Integrating OWL Ontologies and Rules: A Case Study on Landslide Detection

Rosa Aguilar\textsuperscript{1,2} and Edna Ruckhaus\textsuperscript{3}

\textsuperscript{1} Fundación Instituto de Ingeniería, Caracas, Venezuela
rosaa@fii.org
\textsuperscript{2} Universidad Simón Bolívar, Caracas, Venezuela
raguilar@ldc.usb.ve
\textsuperscript{3} Universidad Simón Bolívar, Caracas, Venezuela
ruckhaus@ldc.usb.ve

Abstract. This work is focused on a subset of the geospatial domain that concerns landslide detection, which was represented as a hybrid OWL-log ontology. OWL-log is an existing implementation of the \textit{AL}\textsuperscript{-}log hybrid knowledge representation system where the Description Logics component is extended to the Web Ontology Language OWL DL. Knowledge in hybrid OWL-log ontologies is distributed in its two components: OWL and \textit{Datalog} subsystems. However, a portion of the hybrid ontology knowledge base may be represented in both. We denote this portion as Knowledge Base Intersection (KBI). We studied different KBI representations in the two subsystems, and their relationship to query performance. A query benchmark was defined to support landslide hazard areas detection. Combinations of three factors were studied: KBI representation, query-answering strategy (pre-compilation or dynamic) and dataset size. The results show that KBI representation affects query execution time, specially when knowledge base size increases.

1 Introduction

In order to discover, select and share the large amount of information and services related to geospatial data available on the Web, relevant sources must be described. Ontologies provide a shared or common understanding of this domain that can be communicated among people and application systems. The W3C Consortium has established the Ontology Web Language (OWL), a Description Logics (DL) language as the standard for ontology definition. However, complex domains like the geospatial domain cannot be represented completely with OWL. Geospatial data are characterized by geometry, topology and spatial relationships that define the requirements of their representation. In this work we focus on the subset of the geospatial domain that concerns landslide detection. Several works [9, 15, 3], present application domains where OWL is not expressive enough and where more complex rules are necessary to represent the concepts and restrictions in the specific domain.

Complex interactions in the landslide detection domain are those concerning landslide risks: for instance, a certain area has a high landslide risk because of
its strong slope, scarce vegetation and incoherent rock lithology. The concepts relevant to the landslide detection domain are represented as a hybrid Owl-Rules System supported by the OWL-log ontology System[13].

In general, a hybrid knowledge representation system is composed by two or more knowledge representation subsystems that process fragments of the knowledge base using different reasoning mechanisms under a unique framework. They increase the inferencing power of their component languages, taking advantage of their capabilities w.r.t expressiveness and reasoning efficiency. [12]. \textit{AL-Log} [1] is a hybrid system that involves a simple combination: it combines the DL language \textit{ALC} with positive \textit{Datalog}. The interaction between the subsystems is done through the specification of constraints (DL classes) in the \textit{Datalog} rules which "type" variables appearing elsewhere in the body, so the DL theory acts as an expressive language to structure knowledge for \textit{Datalog} predicates.

\textit{OWL-log} is an implementation of \textit{AL-log} that restricts the \textit{Datalog} atoms to unary or binary where the DL component is extended to the Web Ontology Language OWL DL [10]. \textit{OWL-log} can be seen as a subset of the Semantic Web Rules Language (SWRL), because a constrained \textit{OWL-log} clause corresponds to a SWRL axiom in which only DL Class and Datatype predicates are allowed as constraints in the antecedent of the rules. Two different query-answering strategies have been developed: Dynamic and Precompilation. The Dynamic strategy is based on the notion of constrained SLD-derivation and constrained SLD-refutation [1]. The key idea of the Precompilation implementation is to pre-process all of the DL atoms that appear in the \textit{Datalog} rules, and include them as facts in the \textit{Datalog} subsystem.

Knowledge in hybrid \textit{OWL-log} ontologies is distributed in its two components: the OWL subsystem and the \textit{Datalog} subsystem. However, a portion of the hybrid ontology knowledge base may be represented in any of the subsystems. We denote this portion as the Knowledge Base Intersection (KBI).

