Computing Query Answering With Non-Monotonic Rules: A Case Study of Archaeology Qualitative Spatial Reasoning

Eric Würbel, Mohamed Ellefi, Pierre Drap, Laurent Garcia, Fabien Garreau, Claire Lefèvre, Odile Papini, Igor Stéphan

EPiC Series in Computing

Volume XXX, 2019, Pages 1–14

GCAI 2019. 5th Global Conference on Artificial Intelligence

Query Answering With Non-Monotonic Rules: A Case Study of Archaeology Qualitative Spatial Reasoning *

Mohamed Ben Ellefi[1], Pierre Drap[1], Laurent Garcia[2], Fabien Garreau[2], Claire Lefèvre[2], Odile Papini[1], Igor Stéphan[2], and Éric Würbel[1]

[1] Aix Marseille University, University of Toulon, CNRS, LIS, Marseille, France; - {mohamed.ben-ellefi, drap, papini, wurbel}@univ-amu.fr

[2] LERIA. University of Angers, France - {garcia, lefevre, garreau, stephan}@info.univ-angers.fr

Abstract

This paper deals with querying ontology-based knowledge bases equipped with non-monotonic rules through a case study within the framework of Cultural Heritage. It focuses on 3D underwater surveys on the Xlendi wreck which is represented by an OWL2 knowledge base with a large dataset. The paper aims at improving the interactions between the archaeologists and the knowledge base providing new queries that involve non-monotonic rules in order to perform qualitative spatial reasoning. To this end, the knowledge base initially represented in OWL2-QL is translated into an equivalent Answer Set Programming (ASP) program and is enriched with a set of non-monotonic ASP rules suitable to express default and exceptions. An ASP query answering approach is proposed and implemented. Furthermore due to the increased expressiveness of non-monotonic rules it provides spatial reasoning and spatial relations between artifacts query answering which is not possible with query answering languages such as SPARQL and SQWRL.

1 Introduction

Within the framework of Cultural Heritage (CH), an important issue is to develop efficient tools that fulfill the needs of archaeologists in their investigations. Studies on Underwater Cultural Heritage (UCH) sites induce the use of computerized techniques to handle, preserve and analyze the produced information. Since the 1970s, photogrammetry has been recognized as an essential means for this type of survey [34], [33], [8].

Indeed, photogrammetry is used in the context of UCH to obtain a reliable and accurate representation of sites, which are most often inaccessible or difficult to access. Since the last decade the progress in photogrammetry, computer vision and underwater robotics made possible to obtain realistic facsimiles of the sites. However, these technological advances do not solve the archaeologists' questioning but deport the reflexion from sites to laboratories where cultural heritage photogrammetry meets knowledge representation and reasoning in artificial intelligence. An ontology-profiling method has been proposed for modeling archaeological artifacts [10]. This ontology-based approach consists in modeling cultural heritage through three dimensions: typological, photogrammetrical and spatial. More recently, in order

*This work was supported by the projects ASPIQ (ANR-12-BS02-0003) and GROPLAN (ANR-13-CORD-0014)

to improve the interactions with the user, web tools have been proposed in the semantic web environment allowing user-friendly query answering [9]. These queries are performed via SPARQL services over several linked open datasets corresponding to different archaeological dives in the Xlendi shipwreck. Others approaches tackle the issue of ontology-based semantic image interpretation [14] that combine image features and DL constraints in order to improve a semantic understanding of the content of pictures. However, the construction of an ontology that describes the content of images is out of the focus of this paper.

The aim of this paper is to go a step further in order to get closer to the archaelogists lines of reasoning. When facing 3D underwater surveys, the archaeologists do not perform any computation, they are not fully interested on the accuracy of calculations, in contrast, they focus on the relationships between artifacts to perform qualitative spatial reasoning. The main reason to avoid any calculation is that performing calculations is time consuming, moreover in the absence of data performing calculations is useless.

The archaeologists interact with 3D underwater surveys through queries to an OWL2 knowledge base consisting of an ontology combining underwater archaeological knowledge and photogrammetric measurement knowledge, and a set of observations on underwater archaeological sites [17]. The web-based tools proposed in [9] allow the users to perform advanced SparQL [19] queries to the OWL2 knowledge base concerning typological and photogrammetric features. Furthermore, in order to express spatial queries, rules in SWRL [25] have been added to the OWL2 knowledge base [17]. However, performing qualitative reasoning leads to queries stemming from non-monotonic rules that cannot be expressed in SPARQL nor in SQWRL (SWRL-based query language).

Among the spatial qualitative notions, we concentrate on the notion of closeness between artifacts without any calculation. To this end, we focus on the visibility of artifacts thanks to the high density of artifacts on the archaeological underwater sites. We assume that if an artifact is visible from another one, the latter is close to the former. Visibility between two artifacts means that there is no obstacle between them, therefore the notion of visibility is expressed by a non-monotonic rule and requires a non-monotonic formalism for its representation.

Combining Description logics and non-monotonic features is an important issue, and several approaches have been proposed (eg. [18, 31, 30, 37, 1, 2, 15, 13, 22, 12, 23, 11]). Among them, NoHR, a plug-in for the ontology editor Protégé 5.X, founded on hybrid MKNF [31] under well-founded semantics [27], allows the user to query combinations of ontologies and non-monotonic rules.

Besides, Answer Set Programming [21] (ASP) is an efficient unified formalism for both knowledge representation and reasoning in Artificial Intelligence (AI). It is a non-monotonic logic programming language allowing representation practical to reason with incomplete data. ASP has an elegant and conceptually simple theoretical foundation and has been proved useful for solving a wide range of problems in various domains [35]. Beyond its ability to formalize various problems from AI or to encode combinatorial problems [7, 32], ASP provides also an interesting way to practically solve such problems since some efficient solvers are available [20, 29].

More recently, within the context of the ASPIQ project [1] which aimed at proposing new solutions for querying large scale multi-source heterogeneous information, an extension of Answer Set Programming (ASP) with existential variables in the head of the rules has been proposed [3, 6], called ∃ASP. This extension allows one to encode OWL2 knowledge bases into ASP, more precisely, knowledge bases expressed in tractable sub-languages dedicated to query answering, like OWL2-QL.

