed after a certain time-span. However, this method will, of course, only find errors for which there are rules. Furthermore, the focus of this paper is on detecting errors and not on their correction, which is a much more complex and difficult matter.

Why is this important? Because data are used to confirm or reject hypotheses. If data are incorrect, or inconsistent and messy, this could lead to incorrect results. A second hypothesis may be based on the first one, and when errors are discovered then the entire hypotheses based on the first one are immediately invalidated. This does not necessarily mean that the hypothesis actually was in any way invalid, but people may no longer trust it (or the person who proposed it). In a Big Data approach, it could be claimed that masses of data are used in order to overrule such errors statistically. Unfortunately, this is not always true, and in the case discussed we do not even have the necessary mass of data (of a stable quality). In the domain of archaeology, with many uncertainties and lost information, good data quality is required to ensure that at least the facts that are taken into account are correct.

What is more, with the LOD approach errors may get copied, replicated, or multiplied. Once this has occurred, rectification becomes ever more complicated. In an ideal world this kind of error propagation would not be dramatic, for systems would simply link to the LOD content and once the error

is removed the problem would be solved, except in those cases where the link was made based on the erroneous data. However, LOD is open and used in various ways, and in many cases the provenance link to the source information gets removed or is omitted, so that correcting the LOD will not automatically correct the other systems.

The Current Situation in the Field of Numismatics

Since coins are a mass-produced medium, with more or less standard core data, compared to other categories of material culture, it was relatively easy to define and set up discipline-specific and stable digital representations of the concepts needed for the numismatic field. These representations take the form of HTTP URIs that promote worldwide interoperability between numismatic resources by providing access to reusable information about the concepts, as well as links to and from other fields of study. This way it follows the LOD approach, and also provides its own ontology. But while the work may indeed have been “relatively easy”, it must nevertheless be stressed that the *Nomisma.org* project has already taken seven years to deliver the present volume of LOD concepts and that the work is still ongoing.

Projects can provide their descriptions of numismatic objects using the *Nomisma.org* ontology and the modelling described there (Nomisma.org, n.d. b). These data sets are then published by web projects that are based on *Nomisma.org*. By December 2018, such websites were publishing data sets from 33 institutions containing more than 280,000 descriptions of coins, coin types, and finds of various kinds.

For certain fields of numismatics, there are widely established typologies. This is especially true for the Roman Imperial Coinage and the Roman Republican Coinage, as well as for the coins issued in the name of Alexander the Great. For these, digital type corpora are already online, namely: OCRE (Nomisma.org, n.d. c), a digital type corpus based on Roman Imperial Coinage (RIC), CRRO (Nomisma.org, n.d. a) based on Roman Republican Coinage, and PELLA (Nomisma.org, n.d. d) for the coinage in the name of Alexander. All these online resources use the *Nomisma.org* ontology, and make use of the data sets submitted via *Nomisma.org*. Thus, where a coin from

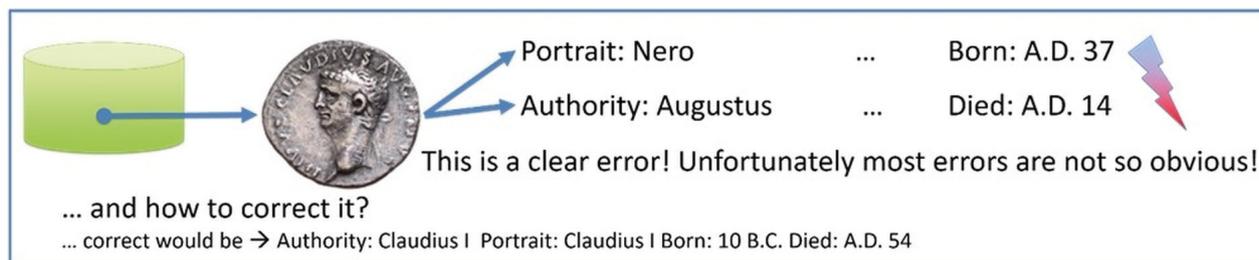


Figure 1. An example of a clear logical error of the kind that generally cannot be corrected automatically.

one of the data sets that employs the concepts in *Nomisma.org* is linked to a specific coin type in one of the typologies, it is listed in OCRE, etc. as an example of the type.