A concept in an \textit{OWL-log} ontology can be related to other concepts using axioms of one of its component’s ontology languages, OWL or Datalog. In this work we determine if the concept together with its related concepts belongs to the KBI. For instance, in our case study some of the related concepts to the DL concept \textit{vegetation} are the following:

\begin{verbatim}
vegetation ⊑ geofeature
forest ⊑ vegetation
⊤ ⊑ \text{Visintervened}.vegetation
vegetation_dense ≡ \exists hasdensity.{dense}
vegetation_dense ⊑ vegetation
⊤ ⊑ \forall hasdensity'.vegetation
\end{verbatim}

In order to determine if \textit{vegetation} can be represented in \textit{Datalog}, we need to analyze the axioms that relate it to other concepts, e.g., the definition of \textit{vegetation_dense} and the domain restriction of the property \textit{isintervened}. In case that the concept \textit{vegetation} belongs to the KBI, and we decide to represent it in the \textit{Datalog} component, we need to add facts in \textit{Datalog} that correspond to
its individuals in DL, and intensional predicates (rules) that correspond to the related concepts axioms in DL.

The goal of the work presented in this paper is to study the relationship between the KBI representation and the efficiency of query evaluation in the OWL-log System, for a case study on landslide detection. The contributions of this work include the representation of the landslide detection domain as an OWL-log ontology, the analysis of the domain concepts in order to determine if they belong to the KBI and an experimental study on the relationship between knowledge representation and query evaluation. Combinations of three factors were studied: KBI representation, query-answering strategy (pre-compilation or dynamic) and dataset size.

This paper is comprised by 7 sections. Section 2 presents the OWL-log System. Next, Section 3 describes the related work. Section 4 presents the methodology used to analyze the KBI. Section 5 presents the case study ontology and the KBI analysis in this domain. Section 6 describes the experimental study, and Section 7 gives the conclusions and future work.

2 OWL-log

In this section we will first present the basis of OWL-log, ALC-log, and then the description of the OWL-log System.

2.1 ALC-log

ALC-log [1] is a hybrid knowledge representation system that consists of two subsystems: (1) a structural subsystem based on the description logic ALC, and (2) a Datalog subsystem based on the deductive database language Datalog without negation. The interaction between the subsystems is done through the specification of constraints in the Datalog rules. These constraints are expressed in terms of ALC classes.

The structural subsystem $\Sigma$ is a pair: $\Sigma = (T, A)$ where $T$ is a set of inclusion axioms and $A$ is a set of assertions on individuals. The Datalog subsystem allows to express Datalog knowledge in terms of Constrained Datalog rules: $\alpha_0 :- \alpha_1, \ldots, \alpha_n & \beta_1, \ldots, \beta_m$. Each $\alpha_i$, $1 \leq i \leq n$, is an atom $r(t_1, \ldots, t_p)$, where $r$ is a Datalog predicate and each $t_k$, $1 \leq k \leq p$ is a term, i.e. either a constant or a variable. Each $\beta_j$, $1 \leq j \leq m$, is a constraint of the form $C(s)$ where $s$ is either a constant or a variable that appears as a Datalog term, and $C$ is an ALC class. $\&$ is a delimiter that separates the body of the Datalog atoms from the constraints.

$^1$ ALC is a DL language that includes classes and roles, class inclusion axioms, assertions about individuals that belong to classes and play roles, and the following constructed and restrictions: Universal and Empty classes, complement, union, intersection, universal restriction of a class by a role, and existential restriction of a class by a role.

Rosa Aguilar and Edna Ruckhaus
Rosa Aguilar and Edna Ruckhaus

A query $Q$ to a knowledge base $K$ is of the form: $q_1, \ldots, q_n \& \beta_1, \ldots, \beta_m$ where each $q_i$, $1 \leq i \leq n$ is an atom, and each $\beta_j$, $1 \leq j \leq m$ is a constraint. The method used for query answering is based on the notion of constrained SLD-derivation where a derivation may terminate with a last query of the form $\& \beta_1, \ldots, \beta_m$ which is called a constrained empty clause.