In this paper, we show how Answer Set Programming formalism allows one to represent the OWL2 knowledge base as well as non-monotonic rules and makes possible for the archaeologists to query the OWL2 knowledge base equipped with non-monotonic rules in order to perform spatial qualitative

[1] http://aspiq.lsis.org

reasoning. Moreover we implement ASP query answering on the Xlendi archaeological site, a real world problem with a large data set within the context of underwater archaeological surveys studied within the GROPLAN project [2].

The paper is organized as follows. Section 2 describes the real world case study, namely the Xlendi wreck survey. Section 3 details the non-monotonic rules for performing spatial qualitative reasoning. Section 4 gives a brief reminder on ∃ASP. Section 5 discusses ASP query answering on the ontological knowledge base enhanced by non-monotonic rules. Section 6 presents the results of the ASP query answering approach: implementation and visualization of the results on the Xlendi archaeological site, before concluding in Section 7.

2 The Xlendi Case study

This paper deals with deep underwater archaeology data management focusing on the Xlendi wreck case-study. The data are acquired by photogrammetry completely diver-less and without any kind of contact with the site. The survey is done by photogrammetry to measure both the sea bed and archaeological artifacts. In this paper we focus on amphorae representation based on 3D data coming from photogrammetric survey. The carried out photogrammetric survey is based on an original approach of underwater photogrammetry that was deployed with the help of a specific instrumental infrastructure provided by COMEX, a partner in the GROPLAN project [16]. This photogrammetry process, as well as the body of surveyed objects, were formalized in an ontological knowledge base expressed in OWL2 [3]. Our approach is based on procedural attachment; the ontology being seen as a dual of the JAVA class structure that manages the photogrammetric survey and the measurement of artifacts. This allows the establishment of a reasoning for the ontologies as well as the intensive calculations using the JAVA programming language with the same interface. Besides, the ontology used to describe the archaeological artifacts from a measurement point of view is aligned with CIDOC-CRM ontology used for museographical objects. The deal here is to be able to use both geometrical and archaeological data in the same ontology [24].

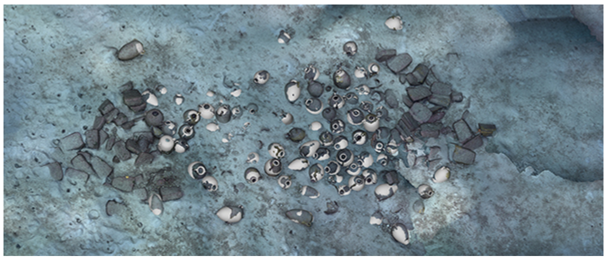

Figure 1: The Xlendi wreck photogrammetric survey with amphorae completion according to their own typology. The ontological KB represents all these reconstructed amphorae (in white on the image).

The ontology has been developed to represent the photogrammetry process used for the survey and the process identification, measure, representation of visible archaeological objects. The final ontology is on one hand an ontology built from a JAVA program modelling the entire photogrammetric process and on the other hand an ontology describing the archaeological artifacts from the point of view of the photogrammetric measure. The goal is to link the measured artifacts with all the observations used

[2]http://www.lsis.org/groplan/
[3]http://arpenteur.org

to measure and identify them. One of the main advantage of the photogrammetric process is to provide several 2D representations of the measured artifacts. This first ontology is built from an existing JAVA code in order to represent the concepts used in photogrammetry and to be able to use a reasoner on the ABox representing photogrammetric data. We need to manage both the computational aspects (often heavy in photogrammetry) implemented in the artifacts measurable by photogrammetry and the ontological representation of the same photogrammetric process and surveyed artifacts. The current implementation is based on a double formalism, JAVA, used for computation, photogrammetric algorithms, 3D visualization of photogrammetric data and patrimonial objects, and OWL for the definition of ontologies describing the concepts involved in the measurement process and the link with the measured objects. The ontology construction in OWL, dual to the JAVA taxonomy, cannot be produced automatically. Each concept of the ontology has been constructed in a concern for the representation of fine knowledge from a specific point of view: measurement. Indeed, the same point of view presides over the development of the JAVA taxonomy, but software engineering constraints are superimposed on a point of view strictly linked to knowledge of concepts. We have abandoned an automatic mapping using JAVA annotation and JAVA beans for a manual extraction even if this is a common way in literature [26, 36]. The main advantage of our approach is that it is possible to perform logical queries on both the ontology and the JAVA representation. We can thus read an ontology, visualize in 3D the artifacts present in the ontology as well as the result of logical queries in a 2D or 3D viewer. Moreover, qualitative 2D spatial relations between artifacts are described within the ontological KB. The plane is divided into 8 cardinal areas, namely n, ne, e, se, s, sw, w, nw[4]. For every pair of artifacts the qualitative cardinal direction relations are computed from the coordinates and orientation of their barycentre and are available. In the rest of the paper, we study the visibility relation between Xlendi shipwreck artifacts represented with non-monotonic rules for ontological KB query answering.

3 Non-Monotonic Rules for Spatial Reasoning

We focus on closeness between artifacts, through the visibility relations between the artifacts due to the high density of artifacts on the wreck site. Visibility between two artifacts means that there is no obstacle between them. The idea is to define a relation $visible$ between two artifacts and this requires non-monotonic rules to represent it.

