1. Introduction

In this article we present a rigorous axiomatization of the notion of ontology from computer and information sciences, using terminology from the mathematical discipline of category theory. Specifically, we cast an ontological schema as a Grothendieck site – a category $\mathcal{C}$ equipped with a notion of “covers.” Such a cover serves to classify a type into an exhaustive list of subtypes. Instances of an ontological schema, and collections thereof, form presheaves or sheaves on $\mathcal{C}$.

One purpose of this paper is to provide another link between information science and category theory, by connecting together central aspects of the two fields. Both Grothendieck sites and ontologies are structures which are designed to provide a world of study by specifying the requisite vocabulary and axiomatic rules that describe and govern that world. We make the similarities between these two notions clear by formulating ontological ideas in category-theoretic language.

In information science literature, the notion of ontology is kept deliberately vague. Different types of problems require different types of ontologies, and it may be best not to pin down a precise definition. In this paper we hope to show that Grothendieck sites are at least strongly related to ontologies, and thus give a jumping-off point for those who wish to either use mathematical theorems and language in their study of ontologies or for those who wish to explicate and clarify distinctions between sites and ontologies.

Another purpose of this paper is to define (“geometric”) morphisms of ontologies, by directly importing the definition from topos theory. In so doing, we provide a new way to deal with the problem of ontology integration. Given two ontologies, there may or may not be a morphism between them, but if they ostensibly “should” be related, then one will probably be able to find a morphism between sub-ontologies (ontologies either with fewer types or fewer rules). Having a precise definition of morphism is an important step toward solving the problems of integrating ontologies.

Finally, this paper lays the groundwork for future work in which we compare ontologies to databases. If one has defined a category of ontologies and a category of databases, one can compare these notions using functors.

The structure of the paper is as follows. In Section ??, we discuss the basics of ontologies and give some examples. In Section ?? we define Grothendieck sites, as well as presheaves and sheaves. In Section ?? we explain how to consider an
ontology as a site and a site as an ontology by giving a dictionary relating their respective vocabularies, and by providing examples. Finally, in Section ?? we discuss morphisms of sites and show their applicability to ontology integration problems.

2. Ontologies

3. Grothendieck Sites

We begin by recalling the definition of a site.

**Definition 3.0.1.** A site is a pair \((C, J)\), where \(C\) is a category and \(J\) is a function that assigns to each \(C \in \text{Ob}(C)\), a set \(J(C)\) in which each element \(T \in J(C)\) is itself a set \(T = \{t_\alpha: c_\alpha \to C\}\) of morphisms in \(C\) with target \(C\); these data are required to satisfy axioms which we present after defining some terms. We call the sets \(T \in J(C)\) trait classifications on \(C\) and individual morphisms \(t \in T\) trait classes on \(C\). We require the following:

1. For each trait class \(t: c \to C\) on \(C\) and each map \(D \to C\) in \(C\), the fiber product \(c \times_C D\) exists in \(C\).
2. For any map \(f: D \to C\) in \(C\) and any trait classification \(T \in J(C)\), the pullback \(f^*T = \{g^*(t) | t \in T\}\) is a trait classification on \(D\).
3. If \(T = \{t_\alpha: c_\alpha \to C\}\) is a trait classification on \(C\) and for each \(t_\alpha \in T\) the set \(U_\alpha = \{u_{\alpha,\beta}: b_{\alpha,\beta} \to c_\alpha\}\) is a trait classification on \(c_\alpha\), then the family of composites

\[
\bigcup_{t_\alpha \in T} \left\{ b_{\alpha,\beta} \xrightarrow{t_\alpha \circ u_{\alpha,\beta}} C | u_{\alpha,\beta} \in U_\alpha \right\}
\]

is a trait classification on \(C\).
4. If \(f: B \to C\) is an isomorphism in \(C\) then the singleton set \(\{f\}\) is a trait classification on \(C\).

**Definition 3.0.2.** Let \((C, J)\) denote a site, \(X\) a topological space, and \(\text{Shv}(X)\) the category of sheaves of sets on \(X\). A functor \(F: C \to \text{Shv}(X)\) is called topological if it preserves the finite limits which exist in \(C\) and if, for every trait classification \(T \in J\), the induced map of sheaves

\[
\left( \prod_{(t_\alpha: c_\alpha \to C) \in T} F(c_\alpha) \right) \longrightarrow F(C)
\]

is surjective.

Given a pair of topological functors \(F, G: C \to \text{Shv}(X)\), we will say that a natural transformation \(a: F \to G\) is topological if, for every trait class \(c \to C\), the induced diagram of sheaves

\[
\begin{array}{ccc}
F(c) & \longrightarrow & G(c) \\
\downarrow & & \downarrow \\
F(C) & \longrightarrow & F(C)
\end{array}
\]

is a pullback square in \(\text{Shv}(X)\).
Example 3.0.3. If \( X \) is a point, then \( \text{Shv}(X) = \text{Sets} \) is the category of sets. A functor \( F: \mathcal{C} \to \text{Sets} \) is topological if it preserves finite limits and takes trait classifications to surjections.

I think that if \( X \) is sober, then a functor \( F: \mathcal{C} \to \text{Shv}(X) \) is local if, for each point \( x \in X \), the induced functor \( F_x: \mathcal{C} \to \text{Sets} \) on stalks (given by composition of \( F \) with \( \text{colim}_{x \in U} \) \( - (U) : \text{Shv}(X) \to \text{Sets} \)) is topological.

Definition 3.0.4. Let \( F: \mathcal{C} \to \text{Sets} \) be a topological functor, and let \( D = \{ (C_1, T_1), \ldots, (C_n, T_n) \} \) denote a finite set of pairs \( (C_i, T_i) \) in which \( C_i \in \text{Ob}(\mathcal{C}) \) is an object and \( T_i \in J(C_i) \) is a trait classification on \( C \). The \( D \)-display of \( F \) is

4. Ontologies as sites, and sites as ontologies

5. Morphisms and ontology integration