2.2 The $\mathcal{OWL}$-log System

The hybrid system $\mathcal{OWL}$-log combines features of both SWRL and $\mathcal{AL}$-log in the following way:

1. $\mathcal{OWL}$-log is an $\mathcal{AL}$-log implementation where the structural component is an OWL DL ontology based in the $\mathcal{SHOIN}(D)$ language.
2. Predicates in $\mathcal{OWL}$-log rules are limited to OWL classes and properties.
3. In $\mathcal{OWL}$-log, each variable occurring in a DL predicate (constraint), must also occur in a Datalog predicate.
4. As predicates in $\mathcal{OWL}$-log rules are OWL classes and properties, they have a maximum arity of 2.

In order to describe $\mathcal{OWL}$-log, we need to define the notions of atomic concept, DL-atom and Datalog atom:

**Definition 1** An atomic concept in an OWL ontology is a class, datatype, object property or datatype property that is not being used in any of the axioms in the ontology.

**Definition 2** Let $\Sigma$ be an OWL-DL ontology and $P$ a Datalog Program. A DL atom in a rule $r$ in $P$ is a predicate that refers to a non-atomic Class or Datatype in $\Sigma$.

**Definition 3** Let $\Sigma$ be an OWL-DL ontology and $P$ a Datalog Program. A Datalog atom in a rule $r$ in $P$ is a predicate that refers to an atomic concept in $\Sigma$.

Similarly to $\mathcal{AL}$-log, we have constrained $\mathcal{OWL}$-log rules where the contraints are DL atoms. We will now define the notion of weak DL-safety:

**Definition 4** Let $\Sigma$ be an OWL-DL ontology and $P$ a Datalog Program consisting of constrained rules. A rule $r$ in $P$ is weakly DL-safe if each variable in a DL atom occurs in a Datalog atom in the rule.

An $\mathcal{OWL}$-log KB can then be formally defined as:

---

2 $\mathcal{SHOIN}(D)$ is a DL language that includes all of the concepts, axioms and restrictions in $\mathcal{ALC}$, and also role inclusion axioms, transitive and inverse roles, nominals, cardinality restrictions and Datatype properties.
Definition 5 Let $K = \langle \Sigma, P \rangle$ be a combined KB where $\Sigma$ is an OWL-DL Knowledge Base and $P$ is a Datalog program consisting of weakly DL-safe OWL-log constrained rules. We say that $K$ is an OWL-log KB if rules in $P$ are of the form:

$h :- p_1, \ldots, p_n \& c_1, \ldots, c_m$ where $h$ and each $p_i$, $1 \leq i \leq n$ are Datalog atoms, and each $c_j$, $1 \leq j \leq m$ is a DL atom.

An example of an OWL-log rule that models a policy on printing documents is the following:

\[ \text{printOnePage}(X) :- \text{containsCartridge}(X), \text{availablePaper}(X,Y), Y > 1 \& \text{printerHP}(X). \]

There are two approaches for defining OWL-log Semantics: (1) The SWRL approach, that is $\text{SHOIN}(D)$ interpretations can be extended to deal with variable bindings, and (2) The AL-log approach, that is, consider a $\text{SHOIN}(D)$ interpretation for the OWL KB and the DL atoms in the rules, and a Herbrand interpretation for the OWL-log rules without the DL atoms.

Following the AL-log approach we have that an interpretation $J = \langle I, H \rangle$ for $K = \langle \Sigma, P \rangle$ is the union of a $\text{SHOIN}(D)$ interpretation $I$ for $\Sigma$ and a Herbrand Interpretation $H$ for $P_D$. $P_D$ is the set of OWL-log constrained rules of $P$, where the OWL-log DL atoms of each rule are deleted.

The OWL-log hybrid reasoner computes answers to queries based on the specification of both subsystems. It implements two different query-answering strategies: Dynamic and Precompilation.