The wreck plane is divided into eight cardinal areas, namely n, ne, e, se, s, sw, w, nw. These areas are denoted by D_i, $0 \leq i \leq 7$ and are clockwise ordered, *i.e.,* $D_0 = n, \cdots, D_7 = nw$. In order to model the fact that an artifact Y is visible from X towards the direction D_i, we consider the three adjacent cardinal areas, $D_{prec(i)}$ with $prec(i) = (8 - ((9 - i) \,\%\, 8)) \,\%\, 8)$ (preceding direction), D_i and $D_{next(i)}$ with $next(i) = (i+1) \,\%\, 8$ (next direction), their inverse (the opposite direction), $D^-_{prec(i)}$, D^-_i and $D^-_{next(i)}$ respectively, and an additional artifact Z, distinct from X and Y, which can represent an obstacle to the visibility from X towards the direction D_i. For any area D_i the rule modelling the notion of visibility of an artifact Y from a given artifact X towards the D_i cardinal direction is the following:

If Y is in direction D_i w.r.t. X and

(1) for any artifact Z such that Z is in direction D_i w.r.t. X, Z is not in direction D^-_i, nor in direction $D^-_{prec(i)}$, and nor in direction $D^-_{next(i)}$ w.r.t. to Y, and

(2) for any artifact Z such that Z is in direction $D_{prec(i)}$ w.r.t. X, Z is not in direction D^-_i, nor in direction $D^-_{next(i)}$ w.r.t. to Y, and

[4] n for $North$, ne for North East, e for East, se for South East, s for South, sw for South West, w for West, nw for North West.

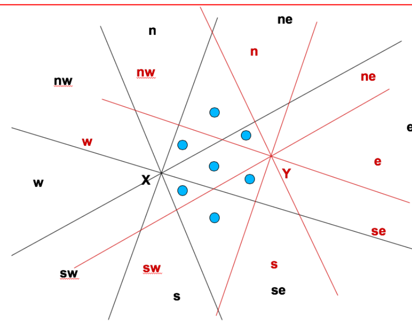

Figure 2: Blue points indicate the artifacts Z which are obstacles to the visibility from X toward the East direction.

(3) for any artifact Z such that Z is in direction $D_{next(i)}$ w.r.t. to X, Z is not in direction D_i^-, nor in direction $D_{prec(i)}^-$ w.r.t. to Y,

then Y is visible from X towards the D_i cardinal direction.

Conditions (1), (2) and (3) represent the three cases where an obstacle has to be excluded and these three conditions have to be satisfied in order to ensure there is no obstacle between X and Y.

Example 1. *Let $i = 2$, $D_i = e$, the rule modelling the notion of visible artifacts towards the East direction is the following:*
If $eof(X,Y)$ and

(1) *for any artifact Z such that $eof(X,Z)$ and not $nwof(Y,Z)$, nor $wof(Y,Z)$, and nor $swof(Y,Z)$, and*

(2) *for any artifact Z such that $neof(X,Z)$, and not $nwof(Y,Z)$ nor $wof(Y,Z)$, and*

(3) *for any artifact Z such that $seof(X,Z)$, and not $wof(Y,Z)$ nor $swof(Y,Z)$,*

then Y is visible from X towards the East direction.

Figure 2 illustrates the notion of visibility towards the East direction. Y is visible from X towards the East direction if there does not exist any obstacle Z, materialized by the blue points, in between X and Y.

4 ASP

We briefly present the ∃ASP formalism which successfully extends ASP with existential variables in the head of rules [6] especially to deal with ontologies. Let $\mathcal{A}$ be the set of all atoms, an ∃ASP program is a set of rules of the form

$$r \;:\; h_1, \ldots, h_v \leftarrow b_1, \ldots, b_m, not\; n_1 \ldots, not\; n_s. \qquad (1)$$

where $m + s, v \geq 1$, and $h_1, \ldots, h_v, b_1, \ldots, b_m, n_1, \ldots, n_s$ are atoms of $\mathcal{A}$. The keyword *not* denotes negation as failure. The set $h_1, \ldots, h_v$ is the head of a rule denoted by $head(r)$, $b_1, \ldots, b_m$ the positive body, which is denoted by $body(r)$, and $not\; n_1, \ldots, not\; n_s$ the negative body denoted by $body^-(r)$.

Thus, $body(r) = body^+(r) \cup body^-(r)$ with $body(r)$ the body of the rule r. Multiple atoms in the head is interpreted like a conjunction of atoms (and not a disjunction as in a disjunctive ASP). An ∃ASP rule is such that each variable appearing in the head of rule without appearing in the positive body is existentially quantified. Intuitively, a rule can be understood as follows: if all the atoms b_i of the positive body of the rule are true, and if none of the atoms n_j of the negative body are true, then the head of the rule can be inferred. For a rule r of type (1), we denote by r^+ the rule $head(r) \leftarrow body^+(r)$. For any set of rules P, $Atoms(P)$ the set of atoms appearing in P. By a slight abuse of notation, for a single rule r we write $Atoms(r)$ to denote the set of atoms appearing in r.

An ∃ASP program can easily be translated into a classical ASP program using two steps, skolemization and expansion. Those steps are meant to transform rules with existential variables and multiple atoms in head into equivalent rules using only universal variables.

Skolemization First step, skolemization is used to remove existential variables in the head of rules. The principle is to transform existential variables into functional terms using only universal variables. Let Y be an existential variable, the skolemized variable Y is a functional term $sk^Y(X_1, \ldots, X_n)$ with $X_1, \ldots, X_n$ all the variables appearing both in the positive body and the head of the rule and sk^Y a unique symbol function. If there is no universal variables in the head then sk^Y is a constant.

Expansion Second step, the expansion is used to remove conjunction of atoms in the head of rules. Let r be an ∃ASP rule ($m + s > 0$, $v > 0$):

$$r = h_1, \ldots, h_v \leftarrow b_1, \ldots, b_m, not\ n_1, \ldots, not\ n_s$$

The expansion of such an ∃ASP rule is the following set of ∃ASP rules :

$$r = \left\{ \begin{array}{l} h_1 \leftarrow b_1, \ldots, b_m, not\ n_1, \ldots, not\ n_s \\ \ldots \\ h_v \leftarrow b_1, \ldots, b_m, not\ n_1, \ldots, not\ n_s \end{array} \right\}$$

The skolemization followed by the expansion of an ∃ASP program is an equivalent standard ASP program. Skolemization must be performed before expansion to keep the link between two existential variable in the same rule.