In a *relational* coin find database, some of the coins will be well preserved and easily identifiable, and so can be easily linked to an existing type corpus such as OCRE. In this case, the link to OCRE would be enough to specify information such as the authority, legend, mint and other type-relevant information, and the individual data on these need not be entered separately into the relational database. However, there are coins that are not so well preserved, and which cannot be attributed to a specific type, and for these it is necessary to store all the type-relevant information separately. When the data are analysed, it will be necessary to include all of this information, and if it has not already been done, it will be necessary to import the relevant information from OCRE. For this reason, many relational databases also store the type-relevant data for coins even where a link to the type in OCRE, etc. exists. This is partly due to the fact that normally several different type corpora have to be dealt with, and it is undesirable to reproduce them all within a database. The result is redundant information, something that normally should be avoided for it means that there is more than one place in which the same information is stored; and when later it is changed in only one place, the information will no longer be consistent. This is the case described here: on the one hand there is the link to the type corpora with its type-relevant information, but at the same time the same information is stored separately in the database. But where provenance information for the data is stored, it is possible to build systems in a way that keep track of the data and ensure consistency across them. However, in relational databases this does not come for free, and would need to be implemented.

In addition, many databases have legacy data,

dating back to before the existence of OCRE, CRRO, etc. When the data were created, it was normal to store the type-relevant information in the individual coin records. But if a link to the new type corpora such as OCRE is added, redundant information is generated. However, in this case redundant information can also be useful, because once it is known to be redundant, it is possible to test if it is identical, and therefore if it is consistent. If it is not, then there is an error that has to be corrected. This is very similar to a checksum for IBAN numbers, etc. As defined in the next section, many rules are based on such redundant data.

It should also be noted that such checks not only ensure data consistency within a database, but when data are checked against other projects, this sometimes leads to the discovery of errors within external projects. With widely used LOD resources, such errors can result in exponential replication. In several cases we have identified such errors in the course of our work: the Portable Antiquities scheme at the British Museum had linked a late Roman mint to a town of the same name in a different location, while within *Nomisma.org* the apparent copying of blocks of text from one entry to another, without changing relevant individual elements, had resulted in the wrong dates being given for a Roman emperor.

Defining and Testing Rules

Rules are categorised here in two groups. Those based on redundant data, as described in the previous section; and pure logical rules, such as conflicts in the time line. A simple example for a logical rule could involve a coin showing a portrait of a different person to the issuing authority, such as a coin struck by an emperor for another member of the imperial family. When the person in the portrait is known to have been born after the death of the authority,

ID	Name	Description	Type
1	Issuer-Chronology	The production time span of a coin must fall within the reign of the assigned authority.	logic
2	Issuer-Portrait-Chronology	The person in the portrait must be born before the death of the authority.	logic
3	Denomination-Material	The denomination implies the material used (there could be exceptions to this rule).	redundant
4	Min-Max-Diameter	In case minimal and maximal diameter are provided, the minimal diameter should be smaller or equal to the maximal diameter.	logic
5	Date_From_To	The production period is normally given by the years defining the start (from) until then end of production (to). These two need to have the correct order.	logic
6	Compare_Types_Local	All coins within the database linked to the same coin type, should be equal with regard to the type-relevant information.	redundant
7	Compare_Types_Remote	The locally entered information should be equal to the type-relevant information provided remotely by a type corpora.	redundant
8	Weight	In many cases the weight of a coin is related to the material and diameter. However, since the thickness is generally not recorded for ancient coins, weight comparisons would provide no more than an indication. Coins of the same material and same diameter should be similar, that is within a defined range. In addition, the condition of a coin has an impact (e.g. corrosion).	logic

Table 1. A high-level description of some of the rules we defined. In reality, some of the rules were divided into a number of sub-rules and additional issues checked that are not mentioned in the description of this table.

there is clearly a time line conflict. Without a time machine, this would simply be impossible. Such a scenario is visualized in Figure 1.