- Dynamic. This method is based on the notion of constrained SLD-derivation where a derivation may terminate with a constrained empty clause $\& \beta_1, \ldots, \beta_m$, and each $\beta_i$ is of the form $C(t)$ or $D(v)$ where $t$ is an object term and $v$ is a datatype term. In order to answer a query we have to collect all its derivations, and every model of the knowledge base $K$ should satisfy the constraints of at least one derivation.
- Precompilation. The key idea of this implementation is to pre-process all of the DL atoms that appear in the Datalog rules, and include them as facts in the Datalog subsystem. Once the pre-processing is done, queries can be answered by the Datalog component using any of the known techniques for Datalog query evaluation.

3 Related Work

The integration of the DL and Datalog formalisms has been the focus of several research initiatives that can be grouped into two main approaches: (1) intersection of DL and Logic Programming (LP) [4, 5], and (2) hybrid systems which consider a KB with two subsystems, the DL and the Datalog subsystem. Among the latter we have AL-log [1], CARIN [7], DL+log [12], Datalog^DL [11] and HEX-programs [2]. Unlike AL-log, in CARIN [7], DL atoms can be unary and binary but may only appear in the antecedent of rules; to gain decidability, CARIN requires Role-safe rules. DL+log is a general framework that integrates
DL with disjunctive Datalog; it realizes a tighter form of interaction between DL-KBs and rules, through the weak safeness condition in Datalog rules. $Datalog^{DL}$ [11] is a family of hybrid languages where the Datalog rules are constrained by DL atoms ranging from $ALC$ up to $SHIQ$. $Datalog^{DL}$ is similar to CARIN with respect to the inclusion of classes and properties in the Datalog rules, but it is less restricted w.r.t. rule safeness conditions (properties should be independent).

HEX-programs are non-monotonic logic programs featuring both higher-order atoms as well as external atoms. Through external atoms, HEX-programs can deal with external knowledge and reasoners of various natures, such as RDF datasets or description-logic bases [2].

Several practical applications of hybrid OWL-rules systems have been developed. Mäss [9] presents semantic rules for the control of data collection for monitoring landslides. He concludes that OWL is not enough for the representation of complex rules that refer to several classes of objects, its attributes and relationships. Subsequently, he proposes the Semantic Web Rules Language (SWRL) for the definition of topological and semantic restrictions that assure the quality of the collected data.

In [3], a hybrid ontology system for the study of brain anatomy is presented. This system uses the combination of two languages: OWL for the structural component, and SRWL for rules. In this paper it is shown that OWL was not enough for expressing dependencies between properties in the ontology, and between the ontology and other n-ary predicates in the domain.

These systems and case studies focus on the problems of reasoning in OWL-rules systems and their expressive capabilities. However, there are no explicit references to the problem of modeling and distributing the knowledge base in its subsystems, nor of its effect on query evaluation.

4 Knowledge Base Intersection (KBI)

The analysis of the KBI in OWL-log can be considered an extension of [14] where a correspondence between DL and Logic Programming is established.

4.1 OWL-DL to Datalog

Each OWL-DL constructor was analyzed. Table 1 illustrates the $allValuesFrom$ constructor. When this constructor is used on the right-hand side of an inclusion axiom it can be represented as a Datalog Rule whereas when the constructor is used on the left-hand side of an inclusion axiom, it can not be represented as a Datalog rule because the first conjunct is not a Horn clause.

Similarly, defined concepts in OWL-DL may be constructed in terms of other primitive or defined concepts. Thus, to determine if an OWL-DL concept corresponds to a Datalog concept, all of its related concepts have to be analyzed as well. Additionally, all of its ABox assertions will correspond to facts in Datalog. For instance, given a class inclusion relationship in OWL-DL:

\[ A \sqsubseteq B_1 \sqcap \ldots \sqcap B_n, \]
### Table 1. Analysis allValuesFrom Constructor

<table>
<thead>
<tr>
<th>DL Semantics</th>
<th>Right Side</th>
<th>Left Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C \sqsubseteq \forall P \cdot D$</td>
<td>$\forall P \cdot D \sqsubseteq C$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LPO Expression</th>
<th>Datalog Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall x, y ((C(x) \land P(x, y)) \rightarrow D(y))$</td>
<td>$h(Y) :- c(X), p(X, Y)$</td>
</tr>
<tr>
<td>$\forall x \exists y ((P(x, y) \lor C(x)) \land (D(y) \rightarrow C(x))$</td>
<td>Cannot be expressed in Datalog</td>
</tr>
</tbody>
</table>

it may be expressed in Datalog as follows:

$b_1(X)$ :- $a(X)$.

