5-2007

Semantic Interoperability of Geospatial Ontologies: A Model-theoretic Analysis

James A. Farrugia

SEMANTIC INTEROPERABILITY OF GEOSPATIAL ONTOLOGIES:

A MODEL-THEORETIC ANALYSIS

By

James A. Farrugia

B.A. East Stroudsburg University, 1986

M.S. The University of North Carolina at Chapel Hill, 1994

A THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

(in Spatial Information Science and Engineering)

The Graduate School

The University of Maine

May, 2007

Advisory Committee:

M. Kate Beard-Tisdale, Professor of Spatial Information Science and Engineering, Advisor

Peggy Agouris, Professor of Remote Sensing, George Mason University

Michael Worboys, Professor of Spatial Information Science and Engineering

Robert Franzosa, Professor of Mathematics

Peter F. Patel-Schneider, Bell Labs Research

SEMANTIC INTEROPERABILITY OF GEOSPATIAL ONTOLOGIES: A MODEL-THEORETIC ANALYSIS

By James A. Farrugia

Thesis Advisor: Dr. M. Kate Beard-Tisdale

An Abstract of the Thesis Presented
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy
(in Spatial Information Science and Engineering)
May, 2007

People sometimes misunderstand each other, even when they use the same language to communicate. Often these misunderstandings happen when people use the same words to mean different things, in effect disagreeing about meanings. This thesis investigates such disagreements about meaning, considering them to be issues of semantic interoperability.

This thesis explores semantic interoperability via a particular formal framework used to specify people's conceptualizations of a given domain. This framework is called an 'ontology,' which is a collection of data and axioms written in a logical language equipped with a model-theoretic semantics. The domain under consideration is the geospatial domain.

Specifically, this thesis investigates to what extent two geospatial ontologies are semantically interoperable when they 'agree' on the meanings of certain basic terms and statements, but 'disagree' on others. This thesis defines five levels of semantic interoperability that can exist between two ontologies. Each of these levels is, in turn, defined in terms of six 'compatibility conditions,' which precisely describe how the results of queries to one ontology are compatible with the results of queries to another ontology.

Using certain assumptions of finiteness, the semantics of each ontology is captured by a finite number of models, each of which is also finite. The set of all models of a given ontology is called its model class. The five levels of semantic interoperability are proven to correspond exactly to five particular relationships between the model classes of the ontologies.

The exact level of semantic interoperability between ontologies can in some cases be computed; in other cases a heuristic can be used to narrow the possible levels of semantic interoperability.

The main results are: (1) definitions of five levels of semantic interoperability based on six compatibility conditions; (2) proofs of the correspondence between levels of semantic interoperability and the model-class relation between two ontologies; and (3) a method for computing, given certain assumptions of finiteness, the exact level of semantic interoperability between two ontologies.

These results define precisely, in terms of models and queries, the often poorly defined notion of semantic interoperability, thus providing a touchstone for clear definitions of semantic interoperability elsewhere.

LIBRARY RIGHTS STATEMENT

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at The University of Maine, I agree that the Library shall make it freely available for inspection. I further agree that permission for “fair use” copying of this thesis for scholarly purposes may be granted by the Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature:

Date:

ACKNOWLEDGMENTS

I gratefully acknowledge the help of all those who helped. This work was partially supported by grants from the National Geospatial-Intelligence Agency (grant numbers NMA-202-97-1-102 and NMA401-02-1-2009, PI: M. Egenhofer) and by a University Graduate Research Assistantship from the University of Maine.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

Practically everywhere software is used, it interacts with other software. For instance, one company's Web browser might run on another company's operating system, and it may connect to a third company's Web server. A meta-search engine on the Internet might exchange data with multiple search engines in order to answer a user's query. Even seemingly standalone programs must interact with the operating systems on which they run. Much of the time this interaction goes smoothly.

Sometimes, however, problems occur. For instance, a document written using a particular word processor on one operating system may, for no apparent reason, change its formatting when it is opened with the "same" word processor running on a different operating system. This problem reveals a certain lack of interoperability between word processors: the formatting of the document is not preserved between the "same" word processor on different operating systems.

Interoperability, in the field of information science, refers to "the ability of various systems to interact with each other no matter the hardware or software being used" Taylor (2004).

When two information systems fail to interoperate, one would like to know why. What is it about the systems and their interactions that causes the problem? A satisfactory explanation would take into account both the nature of the systems themselves and the relevant principles governing their interactions.

For instance, in the above example concerning word processors, one explanation for the inconsistency of formatting might be that one word processor does not have all the fonts that the other one has. This fundamental limitation on the availability of fonts imposes certain constraints on the interoperability of the two systems. These constraints are built into the nature of the each system, and they govern any interaction that deals with conversion of fonts, though they may not become problematic in all these interactions.

In this example, the problem is easily identifiable: the document has one formatting on one operating system (OS), but another formatting on a different OS, even though the 'same' word processing software is being used. One can also clearly describe the desired kind of interoperability: each system (OS and word-processing program) should work with the other so that any document passed between them preserves its formatting. And, assuming that the formatting problem is due to one system's lack of certain fonts, the interoperability problem could be solved by giving both systems identical sets of fonts.

In other cases, even when it is clear that an interoperability problem exists, it may not be easy to *identify* the problem clearly, or to *specify* precisely the desired kind of interoperability, or to *assess* to what extent two systems are interoperable. Even when these challenges can be met, however, it may not be possible to *obtain* the desired interoperability. Nevertheless, we can increase our understanding of the issues involved by exposing the fundamental principles that govern the particular ways that the two systems could, potentially, interoperate.

1.1 Example Showing a Lack of Semantic Interoperability

Figure 1.1: Ball in pool. Pool in yard. Therefore, ball in yard?

Suppose John and Mary are looking in their back yard during the summer, and they observe the ball, the pool, and the yard. (That is, they see these objects and agree on the real-world referents of the terms 'the ball,' 'the pool,' and 'the yard.') And suppose that they both agree that 'The ball is in the pool' and 'The pool is in the yard.'

Suppose that John claims that the statement 'The ball is in the yard' follows from these agreements about meanings, whereas Mary claims that it does not. Specifically, Mary maintains that simply because the ball is in the pool and the pool is in the yard, it does not follow that the ball is therefore in the yard.

This example shows that although John and Mary share a common understanding of some basic aspects of their world, this understanding does not extend to other areas. In particular, it does not extend to what these agreed-upon facts *mean*, where, as the example shows, 'meaning' has to do with logical consequence. One can say that such a difference in understanding reveals a certain lack of *semantic interoperability* between John and Mary. This lack of semantic interoperability would reveal itself in any situation that required an answer to the question, 'Is the ball in the yard?', to which John would answer unequivocally 'Yes,' but Mary would

not. This example thus illustrates the essential issue of semantic interoperability treated in this thesis: when the meanings implicit in different conceptualizations of a common scenario yield different answers to specific questions, to what extent are the conceptualizations semantically interoperable?

Consider, in the case of John and Mary, how one could identify the problem of interoperability, specify the desired kind of interoperability, and assess the extent to which John and Mary are interoperating semantically. The problem can be identified as follows: John and Mary disagree about certain fundamental meanings of their scenario, even though they use the same terms to communicate and they agree on other basic meanings associated with these terms. The desired semantic interoperability could be specified as: John and Mary agree on all meanings related their scenario that. And a rough assessment of the extent to which John and Mary are interoperating semantically could be "somewhat, but not completely."

To this point, the foregoing analysis has

- established that problems of semantic interoperability can exist even when people use the same language to describe a scenario;
- identified a particular lack of semantic interoperability as a disagreement about logical consequence;
- specified one possible kind of desired semantic interoperability; and
- assessed very roughly the extent of semantic interoperability between John and Jane.

In particular, the analysis has not yet dealt with the fundamental constraints in this scenario that govern the semantic interactions between John and Mary. Those constraints are discussed

in Section 1.4. But first, the next two sections provide some overall context for the thesis and discuss the questions that motivated it.

1.2 Context

The context for the research in this thesis is recent work in several related areas: ontologies, Geographic Information Science (GIScience), and the Semantic Web.

1.2.1 Ontologies

The word ‘ontology’ has acquired a distinctive meaning in the last decade or so, especially in the research literature in computer and information science (Guarino, 1998; Smith and Welty, 2001; McGuinness, 2002; Taylor, 2004; Obrst, 2003), where, rather than referring to the philosophical discipline that studies the categories of things that exist (Casati et al., 1998; Sowa, 2000), ‘ontology’ has come to refer to a certain type of written artifact that describes a particular conceptual domain, usually with the intent that it be processable by computers (Guarino, 1998).

1.2.2 GIScience

The University Consortium of Geographic Information Science (UCGIS) has recently identified ontologies as one focus in its research agenda (Egenhofer, 2004). Ontologies have been seen as a tool that might enable the interoperability of Geographic Information Systems (GISs) (Fonseca et al., 2002; Agarwal, 2005). The main purpose of using ontologies in GIScience

is “to define a common vocabulary that will allow inter-operability and minimize any problems with data integration, both between different systems and between users and systems” (Agarwal, 2005, pg. 508). Additionally, government agencies responsible for geospatial data are exploring and developing ontologies (e.g., USGS and the National Geospatial Intelligence Agency (NGA) in the US, and the UK Ordnance Survey (UKOS) in the UK). Issues of semantic interoperability have also been of recent concern to several researchers in GIScience, e.g., Egenhofer (1999); Kuhn (2005a); Hobbs (2006); Agarwal (2005).

Additionally, at least one recent proposal (Egenhofer, 2003) attempts to bridge the research interests of GIScience in ontologies with those of the community of researchers involved in what is becoming known as the Semantic Web (Berners-Lee et al., 2001).

1.2.3 The Semantic Web

‘The Semantic Web’ is a phrase that, unsurprisingly, means different things to different people. A well-known article from 2001 describes it as an extension of the then current Web, “in which information is given well-defined meaning, better enabling computers and people to work in cooperation” (Berners-Lee et al., 2001, pg. 35). A follow-up article has acknowledged the need “for shared semantics and a web of data and information derived from it” (Shaboldt et al., 2006, pg. 96). Both these articles acknowledge the importance of using ontologies on the Semantic Web, with the latter article declaring flatly: “In the past five years, the argument in favor of using ontologies has been won...” (Shaboldt et al., 2006, pg. 96).