The usual semantics for skolemized $\exists ASP$ programs is the same as for normal programs, it is given by the *answer set semantics*. For any program P, a subset $S \subseteq Atoms(P)$ *satisfies* a rule r of the form (1) if $body^+(r) \subseteq S$ and $body^-(r) \cap S = \emptyset$ implies $head(r) \subseteq S$. The Gelfond-Lifschitz reduct [21], of a logic program P by a set of atoms X is the program $P^X = \{head(r) \leftarrow body^+(r) \mid r \in P, body^-(r) \cap X = \emptyset\}$. By definition, an answer set (or stable model) of P is a set of atoms $X \subseteq \mathcal{A}$ such that $X = Cn(P^X)$ where $Cn(P)$ denotes the smallest set of atoms closed under P.

A logic program may have zero, one or several answer sets. Note that a logic program with no default negation (definite logic program) has a unique answer set (minimal Herbrand model). The set of answer sets of a logic program P is denoted by $\mathcal{AS}(P)$ and if $\mathcal{AS}(P) \neq \emptyset$ the program is said *consistent* otherwise it is said *inconsistent*.

5 Combining the ontological KB with non-monotonic rules for query answering

Ontological KB translated into ASP and enhanced by ASP non-monotonic rules In order to encode an OWL2 knowledge base into ASP, the translation requires three steps. (1) The OWL2 knowledge base is first translated into the *existential rules* formalism [5] via a set of tools, called graal [5] [4]. (2) Secondly the resulting set of existential rules is translated into ∃ASP. (3) Finally, the ∃ASP program is translated into a standard ASP program by skolemization of existential variables[6] [6].

The translation is presented in Figure 3. This ASP encoding aims at taking advantage of the efficiency of ASP solvers in order to efficiently query knowledge bases with large data sets. The OWL2

Figure 3: ASP encoding of an OWL2 knowledge base

knowledge base translated into an ASP program $\mathcal{F}$. Cardinal relations between artifacts, coming from the OWL2 knowledge base translation, are represented by atoms of the form $isAt(X, D, Y)$, where X and Y are artifacts and $D \in \{n, ne, e, se, s, sw, w, nw\}$ is a cardinal direction The intended meaning is that X is in direction D from Y. These atoms are derived from the atoms $nof(X, Y)$, $nwof(X, Y)$, $wof(X, Y)$, ..., provided by the OWL2 knowledge base. The ASP program $\mathcal{F}$ is then enriched with non-monotonic rules $\mathcal{R}$ allowing to perform spatial reasoning and the resulting ASP program is $P = (\mathcal{F}, \mathcal{R})$.

Non-monotonic rules The rule presented in Section 3 is translated into ASP rules. The general idea is as follows: when we have the information that $isAt(X1, D, X2)$ for a pair of artifacts $X1$ and $X2$ and a cardinal direction D, we make the hypothesis that they are visible from each other, except if we find that one of the conditions (1), (2) or (3) is violated. But before examining the corresponding encoding, we need some auxilliary ASP rules.

The first set of rules is a set of facts whose aim is to express the clockwise numbering of the cardinal directions :

$$R_1 = \{dirnum(n, 0)., dirnum(ne, 1)., \ldots, dirnum(nw, 7).\}$$

The second set of rules R_2 allows to describe the $next$ and $previous$ direction $D2$ of a direction $D1$ in the order defined by $dirnum$. It also defines the $inverse$ direction $D2$ of a direction $D1$.

$$R_2 = \left\{ \begin{array}{rcl} previous(D1, D2) & \leftarrow & dirnum(D1, N1), dirnum(D2, N2), \\ & & N2 = (8 - ((8 - N1 + 1)\,\%\,8))\,\%\,8. \\ next(D1, D2) & \leftarrow & dirnum(D1, N1), dirnum(D2, N2), N2 = (N1 + 1)\,\%\,8. \\ inverse(D1, D2) & \leftarrow & numdir(D1, N1), numdir(D2, N2), N2 = (N1 + 4)\,\%\,8. \end{array} \right\}$$

where $\%$ stands for the modulo operator.

Now let us figure out how condition (1) presented in section 3 can be violated : it is violated if there exists an artifact Z such that Z is in direction D_i w.r.t. X, and either (i) Z is in D_i^- w.r.t. Y, or (ii) Z is in D_{i-1}^- w.r.t. Y, or (iii) Z is in D_{i+1}^- w.r.t. to Y. These three violation conditions can be translated into three ASP rules which generate an atom $ko(Y, D, X)$, which presence indicates that Y cannot be

[5]https://graphik-team.github.io/graal/

[6]All tools are available on the ASPIQ web site. *http://aspiq.lsis.org/aspiq/?q=fr/node/37*

visible from X in direction D:

$$R_{r_1} = \left\{ \begin{array}{lll} ko(Y,D,X) & \leftarrow & isAt(Y,D,X), inverse(D,Dinv), \\ & & isAt(Z,D,X), Z \neq Y, isAt(Z,Dinv,Y). \\ ko(Y,D,X) & \leftarrow & isAt(Y,D,X), previous(D,D1), inverse(D1,D1inv), \\ & & isAt(Z,D,X), Z \neq Y, isAt(Z,D1inv,Y). \\ ko(Y,D,X) & \leftarrow & isAt(Y,D,X), next(D,D2), inverse(D2,D2inv), \\ & & isAt(Z,D,X), Z \neq Y, isAt(Z,D2inv,Y). \end{array} \right\}$$

We then proceed in the same spirit for condition (2). This condition is violated if there exists an artifact Z such that Z is in direction D_{i-1} w.r.t. X, and either (i) Z is in D_i^- w.r.t. Y, or (ii) Z is in D_{i+1}^- w.r.t. to Y. These two violation conditions can be translated into two ASP rules which generate an atom $ko(Y,D,X)$, which presence indicates that Y cannot be visible from X in direction D:

$$R_{r_2} = \left\{ \begin{array}{lll} ko(Y,D,X) & \leftarrow & isAt(Y,D,X), previous(D,D1), inverse(D,Dinv), \\ & & isAt(Z,D1,X), Z \neq Y, isAt(Z,Dinv,Y). \\ ko(Y,D,X) & \leftarrow & isAt(Y,D,X), previous(D,D1), next(D,D2), inverse(D2,D2inv), \\ & & isAt(Z,D1,X), Z \neq Y, isAt(Z,D2inv,Y). \end{array} \right\}$$

