informatics

MDPI

Article

Conceptualization and Non-Relational Implementation of Ontological and Epistemic Vagueness of Information in Digital Humanities †

Patricia Martin-Rodilla [1,*] and Cesar Gonzalez-Perez [2]

[1] CiTIUS, University of Santiago de Compostela; Jenaro de la Fuente Domínguez, s/n 15782 Santiago de Compostela, Spain

[2] Institute of Heritage Sciences (Incipit) Spanish National Research Council (CSIC) Avda. Vigo, s/n. 15705 Santiago de Compostela, Spain; cesar.gonzalez-perez@incipit.csic.es

* Correspondence: patricia.martin.rodilla@usc.es

† This paper is an extended version of our paper published in TEEM'18, Salamanca, Spain, 24–26 October 2018.

Received: 22 March 2019; Accepted: 30 April 2019; Published: 6 May 2019

Abstract: Research in the digital humanities often involves vague information, either because our objects of study lack clearly defined boundaries, or because our knowledge about them is incomplete or hypothetical, which is especially true in disciplines about our past (such as history, archaeology, and classical studies). Most techniques used to represent data vagueness emerged from natural sciences, and lack the expressiveness that would be ideal for humanistic contexts. Building on previous work, we present here a conceptual framework based on the ConML modelling language for the expression of information vagueness in digital humanities. In addition, we propose an implementation on non-relational data stores, which are becoming popular within the digital humanities. Having clear implementation guidelines allow us to employ search engines or big data systems (commonly implemented using non-relational approaches) to handle the vague aspects of information. The proposed implementation guidelines have been validated in practice, and show how we can query a vagueness-aware system without a large penalty in analytical and processing power.

Keywords: vagueness; non-relational databases; conceptual modelling; imprecision; uncertainty; knowledge representation; digital humanities; ConML

1. Introduction

We generate knowledge from raw data through different mechanisms, such as observation, perception, theorization, and deduction [1], thus producing information models that constitute the starting point of any knowledge generation process. These information models pose a significant impact on the quality and type of knowledge that we are able to generate. When working in the humanities, we also create information models that reflect not only the data that we have but also the possible hypotheses from them in order to fill the knowledge gap. This model-building process is especially relevant when working with information about our past, in which this gap is usually larger. For these reasons, several authors have recently pointed out how relevant models are in the humanities, and identified improvement and evaluation research needs [2,3]. Thus, conceptual modelling techniques have been emerged as a theoretical valid and practical way to represent humanistic knowledge. Conceptual models have been successfully used in humanities projects such as Europeana [4], ARIADNE [5], and DARIAH [6].

Conceptual models describe the world in terms of concepts, their properties, and the relationships amongst them. The main advantage of conceptual modelling, as opposed to other approaches, is its focus on the knowledge-level representation of the domain of discourse, which allows us to obtain simplified and manageable proxies of a relevant scope [3]. Conceptual modelling has been mostly developed under the umbrella of software engineering, and due to this disciplinary heritage, current conceptual modelling techniques lack the necessary mechanisms to represent different subjective opinions or hypotheses [3], and address the ontological or epistemic vagueness that is often part of the part of the world being studied [3]. This is unfortunate, because vagueness plays a crucial role in humanistic models. This is so, firstly, because humanistic studies often deal with our past, which is often described through incomplete and partially unknown information sources and/or fragmented data, and, secondly, because many research practices in the humanities imply a significant degree of vagueness due to their ethnographic and narrative methodologies.

Developing conceptual models that are capable of managing vagueness is difficult, mainly because modelling involves making decisions about the nature, degree, and characteristics of the reality modelled. This difficulty only increases when we try to implement these models as software systems to organize, query, annotate, or search data and assist in the generation of new knowledge. The technologies that we usually employ to do this, either relational or non-relational, are significantly unaware of information vagueness, which only compounds the problem.

In this context, the ConML conceptual modelling language [7] was developed as a simple and affordable tool that can be used by specialists in the humanities without much experience in information technologies, and with special attention towards the implementation of conceptual models as computer artefacts and databases.

In this paper, we present the modelling mechanisms in ConML that explicitly address the representation of vagueness in the humanities. Then, we elaborate by proposing some implementation mechanisms that we can use to carry this improvement over to computer systems, and in particular non-relational store systems. We also provide a complete validation using a real-world humanities project.

The paper is organized as follows: the rest of this section presents a review of existing modelling approaches of vagueness, describing what problems have been found in relation to humanistic information. Section 2 presents the proposed conceptual framework. Section 3 illustrates the proposed approach through its application to a real project in digital humanities, which includes an implementation of a non-relational environment and some examples of data queries involving vagueness resolution. Section 4 discusses the results obtained. Section 5 critically analyses the work and its future possibilities.

1.1. Uncertain Information in Humanities Fields

Data and information modelling applied to humanities is a sub-discipline that has experienced decades of development, due to the need to create models representing humanities data in daily research practices. This need increases exponentially with the recognition of digital humanities as a discipline, and the use of information software systems for storing, indexing, searching, and reasoning about humanistic data. Within this context, there is a large number of works on modelling information in humanities fields [8–10], organized into two underlying categories. On the one hand, humanistic information modelling studies are derived from curation and archives studies, whose practitioners have considerable experience in storing and processing information. These studies have been joined by so-called Linked Open Data approaches [11], which advocate information models that are subsequently shared on the web, converting it into a common database. In all these approaches, the underlying conceptual models usually have a first layer based on an entity–relationship model [12] or similar models and later add layers for the interconnection of models using technologies such as RDF [13]. Common solutions for implementation described here are XML technologies [14], which analyze how the information was obtained or who obtained it (the so-called metadata), or useful annotations for further study of the information contained in the models through information encoding paradigms such as TEI [15]. These conceptual and technological ecosystems for information

modelling in the humanities are very common as a basis for important documentation projects in the field, such as DARIAH [6] or Pelagios [16,17]. Regarding the support for expressing uncertain and imprecise information, neither TEI specification or existing Linked Open Data metamodels explicitly support vagueness (ontological or epistemic). This lack incapacitates these ecosystems regarding the true generation of knowledge in their application domains [18]. In practice, users who need to build software systems based on these models have identified problems with vagueness representation, creating some ad hoc implementations using XML technologies and TEI mechanisms for the representation of vagueness in the metadata part. For instance, some TEI annotation resources have been used (like the TEI Note tag) for representing the certainty degree of some data (adding a possible uncertainty value to the tag) [14] or using XML tags to represent probabilistic aspects [19]. However, these solutions only solve the problem laterally and not modelling the uncertainty as something intrinsic and transversal to the whole model, forcing users to use modelling mechanisms, such as annotation tags, which are not specifically designed for this purpose. Consequently, software searching and indexing systems do not know that these "custom" uses will not be able to index and search while taking vagueness properties into account.

On the other hand, we can find more aligned approaches with the theoretical framework previously presented, not those that use metadata approaches but those that use modelling based on entities and characteristics of the information itself. One of the most well-known works here for digital humanities is CIDOC-CRM (the conceptual reference model impulse by the International Council of Museums) [20], an ISO standard generally applied to the cultural sector that has traditionally been used in archaeological and museum environments, although it has extensions for other humanities uses. The need for modelling aspects of uncertain information has been determined as intrinsic to archaeological practice [21,22] and has also been detected in conceptual analyses carried out on CIDOC-CRM [22,23], although CIDOC-CRM does not support it in its specification [20]. Recently, some authors have started working on an extension of CIDOC CRM to support uncertainty [24], although only covering the uncertainty introduces specific modelling when different users present different points of view or discourses about the information. This approach mixes subjective modelling approaches, and only models some epistemic vagueness scenarios. In addition, we can find other specifications using a thesaurus, an ad hoc creation of ontologies and folksonomies [25], and similar approaches for covering digital humanities' needs in terms of vagueness modelling, but again without any explicit support at a metamodel level.

All these works, and recent international initiatives such as PROgressive VIsual DEcision-Making in Digital Humanities (PROVIDEDH) [26], reveal the need to represent vagueness semantics in the humanities models as part of the intrinsic specification of the modelling mechanisms, avoiding ad hoc solutions. Both large groups of modelling approaches in the digital humanities discussed previously are lacking in this respect. Finally, there are some initiatives for using well-known software engineering modelling technologies to apply uncertainty modelling patterns to the humanities but still are a work in progress. For instance, we can find some isolated examples of using UML [27] to represent information in the humanities, identifying but not addressing the vagueness topic [28]; UML approaches, independent of the application domain, are discussed in the next section. In summary, although vagueness modelling for humanities information is a need that has been detected for decades in many of the works, the existing techniques do not incorporate mechanisms for this within their specifications and are limited to its ad hoc treatment in special cases.

1.2. Existing Approaches Outside Humanities

Modelling aspects of information vagueness represents a field of interest for numerous fields and projects outside humanities disciplines, with different approaches. To facilitate the process of reviewing these approaches for our purposes, we have divided the approaches into three large groups: statistical approaches, strongly mathematical approaches, and software engineering approaches, although some of the reviewed works can be considered hybrids. All these approaches model vagueness explicitly, and some of them have developed techniques and tools that allow for

the explicit treatment of both types of vagueness, which makes them a starting point to analyse their possible application to humanities fields.

First of all, statistics is a particularly relevant discipline in vagueness modelling. Both for ontological and epistemic vagueness, we can find statistical approaches that generally associate probability functions to especially vague attributes of the information that we are modelling. The probability functions could be indicators of the precision (using in inferential statistics) or of the certain degree of the values of the attributes (i.e., error measurements for a given value). These solutions, while explicitly modelling both types of vagueness, assume vagueness as a margin of error function, contradicting our premise of treating uncertain information in the humanities as an intrinsic characteristic of them (that enriches the information) and not as something to mitigate. Thus, we can use these approaches as an idea to explicitly model aspects of vagueness but without giving it semantics of error.

Regarding strongly mathematically approaches, they start from similar paradigms to the previous ones (based on margins of error) such as the interval predictor models [29], models that estimate regions of uncertainty of the contained information. A less error-focused approach corresponds to the fuzzy logic subdiscipline [30,31], which develop specific techniques (e.g., fuzzy sets and probability degrees, rule bases, linguistic summaries as fuzzy descriptions of variables or fuzzy quantifiers, and similarity measures) [31–34] for the modelling of vague aspects of the information. All these techniques contemplate the richness that both types of vagueness bring to the information models and their software applications [32].