... 

$b_n(X)$ :- $a(X)$

In turn, each $B_i$ should have a correspondence in Datalog because in OWL-log rules, DL atoms can only occur in the body of the rule as constraints. Because $A$ now belongs to the body of Datalog rules, it can either be represented in OWL-DL and referenced from the Datalog component as a DL atom (constraint), or translated to a Datalog atom.

#### 4.2 Datalog to OWL-DL

Datalog to OWL-DL translation involves the analysis of extensional and intensional predicates. If $p$ is an extensional unary predicate that is part of a body of a rule:

$h$ :- $p_1, p_2, ..., p_n, p \land C_1, ..., C_k$.

$p$ may be translated to a class $P$ in OWL, with its corresponding ABox assertions. The rule will be modified with $p$ as a constraint as follows:

$h$ :- $p_1, p_2, ..., p_n \land C_1, ..., C_k, P$.

If $p$ is an intensional unary predicate, its correspondence to an OWL-DL axiom or constructor should be determined. For instance, the predicate $p(X) :- r(X, Y)$ corresponds to the domain restriction in OWL-DL $\top \sqsubseteq \forall R^-.P$, if $r$ can be represented as an OWL-DL property. If $p$ is in the body of another intensional predicate then it should be converted to a DL-Atom (constraint).

In the case of binary intensional predicates, only those that are not in the body of another intensional predicate may be translated to OWL-DL, because DL properties are not allowed as constraints in OWL-log rules. N-ary predicates cannot be translated to OWL-DL.

#### 4.3 KBI Stratas

Concepts in the KBI may be partitioned into subsets which we denote as $KBI_p$ or stratas of the KBI. Stratas have a partial order induced by the stratification of concept definitions. A $KBI_p$ is a set of related concepts that depend on an initial concept $C$, and include all those classes and properties related to $C$ through an axiom or constructor in OWL-DL. All of the concepts in one strata should be represented in the same formalism, i.e. should belong to the same OWL-log component.

Given two stratas $KBI_{p1}$ and $KBI_{p2}$, if $KBI_{p1}$ precedes $KBI_{p2}$ then when $KBI_{p1}$ is represented in Datalog, $KBI_{p2}$ may be represented in OWL or Datalog.
Rosa Aguilar and Edna Ruckhaus

while if $KBI_{p1}$ is represented in OWL, then $KBI_{p2}$ should be represented in OWL. This is because concepts in $KBI_{p2}$ are in the body of the OWL-log rules so we can choose to define them as Datalog atoms or as DL atoms. On the other hand, if the concepts in $KBI_{p1}$ are represented as OWL primitive or defined concepts, the concepts in $KBI_{p2}$ should also be represented in OWL, because in OWL-log there is no interface in the direction from the OWL component to the Datalog component.

5 Case Study on Landslide Detection

The ontology was constructed by extending the methodology presented in [6]. The purpose of this ontology is to represent the concepts and relationships relative to the detection of zones that are threatened by landslides. An initial taxonomy includes only primitive concepts, e.g., hazard, geofeature, intervention. Figure 1 illustrates strata of the first level of the concept taxonomy.

![Fig. 1. Taxonomy Ontology Concepts - Primitive Concepts](image)

The relationships between concepts were analyzed, and properties were established for these relationships, e.g., isintervened that refers to the grade of human intervention in a given geographic area; and hasdensity that refers to the density of any type of vegetation. For each class, a dictionary describes its class and instance attributes, and its properties. Following this, defined concepts were developed in terms of other concepts and properties. Then, Horn Rules were defined with respect to risk levels, susceptibility, zones near communication routes and zones in expansion. Table 2 illustrates the Datalog rule ishazardpronearea.