We then apply the same reasoning for condition (3). This condition is violated if there exists an artifact Z such that Z is in direction D_{i+1} w.r.t. X, and either (i) Z is in D_i^- w.r.t. Y, or (ii) Z is in D_{i+1}^- w.r.t. to Y. These two violation conditions can be translated into two ASP rules which generate an atom $ko(Y,D,X)$, which presence indicates that Y cannot be visible from X in direction D:

$$R_{r_3} = \left\{ \begin{array}{lll} ko(Y,D,X) & \leftarrow & isAt(Y,D,X), next(D,D2), inverse(D,Dinv), \\ & & isAt(Z,D2,X), Z \neq Y, isAt(Z,Dinv,Y). \\ ko(Y,D,X) & \leftarrow & isAt(Y,D,X), previous(D,D1), next(D,D2), inverse(D1,D1inv), \\ & & isAt(Z,D2,X), Z \neq Y, isAt(Z,D1inv,Y). \end{array} \right\}$$

Finally, we express that Y is visible from X if it is in direction D w.r.t. X and it does not violate any of the three conditions: $R_{visible} = \{visible(Y,X) \leftarrow isAt(Y,D,X), not\ ko(Y,D,X).\}$

Thus, the full program is $\mathcal{R} = R_1 \cup R_2 \cup R_{r_1} \cup R_{r_2} \cup R_{r_3} \cup R_{visible}$.

ASP query answering To answer a query in ASP programs we make use of conjunctive queries. A conjunctive query on a program is a rule as follows:
$ans(X_1,\ldots,X_k) \leftarrow a_1,\ldots,a_n$ with $k \geq 0$, $n > 0$, $\{a_1,\ldots,a_n\}$ a set of atoms, ans a predicate symbol not in the program nor in the body of the query (ans is called *answer predicate*) and $X_1,\ldots,X_k$ some variables at least in some a_i. In ASP an answer to a query is *valid* if there is at least one answer set to the program, otherwise the answer is $absurd$. If the answer is valid then there are two different cases if we consider either credulous or skeptical reasoning. Let $\mathbf{P} = (\mathcal{F}, \mathcal{R} \cup \{Q\})$ be the program $P = (\mathcal{F}, \mathcal{R})$ queried by $Q = ans(X_1,\ldots,X_k) \leftarrow a_1,\ldots,a_n$. with $k > 0$ and $n > 0$, a conjunctive query and $\mathcal{AS}(Q)$ the set of the answer sets of $\mathbf{P}$. The answer to a query is the set of substitutions of variables by ground atoms, denoted by Σ, such that

(i) for all $as \in \mathcal{AS}(Q)$, $\{\sigma(ans(X_1,\ldots,X_k))|\sigma \in \Sigma\} \subseteq as$, in context of skeptical reasoning or

(ii) there exists an $as \in \mathcal{AS}(Q)$ such that $\{\sigma(ans(X_1,\ldots,X_k))|\sigma \in \Sigma\} \subseteq as$, in context of credulous reasoning.

Query answering is an approach based on the maximum reduction of computation to answer a query. Instead of computing all possible deduction from the TBox, only the rules needed to answer the query are used, rest of the program is not used.

Since OWL2 language is monotonic, the ASP program corresponding to the ontological KB consists of rules with no negation as failure. Moreover, the ontological KB is consistent and the corresponding

ASP program has only one answer set. Therefore ASP query answering is performed by adding an additional rule with a unique predicate symbol, *answer predicate*, not appearing in the program nor in the body of the query and computing the answer set. The ontological KB base describes knowledge and observations about artifacts present on the wreck site, the following example illustrates the queries concerning various features of the artifacts.

Example 2. *An example of query is: "What is the name of amphorae with height between 0.4 meter and 0.6 meter ?" translated with the ASP rule:*
$ans(Y) \leftarrow amphorae(X), hasName(X, Y), hasHeight(X, Z), Z < 600, Z > 400.$

Note that this query could be expressed in SPARQL [19] or in SQWRL [25] and Section 6 shows that ASP query answering gives the same results with a reasonable running time.

When ASP program corresponding to the ontological KB is enriched with non-monotonic rules, as for example the ones proposed above (Section 3) and translated in ASP rules as described in Section 5. ASP query answering is performed by adding the ASP non-monotonic rules and the additional ASP query rule. In this case there may be several answer sets and skeptical or credulous reasoning can be performed. With the non-monotonic rules proposed above for modelling the notion of closeness, querying spatial relationships between artifacts is possible as illustrated by the following example.

Example 3. *Examples of spatial queries followed by their translation in ASP.*
"What are the amphorae which are visible from amphorae a towards $East$?"
$ans(Y) \leftarrow amphorae(a), amphorae(Y), visible\_E(a, Y).$
"What are the amphorae which are visible from amphorae a ?"
$ans(Y) \leftarrow amphorae(a), amphorae(Y), visible(a, Y).$
"What are the amphorae which are visible from amphorae a and are of typology "Ramon" ?"
$ans(Y) \leftarrow amphorae(a), amphorae(Y), hasTypology(Y, \text{"}Ramon\text{"}), visible(a, Y).$

Note that these queries cannot be expressed in SQWRL, in SWRL nor in SparQL since negation as failure cannot be fully represented in these languages. Negation by failure can sometimes be simulated in SQWRL however it requires the use of collections and collection selection operators which is rather cumbersome when dealing with complex queries. Moreover, the results of the query cannot be used and added into the knowledge base. SWRL does not offer the possibility of creating collections. In SparQL 1.1 negation by failure can be simulated by specifying an OPTIONAL graph pattern that introduces a variable and testing to see that the variable is not bound, however this is very hard to use as the complexity increases.

Figure 4 illustrates how the answers to a query are computed with ASP query answering. We first need to translate the OWL2-QL Knowledge base into an ASP logic program (see Figure 3). In the case of classical knowledge bases (without default negation), an equivalent ASP program has only one answer set, we can therefore preload the knowledge base with a first computation of the answer set, then we can add the query and compute the answer set again to obtain the answer.