Finally, approaches from software engineering maintain the differentiation between imprecision and uncertainty that we have detailed in our theoretical framework. In the case of ontological vagueness (imprecision), they try to expressly model the probability and possibility of the existence of entities in the data and information models. In the case of epistemic vagueness (uncertainty), they try to identify modellable characteristics such as set membership, interval membership, incompleteness, and other vague aspects. These works are still in progress (the OMG standardization group for vagueness in UML is still working, and their first ideas are from 2016) [35,36], although some UML modelling solutions based on stereotypes [37] can already be found. In any case, UML does not currently include support for modelling vagueness in its official specification [27].

The three groups of approaches have been applied to represent information and implement software systems in several domains of application (genetics and medicine [38], e-government and infrastructures [39], energy resources, etc.), being less common in models for representing humanistic information. Its treatment of vagueness closely linked to the concept of error and its large mathematical base makes its direct application to humanities fields difficult, where the definition of a probability function or the assumption of an a priori distribution of the data is complex.

With the idea of providing a solution for the explicit modelling of uncertain information in the humanities that is (1) far from this notion of error and (2) simple and intuitive for humanities researchers [40], the modelling language ConML has incorporated specific modelling mechanisms of both types of vagueness. The following section explains in detail the conceptual framework and the mechanisms proposed.

In order to define, characterize, and implement vagueness mechanisms as part of any conceptual model ad their subsequent software systems based on them, it is necessary to make some decisions about the specific treatment of vagueness we adopt and what modelling language is adequate for expressing the models. The following sections introduce both of them.

1.3. Theoretical Framework

Many terms have been used in the literature to refer to the fact that data, or information, is not clear or perfectly defined: imprecision, vagueness, uncertainty, imperfection, etc. A complete conceptual characterisation of what is meant by these terms is rarely provided, so confusion ensues. To avoid this, we provide here a small theoretical framework that hopefully will clarify things and establish the basis for further developments such as the solution proposed in Section 3.

To start with, we acknowledge that many aspects of the world are unclear, imprecise, or not well defined, and when we try to represent them in a model, we are often confronted with the need to either remove or explicitly manage this vagueness. Vagueness comes in two forms:

- Ontological vagueness, or imprecision, which refers to things in the world that are not clear-cut, such as the boundaries of a hill;
- Epistemic vagueness, or uncertainty, which refers to situations where our knowledge about something is unclear or incomplete.

We say that imprecision is ontological because it is an inherent property of some things in the world. For example, a hill is an entity that any of us can conceptualise and reason about, but it lacks clear-cut boundaries, so that it is impossible to determine a line marking the hill's boundary. This fact is independent of the knowledge that we may or may not have about the hill. Contrarily, we say that uncertainty is epistemic because it relates to how much we know about something. For example, I may know the name of this particular hill, or I may ignore it, or I may be roughly certain but not sure about it. This is a subjective phenomenon and definitely not inherent to the hill.

Vagueness, in turn, jointly refers to imprecision and uncertainty. A deeper and complete treatment of vagueness as a knowledge representation concern, including imprecision and uncertainty, can be found in [3] (Chapter 14).

Imprecision, being inherent to the things of the world and independent of our knowledge, depends on what properties we look at. Some properties, such as the names of people or cities, or the height of buildings or people, are not imprecise, as they are clearly established for any particular entity we may consider. For example, I have a clear name and height, regardless of whether you know them or not. This means that a modelling approach that aims to support the expression of imprecision must provide a mechanism to identify which properties or things being represented are subject to this kind of vagueness.

On the contrary, anything may be subject to uncertainty because uncertainty depends on our knowledge about something, regardless of what that something is. As anyone may possibly be more or less knowledgeable about anything, every property of everything is, in principle, equally subject to uncertainty.

Finally, it is worth mentioning the concept of accuracy. Whereas precision refers to how much detail an expression contains (such as 15.25 being more precise than 15.2), accuracy refers to how well an expression represents something, e.g., if I have 15.25 euros in my pocket, the expression 15 is imprecise but is quite accurate, whereas the expression 37.123 is much more precise but far less accurate. Note that precision is a property of expressions alone, regardless of how well they represent anything; contrarily, accuracy is a property of the representational power of expression. In this regard, accuracy is a useful tool to fight uncertainty. For example, imagine that we are required to express the distance between two places in kilometers. If we believe the distance is around 650 km but are unsure of it, we can refrain from attempting to be accurate in order to gain certainty by saying that the distance is between 500 and 900 km. This is certainly not very accurate, but we are probably right as the actual distance falls inside the given interval.

1.4. ConML

ConML is a conceptual modelling language designed for the humanities and social sciences. Using ConML, we can represent the entities in the world as well as their characteristics and the connections among them. We can also represent the relevant categories that we employ to classify these entities, together with the relationships between them. ConML is based on the object-oriented paradigm, as are many other popular modelling languages such as UML [27], but is much simpler so that non-experts in software systems can learn it and use it in under 30 h [18,41].

At the category (type) level, the basic constructs of ConML are class, which represents a category in the world, and feature, which represents a characteristic of a category. There are two kinds of features: attributes that correspond to atomic characteristics, which are expressed through simple values (such as someone's age or the name of a place), and semi-associations, which correspond to

complex characteristics, which are expressed through references to other things, such as a house's owner (which is a person) or a person's birth place (which is a town). In addition, inverse pairs of semi-associations are combined into associations; in this regard, each semi-association of an association corresponds to associations as seen from the point of view of each of the participant classes. In this regard, we can say that, in ConML, classes have features, which can be either attributes or semi-associations, and classes are related to each other through associations, each of which is composed of a pair of inverse semi-associations. For example, we may have a ConML model representing the fact that buildings have an address and a height, and are located in cities, which have a name. Here, building and city are two classes. The building class has two attributes, address and height, whereas the city class has one attribute, name. Furthermore, building and city are related by the association IsLocatedIn.

Attributes in ConML have a data type, which specifies what kind of data may be stored by their instances. Only five simple data types exist in ConML: Boolean, number, time, text, and data. In addition, ConML supports enumerated data types. An enumerated type consists of a list of pre-defined named items, and a value of this type can only hold an existing item. For example, a model may define a styles enumerated type containing the items romanesque, gothic, and neoclassical. An attribute such as building style, defined as having type styles, could only take one of these items as a value. Interestingly, the items in an enumerated type do not need to be arranged as a linear list but can be hierarchically organized to represent subsumption or aggregation, so that every item may have a "parent" or super-item and may have a number of "child" sub-items. For example, we could add decorated gothic and flamboyant gothic under gothic in the styles enumerated to reflect the fact that there are two subkinds of the gothic style.

At the entity (instance) level, the basics constructs of ConML are object, which represents a specific entity in the world as an instance of a class; value, which represents a characteristic of an entity as an instance of an attribute; and link, which represents a connection between two entities as an instance of an association. We can say that, in ConML, objects have values and are connected to each other by links. For example, we may have a ConML model representing the fact that the cathedral in Santiago de Compostela is 32 m high. Here, cathedral and Santiago de Compostela refer to objects instance of building and city, respectively: 34 m is a value instance of Height, and "in" refers to a link between these two objects.

A comprehensive description of ConML is outside the scope in this article but can be found in [3,7].

2. Materials and Methods

This section presents the ConML mechanisms proposed for expressing vagueness as part of digital humanities conceptual models.

Expressing Imprecision and Uncertainty with ConML

ConML features several mechanisms that support imprecision and vagueness. These mechanisms are distinct, but they are often used in combination to express complex facts. In general, imprecision is difficult to treat through cross-cutting mechanisms, as its semantics depend largely on the nature of each imprecise characteristic. On the contrary, uncertainty can be satisfactorily treated through cross-cutting mechanisms in the language, as it is independent of the characteristics being described. The following sections describe each of these mechanisms in turn.

Null and Unknown Semantics

Most modelling or software-oriented languages, as well as most database management systems and languages, provide a null keyword, or equivalent, to express that a piece of data is not available. However, this is ambiguous, because data unavailability may be due to ontological or epistemic reasons. For example, if we read that p.Name = null where p is a person, we should interpret null as meaning epistemic absence, i.e., we do not know p's name. However, if we encounter something like b.Protection¬Level = null, where b is a building, we may interpret this as epistemic or ontological

absence, i.e., we do not know what protection level applies to b, or b has no protection level whatsoever. To avoid ambiguity, ConML offers two different keywords:

- Null, which indicates ontological absence; b.Protection¬Level = null means that no protection level has been established for b;
- Unknown, which indicates epistemic absence; b.Protection¬Level = unknown means that a protection level has been established for b, but we do not know what it is.

In this manner, unknown provides a simple but powerful mechanism to express ignorance of a fact, which is an extreme case of uncertainty.

Null semantics may be applied only to those features that have a minimum cardinality of zero. For example, if the Person.Name attribute in our previous example is defined as having a cardinality of 1 in a class model, then it may not take null values in an instance model in order to maintain type conformance. However, unknown semantics may be applied to any feature, as anything is susceptible of not being known.

Certainty Qualifiers

To cater for finer degrees of uncertainty, ConML incorporates certainty qualifiers. These are labels that may be attached to instances of classes or features to express how certain a statement is, following an exclusive order relation between them. Note that ConML does not define the qualifiers in a quantitative level (e.g., assigning a percentage of certainty to each qualifier), because this assignation could vary between domains of applications or even between implementation solutions, and it could be assigned in next phases of the mode implementation. There are five pre-defined degrees of certainty in ConML:

- Certain. The expressed fact is known to be true. This is indicated by an asterisk * sign;
- Probable. The expressed fact is probably true. This is indicated by a plus + sign;
- Possible. The expressed fact is possibly true. This is indicated by a tilde ~ sign;
- Improbable. The expressed fact is probably not true. This is indicated by a minus - sign;
- Impossible. The expressed fact is known to be not true. This is indicated by an exclamation ! sign.

Certainty qualifiers can be applied to describe existence or predication. When used for existence, they are attached to an instance of class in order to express how certain we are of the existence of such an entity. For example, we may label building b in our previous example as (+), to indicate that the building represented by b probably exists. Similarly, certainty qualifiers can be applied to instances of features to express how certain we are of the associated predication. For example, we may state that b.Height = 34 (*) to indicate that we are sure that the building represented by b is 34 m high.

Abstract Enumerated Items

In previous sections we described the fact that items in an enumerated type can be hierarchically organized to represent subsumption or aggregation between items and sub-items. We can use this varying abstraction level of enumerated items to represent different degrees of vagueness, both ontological (imprecision) and epistemic (uncertainty). Let us imagine that we have a World¬Regions enumerated type having root items Europe and Asia, and then items France, Germany, and Spain under Europe.