5.1 Analysis of the KBI

In this section we analyze the geofeature concept originally modeled in OWL, in order to determine if it belongs to the KBI. Table 3 presents some of the axioms that are related to the concept geofeature. Its concept dependency diagram is shown in Figure 2.
Table 2. Datalog rule \textit{ishazardpronearea}

<table>
<thead>
<tr>
<th>Description</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>high landslide hazard areas</td>
<td>\textit{ishazardpronearea(PF, high):-}</td>
</tr>
<tr>
<td>strong slope(PF), incoherent altered lithology(LI), sparse vegetation(VE), strong scarp(EF), hasspatialrelation(PF, LI), hasspatialrelation(PF, VE), hasspatialrelation(PF, EF).</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Some DL Axioms related to the concept \textit{geofeature}

<table>
<thead>
<tr>
<th>Expression</th>
<th>Concepts</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{geofeature \subseteq spatialthing(X) \sqcap geolocation(X)}</td>
<td>\textit{geofeature}</td>
<td>\textit{spatialthing, geolocation}</td>
</tr>
<tr>
<td>\textit{geolocation \subseteq \forall hasgeoreference.georeference}</td>
<td>\textit{geolocation}</td>
<td>\textit{hasgeoreference}</td>
</tr>
<tr>
<td>\textit{spatialthing \subseteq \forall hasspatialdescription.spatialdescription}</td>
<td>\textit{spatialthing}</td>
<td>\textit{hasspatialdescription}</td>
</tr>
<tr>
<td>\textit{geofeature \equiv spatialthing \sqcap geolocation}</td>
<td>\textit{spatialthing}</td>
<td>\textit{geolocation}</td>
</tr>
<tr>
<td>\textit{local \subseteq georeference}</td>
<td>\textit{local}</td>
<td>\textit{georeference}</td>
</tr>
<tr>
<td>\textit{projected \subseteq georeference}</td>
<td>\textit{projected}</td>
<td>\textit{georeference}</td>
</tr>
<tr>
<td>\textit{geofeature \subseteq \forall xcoordinate}</td>
<td>\textit{geofeature}</td>
<td>\textit{xcoordinate}</td>
</tr>
<tr>
<td>\textit{geofeature \subseteq \forall ycoordinate}</td>
<td>\textit{geofeature}</td>
<td>\textit{ycoordinate}</td>
</tr>
</tbody>
</table>

Initially the strata \( KIB_{\textit{geofeature},0} = \{ \textit{geofeature} \} \).

The \textit{geofeature} axiom (Table 3) can be represented in Datalog as follows:

\textit{geofeature}(X):- \textit{spatialthing}(X), \textit{geolocation}(X).

\textit{spatialthing} and \textit{geolocation} are defined as universal restrictions on the right-hand side of an inclusion axiom (Table 3), so they correspond also to the head of Datalog intensional predicates:

\textit{spatialdescription}(Y):- \textit{spatialthing}(X), \textit{hasspatialdescription}(X, Y).

\textit{geolocation}(Y):- \textit{georeference}(X), \textit{hasgeoreference}(X, Y).

Thus, they may be represented in the Datalog component. Then:

\( KBI_{\textit{geofeature},0} = \{ \textit{geofeature}, \textit{spatialthing}, \textit{geolocation} \} \).

Fig. 2. Dependencies of Geofeature
Rosa Aguilar and Edna Ruckhaus

The analysis of spatialthing involves the analysis of the property hasspatialdescription and the class spatialdescription. They both can be represented in Datalog. Now:

\[ KBI_{geofeature,0} = \{ \text{geofeature}, \text{spatialthing}, \text{geolocation}, \text{spatialdescription}, \text{hasspatialdescription} \} \]

Similarly, geolocation and its related concepts are added to the strata:

\[ KBI_{geofeature,0} = \{ \text{geofeature}, \text{spatialthing}, \text{geolocation}, \text{spatialdescription}, \text{hasspatialdescription}, \text{georeference}, \text{hasgeoreference} \} \]

Classes local and projected are subclasses of georeference. The local inclusion axiom may be represented in Datalog as either the rule:

\[ \text{georeference}(G) :- \& \text{local}(G) \]

or the rule:

\[ \text{georeference}(G) :- \text{local}(G) \]

As local may be either a Datalog atom or a DL atom, a new strata is initiated. Similarly, a new strata is initiated for projected:

\[ KBI_{geofeature,1} = \{ \text{local} \} \]

\[ KBI_{geofeature,2} = \{ \text{projected} \} \]

Thus, if \( KBI_{geofeature,0} \) is represented in Datalog, \( KBI_{geofeature,1} \) and \( KBI_{geofeature,2} \) may be represented in OWL and referenced from the OWL-log rule as a DL atom. If local or projected were modeled in Datalog, they would need to be further analyzed.