Moreover, as the query is added to the final ASP program, we can extend the expressiveness of queries by allowing the use of negation in queries. Thus, we extend the syntax of conjunctive queries, which can now be a rule $ans(X_1, \dots, X_k) \leftarrow a_n, \dots, a_n, not\ a_{n+1}, \dots, not\ a_m$. with $k \geq 0$, $n > 0$, $\{a_1, \dots, a_m\}$ is a set of atoms and $X_1, \dots, X_k$ are variables appearing at least in some a_i, $i \in \{1, \dots, m\}$. Note that variables appearing in a_j, $j \in \{n+1, \dots, m\}$ must also appear in a_i, $i \in \{1, \dots, n\}$ in order for the rules to be safe (see *e.g.* [28] for details about safe rules). Using this extension, we are able to express queries like the one in the following example.

Example 4. *"What are the amphorae which are not visible from amphorae a and are of typology "Pithecusse_366" ?"*
$ans(Y) \leftarrow amphorae(a), amphorae(Y), hasTypology(Y, \text{"}Pithecusse\_366\text{"}), not\ visible(a, Y).$

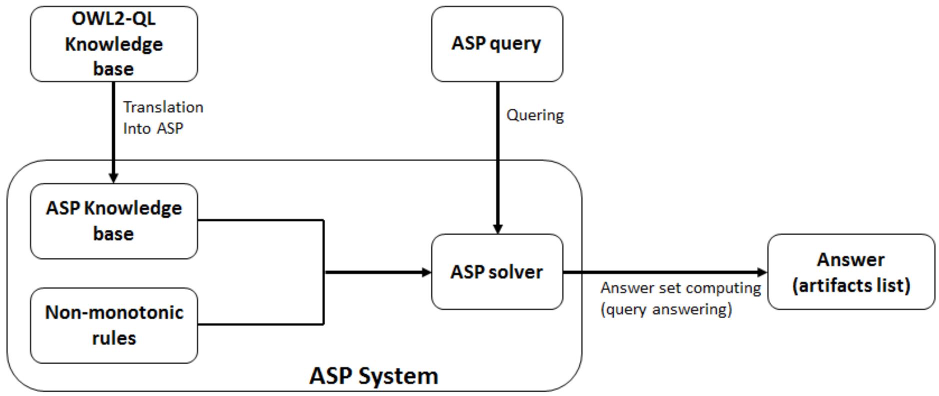

Figure 4: Query answering process for OWL knowledge base and ASP solver.

6 Implementation

Benchmark structure The Xlendi ontological knowledge base is composed of a TBox and an ABox. The TBox defines 69 classes and 124 properties (32 object properties and 92 data type properties). Once it is translated into existential rules, it contains 483 rules of which 60 are integrity constraints and 73 rules with equality in the head. Only one rule contains an existential variable. The translation to ASP only needs to deal with the only existential variable, equality in the head which can be translated into integrity constraints (for this specific ontology) and float numbers which are converted into integers (floats are measurements in meters converted in centimeters and rounded to the centimeter because a better precision is not needed in this benchmark). The ABox is composed of 6210 facts describing 75 amphorae and 55 grinding stones, these objects are described with their position, height, width, orientation, size among others characteristics. The interest of this ontological knowledge base is to be able to query it and to extract data about these objects. We can for example extract data about all the amphorae of a specific typology and compare these amphorae. Moreover, the ABox is composed of 16770 additional facts describing the cardinal relations between the 130 (75 amphorae and 55 grinding stones) artifacts. Once the translation of the TBox and ABox into an ASP program has been performed, the rules expressing the visibility relation have been added, and the grounding of the program has been performed. It is worth mentioning that the program reduces to a set of 50342 definite rules, which indeed describe the only answer set. Thus, all the work is performed at the grounding level, and the solver has virtually no work to do, which explains the excellent results presented hereafter, as this situation is common to all the presented queries.

In order to provide an ASP query answering demonstrator, we opted for a Web solution that marries the Web version of Clingo[7] solver to the 2D Web drawing of Xlendi orthophoto (powered by Raphaël[8]). The Xlendi-ASP Web demonstorator is made available online via URL [9]. This demonstrator provides access to six ASP queries following the different examples presented in Section 5:

- **query 1** What is the name of amphorae with height between 0.4 meter and 0.6 meter ?
- **query 2** Which amphorae are visible from amphora Amphore_A50 towards east ?

[7] https://potassco.org/
[8] http://dmitrybaranovskiy.github.io/raphael
[9] http://www.arpenteur.org/xlendi_asp