Imagine now that that we wanted to express where the prehistoric bell-beaker culture took place. We know that it happened in Europe, but its boundaries are naturally (i.e., ontologically) vague; for this reason, the best thing we can do is use Europe, as France, Germany, or Spain would be too restrictive. The ontologically vague Europe is an acceptable representation of the fact we want to convey, namely, that the bell-beaker culture happened all over Europe but without clear-cut boundaries.

Imagine now that we need to indicate where someone was born, and that we know that it was somewhere in Europe but we are not sure what country. Again, we should use Europe to capture this

fact. By doing this, we would be purposefully injecting some inaccuracy to gain certainty, as explained in previous sections.

As illustrated by the examples, using an abstract enumerated item such as Europe may entail ambiguity, as statements such as Place¬Of¬Occurrence = Europe may mean two different things: the place of occurrence is all of Europe (imprecision), or the place of occurrence is some particular spot in Europe, which we are not sure of (uncertainty). Despite this, the semantics of the expressions are usually sufficient to resolve the ambiguity; for example, Place¬Of-Birth = Europe should be interpreted as an uncertain (rather than imprecise) expression as we know that people are born in a specific spot rather than in a whole continent.

Arbitrary Time Resolution

The time data type introduced in previous sections corresponds to expressions of points along the arrow of time. However, as opposed to other modelling languages, ConML allows expressions of the time data type to contain arbitrary resolution. This means that time points do not necessarily follow the usual pattern of day, month, year, hour, minute, and second, but can be as "thick" or "thin" as needed. Some sample time values in ConML are 8 June 1996 20:45, September 1845, late 20th century, or early neolithic. All these expressions represent "points" in time of different "thickness".

In a similar way as we did with abstract enumerated types, we can use "thick" time points to express imprecision or uncertainty. Furthermore, like in the previous case, the ensuing ambiguity must be resolved by looking at the semantics of each individual expression. For example, a statement such as Moment = 1936 may mean that something was ongoing throughout the complete year 1936 (imprecision), or that it happened at a particular time this year but we are not sure when (uncertainty). A statement such as Date¬Of-Birth = 1936, however, is clearly uncertain rather than imprecise, as we know that people are born on a specific day and time rather than throughout a full year.

The four mechanisms presented cover most of the needs found in terms of humanities information modelling, although it could be possible to define other mechanisms to support imprecision and vagueness as part of ConML (e.g., methods for defining ranges) that we are considering for future revisions. Next section presents a proposal for an implementation of these mechanisms on non-relational data structures, validating ConML mechanisms in a project with real data and showing how the software system manages data queries involving vagueness resolution.

3. Results

3.1. Case Study and Resultant Models

This section describes the application of the solution proposed in previous sections to a real scenario in Digital Humanities. This scenario occurred with a research project carried out at the Institute for Medieval and Renaissance Studies and Digital Humanities (Instituto de Estudios Medievales y Renacentistas y de Humanidades Digitales IEMYRhd) [42], University of Salamanca, Spain. The research project, named DICTOMAGRED [43], analyses historical sources (including oral testimonies, legal documents, literature, etc.), most of them in Arabic, which contain geographical references describing routes through different areas in the Maghreb, their place names, their topography, and other related issues. The main goal of the project is "to provide a software tool for humanities specialists to retrieve information about the location of toponyms in North Africa as they appear in historical sources of medieval and modern times" [43]. Due to the heterogeneous nature of these historical sources, both in type and chronology, multiple needs appeared in relation to the representation of vagueness. In addition, and as in most cases in digital humanities research, vagueness not only helps researchers to better represent the area of study, it also provides additional knowledge about it. For this project in particular, needs included the specification of the degree of certainty of sources in relation to place names, the description of population estimates of the different geographical areas, and the indication of whether these places are now inhabited or not, among others [44]. Figure 1 shows an excerpt of the class model created for the project, focusing on toponyms

(i.e., place names) and relations between them, the related geographical areas, and the historical sources that were employed.

- Toponym: proper name referring to a geographical place. No vagueness is involved;
- Toponym distance: relative distance between two toponyms. This class also holds information related to the reliability of the distance estimation as a separate attribute;
- Geographic area: location of the place referred to by a toponym. If a toponym is still in use, the corresponding geographic area is epistemologically vague but known; if not, the geographic area may be estimated from the historical sources;
- Historical source: any manifestation of a testimony, (textual such as letters, publications, and bibliographical references) or oral testimonies (formal or informal) that allows the reconstruction, analysis, and interpretation of historical events.

ConML allowed us to make decisions about the treatment of vagueness very early in the project while working at the conceptual level, and thus avoid bringing technological dependencies or other implementation decisions to the conceptual model. Thus, the class diagram in Figure 1 lays the foundation for expressing vagueness when taking instances. To illustrate this, we take some instances of the classes in Figure 1, as depicted in Figure 2. Firstly, toponym was instantiated as objects top1, top2, and top3 in order to represent toponyms of interest: Sijilmasa, Aghmat Ourika, and Tamdalt. According with two historical sources (instances of textual historical source that are not presented in the following diagrams for space reasons), Sijilmasa was an important human location founded at 757 B.C. These historical sources place it within the limits of Tiaret, close to a rich gold mine that existed between Sudan and Zawila, on a difficult route. This was a medieval Moroccan center of commerce in the far north of the Sahara in Morocco. The history of the city was marked by several successive invasions of Berber dynasties. Due to their strategic importance, their distance with other important cities and the related routes have been studied for decades. Another important extinct city is Tamdalt, whose records date from the 2nd century B.C.; from Tāmdalt to Siŷilmāsa there are 11 marhalas (stages). Tamdalt is the Ansara river, which was born in the mountain that is ten miles from it, in the Mahgreb, where there is a silver mine. Currently, the name Tamdalt is not in use. Finally, Agmat Ourika was a city located eight days from Siŷilmāsa and three days from Dar'a. From the localities of the Sūs to this city, it takes six days to walk, and many villages of Berber tribes are crossed, whose apogee lay in middle ages. Currently, the known archaeological site Journaa Aghmat in an enclave in the Moroccan Ourika road. All this information is described with vagueness mechanisms in the object-oriented diagram in Figure 2.

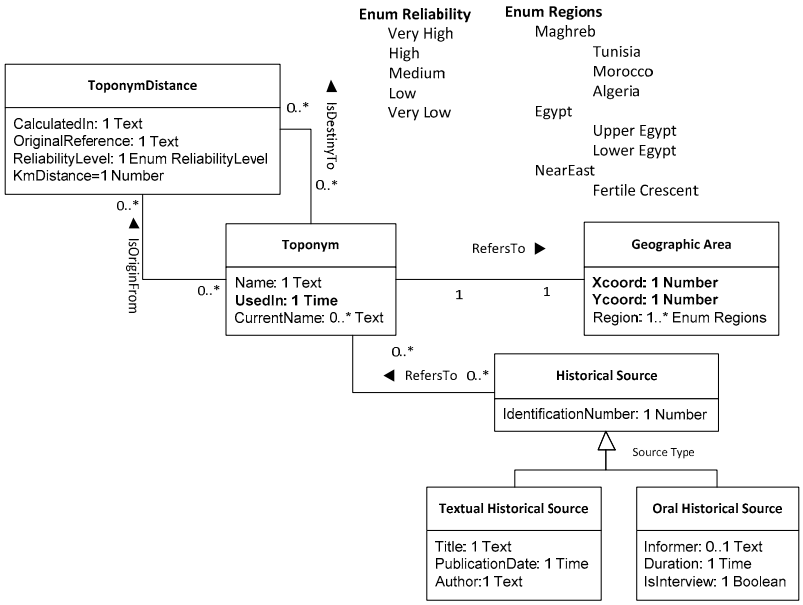

Figure 1. ConML class model for toponym studies in DICTOMAGRED project.

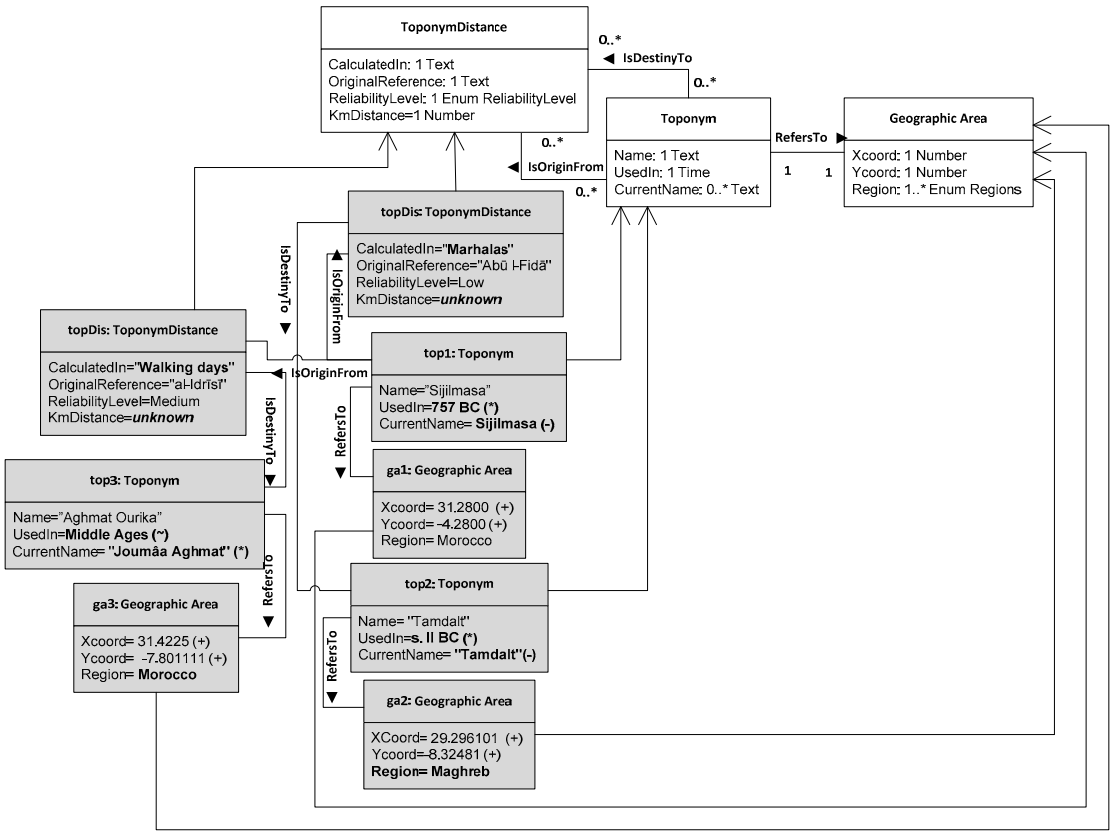

Figure 2. ConML model for Sijilmasa, Tamdalt, and Aghmat Ourika toponyms information in DICTOMAGRED project. In grey, objects created for instantiate the class model, representing imprecise and uncertain information regarding toponym, toponym distance, and geographic area.