Table 1 shows some examples of the rules at a high-level of description that we implemented and used. Some rules are further divided into sub-rules, and may take into account additional issues that are not part of the description in the table. Some of these need to be fine-tuned according to individual cases. For example, rules 6 and 7 depend on the type system being used. Many existing type systems define type-relevant information, and this information should be equal for all coins mapped to the type. However, what exactly must be equal depends on the type system, which means that the rules need to reflect the rules of the type system, and cannot be written for all type systems at once. There are even less rigid approaches to defining type systems (González-Pérez C., 2012), which would mean that a simple equal function would not be appropriate.

For the logical rules in Table 1 in particular, if they are not met there is clearly an error. An exception to

this would be the rule on weight, which could require that two coins of the same material and similar diameter should have a similar weight (but this would not take into account that the thickness of coins can vary, information that is normally not stored, at least not for ancient coins). However, there could be various parameters that have an influence on the weight, for example: corrosion, wear, special treatment such as clipping or halving, and fragmentation. This makes it more complex, and in cases where the rule is violated, it might not automatically mean that there is a real inconsistency. There might just be a special case that is not represented in the rule so far. Here again, type corpora like OCRE can provide an indication of probability, as they provide the ability to retrieve information on the average weight and weight span based on the specimens linked to them.

An additional complication is uncertainty. In the AFE database, most fields can be marked as uncertain, in order to accommodate cases where there is missing or lost information, and this could therefore be classified as epistemic uncertainty. This is some-

thing that each rule needs to be analysed for; whether such an uncertainty flag can have an impact. Other fields provide the possibility of creating multiple entries for cases where it is not certain which of the entries the coin actually corresponds to; but at present no uncertain information of this kind is exported in our dumps to *Nomisma.org*. Other databases might not even allow such granularity. We have proposed ways of modelling uncertainty as LOD (Tolle and Wigg-Wolf, 2015), but this is an ongoing discussion. On top of this, each model reflects an abstraction, and this is true for relational models for the way data are entered via the graphical user interface, for the underlying model (ontology) for LOD, and even in the mind of each domain expert. In each of these models, and in the mapping between, bias and uncertainties can be generated that go beyond epistemic uncertainty.

With regard to geo-positioning, there is an ongoing discussion within the *Nomisma.org* steering committee. Each system has a different approach to dealing with findspot coordinates. The problem is that publishing the exact coordinates of a coin find in a publicly available environment like OCRE, could lead to plundering of the findspot by illegal detectorists. For this reason, different approaches are used when the data are published online. One possibility is to reduce the precision of the coordinates, for example by cutting off the last digits or, as is done for AFE, to only publish online the centre of the area or administrative unit where the findspot is located. For this purpose, a hierarchy for location information was set up within AFE. For online publication there are four levels of administrative units comparable to those provided in *Geonames.org*: federal state, county, parish, parish division. This means that the exact information on the findspot (most accurate) and the site (less accurate), which is required for scientific analysis, is stored locally and is not released as part of the LOD published under *Nomisma.org*. Therefore, rules addressing or requiring the most accurate geo-positioning cannot be applied to the LOD.

However, for coin finds, accurate geo-positioning could be one trigger for identifying duplicates between different datasets. For example, AFE and KENOM (KENOM, n.d.) contain overlapping data from literature that was manually entered into both

systems. At present, at least the known cases are handled by indicating them in the LOD data: we included *skos:exactMatch* fields for AFE data in order to store this information and to export it into the LOD we provide.

Some AFE-specific rules for geo-positioning were set up within the AFE database, whereby the geo-data for the findspot was checked against the geo-data for its upper level (i.e. administrative unit), in order to see whether they correspond, or the findspot should, for example, instead be attributed to another administrative unit (a problem particularly for common names such as Neustadt). During these tests many errors were discovered, especially for one particular federal state. We realised that there had been an administrative-territorial reform for the federal state that was not reflected in our data. This is a significant problem due to the necessity to remain up-to-date. But because in the literature, which is often used when entering data into the database, there are references to the old administrative units and areas, it is important that this information is not deleted. However, storing and maintaining the evolution of the administrative divisions would be a project in itself. Even *Geonames.org* does not provide their full history; they simply mark old administrative divisions as historic (with an H). For example, in *Geonames.org* some Roman provinces such as Germania Superior are marked as a historical first-order administrative division - ADM1H. A solution for this would be beneficial for the entire archaeological digital world, and would make it easier to build bridges between the different archaeological subdomains. In our view this problem is similar to the challenge posed by periods, and systems addressing this issue similar to the Web service *iDAI.chronontology* (*iDAI.chronontology*, n.d.) could be a solution.