\textit{geofeature} is the domain of datatype properties coordinateX and coordinateY. These axioms may be expressed in Datalog, so the properties are added to the strata \( KBI_{geofeature,0} \).

\textit{geofeature} participates in several inclusion relationships which conform new stratas. Figure 3 shows the KBI stratas and their precedence relation. Finally, we can assert that the class \textit{geofeature}, together with its related concepts and properties, belongs to the KBI.

Fig. 3. Stratification KBI Geofeature
6 Experimental Study

This study pursues the analysis of the relationship between the representation of the KBI in each component of OWL-logic and the efficiency of query processing. Three factors were taken into consideration: size of the KB, reasoning strategy of OWL-logic, and representation of the KBI. The experimental study is factorial complete with $k = 3$ factors without replications. The design is as follows:

1. Representation of the KBI. KBI in Datalog or KBI in OWL-DL.
2. Reasoning Strategy. Precompilation or Dynamic.

The generation of triples and instances was done by extending the code of the LUBM benchmark generator [8] in order to consider the case study domain. Individuals and triples were randomly generated. The experiments were executed in a Pentium IV, 3GB RAM, Windows XP computer. A benchmark of ten queries was developed aimed to discover zones that may be flooded, landslide hazard zones, and zones that could be subject to urban growing. For instance, the query to detect if a zone may be subject of flooding, will need to discover zones close to an important body of water, so the query will search a class hierarchy, use the spatial distance concept, and access two rules that express complex interactions between classes and properties.

Execution times for the dynamic and precompilation strategies combined with the other two factors, show that q1 through q6 had similar behaviour, and q7 through q9 also. Therefore, results were detailed for three queries: q1, q7 and q10. Table 4 shows the execution time in msg. for the ten queries with the precompilation strategy. q1 is a simple query which covers a class hierarchy but does not need to query complex interactions between concepts. q7 evaluates a Datalog rule, covers complex interactions between concepts and makes use of the spatial distance relationship. q10 is similar in complexity to q7 but it makes use of additional concepts like intersection and contention.
Table 5 shows the results for query q1 in each experiment. The results indicate that for these types of queries, the precompilation strategy produces shorter evaluation times than the dynamic strategy. An ANOVA analysis indicates that the representation of KBI by itself does not affect the execution time, but the interaction of this factor with the dataset size and the evaluation strategy does affect it. Figure 4 shows this interaction and its effect. For small and medium-sized datasets, with the KBI in OWL, there are shorter evaluation times than with the KBI in Datalog. On the other hand, for large datasets, better results are obtained when the KBI is in Datalog. The rules that need to be evaluated in q1 when the KBI is in Datalog are the following:

\[
\begin{align*}
\text{inventory}(X) : & :- \text{landslideinventory}(X), \\
\text{landslideinventory}(X) : & :- \text{eventoccured}(X, \text{landslide}).
\end{align*}
\]

These rules have only one goal in their body; in this case the evaluation process retrieves all the facts of \text{eventoccured}. If the \text{inventory} concept is represented in OWL, and the dynamic strategy is used, the evaluation process consists of querying the KB using the DL reasoner. The DL reasoning algorithm has a higher complexity than the Datalog evaluation algorithm, and the effect of this is more visible in large datasets.