In an effort to endow Web markup languages with unambiguous semantics, so that computers can draw valid inferences about tagged Web resources, the World Wide Web Consortium (W3C) has recently developed the markup languages RDF (Klyne and Carroll, 2004) and

OWL (Patel-Schneider et al., 2004). Each of these languages has a model theory that supplies its semantics (Hayes, 2004; Patel-Schneider et al., 2004), and each potentially supports pieces of the vision of the Semantic Web (Berners-Lee et al., 2001; Shaboldt et al., 2006).

A key part of that vision is the expectation that ontologies will be able to "interoperate semantically" without the need for humans in the loop. One early definition of semantic interoperability on the Web is given in Heflin and Hendler (2000), where the authors say: "To achieve semantic interoperability, systems must be able to exchange data in such a way that the precise meaning of the data is readily accessible and the data itself can be translated by any system into a form that it understands" Heflin and Hendler (2000, pg. 111). This approach describes, somewhat generally, semantic interoperability in terms of some kind of translation of meanings. Such a 'translation approach' to semantic interoperability, though interesting and potentially useful, is not the approach taken in this thesis.

A different and more recent view of the Semantic Web might be summarized as "the meaning of a Web resource or term is whatever a collection of people say it is." Sites that let users tag resources with free-form vocabulary (e.g., Amazon.com and librarything.com) help to create a linked network of human-generated and human-understandable meanings that is amenable to certain kinds of machine processing, though not to the kind of machine-understandable processing envisioned in Berners-Lee et al. (2001), for instance. There are good arguments supporting the 'human-understandable' approach to semantics on the Web. However, such arguments are not considered here, since this thesis deals with machine 'understanding' of semantics, not people's understandings of what particular tags mean to them.

1.2.4 This Thesis in Context

Given these related research activities in ontologies, GIScience, and the Semantic Web, the time is ripe for foundational research into the semantic interoperability of geospatial ontologies. This thesis defines a *geospatial ontology* to be a collection of data and axioms (about a geospatial domain) written in a logical language equipped with a model-theoretic semantics. This definition of an ontology, though perhaps not the first definition that most researchers in GIScience would think of — see, however, e.g., Casati et al. (1998); Smith (1996); Cohn and Varzi (2003) for noteworthy exceptions — is, nonetheless, well established in the computer-science and Semantic-Web research communities. Sowa (2000), for instance, calls this kind of ontology an axiomatic ontology. Such an ontology is also, depending on the complexity of the relationships it seeks to specify, amenable to specification in Web markup languages like RDF and the Web Ontology Language OWL, but not in a language like GML (Cox et al., 2003), since RDF and OWL have an associated model theory, whereas GML does not.

The research in this thesis seeks to fill a particular hole in current research and practice in GIScience (and more broadly in information science): the lack of a clear operational definition, with worked-out examples, of what it means to say that two formal geospatial ontologies are semantically interoperable. The lack of such a definition leads to ambiguities and a lack of clarity when people discuss semantic interoperability, which in turn leads to a lack of consensus on how to recognize and deal with problems of semantic interoperability. The research in this thesis is central to the intersection of research in ontologies, the Semantic Web, and semantic interoperability issues in GIScience (Figure 1.2).

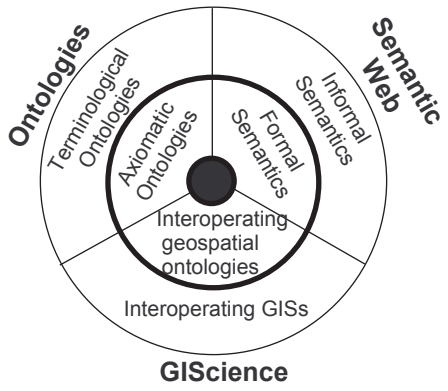

Figure 1.2: Thesis Research in Context

This thesis's approach to semantics is consonant with the judgment of Kuhn (2005b), that: "[t]he only sensible use of the term 'semantics' refers to *expressions in a language*" (Kuhn, 2005b, pg. 50) (italics in original). Kuhn explains that there are many different languages used by information systems, and that "[c]oping with geospatial semantics means, eventually, building ontologies specifying the meaning of expressions in most or all of these languages" (Kuhn, 2005b, pg. 50). The research in this thesis is in keeping with this sentiment, but its focus differs from Kuhn's. Whereas for Kuhn, it is the *service interfaces* (the interfaces among software components that are captured by a particular signature) that need to interoperate semantically, for this thesis, it is the ontologies themselves and the inferential software that supports them that need to interoperate semantically.

The particular line of research in this thesis is not meant to deny that people and machines disagree about meanings in a multitude of different, sometimes disconcerting or fascinating ways, not all of them related to issues of logical consequence. In a geospatial context, disagreements about meanings of spatial relationships arise from issues of scale (e.g., a person

being ‘in’ a building, versus a province being ‘in’ a country), imprecise information (e.g., being ‘near’ a restaurant), boundaries (e.g., is the boundary ‘part of’ the object or region which it bounds), etc. These potential sources of semantic discrepancies about spatial relationships are real, but they are not the central focus of this research. Rather, this thesis addresses what it means for two geospatial ontologies, which specify the meanings of spatial relations via model-theoretic semantics, to be semantically interoperable.

Semantic interoperability, in the context of logic, has been addressed by researchers in the areas of logic and computer science (Meseguer, 1998; Carnielli and D’Ottaviano, 1997; Goguen, 2005), ontologies and the Semantic Web (e.g., Obrst (2003); Grüninger (2004); Grüninger and Kopena (2005); Schorlemmer and Kalfoglou (2004)), as well as by researchers in geospatial semantics (e.g., Hobbs (2006); Adams (2006)), but not in the same way, or with the same kind of detail, as is done in this thesis (Chapters 2 and 3). Specifically, assuming the specification of spatial relations in two axiomatic geospatial ontologies, this thesis addresses a process for determining the level of semantic interoperability between the two ontologies, based on particular considerations of semantics and queries (Section 1.6).

1.3 Motivating Questions

The questions that motivated this research are:

1. What does ‘semantic interoperability’ mean?

2. How can the claim that two geospatial ontologies are, or are not, semantically interoperable be understood?

3. What is the connection between the results of queries to two geospatial ontologies and whether or not these ontologies are semantically interoperable?

4. Is there anything “special about spatial” when considering the semantic interoperability of geospatial ontologies?

The whole thesis is an investigation into question 1. The second and third questions are answered by the theoretical framework developed in Chapters 3 and 4, an implementation of which is given in Chapter 5. The fourth question is addressed in Section 2.6.

The specific research question of this thesis (Section 1.7.1) came about after considering certain details in the framework developed in Chapter 3. An overview of some of these details, sufficient for framing the research question, is given in the next three sections.

1.4 Implicit Meanings, Logical Consequence, and Semantic Interoperability

The example from Section 1.1 showed that John and Mary agree on two basic statements of their pool scenario, but disagree on whether a particular third statement followed as a logical consequence of the agreed-upon two.

Logical consequence is a formal relationship between formulas (in a formal language) that is intended to capture the intuitive notion of one statement following logically from others ((Ebbinghaus et al., 1994; Etchemendy, 1990)). The disagreement between John and Mary about whether ‘The ball is in the yard’ is, when formulated appropriately, a disagreement about logical consequence: John’s claim that ‘The ball is in the yard’ follows from (i.e., is a

logical consequence of, is entailed by) ‘The ball is in the pool’ and ‘The pool is in the yard.’ But Mary claims that the former statement is not entailed by the latter two. For John, part of what ‘in’ *means* in this scenario is the entailment that ‘The ball is in the yard.’

Note that John’s and Mary’s disagreement about whether the ball is in the yard occurs in spite of the fact that they use the same language and terms to describe their scenario. That is, on the surface it appears that there is no ‘heterogeneity’ to their semantics, and thus one might conclude that there should be no lack of semantic interoperability. However, as has been show, the lack of semantic interoperability can still take place. In this case, one could attribute this lack of semantic interoperability to an axiom that John holds to be true but fails to make explicit (i.e., that ‘in’ is transitive), rather than to any superficial difference in the vocabulary used to describe the scenario.

Computers working from formal specifications can exploit both explicitly stated assertions and derived implicit meanings in their analysis of semantics and semantic interoperability. One way they accomplish this is by considering the various possible truth values, stated or implied, of the set of possible statements about a given scenario. This thesis uses such an approach.

In the pool example there are three entities and one relation. Considering just simple statements like ‘The ball is in the pool,’ there can be nine such statements in this scenario, and each can be assessed as true or false.

A specification of the conceptualization of this scenario would thus implicitly consider 2^9 or 512 cases. Mary explicitly acknowledges 2 of the 9 statements to be true; thus, in the specification of Mary’s conceptualization just 2^7 or 128 different possibilities are left implicit. In Figure 1.3 these cases are collapsed into one 3x3 table, but it is useful to think of each

case as its own 3x3 table with definite values of T or F for each cell. The T/F in seven of the nine cells in Figure 1.3 indicates that Mary has left these values unspecified and that any combination of T's or F's in these cells is consistent with her view of the pool scenario.

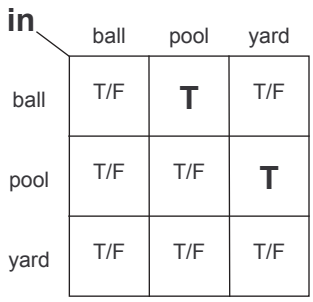

Figure 1.3: Interpretations Consistent with Mary's Conceptualization

To specify John's conceptualization, there also appear to be 128 different cases to consider. However, recall that John also claims that 'The ball is in the yard' (i.e., that for him this claim follows logically from 'The ball is in the pool' and 'The pool is in the yard.') Therefore John's specification shows a T in the ball/yard cell (Figure 1.4). Thus there are 64 possible 3x3 tables with T's in the ball/pool, pool/yard, and ball/yard cells. Another way to consider this third claim of John's is to say that 'The ball is in the yard' is true in any case (i.e., in any 3x3 table) that is consistent with 'The ball is in the pool' and 'The pool is in the yard.'