Systems We Worked With

We started to explore and test existing rule engines. This included JESS (JESS, n.d.) and Drools (Drools, n.d.). Both are Java-based rule engines, but since the licence for Drools is more open and it has a more flexible connection interface, the decision was taken to concentrate on Drools. For experienced Java programmers it is easy to handle. It is possible to con-

Experiments and Results

Tools / Language	 rapidrep	 Drools	SWRL (with Protégé and Pellet) <small>Semantic Web Rule Language (W3C Member Submission)</small>
Type	proprietary software	open source	Language
Data Model	self defined, relational	self defined, relational	incorporated into an ontology
Skills needed	Java, SQL, Excel	Java, SQL, Hibernate	RDF/OWL
Time to first prototype	- (needs internal Model and ETL)	-- (needs internal model, ETL and setup of the system)	++
Performance	++	++	- Protégé is not built to handle huge amounts of data
Remarks/Experience	Nice overview of results (in Excel) easy to understand	Needs IT-experts to set it up Very flexible and powerful	Needs understanding of Open World Assumption

Figure 2. Our experiences with the tools we tested.

nect to and load data from various sources, including the possibility of connecting to RDF-data via Apache Jena. However, the link to RDF-data needs to be set up manually, and the data is then transformed into an internal data model, which means that it is necessary to define one model and to map and dump the different sources into it, as with the normal Data Warehouse approach. This requires extra effort and can lead to errors within the mapping.

In addition, we worked with RapidRep (RapidRep, n.d.), a commercial tool. One of its strengths is that for non-programmers it provides many functionalities via a front end, so helping close the gap between domain experts and programmers. Rules are defined in Excel sheets that are understandable for both groups. However, here also it is necessary to set up an internal relational model, and RapidRep does not support RDF semantics natively.

Both Drools and RapidRep worked and performed very well. But as LOD, RDF and the representation for coins based on *Nomisma.org* has become more widely used in the community, we concentrated on solutions that include the semantics that are already given and supported. Reasoners like Pellet (Sirin et al., 2007) understand natively properties like *owl:sameAs*, etc. With SWRL (Horrocks et al., 2004a) we found an approach where the rules can still be expressed at a high level. Unfortunately, SWRL is only a W3C member submission, and not a recommendation, and it is therefore only supported by a few reasoners, including Pellet. We also did a number of tests within Protégé (Protégé, n.d.) where Pellet can be included as a plug-in reasoner.

An example of a rule would be one stating that the authority of a coin needs to be active at least until the end of the period of production of a coin type. In SWRL it would be encoded as:

```
nmo:hasEndDate(?X, ?D) ∧ nmo:hasAuthority(?X, ?A) ∧ nmo:hasEndDate(?A, ?T)
∧ swrlb:greaterThan(?D, ?T) → hasError(?X, R4)
```

Translation: If an object X has the property *nmo:hasEndDate* with the value D, and X has a connection to the object A via the property *nmo:hasAuthority* and the value T, and if D is greater than T, we have an error. The reasoner in this case adds (infers) an additional statement into the data, the property *hasError* (from the local namespace) to the object X with the value R4.

When working with reasoners, it is important to understand their underlying concept. With the Pellet reasoner that is part of the OWL world, the Open World Assumption (OWA) was our basis. This means additional information might exist, but that the reasoner only takes things for granted that are explicitly pointed out. When writing our first rules it took some time until this lesson was learned. When comparing objects and trying to check if two objects are different by using OWLs *differentFrom*(?x,?y), the results were not always what was expected. In order to take into account the OWA, we had to explicitly include into the model that, for example, Nero and Augustus are different persons. Just having two different URIs does not mean that they are different, since it is possible that someone could add *owl:sameAs*(?x,?y) for them. Thus, the open world assumption could be explained as: Anything that is not stated explicitly could be true, while the closed world assumption would be the opposite approach. Sequeda (2012) provides more details on this.