<table>
<thead>
<tr>
<th>Size KB</th>
<th>OWL</th>
<th>Datalog</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dynamic</td>
<td>Precompiled</td>
</tr>
<tr>
<td>Small</td>
<td>2174.4</td>
<td>414.5</td>
</tr>
<tr>
<td>Medium</td>
<td>2664.7</td>
<td>536.9</td>
</tr>
<tr>
<td>Large</td>
<td>3769.8</td>
<td>977.5</td>
</tr>
</tbody>
</table>

q7 has lower execution times when the dynamic strategy is used and the KBI is in OWL. To analyze the effect of each factor, and of their combination, on the execution time, an ANOVA analysis was developed and is presented in Table 6. We can observe that all the factors and interactions are significant.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Degrees of Freedom</th>
<th>Total square</th>
<th>Average square</th>
<th>Value of F</th>
<th>P Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>KBI Representation</td>
<td>1</td>
<td>375195</td>
<td>375195</td>
<td>1633.6</td>
<td>2.24E-03 ***</td>
</tr>
<tr>
<td>Strategy</td>
<td>1</td>
<td>1197659</td>
<td>1197659</td>
<td>5214.5</td>
<td>2.20E-04 ***</td>
</tr>
<tr>
<td>Size</td>
<td>1</td>
<td>1385368</td>
<td>1385368</td>
<td>6031.7</td>
<td>1.65E-04 ***</td>
</tr>
<tr>
<td>KBI Rep.:Strategy</td>
<td>1</td>
<td>417161</td>
<td>417161</td>
<td>1816.3</td>
<td>1.81E-03 ***</td>
</tr>
<tr>
<td>KBI Rep.:Size</td>
<td>1</td>
<td>261159</td>
<td>261159</td>
<td>1137.1</td>
<td>4.61E-03 ***</td>
</tr>
<tr>
<td>Strategy:Size</td>
<td>1</td>
<td>902459</td>
<td>902459</td>
<td>3929.2</td>
<td>3.88E-04 ***</td>
</tr>
<tr>
<td>KBI Rep.:Strategy:Size</td>
<td>1</td>
<td>260068</td>
<td>260068</td>
<td>1132.3</td>
<td>4.65E-03 ***</td>
</tr>
<tr>
<td>Residuals</td>
<td>4</td>
<td>919</td>
<td>230</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The results for q10 can be observed in Figure 4. The results show that shorter execution times are obtained when the KBI is in Datalog, and the answering strategy is Dynamic.
In more complex queries, like q7 and q10, it is advisable to use the dynamic evaluation strategy. The worst results for these queries were obtained when the KBI is in Datalog and the strategy is Precompilation. To evaluate a user query in Datalog, basic facts have to be used to identify derived knowledge. Certain query orderings and Datalog rule orderings may induce a number of inferences that are not relevant to the results; thus, the query execution time is impacted by the order or plan in which the query is evaluated, the order of the goals in the body of the ontology’s Datalog rules and the size of the dataset. OWL-log does not provide yet query and rule reordering techniques.

On the other hand, the dynamic strategy will use the DL reasoner during query evaluation, and its higher complexity will affect the query evaluation time when the size of the OWL-DL component is large. Thus, the better results for complex queries like q7 and q10 were obtained when the dynamic strategy was used but the DL component was not large.

7 Conclusions and Future Work

In this work we established that representation of the KBI into the two components of hybrid OWL-rules ontologies affects query evaluation time. In particular, we worked on an ontology in the domain of landslide detection where its KBI was determined according to KBI, stratas of related concepts that can be represented in both OWL and Datalog components.

The experimental study results were submitted to a descriptive and ANOVA analysis to better understand the effect of each of the three factors considered; representation of the KBI affects query execution time, specially when the ontology size increases. In general, better execution times were obtained when the KBI is in Datalog, and the evaluation strategy is Dynamic. However, it should be noted that the results could be affected by the fact that the experiments
Rosa Aguilar and Edna Ruckhaus

were conducted on synthetic ontologies where concepts and instances are uniformly distributed. In the future we plan on conducting experiments on real life ontologies.

Geographical data widely use spatial relationships. In this work we modeled explicitly the spatial relationships between the different elements, but their symmetry and transitive properties were modeled in Datalog. This implies that we frequently had to infer relationships that had already been inferred in previous queries so we propose to materialize all the spatial relationships. Future work includes improving query execution time with query optimization techniques that include cost-based join-ordering strategies and Magic-Sets rewriting.

References