This kind of analysis of logical consequences underlies the definitions and computations of semantic interoperability used in this thesis.

One reason John says that 'The ball is in the yard' follows from 'The ball is in the pool' and 'The pool is in the yard' might be that he considers it to be axiomatic that for all entities X, Y, and Z in the domain of interest, 'If X is in Y and Y is in Z, then X is in Z.' If this axiom were codified and made part of John's explicit specification of his conceptualization (i.e., part

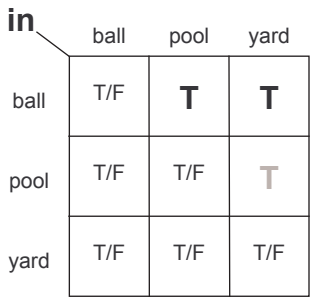

Figure 1.4: Interpretations Consistent with John's Conceptualization

of John's ontology) a computer could derive his implicit entailment that 'The ball is in the yard.' Once that entailment is derived, it can be compared to those statements that follow logically from Mary's specification of her conceptualization (i.e., Mary's ontology).

The nature of the discrepancy between the sets of entailed statements of two different ontologies reveals fundamental constraints on the semantic interoperability of the two ontologies.

1.5 Same Symbols, But Different Meanings

1.5.1 Need for Formalism

Sections 1.1 and 1.4 discussed implicit meanings and logical consequences, but to treat these topics in a way that is amenable to machine processing, some formal machinery is needed to handle the formal counterparts of implicit meanings and logical consequences.

This thesis assumes information systems can encode conceptualization like those of John and Mary in the above example, and it investigates precise descriptions for the semantic interoperability between such conceptualizations.

To identify the fundamental constraints on interoperability, one needs to describe precisely the differences in meaning that underlie the lack of semantic interoperability and indicate the kind of semantic interoperability at issue. This capability is established through an appropriate choice of formalisms (Chapter 3), which allows the rigorous treatment of the scenarios posed in this thesis. Specifically, because the scenarios involve logical consequences and entailment as fundamental aspects of semantics, and because the aim is to understand semantic interoperability via automated means, the choice of formalism must be one that machines can use to compute logical consequences. This approach yields unambiguous definitions of different levels of semantic interoperability (Chapter 3), which in turn facilitate the calculation of the degree of semantic interoperability between two geospatial ontologies.

As mentioned in Section 1.2.4, for the purposes of this thesis, a *geospatial ontology* is a collection of data and axioms concerning the spatial properties of geographic objects and relations. The semantics of a geospatial ontology is specified by model-theoretic semantics (Hodges, 1997; Manzano, 1999), where the meanings of the symbols in the ontology are given by *models*, in the way this term is used in database theory (Vianu, 1997) and logic (Manzano, 1999). In this thesis, *models* are sets with certain relations defined on them.

The next two sections describe in some detail the particular kind of semantic heterogeneity studied in this thesis: that of geospatial ontologies differing only in the semantics of their spatial relation symbols (the common symbols used to denote spatial relations, such as *in* or *through*). Nevertheless, the approach used can be generalized — within the setting of formal ontologies — to *any* semantic differences that are reflected in the models of the ontology.

This last remark deserves to be amplified. First, the approach used in this thesis analyzes an ontology's semantics in terms of its models (Chapter 3); therefore, any differences in semantics that are reflected as in the models of the ontology are amenable to the kind of analysis used in this thesis. Many such differences, however, lie outside the scope of the present investigation. For example, a difference in semantics that is due to two different words being used to name the spatial relation 'in' (e.g., *in* and *inside*) lies outside the scope of this thesis, though one could still conduct an analysis of semantic differences based on the differences in the models of the ontologies. Second, even though all the examples in this thesis deal with so-called 'populated ontologies,' (i.e., ontologies that include instance data like 'the pool,' 'the ball,' etc.), the analysis done in Chapters 3-5 is still applicable for ontologies that lack such instance data, i.e., ontologies that consist of just relationships between classes, and axioms.

The semantic heterogeneity between two geospatial ontologies is reflected directly in the degree or level of semantic interoperability between these ontologies (Chapter 3). A detailed specification of different levels of semantic interoperability is postponed until the necessary formalisms and examples have been presented (Chapters 2 and 3).

What follows is a consideration of models in the context of simple relational databases: collections of data tuples without any axioms. Afterward, the basic concepts of semantics and models are extended to apply to ontologies (which contain both data tuples and axioms).

1.5.2 Database Examples

Consider two very simple relational databases, where the only relation that is modeled is the spatial relation *in*, the only entities under consideration are Route 2, Orono, Bangor, and Maine, and there are no attributes for any of these entities. We assume the databases have the following information in common:

1. the 'real-world' domain, whose entities are Route 2, Orono, Bangor, and Maine;
2. the data domain: $\{Route2, Orono, Bangor, and Maine\}$;
3. the data model — in this case, the relational data model;
4. the database schema, which consists of a single binary relation 'in', without attributes (i.e., there is no special name given to the set of elements that could be in the first or second position of the tuples for *in*); and
5. the type of the data (character data).

Two different people might plausibly create the databases DB_1 and DB_2 (Figure 1.5), according to their understandings of what the spatial relation *in* means. The reader should not consider these databases to be simplified versions of Geographic Information Systems (GISs) or spatial databases (Rigaux et al., 2002), which specify or derive spatial relations via coordinate systems or via topological relations of spatial parts.

The examples in this section serve to illustrate the central ways in which an ontology differs from a conventional database: (1) the ontology specifies axioms in addition to data; (2) the semantics of an ontology typically commits to more than one fixed 'way the world could

be'; and (3) the analysis of the semantics of an ontology is inextricably linked to the idea of 'more than one way that the world could be.'

Consider DB_1 (Figure 1.5). The person who created DB_1 has a notion of *in* that does not include either the relationship "Route 2 is in Orono" or that "Route 2 is in Bangor," although this person's notion of *in* does include the relationships "Orono is in Maine," "Bangor is in Maine," and "Route 2 is in Maine." (Such an argument glosses over considerations of the open-world and closed-world assumptions (Reiter, 1978), which are discussed further in Chapters 2 and 3.)

DB_1: Tuples for $in(x, y)$	DB_2: Tuples for $in(x, y)$
	$(Route2, Orono)$
	$(Route2, Bangor)$
$(Orono, Maine)$	$(Orono, Maine)$
$(Bangor, Maine)$	$(Bangor, Maine)$
$(Route2, Maine)$	$(Route2, Maine)$

Figure 1.5: Two databases that model *in* differently

The difference in the two persons' notions of *in* is thus reflected, albeit imperfectly, by the different sets of tuples used to populate their respective databases. More specifically, the elements in the domain of the database, plus the data tuples in the *in* relation constitute a *model* (Manzano, 1999; Vianu, 1997) of the part of the real world under consideration. Once people create these tuples in the database, they effectively turn over any treatment of the meaning of *in* to the system that works with these tuples. Consequently, from the perspective of the database management system, the two different meanings of *in* are determined *solely* by the data elements and the tuples themselves. Whether or not the resulting databases can interoperate semantically is thus a function of the tuples they contain for the relation *in*.

Regardless of whether the two people who created DB_1 and DB_2 have accurately captured their intended meanings of *in*, *so far as the two databases are concerned*, the meaning of *in* is determined solely by the set of tuples in their respective relational tables for *in*, i.e., by their *models* of *in*. In this example, the tuples themselves, along with relational symbol *in*, comprise the only machine-processable information available for processing the meaning of *in*. There may be myriad other senses of 'in' that are not captured by this specification, but the point here is that the only thing a computer has to work with is the specification it is given, along with the rules for processing it.

Saying that two databases *mean* different things by the same relational symbol amounts to saying that the databases have different sets of tuples in their respective tables for this relational symbol. Schematically, this situation is depicted in Figure 1.6.

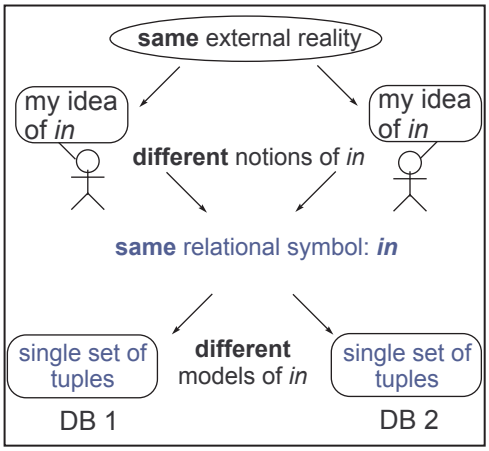

Figure 1.6: Same symbol, different meanings

The following assumptions underlie the framework of the above analysis of differences between DB_1 and DB_2.

- The two people are modeling the "same" external reality.
- The two people have their own individual notions of what the spatial relation *in* means.
- They use the same symbol—*in*—to represent this spatial relation in their databases DB_1 and DB_2.
- The set of tuples in one database may or may not overlap with the set of tuples from the other database. In the example of Figure 1.5, DB_1 and DB_2 have different, overlapping sets of tuples, which reflect the different notions their creators have of the spatial relation *in*.
- When one considers the differences in the specifications of the databases, one is no longer dealing with the notions of *in* that the two human modelers have. Rather, the only information that is available about the meaning of *in* is that which can be gleaned solely from the databases themselves. When the focus is on the databases themselves and not on what the human modelers may have intended, the set of tuples from the *in* table becomes the only tangible artifact available to determine what *in* means. That is, once the human modeler is out of the picture, the set of data tuples of a database essentially *is* the meaning of the relation *in*.
- The set of tuples in DB_1 or DB_2 *is* the *model* of the world that each person has created, whether or not this model faithfully depicts all the nuances of the relation *in* that the database creator might have.

DB_1 and DB_2 are *semantically heterogeneous* because even though they use the same symbol to describe the relation *in*, they mean different things by it. Given that DB_1 and DB_2 are semantically heterogeneous, to what extent might these databases nevertheless be semantically interoperable? There are many plausible ways to answer this question, among which are that

- their sets of tuples overlap;
- they give the same answers to the same questions;
- queries to the databases do not involve Route 2; and
- queries to the databases do involve Maine.