In order to query the resulting error messages with SPARQL in Protégé, it is useful to store the model with the inferred statements. This can be done in Protégé within the File menu under “Export

ID	Rule	Name	No of cases	Reference Query	Reference Size	Ratio	Query Type	Query
6,7	1	Portrait	205	coins_type	4261	0,048	Inconsistent	w3.org/2003/01/
6,7	2	Start Date and End Date fitting	405	coins_type	4261	0,09505	Inconsistent	w3.org/2003/01/
5	3	Start Date after End Date	5	coins AND types	11717	0,000	Inconsistent	w3.org/2003/01/
6,7	4	Denomination	45	coins_type	4261	0,011	Inconsistent	w3.org/2003/01/
6,7	5	Mint	1	coins_type	4261	0,000	Inconsistent	w3.org/2003/01/
6,7	6	Material	54	coins_type	4261	0,013	Inconsistent	w3.org/2003/01/
4	7	Diameter	20	coins	8272	0,002	Inconsistent	w3.org/2003/01/
/	8	Diameter Weight existing	1196	coins	8272	0,14458	Missing	w3.org/2003/01/
/	9	Start - End Date - missing	32	coins_type	4261	0,008	Missing	w3.org/2003/01/
/	10	Tests Diameter Weight	413	coins	8272	0,050	Missing	w3.org/2003/01/
8	11	Diameter Weight	13	coins	8272	0,002	Outlier	w3.org/2003/01/

ID	Rule	Name	No of cases	Reference Query	Reference Size	Ratio	Query Type	Query
6,7	1	Portrait	181	coins_type	5209	0,035	Inconsistent	w3.org/2003/01/
6,7	2	Start Date and End Date fitting	103	coins_type	5209	0,020	Inconsistent	w3.org/2003/01/
5	3	Start Date after End Date	0	coins AND types	12719	0,000	Inconsistent	w3.org/2003/01/
6,7	4	Denomination	28	coins_type	5209	0,005	Inconsistent	w3.org/2003/01/
6,7	5	Mint	0	coins_type	5209	0,000	Inconsistent	w3.org/2003/01/
6,7	6	Material	2	coins_type	5209	0,000	Inconsistent	w3.org/2003/01/
4	7	Diameter	9	coins	9155	0,001	Inconsistent	w3.org/2003/01/
/	8	Diameter Weight existing	2116	coins	9155	0,23113	Missing	w3.org/2003/01/
/	9	Start - End Date - missing	9	coins_type	5209	0,002	Missing	w3.org/2003/01/
/	10	Tests Diameter Weight	0	coins	9155	0,000	Missing	w3.org/2003/01/
8	11	Diameter Weight	13	coins	9155	0,001	Outlier	w3.org/2003/01/

Figure 3. Overview and metrics of two different executions on CNT data. The second run (bottom) was executed about one month after the first; as can be seen, the reference sizes had also increased in this time. The ID column on the left refers to the IDs of Table 1.

inferred axioms as ontology ...”. For small amounts of data this worked very well. However, with increasing data volume more memory must soon be provided to Protégé by increasing the heap size. This can be done by changing the Java call parameter -Xmx while starting Protégé (under Windows this can be done in the relevant run.bat file, for example: -Xmx 4G sets the heap size to 4 gigabytes). However, it is important to note that Protégé is designed as an editor, and not as a database. With realistic data sizes, performance troubles are encountered even with increased memory, and currently we are investigating how to overcome these performance issues; either by using a different tool setup, or by using approaches such as SQWRL (O’Connor and Amar, 2009).

Apart from the rule-based approach, pure SPARQL, OWL, or SPARQL SPIN can also be used, as is explained by the World Wide Web Consortium (W3C, n.d.). However, the built-in functions of SWRL, such as *swrlb:greaterThan*, turned out to be very useful, resulting in shorter and more readable rules. An overview of existing functions can be found under section eight of Horrocks et al. (2004b). Figure 2 summaries our experiences to date.