Vagueness is expressed throughout the model in Figure 2 as follows:

Three objects (top1, top2, and top3) represen the three toponyms involved on this scenario. For each object, time arbitrary resolution is used to express when each toponym was initially used. In addition, certainty qualifiers are employed to describe how certain we are about these datings: for Sijilmasa and Tamdalt, the asterisk in parenthesis at the end of the UsedIn attribute line indicates that we are sure that the toponym was in use on these dates for the reliable historical sources; for Aghmat Ourika, we use a tilde sign to indicate that we are not sure that it was used at the middle ages. In addition, the current name for Aghmat Ourika is Journâa Aghmat, the current name of the archaeological site with a certain qualifier sign, as no place name exists today in references to the other archaeological sites; Sijilmasa and Tamdalt, on the contrary, maintains their original names but with a minus sign, because is false that the old toponyms are now in use.

Parallel objects ga1, ga2, and ga3 represent the geographical areas where we currently place each toponym. These objects also employ certainty qualifiers for the values of XCoord and YCoord attributes in order to express the certainty of the coordinates. Abstract enumerated items are also employed with the region attribute. In the case of Sijilmasa and Aghmat Ourika, since there are well-known archaeological places in the center of Morocco, we can safely state them in Morocco. In the case of Tamdalt, it is an inhabited archaeological site near frontiers at present, so the level of certainty about the region is low, and therefore the very general Maghreb value is chosen since we cannot be more specific.

Regarding both topDis1 and topDis2 objects, vagueness is explicitly treated through the reliability level enumerated type, which allows us to state that the distance of “marhalas” (a stage or period in different Arabic languages and dialects) presents low reliability, whereas “walking days”

presents medium reliability. Additionally, we cannot specify a distance in km, so unknown is used as a value for km distance.

As we can see in the DICTOMAGRED conceptualization [44], the use of explicit vagueness modelling mechanisms (both ontological and epistemic) allows us to capture relevant information needs in digital humanities research. In addition, it allows us to develop a software system while taking into account these specificities in the information.

3.2. Implementation

The final aim of the vagueness inclusion in DICTOMAGRED project includes the development of indexing and searching mechanisms according to different levels of information uncertainty, for example, searching only toponyms in current usage or accessing those that are on camel-days journeys or marhalas measurements of estimated distance with a high confidence by the historical sources. A non-relational storage structure has been chosen for the software system, since it allows us to maintain acceptable rates for indexing and searching information.

Non-relational databases present particularities that we need to manage when implementing the vagueness mechanisms. In order to define this implementation proposal as universally as possible, we have decided to work with key-value structures for the expression of information, since they are the simplest and most commonly employed structure in all non-relational databases. Additionally, key-value principle is used as basis for document-based structures, which are also commonly non-relational schemas in which the data entities are grouped in documents as objects, which are composed by keys (properties) and values. These documents are usually formatted following JSON syntax [45,46]. For complete information about the non-relational terminology used here for describing implementation mechanisms, please consult [47].

In addition, it should be pointed out that non-relational databases are the most widely currently used structure for application development, due to their performance in terms of indexing and searching performance, real-time data management, and connectivity (for example, for mobile or distributed applications). Digital humanities software systems also require these indexing and searching performance capabilities. Next, we detailed the non-relation implementation designed for each vagueness mechanism defined:

1. Null and unknown semantics. Most of the non-relational systems do not allow one to create specific reserved words that could implement the need for null and unknown semantic for expressing vagueness. Some systems use numeric values such as zero, negatives values, or empty strings to represent null and/or unknown values. Other values are sometimes used as "magic" values for these semantics. However, these practices often introduce ambiguity and confusion, as zero and empty strings may constitute acceptable values for associated attributes. It is also common practice to create specific informational objects in the database structure for null or unknown semantics. This is a possible solution in systems where the object structure is still supported, such as MongoDB [48]. However, this solution is not possible in all non-relational systems. As we need specific semantics elements for representing absence of facts and absence of information universally, we have defined a node in our non-relational structure for each of them, encapsulating in specific references in the non-relational software systems the semantic required. Figure 3 shows the non-relational node and the key-value structures defined for null and unknown semantics and their use in a specific toponym information description in DICTOMAGRED.

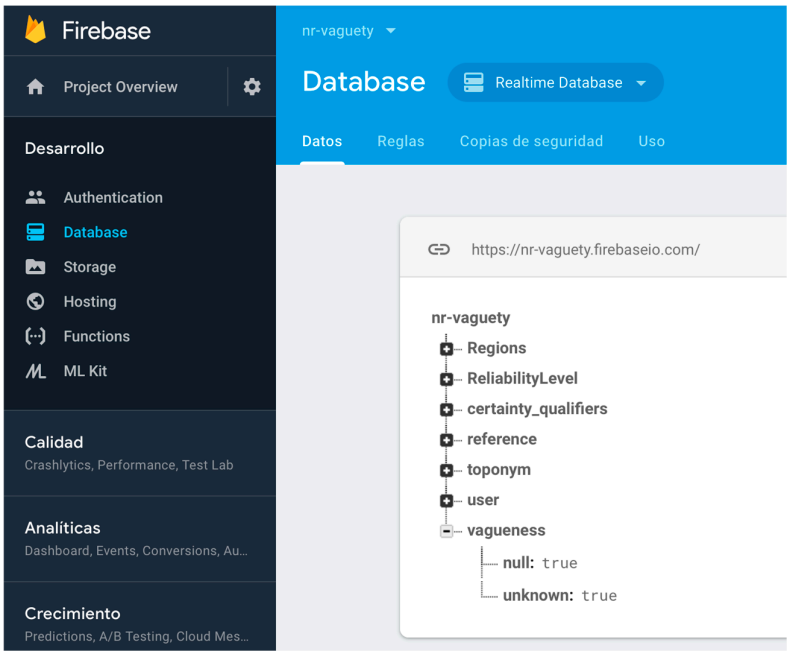

Figure 3. Firebase console showing the data node for defining null and unknown semantics.

2. Certainty qualifiers. As we previously detailed, a certainty qualifier offers some "extra" information about a specific value of an attribute defined in the conceptual model (i.e., in b.Height = 34 (~), "34" is the value and the certainty qualifier indicates extra information; we are not very sure about the height given value). Thus, it is necessary to firstly define in the non-relational structure the certainty qualifiers as specific references that we can add to any key-value previously defined. A node with all possible certainty qualifiers is defined as part of the non-relational structure, separated from any other information node. With this solution, it is possible to correlate another key-value structure to the value "34" itself (following the example), for indicating the certainty qualifier. Figure 4 shows the nodes added and their use in a specific toponym information description in DICTOMAGRED.

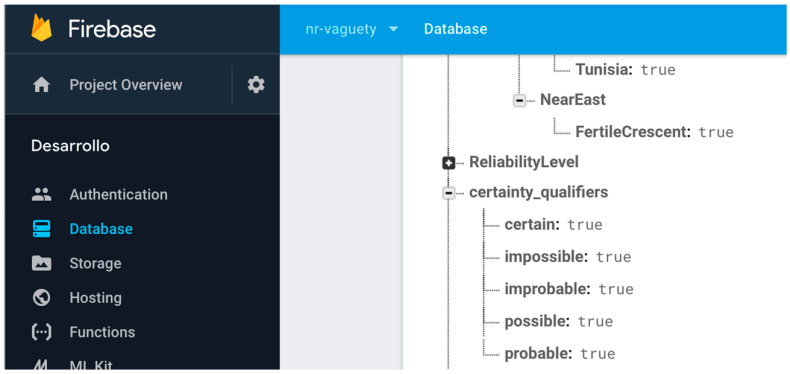

Figure 4. Firebase console showing the data node for defining certainty qualifiers in DICTOMAGRED implementation.

3. Abstract enumerated items. Some systems use numeric values for representing levels of abstraction in a hierarchical structure of items. Other values are sometimes used as ad hoc formatted values for these semantics, as chains of strings separated by special characters like "." or "/" for representing the entire path of the enumerated item value (Region = Magreb.Morocco). However, these practices often introduce ambiguity and confusion in the information, as they may constitute acceptable values for the associated attributes or responds to arbitrary

implementation decisions. It is also common practice to create implement abstract enumerated items as in the previous certainty qualifiers mechanism, defining a hierarchical node in the non-relational structure and putting the most concrete value of the hierarchy (Region = Morocco). Then, the software system iterates this node in order to obtain at what level of abstraction the value is described. The final possibility is to define the hierarchical node but putting as Boolean values of the attribute all the levels involved (Magreb = true; Morocco = true). Both last solutions follow a non-relational structure and are operational for implementing abstract enumerated items. However, iterating the node each time we want to solve the abstraction information is inefficient in non-relational environments, so finally we chose the Boolean values structure. Figure 5 shows the non-relational node defined for the regions enumerated type and their items, and their use in a specific toponym information description in DICTOMAGRED.