What these plausible answers show is that *in assessing to what extent two such databases might be semantically interoperable, it makes sense to consider both the sets of tuples in the databases and the queries put to the databases*.

To determine to what extent two *ontologies* (as opposed to two databases) are semantically interoperable, one also needs to consider the axioms of the ontologies and the models that result from including axioms along with data. The next section introduces these considerations. A fuller discussion of some needed technical background is given in Chapter 2.

1.5.3 Ontology Examples

Consider again two people modeling the relation *in* over the four entities of Route 2, Orono, Bangor, and Maine. Instead of explicitly providing all the tables of data via tuples, they now also specify axioms that they think *in* should satisfy. These axioms enhance the explicit data

tuples for *in* by adding other conditions that *in* must satisfy. Such a combination of data and axioms is an example of an axiomatic ontology. Assume further that the two people use the same language (e.g., the language of first-order logic) to specify the axioms for *in*.

Schematically, these assumptions are represented in Figure 1.7. Note the three major differences between this figure and Figure 1.6. First, Figure 1.6 deals with databases; second, it makes no mention of a logical language; and third, it shows just one model (i.e., set of tuples) for each database. By contrast, Figure 1.7 deals with ontologies, it mentions the logical language (used for specifying axioms and for drawing inferences), and it shows more than one model (set of tuples) for each ontology.

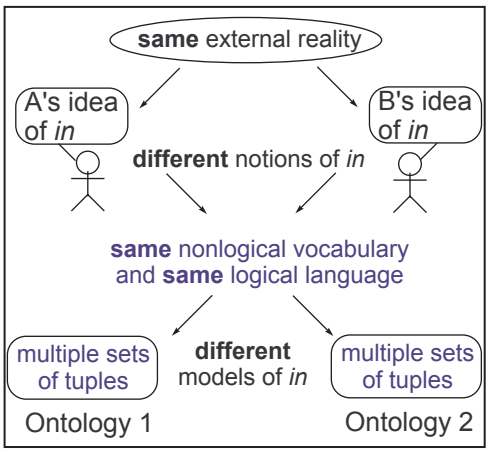

Figure 1.7: Same symbols and logical language, different meanings

Consider the following points.

- The two people are modeling the same external reality.
- The two people have their own individual notions of what the spatial relation *in* means.

- They use the same symbol—*in*—to represent this spatial relation.
- This same symbol and the same logical language are used by both ontologies.
- Each ontology has multiple sets of tuples, each of which is a model of the spatial relation *in*, reflecting the individual's notion of *in*.
- Once the focus is on the ontologies (as machine-processable written artifacts), one is no longer dealing with exactly which notions of *in* the two human modelers have had in mind. Instead, the ontology is taken to be the best available approximation to their notions, and the focus is on the meaning of *in* as that meaning is specified via the semantics of the ontologies themselves.

In the case of the example databases (Figure 1.6), *the model* of a given database consists of just the data elements along with a single set of tuples for the relation *in*. Under the viewpoint taken this thesis, this unique model defines what *in* means in that database.

In the case of an ontology, *a model* also consists of the data elements along with a single set of tuples for the relation *in*. The significant difference between an ontology and a simple database is that an ontology usually has multiple models, expressed implicitly by the combination of data and axioms, whereas a database has just a single model, specified exactly by the data tuples.

Each of the ontologies in Figures 1.8 and 1.9, for instance, has more than one model, because the axioms and the data of the ontology do not uniquely constrain the set of tuples that specify the semantics of the relation *in*. The fact that ontologies in general typically have more than one model is central to any definition of semantic interoperability between ontologies. This is so in spite of the fact there may be other, human-significant aspects of the

relation *in* that are not captured by the ontology. But because ontologies do not capture any aspects of meaning outside the data and the axioms (and the formal framework of reasoning in which they are embedded), such meanings are not amenable to the formal analysis used in this thesis.

Axioms for in	Data Tuples for in
$\forall x,\, in(x, x)$	(Route 2, Orono) (Orono, Maine)

Figure 1.8: Ontology O_1 for *in*

Axioms for in	Data Tuples for in
$\forall xyz,\, in(x, y) \wedge in(y, z) \rightarrow in(x, z)$	(Route 2, Orono) (Bangor, Maine) (Orono, Maine)

Figure 1.9: Ontology O_2 for *in*

Consider the ontologies O_1 and O_2 specified in Figures 1.8 and 1.9 above. Each contains a single axiom and some tuples of data. The axiom of O_1 states that every entity is in itself, and the axiom of O_2 states that the relation *in* is transitive. More significantly, the particular axioms and data tuples of the ontologies combine to specify (implicitly) the *models* of each ontology.

A *model for a given ontology* is a depiction of how the world could be configured that conforms to the data and the axioms. (Chapter 3 gives a more precise definition.) For instance, in any model of O_1 the following relationships must hold: "Route 2 is in Route 2," "Orono is in Orono," "Bangor is in Bangor," and "Maine is in Maine", because these relationships are dictated by the axiom. Similarly, in any model of O_1 the relationships "Route 2 is in Orono" and "Orono is in Maine" must hold, because these relationships are dictated by the data.

A convenient way to picture the models of O_1 is to use graphs, since the entities of Route 2, Orono, Bangor, and Maine can be represented by vertices in a graph, and the single binary relation *in* can be depicted by the directed arrows in the graphs. In the top half of Figure 1.10, two of the models of O_1 are shown as graphs, where the vertices labeled 'R2,' 'O,' 'B', and 'M' stand for Route 2, Orono, Bangor, and Maine, respectively. Similarly, the bottom half of Figure 1.10 shows two models of O_2.

Figure 1.10 suggests that O_1 and O_2 do not have the same sets of models (because model 1 of O_1 is different from model 1 of O_2). This turns out to be the case, since model 1 of O_2 could never be a model of O_1 (since the *in* relation in model 1 of O_2 is not reflexive), and so the sets of models of the two ontologies cannot be the same. Because O_1 and O_2 have different sets of models, they have different semantics, in particular, different semantics for the spatial relation *in*.

Even though both O_1 and O_2 use the same relation symbol, and the same logical language to express axioms, the *meanings* that the two ontologies give to the symbol *in* are different. This is true in spite of the fact that the ontologies have at least one model in common — the the right-hand model in Figure 1.10.

Figure 1.10 exemplifies another important difference between the semantics of databases and the semantics of ontologies. Although in both cases the meaning of symbols is formalized via models, if two *databases* have a model in common, then they have the same semantics (based on the discussion in the previous section), because the model for a given database is unique. Two *ontologies*, however, may have one or more models in common and yet have *different* semantics, since an ontology generally has more than one model.

One might suppose that the difference in the semantics of *in* between O_1 and O_2 is due to the fact that O_1 has one axiom and O_2 has a different axiom. But that fact provides only a partial explanation for the different semantics of O_1 and O_2. A fuller explanation is that the different meanings are due to each ontology's particular combinations of axioms and data that result in different sets of models for the two ontologies. There are three ways that two ontologies might have different sets of models: (1) they have the same data and different axioms; (2) they have the same axioms but different data; or (3) they have different axioms and different data, which is the case of O_1 and O_2. What is significant about the semantics of an ontology is not the data tuples or axioms considered individually, but rather the way that the data and axioms collectively determine the models of the ontology (see Section 3.1).

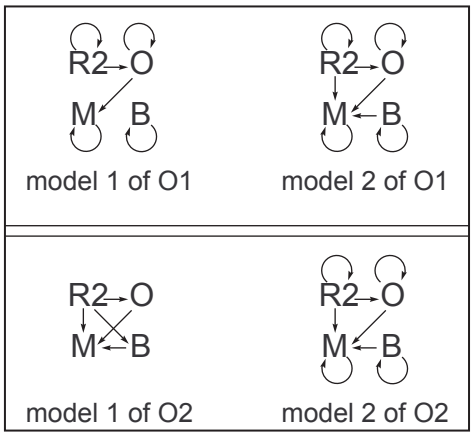

Figure 1.10: Two models of Ontology 1 and of Ontology 2

Since the semantics of the ontologies is determined by their models, and since O_1 and O_2 have different sets of models, one can say that they are *semantically heterogeneous*. Given this particular view of semantic heterogeneity, to what extent might the ontologies nonetheless be semantically interoperable?

To answer this question, consider the following. The semantic interoperability of ontologies has to do with implicit logical consequences. These consequences can be probed by determining whether a given query (statement in the same logical language used to specify the ontology) evaluates to True in all models of the ontology. So, it makes sense that an analysis of semantic interoperability takes into account both models and queries.

Thus, among the plausible answers to the above question are that

- the sets of models of O_1 and O_2 overlap;
- queries to each ontology target one or more of the models they have in common;
- a given query Q posed to O_1 gives the same answer when posed to O_2.

Chapter 3 explores in-depth an answer to this question based on the sets of models of the two ontologies and the queries put to the ontologies.

1.6 Assessing Semantic Interoperability

Three points emerge from the analysis in Sections 1.3 - 1.5.

1. Two geospatial ontologies that deal with the same domain and use the same language to describe this domain can nevertheless differ in their semantics.
2. Because the semantics of these ontologies is defined via model-theoretic semantics, the differences in meanings between the ontologies are reflected in differences between the sets of models of the two ontologies.

3. Analysis of the semantic interoperability of two ontologies should take into account both the connections between the models of two ontologies and the results of queries put to these two ontologies. These connections relate the semantics of the ontologies (i.e., the models of the ontologies) to the implicit logical consequences specified by the ontologies, which can be probed by evaluating particular queries in the ontologies. That is, whether or not two ontologies interoperate semantically is a function of the relationship between their models and their logical consequences (statements satisfied by all models of a given ontology), and these relationships can be probed via queries.

Assessment of semantic interoperability of geospatial ontologies in this thesis is therefore based on:

1. two similar (but not identical) geospatial ontologies that deal with the same domain using the same language to describe this domain, but differing in the semantics of the spatial relation symbols used to describe this domain;

2. the models of these ontologies, because the semantics of the spatial relations in these ontologies are defined via their models; and

3. the queries that can be put to the ontologies, for two reasons. First, ontologies are used (at least in part) to answer queries. Second, the result of a query to an ontology depends on the models of the ontology in the sense that a query evaluates to true if it is true in all models of the ontology (Section 3.1).