Results in Application to Other Datasets

During our experiments on the rule systems, we used the results to improve the AFE data and to correct inconsistencies. We then chose another dataset for application of the rules in order to a) see how useful the rule system is, and b) demonstrate that it can be applied to other datasets without major adjustments. In order to do so, we selected data from the project *Corpus Nummorum Thracorum* (CNT, n.d.). The CNT database contains a virtual meta-collection of ancient coins of Thrace, a region that covers parts of modern Bulgaria, Northern Greece and European Turkey, and consists of data about Thracian coins located in museums and private collections from all over the world. The goal of CNT is to generate a typology of Thracian coins.

One of the challenges was that there might be exceptions to our rules or the domain experts might want to add comments for later usage or explanation, but that at the same time different persons were working on the system. In order to handle this, and to come up with a method that can be employed independently of the actual system in use, a proto-

type solution based on Excel files was implemented, with a separate spreadsheet for each main rule. Each execution of the defined rules at a given point in time results in an Excel file. The different domain experts can then work on separate copies of the file and add comments to the different cases. The next time the rules are executed, the spreadsheets are read once more and comments on cases that still exist are transferred to the new spreadsheet. This turned out to be very practical method, avoiding the need to check things repeatedly. In addition, some metadata, for example the first time a case was discovered, can be carried over.

The Excel file also contains an overview with a small metric. Currently, this shows eleven different rules (some of the main rules include sub-rules). These are categorized into rules addressing inconsistencies, missing data, and the highlighting of outliers. Depending on the rule, the reference size as shown in Figure 3 differs, containing either the number of coins linked to a type (*coins_type*), all coins (*coins*), or the sum of all coins and defined coin types (*coins AND types*). The Ratio is calculated by dividing the *No of cases* by the *Reference Size*.

CNT deals with Greek coinage so that rules 1-3 in Table 1 were not included. The reason for this was that many issuers are not yet integrated into *Nomisma.org* and the accordingly their chronological data still needs to be established by the domain experts. Furthermore, inferring the material of coins based on the denomination is less reliable for Greek coins than it is for later periods.

The other rules in Table 1 were included. The resulting mappings are presented in Figure 3 below. The Query column on the right contains the full SPARQL query used to generate the results on the *Nomisma.org* RDF dump.

Summary and Conclusion

One goal of LOD is to link existing systems. Within the field of numismatics this process is still in progress, and currently the individual systems are responsible for their own data quality. This can be ensured to some extent by good modelling of the data using the features of the underlying system. Data entry on the front-end is also a good place for defining rules, and so avoiding errors being included in the data in

the first place (both should be employed, and not just one). For example, a number of Roman emperors struck coins for other members of the imperial family, or for predecessors to whom they wished to proclaim a particular relationship. Therefore, within the front-end of AFE, the usual order found in other databases of entering the data on the issuer and the person portrayed was reversed so that the person portrayed is entered first: a list of possible issuers then appears, thus avoiding incorrect combinations of issuer and portrayed. There might be additional ways of entering data, e.g. by imports. The bottom line is that data quality needs to be addressed at different levels and should be applied as soon as possible. What is more, this can also make things more comfortable for people working with the system.

Previously, since the modelling and front-end were different for each system, it was simply not possible to exchange or reuse data between them. Thanks to *Nomisma.org*, it is now possible to use the same modelling (at least for the 33 institutions that use it already), and therefore rules defined on this level can be shared and evolved in a collaborative way. Our goal here is to set up and publish a prototype version of rules that can be reused by others without adoption efforts. The community can test and improve them in order to also handle special cases. The result would be an improvement of the rules for all, and an increase in quality within the open data. Once the rules are accepted by the community, they can also serve as the basis for a quality metric. This is not *per se* new, and has already been mandated, for example by Hyvönen et al. (2014).

Several errors were found within the AFE data that could be corrected. The same was true in the case of CNT, and we are sure there are still errors to be found. The moral of the story is to beware of trusting the data of others without applying (your own) rules: as you will see once you have carried out this exercise.

Acknowledgements

We would like to thank Tetiana Goncharenko, Linda Homeier, Walaa Karakich and Christian Schöneberger for their support, work, inspiration and feedback on the topic.