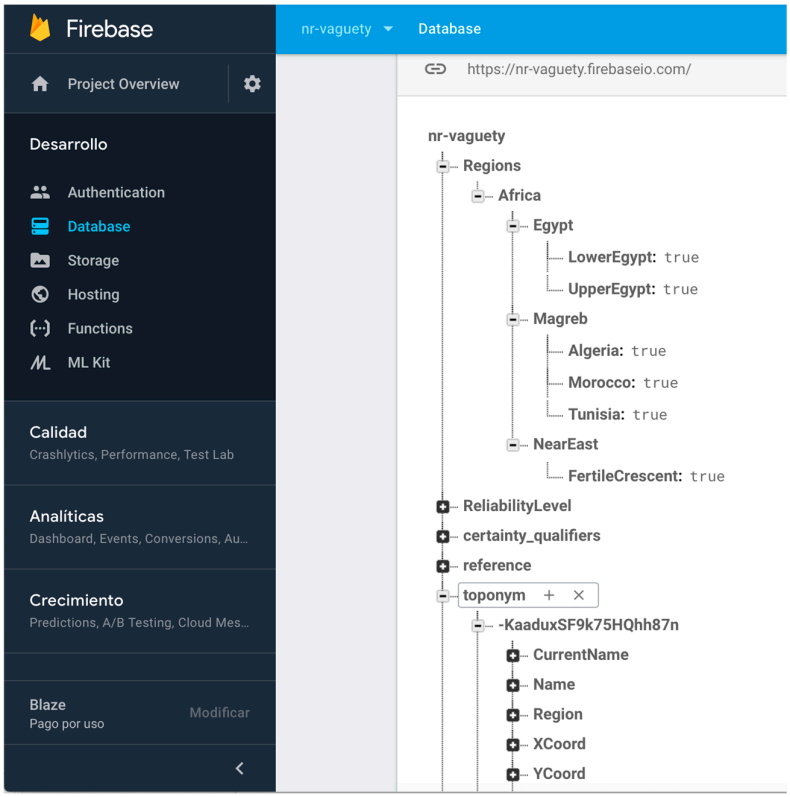

Figure 5. Firebase console showing the regions data node implementing the abstract enumerated items mechanism.

4. Arbitrary time resolution. Most of the non-relational systems use the timestamp mechanism to represent temporal values (number of milliseconds after 1st January 1970). The need for representing previous dates at any granularity level in digital humanities makes timestamps use impossible for humanities information. There are some non-relational systems, such as MongoDB [48], that present specific data types for dates but with a very rigid format guided by ISO 8601 standards, which also presents other problems for humanities information, such as absence of support of Julian calendar or problems in data conversions between other date systems, such as Hegira (used in DICTOMAGRED project), Chinese calendar, etc. These limitations encouraged us to implement class library supporting the arbitrary resolution inherent to the time data type in ConML, which allows for some of the most usual forms of time representation, including simple and incomplete dates (and times), years, decades, and centuries. Now, we have implemented part of the functionalities of the class library in the non-relational environment for DICTOMAGRED. Similar to the certainty qualifiers implementation, we have defined a node in the non-relational structure with a hierarchical conceptualization of

vagueness points in a timeline that we want to manage (years, decades, centuries, time eras, etc.). Then, we included a key-value structure referring to the specific point in time used for solved a given value. For instance, UsedIn = middle ages contains a key-value structure indicating that the value "middle ages" needs to be interpreted as the "Age" level of granularity in time. Figure 6 shows the non-relational node defined for the arbitrary time resolution, and its use in a specific toponym information description in DICTOMAGRED.

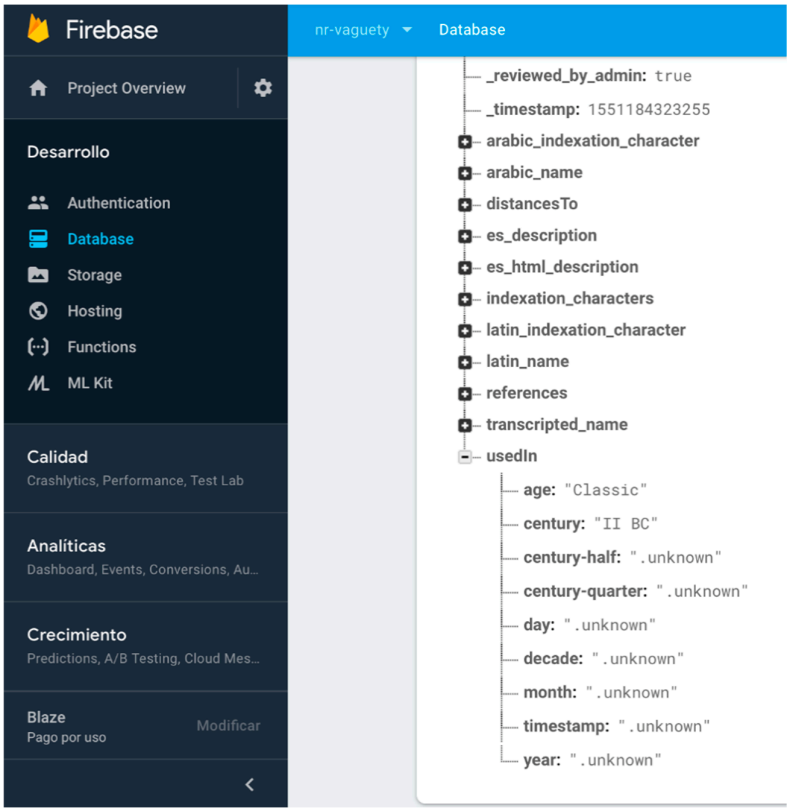

Figure 6. Firebase console showing UsedIn attribute implementation according the arbitrary time resolution mechanism.

Note that, although we explained the implementation proposal by each vagueness mechanism, it is possible (and desirable) to combine the mechanisms, exploiting the expressiveness of the ConML vagueness mechanisms and the potential of the non-relational structure. Thus, it is possible to express in a non-relational structure that one specific toponym was used in the second century (S.II B.C.) with highly confidence (using certainty qualifiers) while other was used in middle ages with a lower confidence.

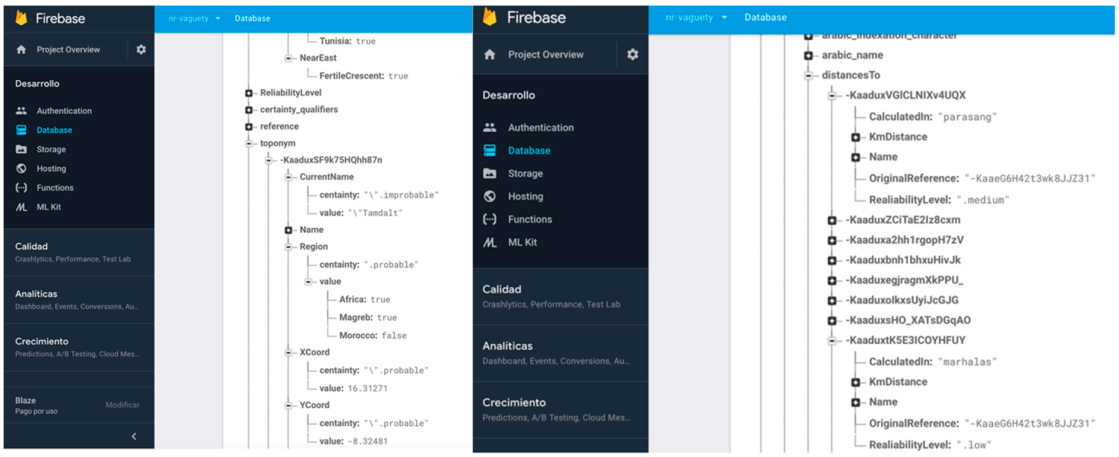

Figure 7. Firebase console showing final implementation details. At right, the values marhalas or parasangs (Iranian past measure unit for distance) as vague measurement units for distance in the DICTOMAGRED data model. At left, the final values for the specific Tamdalt toponym supporting vague information.

All the implementation details in non-relational structure shows are implemented in DICTOMAGRED, including vague measurements for distances or vague locations (see Figure 2 and Figure 7). The project uses a web-based environment with non-relational real-time database provided by Firebase services [49]. Firebase is a mobile and web application development platform run by Google since 2014 that allow us to personalize the non-relational database implementation with indexing and searching integrated services, as well as other functionalities (real time maintenance, cloud services, etc.). It is important to highlight that the implementation proposal presented here is defined in terms of the conceptual model previously defined and following a non-relational data structure, but independently of the specific non-relational environment chosen. Thus, as well as on Firebase, the following implementation could also be adopted as part of any other well-known non-relational environment based on key-value or document-based structures, such as MongoDB, Amazon DynamoBD, CouchBase, Oracle noSQL, etc. [47,50]. Following this premise, the specific modelling and implementation decisions made during this work present some homogeneity for all mechanisms, in order to ensure that the implementation proposal defined here is as universally applicable as possible for non-relational contexts with expression of informational vagueness needs, both ontological and epistemic. In addition, we employed a search system service provided from Algolia [51] via a RESTful JSON API for implementing the non-relational queries, although Firebase supports the main programming languages (including Javascript, PHP, or Python, among others) that will allow us to integrate the DICTOMAGRED system via web. The following subsection shows the experiments carried out within the DICTOMAGRED project defining specific queries that include aspects of vagueness and illustrating how the DICTOMAGRED software system manages vagueness in its query results.

3.3. Query-Based Vagueness Resolution Results

Three queries have been defined according to the specific vagueness needs of the case study shows in Figure 2 from DICTOMAGRED, expressed first in natural language and subsequently executed in the Algolia search systems accessing the Firebase-defined structure:

- QUERY A: Searching for all Dictomagred toponyms located in Maghreb region whose current name is improbable**.** This means that the toponym is probably not in use regarding current maps of populations and cities. QUERY A involves tow vagueness mechanisms: abstract enumerated items to solve the hierarchical levels of the information about the regions attribute, and certainty qualifiers to evaluate what values of the current name present an improbable qualifier.

- QUERY B: Searching for all DICTOMAGRED toponyms whose distance from Sijilmasa is unknown. This means that the system evaluates the instances of Toponym Distance where km Distance is unknown and shows the correspondence toponyms involved in these instances as origin or destinies. This query allows us to test the resolution of unknown references.
- QUERY C: Searching for all toponyms used in middle ages or in the second century B.C. This means that the software system has to query *UsedIn* attribute value at two levels of abstraction for solving the query employed arbitrary time resolution (note that both points in time present different levels of granularity and neither of them adjusts to classic timestamps of data formats employed in ISO 8601 standard or similar references).

Note that all queries require, at least, the use of one vagueness mechanism or even combined versions of them, in order to offer to the DICTOMAGRED users (mainly researchers on Arabic language; Magreb topography, history, and/or archaeological remains; etc.) responses to their research questions. Next Figures 8, 10, and 12 show how these queries are executed, and Figures 9, 11, and 13 show the corresponding results consulting our Firebase non-relational database using the Algoria search engine. Note that, for executing a query in the Algolia dashboard, it is necessary to define as filters or facets [51] the parameters that the query requires, in our case region as Maghreb and current name certainty as improbable in the query A (Figure 8), km distance as unknown in the query B (Figure 10) and UsedIn as middle ages or second century B.C. in query C (Figure 12).