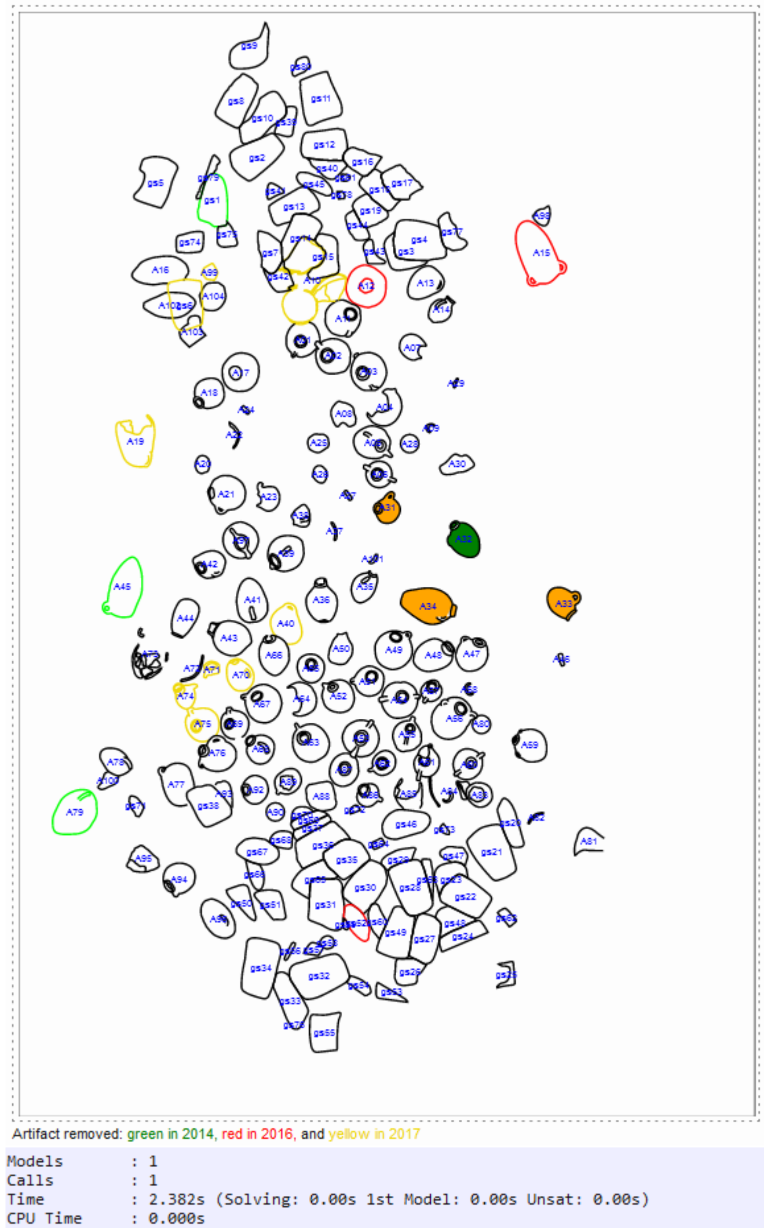

Figure 5: A web demonstration of query answering corresponding to query 3. “Amphore_A32” is colored in green. “Amphore_A34”, “Amphore_A31” and “Amphore_A33” are the answer set and are colored in orange.

- **query 3** Which amphorae are visible from amphora Amphore_A32 ?
- **query 4** Which amphorae are of type *“Ramon-T2111-69”* and are visible from amphora Amphore_A08 ?

- **query 5** Which are the amphorae that are **not** visible from "Amphore_A21", and are of typology *"Pithecusse_366"* ?.

- **query 6** Get all relations of visible artifacts couples in Xlendi.

Table 1 present some information about the size and run time of these queries.

query	# of atoms	# of rules	exec time for WEB demo (s)	exec time for native solver (s)
1	52949	52949	2.792	0.398
2	52908	52908	2.787	0.399
3	52910	52910	2.776	0.397
4	52908	52908	2.784	0.398
5	52913	52913	2.781	0.399
6	50342	50342	2.644	0.378

Table 1: ASP program size and performance measurements for the six example queries.

In the following we will discuss the answer set corresponding to the query 3: $ans(Y) \leftarrow amphorae(\text{“}Amphore\_A32\text{”}),\ amphorae(Y),\ visible(\text{“}Amphore\_A32\text{”}, Y)$. Figure 5 presents a demo of the query 3 where the answer set colored in orange. Since the demo is implemented in a Web program, solving an ASP program with Clingo in a web browser is much slower compared to the desktop version. For example, running query 3 in the Web version will take 2.4 seconds, while it takes less than 0.4 seconds using the desktop version of Clingo.

A visual check of Figure 5 depicts that "Amphore_A32" is surrounded by "Amphore_A34", "Amphore_A31", "Amphore_A33" and "Amphore_A30". However the answer set does not include Amphore_A30 in the query result. This is due to the following three facts:
$nwof(\text{“}Amphore\_A32\text{”}, \text{“}Amphore\_A30\text{”})$, $nwof(\text{“}Amphore\_A32\text{”}, \text{“}Amphore\_A31\text{”})$, and
$nof(\text{“}Amphore\_A31\text{”}, \text{“}Amphore\_A30\text{”})$ Indeed, according to the condition (1) in the rule modelling the notion of visibility in Section 3, "Amphore_A30" cannot be visible from "Amphore_A32". Moreover, it should be noted that spatial relations between Xlendi artifacts are computed according to their barycenter (not according to their shape) which can be confusing during the visual check of some artifacts.

7 Conclusion

The paper proposes an ASP query answering approach for ontology-based knowledge bases equipped with non-monotonic rules within the framework of Cultural Heritage. It focuses on the Xlendi wreck case study. The knowledge base initially represented in OWL2-QL is translated into an equivalent Answer Set Programming (ASP) program and is enriched with a set of non-monotonic ASP rules. The increased expressiveness of non-monotonic rules allows for querying spatial relations like closeness between artifacts, which is not possible with languages such as SPARQL and SQWRL. The proposed approach is implemented and an ASP query answering demonstrator is provided on line, combining a Web version of Clingo solver and a 2D Web drawing of Xlendi orthophoto.

Underwater archaeology investigates a wreck site layer by layer, and the study of a layer leads to its destruction in order to study the underlying one. Since a 2D representation corresponds to a layer, a natural future work is the extension of cardinal relations to 3D spatial relations in order to query a 3D representation of the cargo which is closer to reality.

References

[1] Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. Journal of Automated Reasoning 14(1), 149–180 (1995)

[2] Baader, F., Hollunder, B.: Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. Journal of Automated Reasoning 15(1), 41–68 (1995)

[3] Baget, J., Garreau, F., Mugnier, M., Rocher, S.: Extending acyclicity notions for existential rules. In: ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014). pp. 39–44 (2014)

[4] Baget, J., Leclère, M., Mugnier, M., Rocher, S., Sipieter, C.: Graal: A toolkit for query answering with existential rules. In: Rule Technologies: Foundations, Tools, and Applications - 9th International Symposium, RuleML 2015, Berlin, Germany, August 2-5, 2015, Proceedings. pp. 328–344 (2015)

[5] Baget, J., Leclère, M., Mugnier, M., Salvat, E.: On rules with existential variables: Walking the decidability line. Artif. Intell. 175(9-10), 1620–1654 (2011)

[6] Baget, J., Garcia, L., Garreau, F., Lefèvre, C., Rocher, S., Stéphan, I.: Bringing existential variables in answer set programming and bringing non-monotony in existential rules: two sides of the same coin. Ann. Math. Artif. Intell. 82(1-3), 3–41 (2018)