1.7 Research Question and Goal

1.7.1 Research Question

The research question addressed in this thesis is:

When two geospatial ontologies use the same language to describe the same domain, but differ in the model-theoretic semantics of their primitive spatial-relation symbols, in what sense and to what extent are the ontologies semantically interoperable?

Our focus on one specific kind of semantic heterogeneity (differences in the interpretation of primitive spatial relation symbols) presupposes a large amount of similarity between the two ontologies. Section 4.1 explains exactly what is assumed to be common between the ontologies.

Note that the terms 'meaning' and 'semantics' are used in this thesis in a very particular way. The reader should understand that the meanings considered here are 'internal' to some computer-based information system (Lipski, 1981), but not necessarily internal to the minds of the human beings who design and use these information systems.

1.7.2 Research Goal

The research goal is to create a method to assess the extent of semantic interoperability between two geospatial ontologies, based on the considerations given at the end of Section 1.6.

This research goal is achieved by:

1. defining six semantic compatibility conditions between ontologies (Chapter 3);

2. using the six compatibility conditions to specify, in terms of models and queries, under

what conditions two geospatial ontologies are: (1) completely semantically interoperable; (2) partially semantically interoperable; or (3) not at all semantically interoperable (Chapter 3); and

3. developing (Chapter 4) and implementing (Chapter 5) an algorithmic and a heuristic procedure that allow, under suitable assumptions and in certain cases, the calculation of a unique level of semantic interoperability between two ontologies.

1.8 Intended Audience and Scope

1.8.1 Intended Audience

This thesis is intended for researchers, software developers, and users of geospatial ontologies of the kind that might soon be available on the Web. It is of potential interest to non-commercial developers of spatial and terminological ontologies, as well as commercial spatial database vendors. It is also of potential interest to computer scientists, knowledge engineers, and database specialists. In particular, those researchers working in the areas of ontologies, semantics, and semantic interoperability, and the Semantic Web can find in this thesis the clarification of several fundamental notions dealing with semantic interoperability of ontologies.

1.8.2 Scope

The scope of this thesis is limited as follows.

- This thesis does not consider the psychology behind the modeling of the semantics of spatial relations. Instead, the point of departure for the analysis conducted here is the existence of geospatial ontologies, processable by computers.
- This thesis does not consider, except as background information, the many different ways there are to classify semantic heterogeneities in ontologies. Instead, the investigation is narrowed to focus on the ramifications of one kind of semantic heterogeneity.
- This thesis does not consider the semantics of traditional GISs or other ontological semantics that are not based on model-theoretic semantics.
- This thesis does not aim to create methods to align, integrate, or merge ontologies.
- This thesis does attempt to pursue the most abstract formulations (e.g., (Schorlemmer and Kalfoglou, 2004; Grüninger, 2004)) of the issues investigated here.

1.9 Results and Contributions

1.9.1 Results

1. A formal analysis of the semantic interoperability of geospatial ontologies based on models and queries—a unique research contribution;
2. The creation of six semantic compatibility conditions between ontologies; each compatibility condition is defined in terms of models and queries;

3. Definitions, in terms of compatibility conditions, of five specific levels of semantic interoperability of geospatial ontologies;

4. An algorithm for computing, given certain assumptions, the exact level of semantic interoperability between two ontologies;

5. An method for computing, given certain other assumptions, the possible levels of semantic interoperability between two geospatial ontologies; and

6. Implementations of these two algorithms using SEM (Zhang and Zhang, 1995) and Perl. SEM is used as a black box to generate the models of each ontology, once these ontologies are specified in the appropriate format (see Chapter 5.) Perl is then used on the sets of models generated for the ontologies, to determine what relation (e.g., overlap, contains) holds between the model classes of the ontologies.

1.9.2 Contributions

The chief research contribution of this thesis is the novel analysis—in terms of models and queries—of semantic interoperability.

Additional contributions are:

- *Conceptual clarity*. By narrowing the focus to ontologies that use the same language and symbols to specify constraints on a given geospatial domain, this thesis establishes with detailed examples a conceptual foundation for the analysis of semantic interoperability, based on models and queries.

- *Foundation for further study*. Taking advantage of the conceptual clarity achieved by narrowing the focus to ontologies that use the same language and symbols to specify constraints on a given geospatial domain, this thesis positions researchers to undertake additional analyses of semantic interoperability for cases where other aspects of the formal setup are different, e.g., the signatures of the ontologies differ, or the logical languages themselves are different.

The research contributions give researchers and users of geospatial ontologies sufficient background to be able to:

- understand the role of models and queries in the definition of semantic interoperability;
- appreciate the issues involved in formalizing intuitive notions like ‘semantic interoperability’;
- scrutinize claims of semantic interoperability of geospatial ontologies;
- define ‘semantic interoperability’ in a way that is appropriate for formal geospatial ontologies (i.e., via models and queries);
- create a definition of semantic interoperability that actually takes into account the different (model-theoretic) semantics of spatial relations; and
- understand the relationships between different levels of semantic interoperability and the patterns of possible query results.

1.10 Summary

People sometimes mean different things when they use the same words to describe what appears to be the same objective reality. Such differences in meaning can arise if people differ in the implicit meanings they assign to certain words. If people can codify in a formal ontology their explicit conceptualizations then the implicit differences in meanings of their conceptualization, and the ramifications of these differences, can be analyzed computationally.

This thesis explores differences in the meaning of basic spatial relations (e.g., 'in' and 'through') as these differences are specified in geospatial ontologies. It considers two similar geospatial ontologies that differ in the semantics of these primitive spatial relation symbols (e.g., *in* or *through*) and asks to what extent the two ontologies are 'semantically interoperable.'

To assess to what extent two such geospatial ontologies are semantically interoperable, this thesis focuses on the models of the ontologies and on the queries that can be put to the ontologies. It develops the notion of 'compatibility conditions' in terms of models and queries and uses these compatibility conditions to define different levels of semantic interoperability between two ontologies. It develops an algorithm and a heuristic for determining the level of semantic interoperability between sample geospatial ontologies. Finally, it connects the results of this thesis to possible jumping-off points for further research.

Chapter 1 has shown that even when two ontologies are identical in everything but the model-theoretic semantics of their spatial-relation symbols, the question of in what sense, or to what extent they are semantically interoperable is still an open one. This thesis answers that question (Chapters 3-5, Section 6.2).

1.11 Organization of Remaining Chapters

Chapter 2 reviews related work and provides the technical foundations needed to understand the subsequent analyses in Chapter 3. Chapter 3 defines semantic heterogeneity of geospatial ontologies in terms of models, and defines six compatibility conditions that characterize the semantic interoperability between ontologies. Chapter 3 then uses these compatibility conditions to specify under what conditions two geospatial ontologies are completely semantically interoperable, partially semantically interoperable, or not at all semantically interoperable. Chapter 4 develops procedures for determining the level or levels of semantic interoperability between two ontologies. Chapter 5 implements these procedures, and Chapter 6 provides a summary, conclusions, and directions for future work.

Chapter 2

PERSPECTIVES ON SEMANTIC INTEROPERABILITY

Chapter 1 framed the research question of this thesis: When two geospatial ontologies use the same language to describe the same domain, but differ in the model-theoretic semantics of their spatial-relation symbols, in what sense and to what extent are the ontologies semantically interoperable?

In framing this research question, Chapter 1

- explored a particular kind of semantic heterogeneity between formal geospatial ontologies, namely, that of using the same spatial-relation symbol to mean different things, where meaning comprises explicit data and axioms, as well as their logical consequences;
- showed how this kind of semantic heterogeneity is reflected in the *models* of the ontologies; and

- argued that to answer the research question appropriately one needs to consider both the models of the ontologies and the queries posed to the ontologies.

Chapter 2 fills out this framework by providing the definitions and context needed for the approach used in Chapters 3-5 to answer the research question. Section 2.1 discusses the modern notion of *ontologies* and their uses in helping come to terms with different semantic conceptualizations of the 'same' objective reality. Section 2.2 discusses the nature and role of ontologies in GIScience (i.e., geospatial ontologies). Section 2.3 focuses on how geospatial ontologies are used to model geospatial relations. Section 2.4 zeros in on the particular semantic difference (heterogeneity) treated in this thesis: the case of two ontologies that have different semantics for the same primitive spatial-relation symbols. The chapter is rounded out by two additional context-setting sections: Section 2.5, which discusses relevant related approaches to semantic interoperability, and Section 2.6, which discusses to what extent the spatial nature of the conceptual domain studied here is 'special' for the treatment of semantic interoperability given in this thesis.

2.1 Ontologies

The word 'ontology' has acquired a distinctive meaning in the last decade or so, especially in the research literature in computer and information science (Guarino, 1998; Smith and Welty, 2001) The word 'ontology,' rather than referring just to the philosophical discipline that studies the categories of things that exist (Casati et al., 1998; Sowa, 2000), has come to refer to a certain type of artifact written in natural language or in a particular formal notation, usually with the intent that it be processable by computers (Guarino, 1998).

One of the most commonly quoted definitions of an ontology is that it is "an explicit specification of a conceptualization" (Gruber, 1993, pg. 199).

The unifying ideas behind this definition and the many other similar ones used in computer and information science are:

- an ontology describes what some person or group of people thinks are essential categories, entities, features, properties, or relations in some domain of interest;
- an ontology does not exist solely in people's heads; it can also be, and is most often used as, a written artifact;
- an ontology is usually created so that it can be manipulated by computers, with the goal that implicit information in the ontology can be extracted and used; and
- an ontology may or may not use axioms to describe a particular domain of interest.

The philosophical discipline of ontology deals with categories of things that exist and relationships among these categories. Naturally, people will often disagree about these categories and relationships. However, even when people agree on the fundamental categories and relationships under consideration, and when they specify these categories and relationships in the same language (natural or artificial), they can still misunderstand each other.