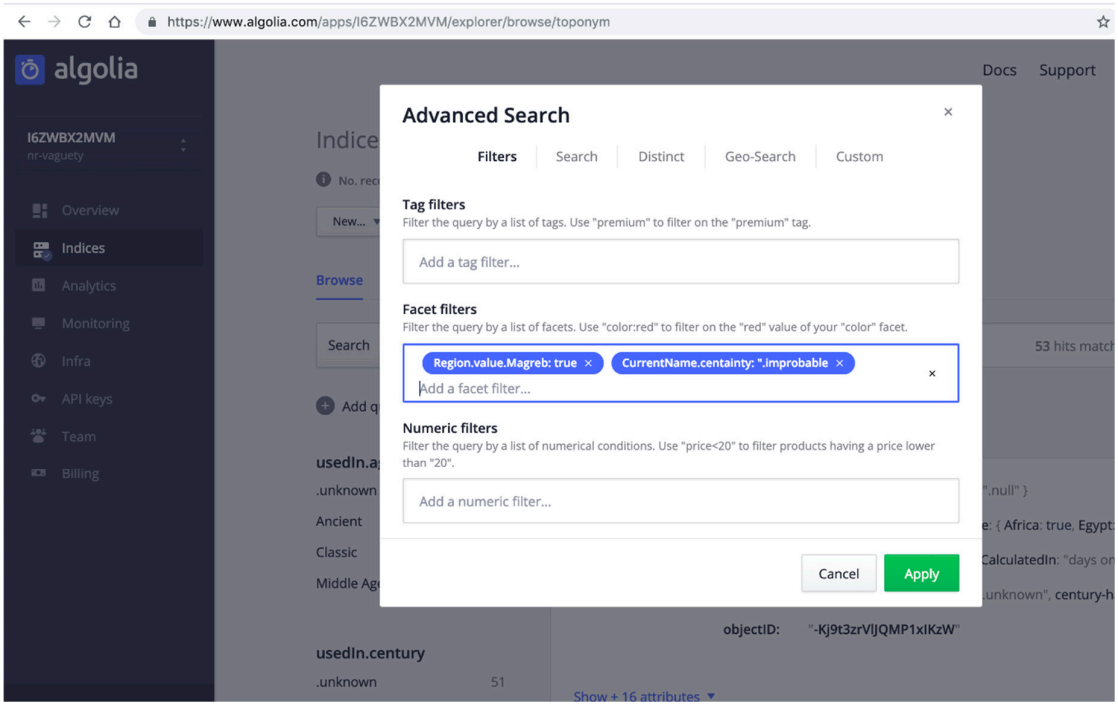

Figure 8. Query A execution through Algolia search engine. We have added two facets with the two requirements of the query about the region and the certainty in the current name use of the toponyms.

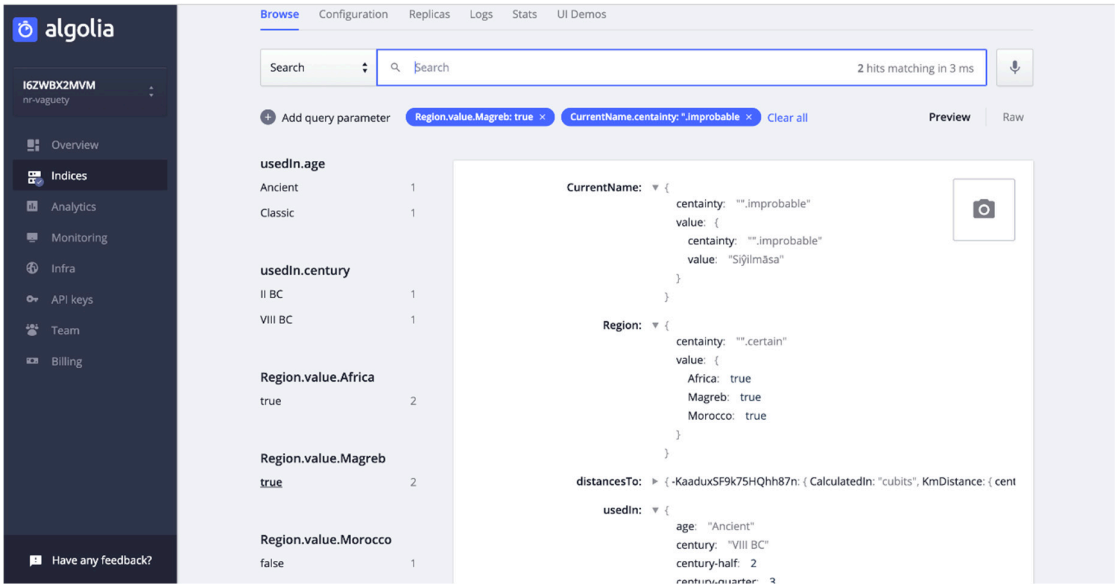

Figure 9. Results for query A.

In the first case, query A results offered all toponyms situated specifically at Maghreb whose current name certainty is improbable. The system recovers two toponyms with the following conditions: Sijilmasa and Tamdalt (see in Figure 2 the correct values of these toponyms according with the query requirments.). Figure 9 shows the results for the query, showing the data for Sijilmasa toponym.

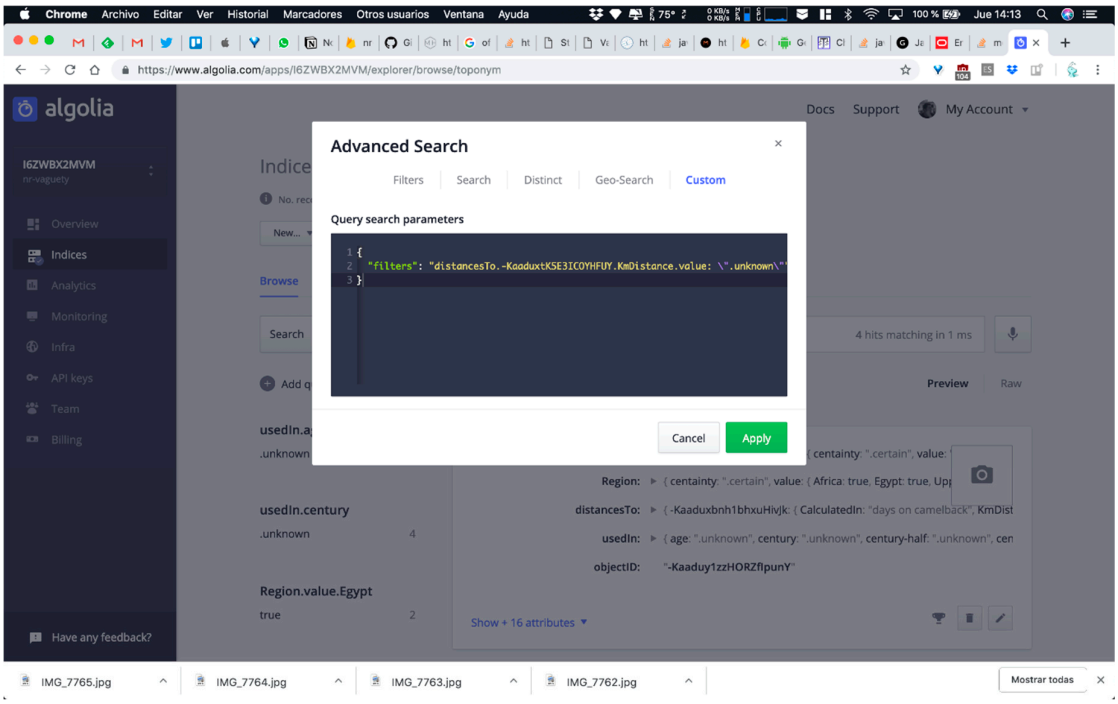

Figure 10. Query B execution using Algolia search engine. We have added custom expression on the Algolia console referring to Sijilmasa internal code as the reference point for recovering distances to it.

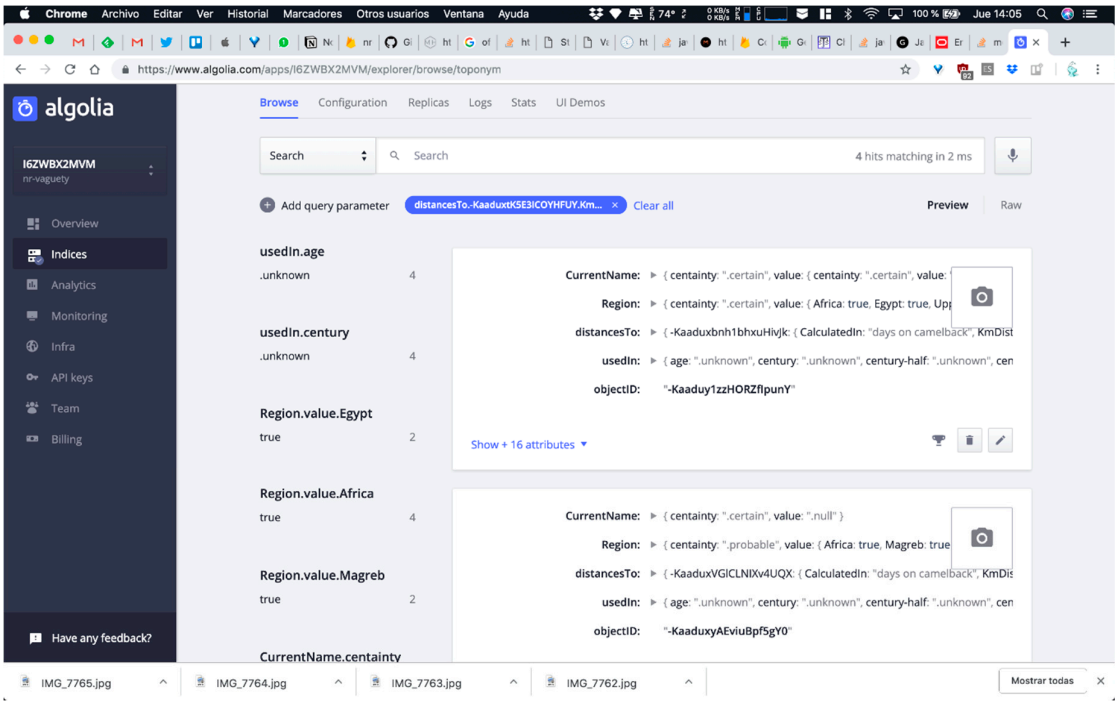

Figure 11. Results for Query B.

Regarding query B results, the systems recovers four toponyms (two of them are part of our example) whose distance in kilometers from *Sijilmasa* is unknown.