[7] Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2003)

[8] Bass, G.F., Rosencrantz, D.: Submersibles in underwater search and photogrammetric mapping. In: Underwater archaeology. A nascent discipline, vol. Museums and monuments n. 13, pp. 271–283. UNESCO, Paris (1972)

[9] Ben Ellefi, M., Drap, P., Papini, O., Merad, D., Royer, J., Nawaf, M.M., Nocerino, E., Hyttinen, K., Sourisseau, J., Gambin, T., Castro, F.: Ontology-based web tools for retrieving photogrammetric cultural heritage models. In: ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 422,. pp. 31–38 (2019)

[10] Ben-Ellefi, M., Papini, O., Merad, D., Boï, J., Royer, J., Pasquet, J., Sourisseau, J., Castro, F., Nawaf, M.M., Drap, P.: Cultural heritage resources profiling: Ontology-based approach. In: Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018. pp. 1489–1496 (2018)

[11] Bonatti, P.: Rational closure for all description logics. Artificial Intelligence 274, 197–223 (2019)

[12] Bonatti, P.A., Faella, M., Petrova, I.M., Sauro, L.: A new semantics for overriding in description logics. Artificial Intelligence 222, 1–48 (2015)

[13] Bonatti, P.A., Lutz, C., Wolter, F.: The complexity of circumscription in dls. Journal of Artificial Intelligence Research 35, 717–773 (2009)

[14] Donadello, I., Serafini, L.: Integration of numeric and symbolic information for semantic image interpretation. Intelligenza Artificiale 10(1), 33–47 (2016)

[15] Donini, F.M., Nardi, D., Rosati, R.: Description logics of minimal knowledge and negation as failure. ACM Transactions on Computational Logic (TOCL) 3(2), 177–225 (2002)

[16] Drap, P., Merad, D., Hijazi, B., Gaoua, L., Nawaf, M., Saccone, M., Chemisky, B., Seinturier, J., Sourisseau, J.C., Gambin, T., Castro, F.: Underwater photogrammetry and object modeling: A case study of xlendi wreck in malta. Sensors 15(12), 29802 (2015), http://www.mdpi.com/1424-8220/15/12/29802

[17] Drap, P., Papini, O., Sourisseau, J., Gambin, T.: Ontology-based photogrammetric survey in underwater archaeology. In: The Semantic Web: ESWC 2017 Satellite Events - ESWC 2017 Satellite Events, Portorož, Slovenia, May 28 - June 1, 2017, Revised Selected Papers. pp. 3–6 (2017)

[18] Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. Artif. Intell. 172(12-13), 1495–1539 (2008)

[19] Garlik, S., Seaborne, A.: Sparql 1.1 query language. In: W3C: Available at https://www.w3.org/TR/sparql11-query/ (2013)

[20] Gebser, M., Kaufmann, B., Schaub, T.: Conflict-driven answer set solving: From theory to practice. Art. Intell. 187, 52–89 (2012)

[21] Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proc. of ICLP'88. pp. 1070–1080 (1988)

[22] Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: A non-monotonic description logic for reasoning about typicality. Artificial Intelligence 195, 165–202 (2013)

[23] Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: Semantic characterization of rational closure: From propositional logic to description logics. Artificial Intelligence 226, 1–33 (2015)

[24] Hiebel, G., Doerr, M., Hanke, K., Masur, A.: How to put archaeological geometric data into context? representing mining history research with cidoc crm and extensions. International Journal of Heritage in the Digital Era 3(3), 557–578 (2014)

[25] Horrocks, I., Patel-Scneider, P., Boley, H., Tabet, S., Grosof, B., Dean, M.: Swrl: A semantic web rule language combining owl and ruleml. In: W3C: Available at https://www.w3.org/Submission/SWRL/ (2004)

[26] Jezek, P., Moucek, R.: Semantic framework for mapping object-oriented model to semantic web languages. Frontiers in Neuroinformatics 9(3) (2015)

[27] Knorr, M., Alferes, J.J., Hitzler, P.: Local closed world reasoning with description logics under the well-founded semantics. Artif. Intell. 175(9-10), 1528–1554 (2011)

[28] Lefèvre, C., Béatrix, C., Stéphan, I., Garcia, L.: Asperix, a first order forward chaining approach for answer set computing. Theory and Practice of Logic Programming (2015)

[29] Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV system for knowledge representation and reasoning. ACM Trans. Comput. Log. 7(3), 499–562 (2006)

[30] Lopes, C., Knorr, M., Leite, J.: Nohr: Integrating XSB prolog with the OWL 2 profiles and beyond. In: Logic Programming and Nonmonotonic Reasoning - 14th International Conference, LPNMR 2017, Espoo, Finland, July 3-6, 2017, Proceedings. pp. 236–249 (2017)

[31] Motik, B., Rosati, R.: Reconciling description logics and rules. J. ACM 57(5), 30:1–30:62 (2010)

[32] Niemelä, I.: Logic programs with stable model semantics as a constraint programming paradigm. AMAI 25(3-4), 241–273 (1999)

[33] Pollio, J.: Underwater Mapping with Photography and SONAR. Photogramm Eng 37(9), 955–968 (1971), http://www.scopus.com/inward/record.url?eid=2-s2.0-0015113096&partnerID=40&md5=579037c0828eeaa2af59031aa33088ea

[34] Rosencrantz, D.M.: UNDERWATER PHOTOGRAPHY SYSTEMS. Photogramm Eng 37(9), 969–974 (1971), http://www.scopus.com/inward/record.url?eid=2-s2.0-0015113099&partnerID=40&md5=ec6aaac4230ff8982c523d458bfc2237

[35] Schaub, T.: Here's the beef: Answer set programming ! In: Proc. of ICLP'08. pp. 93–98 (2008)

[36] Stevenson, G., Dobson, S.: Sapphire: Generating Java Runtime Artefacts from OWL Ontologies, pp. 425–436. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

[37] Straccia, U.: Default inheritance reasoning in hybrid kl-one-style logics. In: IJCAI. vol. 93, pp. 676–681 (1993)