One widely recognized way that people misunderstand each other is by using different terms to denote the same modeling constructs. For instance, depending on the modeling language used, a 'category' might be termed a 'kind,' a 'class,' a 'concept,' or a 'unary relation.' Similarly, a relationship between categories might be termed a '(binary) relation,' a 'slot,' a 'role,' or a 'property' (Gomez-Perez et al., 2004, pg. 203). A useful discussion of how different terms are used to refer to the same notion is given in Lassila and McGuinness (2001). This

thesis uses a common language and modeling constructs, so that when it examines the semantic interoperability between two ontologies, this extra source of variability (different ways of modeling ontologies) does not cloud the issue.

A second widely recognized way that people can misunderstand each other even when they agree on the basic categories and relationships under consideration is by using the same words to mean different things. This thesis studies this second kind of potential misunderstanding, which the development and use of ontologies was largely designed to overcome.

Modeling constructs such as 'class' or 'role,'can be considered to be the tools people use to *build* ontologies (Beard-Tisdale, 2006). These constructs should be distinguished from the relations *described by* ontologies that use these tools. For example, a binary relation (sometimes termed a 'slot,' a 'role,' or a 'property') is an ontological modeling tool (one could call it a modeling primitive) that can be used in a given ontology to model any number of domain-specific relations that happen to be binary. In the spatial domain, for example, a binary relation can be used to model a variety of common spatial relations (e.g., 'meets' or 'overlaps', or 'in' or 'through'). A ternary relation can be used to model other spatial relations, like 'between' (as in 'A is between B and C'). If an ontology does not have an appropriate modeling tool (e.g., a binary relation), then it will not be able to model certain domain-specific notions.

Since ontologies deal with categories of things that exist and the relationships between these categories, most if not all ontology modeling languages have the ability to model classes, the subclass relation, and an instance-of-a-class. As a result, a main use of ontologies is to organize knowledge hierarchically and to determine whether a given class is a subclass of another class, or whether a given entity is an instance of a particular class (Sowa, 2000).

In the last few years some relatively sophisticated languages have been used to model ontologies, with the result that instead of serving as simple classification tools, ontologies can now specify information about the attributes of classes and instances, as well as cardinality constraints on members of the domain (Gomez-Perez et al., 2004). According to Gomez-Perez et al. (2004, pg. 203), ontology languages usually allow the expression of: (1) *concepts*, (2) *attributes*, which can be divided into instance attributes and class attributes, (3) *taxonomies* of concepts, (4) *relations*, (5) *axioms*, and sometimes (6) *functions*. (The ontology language L used in this thesis (Chapter 3) allows the expression of concepts, relations, and axioms.)

Ontologies exist across a range of complexity (Lassila and McGuinness, 2001) from taxonomies that allow the modeling of class-subclass relations to knowledge bases that allow certain kinds of inferences (Guarino and Giaretta, 1995). Correspondingly, ontology modeling languages range in complexity with respect to the modeling constructs they can specify (Gomez-Perez et al., 2004). On the expressive end of the range are languages that can specify a wider breadth of explicit and implicit knowledge of a domain and can use automated reasoning tools to obtain inferences (Obrst, 2003).

That is, ontologies can allow more kinds of inferences about the classes and elements in a domain than simply whether class A is a subclass of class B, or whether x is an instance of a class C. Exactly which inferences can be made for a given ontology is a function of the logical language used to specify the constructs of the ontology and the semantics used to assign a formal meaning to these constructs.

The languages used to specify axiomatic ontologies typically have a model-theoretic semantics that gives the modeling primitives and language statements a well-defined semantics. These languages also permit: (1) relatively complex constraints on the categories and entities

in a given domain to be specified via axioms; (2) inferences to be carried out via automated reasoning; and (3) the specification of queries that can go beyond questions of class-subclass or class-instance relationships. These three capabilities are characteristic of 'knowledge bases.' Thus, the line is blurred between ontologies as classification aides and ontologies as knowledge bases that can support more complex kinds of constraints and queries. For a further discussion of the relationship between ontologies and knowledge bases, see Guarino and Giaretta (1995).

2.2 Ontologies in GIScience: Geospatial Ontologies

The use of ontologies in GIScience (Casati et al., 1998; Smith and Mark, 1998; Mark et al., 1999; Fonseca, 2000) has come about in recent years to deal with certain problems of interoperability (Sondheim et al., 1999; Egenhofer, 1999; Goodchild et al., 1999) having to do with the semantics (Sheth, 1999; Egenhofer, 2003) of spatial data (Shekar and Chawla, 2002).

The focus on ontologies to help solve interoperability problems in GIScience has its roots in research efforts involving interoperating Geographic Information Systems (GISs) (Goodchild et al., 1999). According to one researcher, the goal of interoperating GISs is "to achieve an automated process that will allow us to use data and software services across the boundaries that their collectors and designers envisioned" (Egenhofer, 1999). Egenhofer goes on to note that the difficulties "are primarily in the semantics of the diverse applications," that "[c]ompatible semantics of geospatial information are a key characteristics of interoperating GISs," and that "powerful methods to capture and describe geospatial semantics are critical" (Egenhofer, 1999, pg. 1).

One recent article summarizes: “The primary purpose of using ontology in GIScience is to define a common vocabulary that will allow inter-operability [*sic*] and minimize any problems with data integration, both from different systems and between users and systems” (Agarwal, 2005, pg. 508). Yet, as the examples in Chapters 1, 3, and 5 show, even when people use a common vocabulary, they can still fail to interoperate semantically.

Ontologies have been identified as a research area by the University Consortium of Geographic Information Science (Egenhofer, 2004), and by several governmental agencies (e.g., the National Geospatial Intelligence Agency (NGA) and the USGS in the United States, and the UK Ordnance Survey in the UK).

Yet, in spite of this recent research activity in GIScience on ontologies and interoperability, the notion of semantic interoperability is avowedly “hard to pin down,” because it is “somewhat redundant, there is no accepted formal definition, there are no benchmarks or commonly agreed challenges, the role of humans in the loop is unclear, and the acronym inflation around the semantic web obscures rather than highlights the deeper research issues” (Kuhn, 2005a, pages. 1-2).

Further, as of August, 2006 NGA has no standard geospatial ontology with which to interoperate semantically (Adams, 2006).

That dealing with semantic interoperability of geospatial ontologies should prove to be challenging is not surprising. There are many different ways to understand, and potentially resolve, various kinds of semantic heterogeneity that can stand in the way of semantic interoperability. Following is a discussion of the approaches to semantic interoperability in GIScience most relevant to the approach taken in this thesis.

Bishr (1998) distinguishes two kinds of semantic heterogeneity: *cognitive heterogeneity*, which is due to differences in the mental models being represented in information systems, and *naming heterogeneity*, which occurs when different words refer to the same real-world entity. Bishr creates a framework for comparison of the interoperability of Geographic Information Systems that consists of different levels, ranging from the lowest level of network protocols to the highest levels of data models and application semantics (Bishr, 1998, pg. 300). He then proceeds to analyze the problems of heterogeneity and interoperability at the levels of data models and application semantics, and he offers some solutions for these problems.

The issue of resolving semantic heterogeneities in ontologies for geographic information processing has also been discussed in Visser et al. (2002). The authors use "the term *semantic integration* or *semantic translation* to denote the resolution of semantic conflicts that disable a one-to-one mapping between concepts or terms" (Visser et al., 2002, pg. 7). The authors' approach considers contextual information related to the concepts or classes to which entities belong, and they seek to explicate this context by defining the necessary and sufficient conditions for an entity to belong to a given class. A further consideration in their approach is "how and what kind of context knowledge has to be considered in the translation process because the choice of the representation has mayor[sic] impacts on the classification method to choose and the expected results" Visser et al. (2002, pg. 8).

Hakimpour and Timpf (2002) also deal with semantic interoperability of geospatial ontologies. Their approach discusses how to merge ontologies based on the semantic similarity of intensional definitions of *terms*. Their analysis of semantic similarity in terms of set-theoretic relations (e.g., overlap, contains) appears somewhat similar to our analysis of the 5 model-class relations introduced in Chapter 3.

The classification of semantic heterogeneity by Bishr is not applicable to this thesis, for three reasons. First, although it could be argued that the heterogeneity exhibited in the sample ontologies O_1 and O_2 from Chapter 1 (Figures 1.8 and 1.9) is an example of Bishr's cognitive heterogeneity, Bishr's approach offers no way to assess the level of semantic interoperability of systems that exhibit such kind of heterogeneities. This thesis, by contrast, does offer a framework (Chapter 3) for assessing the level of semantic interoperability between ontologies. Second, the kind of semantic heterogeneity exhibited by O_1 and O_2 is clearly not Bishr's 'naming heterogeneity,' where different words refer to the same real-world entity. Rather, this thesis considers the situation in which the *same* word has *different* meanings. Third, Bishr's classification of semantic heterogeneity is connected to the different levels of abstraction people use to model the conceptual and symbolic worlds. The focus in this thesis, by contrast, is on differences in the actual specifications of ontologies (and their models), rather than on issues relating to how people conceptualize their worlds.

Visser et al. (2002) do deal with differences of semantics in the framework they use; however, their framework is not easily comparable to our approach, for three reasons. First, their approach deals with mappings between ontologies that match concepts or terms, whereas our approach does not involve any mappings between ontologies. Second, under their approach it is not clear which components of different ontologies (e.g., languages, vocabulary, semantics) they consider to be the same for two given ontologies and which they consider to be different. In our approach, on the other hand, these similarities and differences are made explicit (see Figure 2.2 and Section 4.1). Third, their notions of semantic integration and semantic translation are not based, or at least not obviously based, on the models of the ontologies, whereas this thesis uses models directly to define and calculate levels of semantic interoperability.

Although Hakimpour and Timpf (2002) deal with semantic interoperability of geospatial ontologies and although they use an approach based on set-theoretic relations like overlap and contains, it is not clear that models and model classes are the basis of their analysis.

2.3 Semantics of Spatial Relations

Within the realm of spatial data, researchers have paid particular attention to representing and reasoning with spatial relations (e.g., Cohn and Hazarika (2001)). Approaches to spatial relations have fallen into two major areas: topological representations and algebraic manipulations (e.g., Egenhofer and Franzosa (1991)), and logical treatments (e.g., Randell et al. (1992)). Cohn and Hazarika (2001) provide a recent overview of these broad areas of qualitative spatial reasoning, as well as references to additional work by Egenhofer and Cohn.