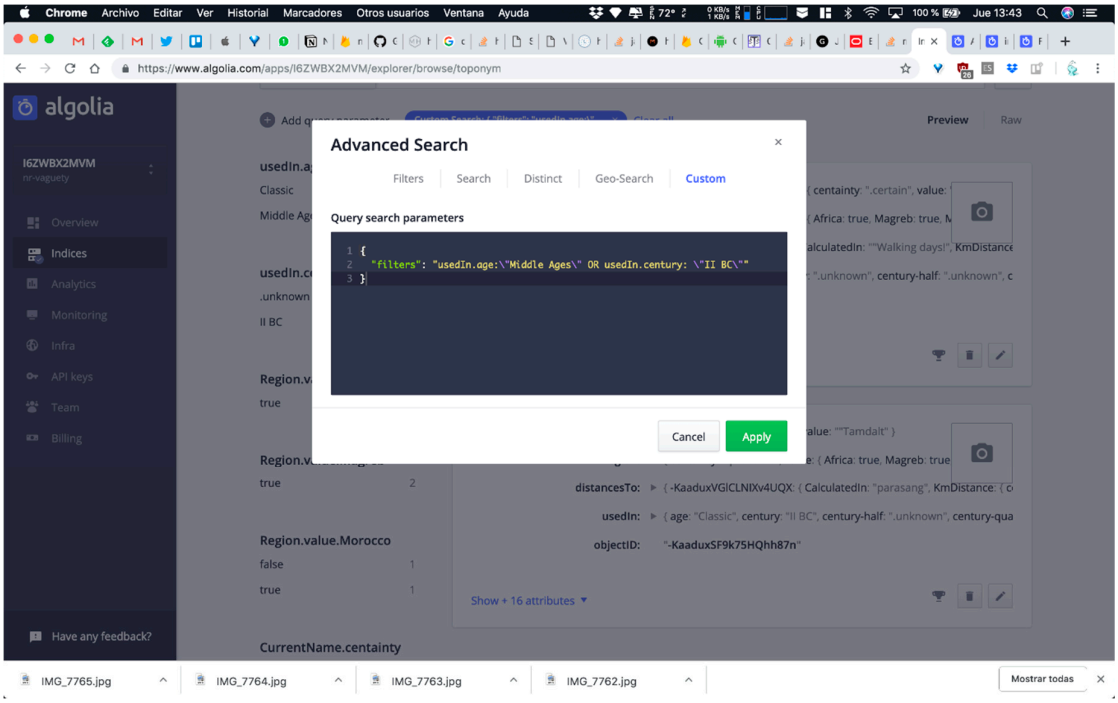

Figure 12. Query C execution through Algolia search engine. We have added a custom expression on the Algolia console with an OR expression for executing it.

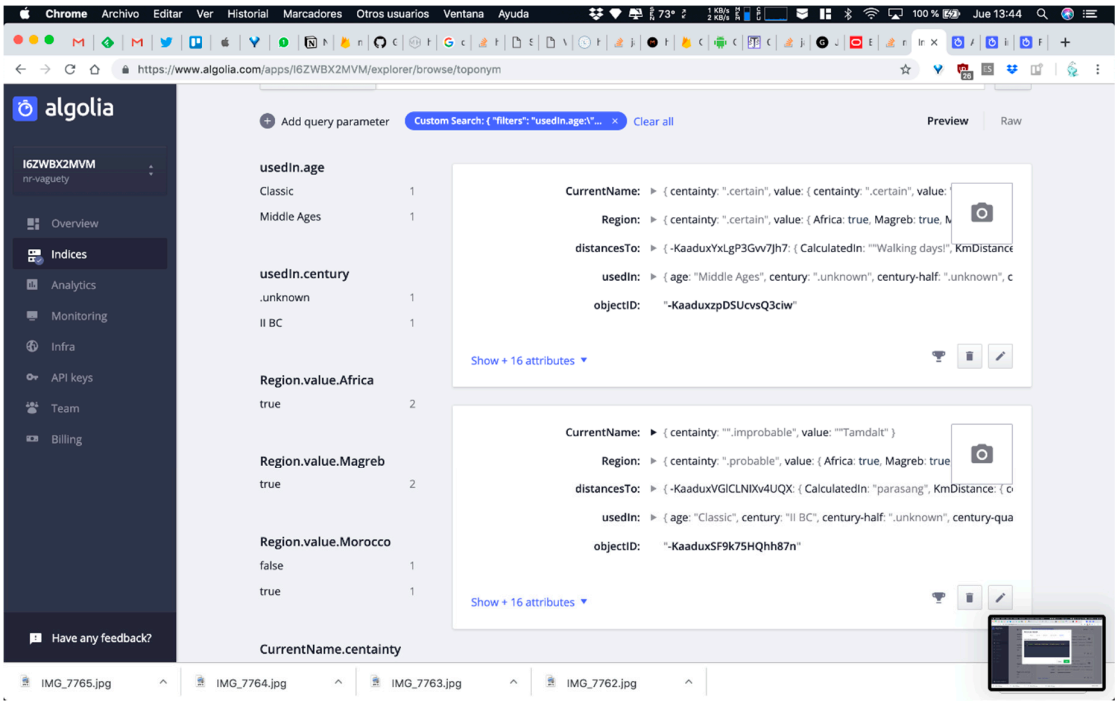

Figure 13. Results for query C.

Finally, query C involved the execution of two combined searching structures due to the fact we have to manage toponyms used in the middle ages or used in the second century BC. Logical operators are common in relational database structures, but less supported in non-relational systems. Algolia allow us to use OR logical operator thanks to the custom search console including in their dashboard. Query C results recovers 20 toponyms used in these periods of time, including two presented in our case: Aghmat Ourika and Tamdalt.

In summary, the previous implementation of the four ConML mechanisms for expressing vagueness in the Firebase non-relational database allowed us to define searches that include vagueness references in their specification, taking advantage of the capabilities of non-relational systems.

4. Discussion

The results obtained for the A, B, and C queries defined and the Firebase-based software system created for the presented implementation show that the non-relational implementation of the vagueness mechanisms is possible with vagueness resolution in the query system.

Note that, apart from the specific example that we wanted to show in this paper (represented in Figure 2), the software system manages all toponyms, retrieving those that meet the established vagueness criteria. It will be also possible to concretize the results only for our case study, using the filtering mechanisms on the original results. This filtering service is provided by Firebase (and almost all non-relational structure software systems) and could analyse only the case of Sijilmasa and related toponyms.

Because DICTOMAGRED [43] has a manageable number of nodes in its non-relational structure (currently DICTOMAGRED manages 53 toponyms with five hierarchical levels of information in their tree non-relational structure, which constitutes around 300 nodes of information), we could validate with the project researchers that the coverage of the implementation is total, that is, the conceptual model and the vagueness mechanisms created represent both the research needs of the project and the data source, obtaining accurate results (data that meet the conditions of indexing and searching) for queries A, B, and C. This type of expert-guided validation is only possible with a

manageable number of nodes, which are easily verifiable by humans. In other contexts, a solution based on monitoring the coverage of the algorithm automatically will be necessary.

Finally, it is important to highlight the need for the initial conception of vagueness support from the first stages of design of each project or concrete application. As can be seen, the queries that we have designed already take into account the possibilities of expressing the vagueness of the software system since they arise from the previously conceptual model created. Without this previous conceptual design, the queries designed would probably not follow the vagueness logic of the model.

We believe that the presented implementation constitutes an important advance for the support of vagueness in digital humanities at a conceptual level. Specially, and going back to the motivation of this work, the explicit addressing of the value added for the vague information in the humanities is treated through the proposal presented, providing mechanisms for future projects with same needs to deal with the vagueness in their implementation, instead of adapting non-vague support solutions. In addition, and due to the performance advantages of non-relational systems, there are currently more applications and projects in digital humanities that choose non-relational structures to manage their data. This implementation can serve as a relevance reference for these type of projects and applications with clear vagueness management needs.

5. Conclusions

Imprecise and uncertain information constitutes an intrinsic characteristic of the digital humanities research practice, and, when properly modelled and expressed, may comprise a valuable asset. This paper has reviewed most well-known approaches to the modelling of vagueness, and presented a theoretical framework and specific modelling mechanisms in ConML for the expression of ontological and epistemic vagueness in the digital humanities. As illustrated by an application to a real project, these mechanisms allow researchers to express imprecision and uncertainty in their own models. In addition, the implementation proposal presented allows them to fulfill their vagueness needs without a large penalty in analytical and processing power, thanks to the non-relational structures. As far as we know, this is the first implementation proposal for vagueness in digital humanities that offers a software solution for vagueness from the conceptual model design to the implementation in a real digital humanities project, dealing with specific examples of vagueness needs.

Due to this innovative component, critical analysis is also needed. Some suggestions for making improvements are identified as part of our future roadmap.

The first aspect is that, in contrast to data coverage validation that we have already mentioned in the previous section, we do not have data about the performance of the software system (time for solving a query, etc.). It has not been considered necessary to measure them because, below a specific volume of information (as DICTOMAGRED volume), it is difficult to obtain reliable measures in performance. As a future plan, it is necessary to evaluate the implementation presented with a greater volume of nodes, so that the performance in some searches could be compromised. This is especially relevant in queries that involve the vagueness mechanism of arbitrary time resolution or that involve more than one vagueness mechanism at the same time. In addition, we plan to compare the performance results obtained with implementations in relational structures, in order to stablish some criteria or guidelines that will help engineers and digital humanities project managers in making decisions about their implementation data structure based on the informational needs of each project or application.

Secondly, some of the defined vagueness mechanisms are closely related to new implementation techniques related to fuzzy logic. For instance, certainty qualifiers could be seen as fuzzy characterizations of information. For theses reason, we are also considering fuzzy sets and levels of set membership [32–34] or similar rule-based logic mechanisms [32] for improving specific details of the implementation of vagueness.

Finally, the application of the proposal presented here, both at a conceptual level and at the implementation level, to heterogeneous projects or application on digital humanities will allow us to test the vagueness mechanisms expressiveness and the implementation in a variety of humanistic contexts and realities.

These future works will allow us to improve applications and specific implementations of the preset proposal for cases with greater demands for vagueness in the digital humanities.

Author Contributions: Conceptualization, C.G.-P.; Data Curation and Software Implementation, P.M.R.; Methodology, Validation, Formal Analysis and Writing, C.G.-P. and P.M.R.