References

- AFE** n.d. *Antike Fundmünzen in Europa*. Available at: <http://afe.fundmuenzen.eu/> (Accessed: 30 December 2018).
- iDAI.chronontology** n.d. Available at: <http://chronontology.dainst.org/> (Accessed: 30 December 2018).
- CNT** n.d. *Corpus Nummorum Thracorum*. Available at: <https://www.corpus-nummorum.eu/> (Accessed: 30 December 2018).
- Drools** n.d. *Drools - Business Rules Management System* (Java™, Open Source). Available at: <https://www.drools.org/> (Accessed: 30 December 2018).
- González-Pérez, C 2012** Typeless Information Modelling to Avoid Category Bias in Archaeological Descriptions. In: Chrysanthi, A., Murrieta Flores, P., Papadopoulos, C. (eds) *Thinking beyond the Tool: Archaeological computing and the interpretive process*. Archaeopress, pp. 72-87. ISBN 978 1 4073 0927 9
- Horrocks, I, Patel-Schneider, P, Boley, H, Tabet, S, Grosf, B, and Dean, M 2004a** SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Available at: <https://www.w3.org/Submission/SWRL/> (Accessed: 30 December 2018).
- Horrocks, I, Patel-Schneider, P, Boley, H, Tabet, S, Grosf, B, and Dean, M 2004b** *Built-Ins for SWRL*. Available at: <http://www.daml.org/swrl/proposal/builtins.html> (Accessed: 30 December 2018).
- Hyvönen, E, Tuominen, J, Alonen, M, and Mäkelä, E 2014** Linked Data Finland: a 7-star model and platform for publishing and re-using linked datasets. In: Presutti, V., Blomqvist, E., Troncy, R., Sack, H., Papadakis, I., and Tordai, A. (eds) *The Semantic Web: ESWC 2014 Satellite Events*. Basel: Springer International Publishing, pp. 226–230. Series; LNCS 8798. doi: 10.1007/978-3-319-11955-7_24.
- JESS** n.d. *Jess, the Rule Engine for the Java Platform*. Available at: <http://www.jessrules.com/> (Accessed: 30 December 2018).
- KENOM** n.d. *Kooperative Erschließung und Nutzung der Objektdaten von Münzsammlungen*. Available at: <https://www.kenom.de> (Accessed: 30 December 2018).
- Nomisma.org** n.d. a *CRRO Coinage of the Roman Republic Online*. Available at: <http://numismatics.org/crro/> (Accessed: 30 December 2018).
- Nomisma.org** n.d. b *How to contribute data*. Available at: <http://nomisma.org/documentation/contribute> (Accessed: 30 December 2018).
- Nomisma.org** n.d. c *OCRE Online Coins of the Roman Empire*. Available at: <http://numismatics.org/ocre/> (Accessed: 30 December 2018).
- Nomisma.org** n.d. d *PELLA Coinage of the Kings of Macedonia*. Available at: <http://numismatics.org/pella/> (Accessed: 30 December 2018).
- O'Connor, M, and Das, A 2009** SQWRL: a query language for OWL. In *Proceedings of the 6th International Conference on OWL: Experiences and Directions - Volume 529*, pp. 208-215. CEUR-WS. org.
- RapidRep** n.d. *Software für Testautomatisierung, Model Based Testing, Reporting – RapidRep*. Available at: <http://www.rapidrep.com/de/> (Accessed: 30 July 2017).
- Sequeda, J 2012** *Introduction to: Open World Assumption vs Closed World Assumption – DATAVERSITY*. Available at: <http://www.dataversity.net/introduction-to-open-world-assumption-vs-closed-world-assumption/> (Accessed: 30 December 2018).
- Sirin, E, Parsia, B, Grau, B C, Kalyanpur, A, and Katz, Y 2007** Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2), 51-53.
- Tolle, K and Wigg-Wolf, D 2015** Uncertainty handling for ancient coinage, in Gilligny, F, Djindjian, F, Costa, L., Monscati, P., and Robert, S. (eds) *CAA2014. 21st Century Archaeology Concepts, Methods and Tools. Proceedings of the 42nd Annual Conference on Computer Applications and Quantitative Methods in Archaeology*. Oxford: Archaeopress, pp. 171–178.
- W3C** n.d. *Examples of RDF Validation*. Available at: <https://www.w3.org/2012/12/rdf-val/SOTA> (Accessed: 30 December 2018).