Common to both these veins of research are the two fundamental ideas of *constraints* and *consequences*. For instance, when using topological representations and algebraic manipulations to represent and reason about spatial relations, researchers make their treatments of spatial relations precise by *constraining*, via a topological specification, 'what it means' for one spatial region to be *contained* in another. Then, the *consequences* of these constraints are worked out via algebraic manipulations, so that, for example, if region A is contained by region B and region B is contained by region C, then region A is also contained by region C.

The ideas of constraints and consequences also apply to the kinds of qualitative reasoning in the second research vein mentioned above (e.g., Renz (2002)). In this approach, logical formulations are provided to *constrain* 'what it means' for, say, one region to be a non-connected proper part of another region. Then, the machinery of logical inference is brought to bear to determine the *consequences* of the given constraints.

This thesis similarly treats the semantics of spatial relations from within the framework of 'constraints and consequences.' Under this view, the *constraints* on the semantics of spatial relations are given by the data tuples and axioms in an ontology. For example, if one data tuple of one ontology represents the claim that 'the ball is in the pool,' this data tuple effectively *constrains* the meaning of the spatial relation *in* in such a way that (for this ontology) the ball is indeed *in* the pool. (In subsequent examples in Chapter 3, we shall see that axioms of an ontology also constrain the semantics of the spatial relations.) As for the *consequences* associated with the spatial relations, these, too, must be considered 'part of' their meaning. For instance, the discussion in Chapter 1 about whether the ball is in the yard dealt with the (implicit) consequences of what it *means* to say that the ball is in the pool and the pool is in the yard.

Following is a brief overview of the formal apparatus used in this thesis to treat the constraints and consequences that together determine the semantics of spatial relations.

For the purposes of this thesis, a *geospatial ontology* is an axiomatized ontology of the geospatial domain. A geospatial ontology differs from a conventional Geographic Information System (GIS) in four fundamental ways.

1. A conventional GIS is a software system that contains data about geographic entities and relations that are tied to one or more coordinate systems (Rigaux et al., 2002),

whereas a geospatial ontology is a collection of axioms and geospatial data, which is often assumed to be supported by querying and inferencing software, but whose entities and relations are not necessarily tied to a particular coordinate system. (The geospatial ontologies in this thesis do not use a coordinate system.)

2. Even when a conventional GIS does use constraints on geographic entities and relations, these constraints are not subsequently interpreted in models (Section 3.1), whereas in a geospatial ontology they are.

3. A geospatial ontology (of the kind dealt with in this thesis) uses a logical language to specify its data and axioms.

4. A geospatial ontology can use an automated reasoner to carry out inferences from its axioms and data.

By the definition above, the sample ontologies in Chapter 1 are geospatial ontologies. The language they use is a subset of the language of first-order logic. Their axioms are expressed via two kinds of vocabulary: a logical vocabulary, and a non-logical vocabulary. The logical vocabulary consists of a finite set of variables (e.g., $x, y, z,$ etc.), logical operators $\vee$, $\wedge$, and $\neg$, the universal quantifier ($\forall$) ('for all'), and the existential quantifier ($\exists$) ('there exists'). The non-logical vocabulary consists of a finite set of constant symbols and a finite set of relation symbols. In the examples of Chapter 1, the constant symbols are *Route 2, Orono, Bangor,* and *Maine* that denote Route 2, Orono, Bangor, and Maine, respectively. The sole relation symbol in the non-logical vocabulary is *in*. In general, a nonlogical vocabulary can also include function symbols, although for the ontologies in this thesis no function symbols

are used. (See Chapter 3 for a more detailed discussion of the logical language used in the analyses of this thesis.)

The language used in the examples of Chapter 1 is capable of expressing certain claims about the four geographic entities under consideration: Route 2, Orono, Bangor, and Maine. In particular, that language allows us to express various claims about, or constraints on, how those entities are related to each other via the spatial relation 'in.' These claims are expressed by axioms written in the language, as well as by the data tuples that specify (the extension of) the relation *in* (Figures 1.8 and 1.9).

The geospatial nature of the ontologies comes from the fact that the entities in the ontology are geographic entities, and the relation in the ontology is the spatial relation *in*. This spatial relation can be axiomatized to have various properties, such as the antisymmetry of 'in': 'If X is in Y, and Y is in X, then X equals Y'.

Two people who create ontologies specifying spatial relations can legitimately differ on which properties that they believe these spatial relations should satisfy. These differences in properties are reflected in the different axioms and data tuples that the two people or groups use to create their ontologies, and the differences in data and axioms are further reflected in the (possibly) different *semantics* of the two ontologies, which are specified in the (possibly) different *models* of the ontologies. Thus, the 'semantic consequences' of the legitimate differences that people may have about the properties of spatial relations can be explained by examining the models of the ontologies. One such 'semantic consequence' is whether or not the resulting ontologies can interoperate semantically, in some specified sense.

This thesis is not concerned with finding a 'correct' characterization of *in* or any other spatial relations. Rather, given that two people can legitimately create different ontologies for

spatial relations, it investigates to what extent these ontologies could be semantically interoperable (in the precise sense specified in Chapter 3).

2.4 Our Target: Different Semantics of Primitives

The treatment of semantic interoperability most directly relevant to this thesis is that of Klein (2001). Klein's classification scheme identifies a specific semantic heterogeneity dealt with in this thesis: two ontologies differing (only) in their semantics of primitives (Figure 2.1). Klein does not, however, discuss how to ascertain or compute any degree or level of semantic interoperability between two such ontologies.

The kind of semantic heterogeneity dealt with in this thesis, where the same spatial-relation symbol used in two ontologies has a different formal semantics, appears to be exactly what Klein describes as a language-level mismatch that deals with different semantics of primitives. In describing this mismatch Klein says: "Despite the fact that sometimes the same name is used for a language construct in two languages, the semantics may differ; e.g., there are several interpretations of `A equalTo B`." It is possible that Klein may have implicitly considered the case in which the two languages are different; in this thesis, however, we explicitly state that the languages themselves (and the symbols used in them) are the same for the two ontologies; only the model-theoretic semantics of the spatial relation symbols differs.

Klein begins by distinguishing two broad types of mismatches and several subtypes within each type (Figure 2.1).

His two broad types of mismatch are *language-level mismatches* and *ontology-level mismatches*. At the language level, the mismatches are "between the *mechanisms* to define

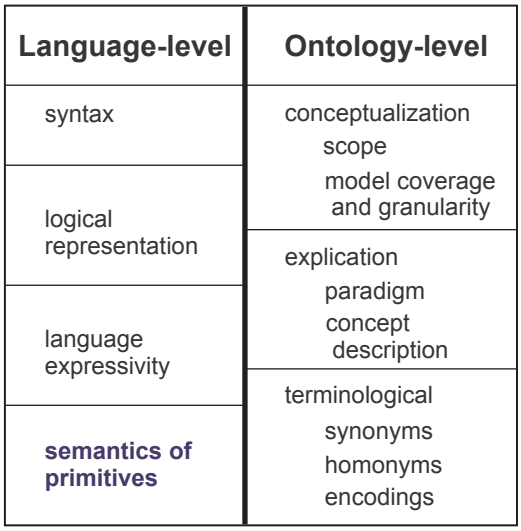

Figure 2.1: Types of ontology mismatches (from Klein, 2001)

classes, relations and so on." These mismatches are at "the level of the language primitives that are used to specify an ontology." At the ontology level, the mismatches are differences in the way the domain is modeled and occur when two or more ontologies describe "(partly) overlapping domains" (Klein, 2001).

Of course, Klein's classification is in some sense arbitrary. What makes his classification relevant to and useful for this thesis, though, is that there is a strong correspondence between his categories and the framing of the research question in Chapter 1 (Section 1.7). We do not repeat Klein's explanations of his categories and subcategories, but focus on one particular category of mismatch that he identifies: — a difference in the semantics of primitives, which in this thesis translates to a difference in the semantics of primitive spatial relation symbols used to model binary spatial relations.

In this thesis, a 'primitive spatial relation symbol' is a text string (e.g., 'in,' 'on,' 'through,' 'between,' 'meets,' 'overlaps,' 'connected to,' etc.) used as a predicate symbol in the logical

language used to specify the ontology. This symbol is ‘primitive’ in the following senses. First, it is treated as a single symbolic unit (a predicate symbol) in the logical language. Second, as a single unit, it is intended to capture the meaning of a basic (i.e., a primitive) relation. It ‘captures’ this meaning by means of its semantic specification in the ontology. This semantic specification is achieved through the conjunction of data and axioms in which the symbol plays some role. Third, the axioms that help specify the semantics of this symbol do not contain any other spatial relation symbols. Thus, by ‘semantics of a primitive spatial relation symbol’ we mean the semantics (i.e., the models) that result from the use of this symbol in the data and axioms of the ontology.

In describing this form of semantic heterogeneity, Klein says: “Despite the fact that sometimes the same name is used for a language construct in two languages, the semantics may differ; e.g., there are several interpretations of `A equalTo B`.”

Figure 2.3 illustrates how the general framework of this thesis (depicted in Figure 2.2) corresponds to Klein’s framework (Figure 2.1).

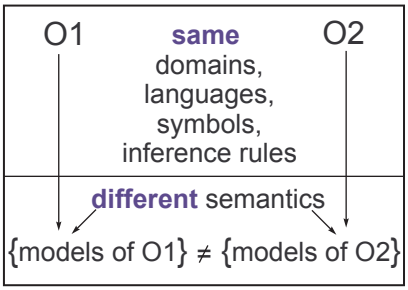

Figure 2.2: The framework assumed in this thesis

To that the ontologies use the same language is essentially to say that there are no differences in what Klein calls the language-level logical representation and language expressivity.