Funding: This research was partially funded by Spanish Ministry of Economy, Industry, and Competitiveness under its Competitive Juan de la Cierva Postdoctoral Research Programme, grant FJCI-2016-28032.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ackoff, R.L. From data to wisdom. *J. Appl. Sys. Anal*. **1988**, *16*, 3–9.
2. Ciula, A.; Eide, Ø. Modelling in digital humanities: Signs in context. *Digit. Scholarsh. Humanit*. **2016**, *32* (suppl_1), i33–i46.
3. Gonzalez-Perez, C. *Information Modelling for Archaeology and Anthropology: Software Engineering Principles for Cultural Heritage*; Springer International Publishing: Berlin, Germany, 2018. doi: 10.1007/978-3-319-72652-6.
4. Europeana. Europeana project 2008–2015 [26/04/2016]. Available online: http://www.europeana.eu/ (accessed on 22 March 2019).
5. ARIADNE. ARIADNE Project 2013. Available online: http://ariadne-infrastructure.eu/ (accessed on 22 March 2019).
6. DARIAH-EU. Digital Research Infrastructure for the Arts and Humanities (DARIAH) 2007–2015 [26/04/2016]. Available online: https://dariah.eu/ (accessed on 22 March 2019).
7. Incipit. ConML Technical Specification. ConML 1.4.4 2015. Available online: http://www.conml.org/Resources_TechSpec.aspx (accessed on 22 March 2019).
8. Flanders, J.; Jannidis, F. Data modeling. In *A New Companion to Digital Humanities*; Schreibman, S., Siemens, R., Unsworth, J., eds.; Wiley: Hoboken, NJ, USA, 2015.
9. Flanders, J.; Jannidis, F. Knowledge Organization and Data Modeling in the Humanities. Available online: https://www.wwp.northeastern.edu/outreach/conference/kodm2012/flanders_jannidis_datamodeling.pdf (accessed on 22 March 2019).
10. Hedges, M. Grid-enabling humanities datasets. *Digit. Humanit. Q*. **2009**, *3*, 4.
11. Linked Data. Available online: http://linkeddata.org/ (accessed on 22 March 2019).
12. Chen, P.P.-S. The entity-relationship model: Toward a unified view of data. In *Readings in Artificial Intelligence and Databases*; Elsevier: Amsterdam, The Netherlands, 1988; pp. 98–111.
13. W3C. RDF Schema 1.1. W3C Recommendation 25 February 2014. Available online: https://www.w3.org/TR/rdf-schema/ (accessed on 22 March 2019).
14. Hunter, A.; Liu, W. Representing and Merging Uncertain Information in XML: A Short Survey. Available online: http://www0.cs.ucl.ac.uk/staff/A.Hunter/papers/saj.pdf (accessed on 22 March 2019).
15. Consortium, T. Text Enconding Initiative (TEI) 2016. Available online: http://www.tei-c.org/index.xml (accessed on 22 March 2019).
16. Isaksen, L.; Simon, R.; Barker, E.T.E.; de Soto Cañamares, P. Pelagios and the emerging graph of ancient world data. In Proceedings of the 2014 ACM conference on Web science; Bloomington, Indiana, USA, 23–26 June 2014; pp. 197–201.
17. Commons, P. Pelagios Commons WebSite (Pelagios 6 Project). Available online: http://commons.pelagios.org/ (accessed on 22 March 2019).
18. Gonzalez-Perez, C.; Martín-Rodilla, P. Teaching Conceptual Modelling in Humanities and Social Sciences. *Digit. Humanit. Mag*. **2017**, *1*, 408–416.
19. Chirico, R.D.; Frenkel, M.; Diky, V.V.; Marsh, K.N.; Wilhoit, R.C. ThermoML—An XML-Based Approach for Storage and Exchange of Experimental and Critically Evaluated Thermophysical and Thermochemical Property Data. 2. Uncertainties. *J. Chem. Eng. Data* **2003**, *48*, 1344–1359.
20. ISO. ISO 21127:2006 Information and Documentation—A Reference Ontology for the Interchange of Cultural Heritage Information 2006. Available online: https://www.iso.org/standard/34424.html (accessed on 22 March 2019).
21. De Runz, C.; Desjardin, E.; Piantoni, F.; Herbin, M. Using Fuzzy Logic to Manage Uncertain Multi-modal Data in an Archaeological GIS. Available online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.108.7063 (accessed on 22 March 2019).

22. Tolle, K.; Wigg-Wolf, D. Uncertainty … ? ECFN Meeting 2014—Basel Goethe University 2014. Available online: http://ecfn.fundmuenzen.eu/images/Tolle_Wigg-Wolf_Uncertainty.pdf (accessed on 6 May 2019).
23. Christensen-Dalsgaard, B.; Castelli, D.; Jurik, B.A.; Lippincott, J. Research and Advanced Technology for Digital Libraries. In proceedings of 12th European Conference, ECDL 2008, Aarhus, Denmark, 14–19 September 2008.
24. Van Ruymbeke, M.; Hallot, P.; Billen, R. Enhancing CIDOC-CRM and compatible models with the concept of multiple interpretation. *Remote Sens. Spat. Inf. Sci.* **2017**, *4*, 287. doi:10.5194/isprs-annals-IV-2-W2-287-2017.
25. Ore, C.-E.; Eide, Ø. TEI and cultural heritage ontologies: Exchange of information? Lit. Linguist. Comput. **2009**, *24*, 161–172.
26. PROVIDEDH. PROgressive VIsual DEcision-Making in Digital Humanities (PROVIDEDH) Project 2019. Available online: https://providedh.eu (accessed on 22 March 2019).
27. ISO/IEC. Information Technology—Object Management Group Unified Modeling Language (OMG UML) Part 1: Infrastructure. ISO/IEC 19505-1:2012. Available online: https://www.iso.org/standard/32624.html (accessed on 22 March 2019).
28. Malta, M.C.; González-Blanco, E.; Cantón, C.M.; Del Rio, G. A Common Conceptual Model for the Study of Poetry in the Digital Humanities. Available online: https://dh2017.adho.org/abstracts/148/148.pdf (accessed on 22 March 2019).
29. Lacerda, M.J.; Crespo, L.G. Interval predictor models for data with measurement uncertainty. In proceedings of 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017.
30. Zadeh, L.A. Fuzzy logic= computing with words. IEEE T. Fuzzy Sys. **1996**, *4*, 103–111.
31. Zadeh, L.A. A Summary and Update of "Fuzzy Logic". In proceedings of 2010 IEEE International Conference on Granular Computing, San Jose, CA, USA, 14–16 Aug. 2010.
32. Bouchon-Meunier, B. Strengths of Fuzzy Techniques in Data Science. Available online: https://hal.sorbonne-universite.fr/hal-01676195/document (accessed on 22 March 2019).
33. Zhou, H.; Wang, J.-Q.; Zhang, H.-Y. Multi-criteria decision-making approaches based on distance measures for linguistic hesitant fuzzy sets. *J. Oper. Res. Soc.* **2018**, *69*, 661–675.
34. Faizi, S.; Rashid, T.; Sałabun, W.; Zafar, S.; Wątróbski, J. Decision making with uncertainty using hesitant fuzzy sets. *Int. J. Fuzzy Sys.* **2018**, *20*, 93–103.
35. OMG. Project Portal for OMG® Uncertainty Modeling (UM) 2017. Available online: http://www.omgwiki.org/uncertainty/doku.php?id=Home (accessed on 22 March 2019).
36. Yue, T.; Ali, S.; Selic, B. Available online: Standardizing Uncertainty Modeling at OMG. http://www.cister.isep.ipp.pt/ae2016/presentations/utest2.pdf (accessed on 22 March 2019).
37. Xiao, J.; Pinel, P.; Pi, L.; Aranega, V.; Baron, C. Modeling uncertain and imprecise information in process modeling with UML. In proceedings of Fourteenth International Conference on Management of Data (COMAD), Mumbai, India, 17–19 December, 2008.
38. Jackson, C.H.; Bojke, L.; Thompson, S.G.; Claxton, K.; Sharples, L.D. A framework for addressing structural uncertainty in decision models. *Med. Decis. Mak.* **2011**, *31*, 662–674.
39. Ottomanelli, M.; Wong, C.K. Modelling uncertainty in traffic and transportation systems. *Transportmetrica* **2011**, *7*, 1–3.
40. Sarma, A.D.; Benjelloun, O.; Halevy, A.; Nabar, S.; Widom, J. Representing uncertain data: Models, properties, and algorithms. *VLDB* **2009**, *18*, 989–1019.
41. Martín-Rodilla, P.; Gonzalez-Perez, C. Assessing the learning curve in archaeological information modelling: Educational experiences with the Mind Maps and Object-Oriented paradigms. In proceedings of 45th Computer Applications and Quantitative Methods in Archaeology (CAA 2017), Atlanta, GA, USA, 13–16 March 2017.
42. IEMYR. Instituto de Estudios Medievales y Renacentistas y de Humanidades Digitales IEMYRhd 2018. Available online: http://iemyr.usal.es/ (accessed on 22 March 2019).
43. Dictomagred. DICTOMAGRED: Diccionario de Toponimia Magrebí 2018. Available online: https://dictomagred.usal.es/ (accessed on 22 March 2019).
44. Rodríguez, M.A.M. Paisajes, espacios y objetos de devoción en el Islam. Available online: https://dialnet.unirioja.es/servlet/libro?codigo=708334 (accessed on 22 March 2019).
45. Sharp, J.; McMurtry, D.; Oakley, A.; Subramanian, M.; Zhang, H. *Data access for highly-scalable solutions: Using SQL, NoSQL, and polyglot persistence*; Microsoft Patterns & Practices: Redmond, DC, USA, 2013.

46. Freitas, M.Cd.; Souza, D.Y.; Salgado, A.C. Conceptual Mappings to Convert Relational into NoSQL Databases. In Proceedings of the 18th International Conference on Enterprise Information Systems; Rome, Italy, 25–28 April, 2016.
47. What are NoSQL Databases? Available online: https://aws.amazon.com/nosql/ (accessed on 22 March 2019).
48. MongoDB. Available online: https://www mongodb com/ (accessed on 22 March 2019).
49. Inc. G. Firebase 2019 [01/03/2019]. Available online: https://firebase.google.com/ (accessed on 22 March 2019).
50. Abramova, V.; Bernardino, J. NoSQL databases: MongoDB vs cassandra. In Proceedings of the international C* conference on computer science and software engineering, Porto, Portugal, 10–12 July 2013.
51. Algolia. Algolia Website 2019. Available online: https://www.algolia.com/ (accessed on 22 March 2019).