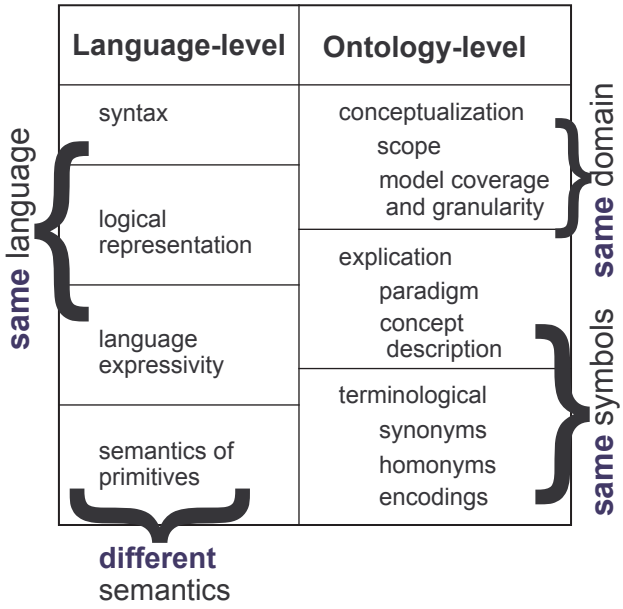

Figure 2.3: Comparison of the classification used in this thesis with Klein's

Similarly, when we say that the ontologies have the same domain, we are essentially saying that there are no differences in what Klein calls the ontology-level category of conceptualization, which has the sublevels of scope and model coverage.

Further, when we say that the ontologies *do* differ in their *semantics*, we are saying that there are differences in the language level of what Klein calls the semantics of primitives, and which we take to mean different model-theoretic semantics for the primitives. We mean by 'different model-theoretic semantics for the primitives' the different model sets that result from the different ways two ontologies specify constraints (via axioms and data tuples) on the primitive spatial-relation symbols in the ontologies.

Three points should be made here: (1) some subtleties were glossed over in making the correspondences above; and (2) one of Klein's subcategories of mismatch cannot quite be squeezed into or out of our framework: the paradigm mismatch. From what Klein says,

paradigm mismatches would seem to deal with differences in representational primitives for a given domain (such as points versus regions for the spatial domains, or points versus intervals for the temporal domain). Although we assume in this thesis that the ontologies do not differ in this regard, it is hard to state this using just our categories of domains, languages, symbols, and inference rules (Figure 2.2); and (3) Klein's classification (rightly, we would argue) does not treat possible differences in inference rules, since they are beyond the scope of his concerns.

Finally, although there can clearly be many different ways to classify semantic heterogeneities in ontologies, we are not concerned in this thesis with comparing the strengths and weaknesses of different classifications. Rather, the primary purpose of this section has been to make clear—by situating the thesis in the context of a similar framework—which pieces of our framework are assumed to be the same across ontologies and which are assumed to be different.

2.5 Semantic Interoperability: Other Views

The phrase 'semantic interoperability' does not mean the same thing to everyone across different communities. Even within the same community, the meaning of this term can vary widely. Fortunately, though, researchers use a similar approach in framing their discussions of semantic heterogeneity and semantic interoperability. After describing this approach, we provide a context for the above discussions of semantic interoperability in GIScience by presenting an overview of related work on semantic interoperability in fields other than GIScience.

2.5.1 A Common Approach

In framing their discussion of semantic heterogeneity and semantic interoperability, researchers typically do the following.

1. They narrow the scope of issues under consideration. In GIScience, for instance, one might limit one's scope to models of the geographical world that are based on fields, or to models that are based on objects. (Note: the word 'model' is used here and in the next two paragraphs in a general sense; not in the specific sense of model-theoretic semantics.)

2. They create models within that scope that are somehow comparable. So, they might create a field-based model and an object-based model for a given part of the geospatial world.

3. They identify a certain kind of semantic heterogeneity (e.g., differences in meaning that result from field-based versus object-base models of the "same" geographical reality).

4. They specify a desired kind of semantic interoperability and then try to discover to what extent the ontologies are already semantically interoperable in order to manipulate the ontologies, or force them to be semantically interoperable in a certain way. For example, a researcher might define the field-based and object-based models to be completely semantically interoperable if there is a way of converting data, queries, and query results from the field-based model to "equivalent" data, queries, and query results in the object-based model, and vice versa. Then, the researcher might develop either an algorithm

that ascertains whether the two models are already completely semantically interoperable, or an algorithm that operates on the two models and forces them to be completely semantically interoperable.

The first two of these steps are often taken jointly, by focusing on particular problems of semantic heterogeneity that arise when different information systems handle meanings differently.

There are many ways in which ontologies can be, in a broad sense, semantically heterogeneous. Among these are: (1) whether they are general or "upper-level" ontologies or lower-level, domain-specific ontologies; (2) whether they consider the most elementary components to be the same kinds of entity (e.g., points versus regions in ontologies of space); (3) whether they treat the same domains, (4) whether they treat a given domain at the same level of granularity; (5) whether they use the same ontology language to model their domain(s), and so on. Under a broad interpretation of the word 'semantics,' an ontology's semantics is affected by all of the above choices.

One can begin to understand repercussions of these choices by exploring some of the treatments and classifications of semantic heterogeneity that have been proposed in various related areas: databases, GIScience, and information/computer sciences.

2.5.2 Database Interoperability

In 1990, Sheth and Larson acknowledged that the problem of semantic heterogeneity was "poorly understood" and that there was "not even an agreement regarding a clear definition of the problem" (Sheth and Larson, 1990, pg. 187). Since then, some progress has been made in identifying certain types of semantic heterogeneity, in specifying the desired kind of

semantic interoperability, and in manipulating databases to achieve certain degrees of semantic interoperability. Yet significant problems remain.

Representative Approach: Merging Schemas. Substantial progress on treating semantic heterogeneities has been made in the area of database schemas, which specify the vocabulary and structures used to describe a given domain. Research has shown that certain semantic heterogeneities among databases that use different schemas can be resolved by matching the compatible components of their schemas and then integrating these components into a global schema (Rahm and Bernstein, 2001; Hakimpour and Geppert, 2001, 2002).

Because a database schema deals with vocabulary for talking about a given domain, the schema has a direct effect on how the database deals with meanings. To the extent that problems with schema heterogeneities are overcome, one can say that the databases have been made more semantically interoperable. But, although the kind of interoperability achieved at the schema level does enable users to employ a common vocabulary to ask and get answers to questions, it suffers from several shortcomings.

First, simply because two databases have had their schemas merged or mediated and so can be queried with the same vocabulary, does not mean that they are semantically interoperable, in the sense that their *meanings* are now compatible with each other. To suppose so is akin to supposing that because two people speak the same language, they therefore mean the same things by the words they use.

Second, semantic interoperability at a schema level is just one of several kinds of semantic interoperability that merit investigation.

Third, this kind of interoperability is not complete in the sense that it is not generally possible to automatically determine “all matches between two schemas, primarily because most

schemas have some semantics that affects the matching criteria but is not formally expressed or even documented" (Rahm and Bernstein, 2001, pg. 337).

Fourth, resolving schema heterogeneities sheds no light on how query results depend on the different semantics. That is, even if queries can now be *posed* using a common vocabulary, schema integration by itself says nothing about how the *answers* to queries are related to the semantics of the databases.

Fifth, the semantics of a conventional database is determined by the unique model of that database, whereas the semantics of an ontology (which includes axioms as well as data tuples) is determined by (most often) multiple models of the ontologies. In the case of ontologies, therefore, more is needed to achieve semantic interoperability than just a common framework for asking and answering queries. Specifically, what is needed is a detailed analysis of the differences in semantics between two ontologies, an analysis that considers both the models of each ontology and possible queries to them.

In 2006, all of these issues and more still confront researchers in semantic interoperability.

2.5.3 Information Science

The classification of ontology mismatches by Visser et al. (1998) and the framework for understanding differences between ontologies created by Klein (2001) demonstrate that, as recently as a few years ago, researchers were still trying to clarify their understandings of the different kinds of heterogeneity between ontologies. As Schorlemmer and Kalfoglou (2004) state, "Semantic interoperability and semantic integration are much contested and fuzzy concepts, which have been used over the past decade in a variety of contexts and works" (Schorlemmer and Kalfoglou, 2004, pg. 46). Recent work by the European Knowledgeweb project does

provide an "[i]ntegrated view and comparison of alignment semantics" (Hitzler et al., 2006), although at a level of abstraction that is higher than that used in this thesis.

Translational Approach to Semantic Interoperability. The notion of semantic interoperability that underlies most discussions in the research literature can be gleaned from the following quotation from Heflin and Hendler (2000): "To achieve semantic interoperability, systems must be able to exchange data in such a way that the precise meaning of the data is readily accessible and the data itself can be translated by any system into a form that it understands" Heflin and Hendler (2000, pg. 111).

In this context, achieving semantic interoperability requires that data in one system be translated to another system such that the translated meaning in the second system means "essentially the same thing" as the original meaning does in the first system. Let us agree to call this approach to semantic interoperability the "translational approach" to semantic interoperability.

Grüninger and Kopena (2005) follow this approach. Their goal is to develop technology supporting semantic integration: "two software systems can be semantically integrated through a shared understanding of the terminology in their respective ontologies" Grüninger and Kopena (2005, pg. 11).

The notion at play in the translational approaches to semantic interoperability appears to be based on the notion of isomorphism: a one-to-one and onto (hence, invertible) mapping of elements between two structures such that certain significant properties are preserved by the mapping (and its inverse). A classic kind of isomorphism is that which occurs between two graphs, where the mapping takes vertices of one graph into vertices of the second graph in such a way that the adjacency relationship that exists among vertices of the first graph is

"preserved" by the isomorphism: i.e., the mapped vertices in the second graph have the same adjacency relationships as do the original vertices in the first graph. When two graphs are isomorphic, they are "essentially the same graph" except for the labeling of their vertices.

The translational notion of semantic interoperability has benefits and drawbacks. One benefit is that it offers a specific view of when two ontologies are semantically interoperable: namely, when there exists an isomorphism between them. Another benefit is that *if* such an isomorphism exists, then the ontologies are "fully" semantically interoperable, in the sense that all relevant meaning is preserved by the isomorphism.

But along with these benefits come several drawbacks. First, there are quite a few ways to define an isomorphism between ontologies, and one needs to be careful to specify exactly the domain and range of the mapping, and exactly what properties remain preserved by this mapping. For instance, in trying to create an isomorphism between ontologies, should the mapping be between elements in the ontologies' signatures, or between the entities in the conceptual domain? And, in either case, what is it that is preserved by the mapping (isomorphism)? Is it the model classes of the ontologies?

Second, even if a careful and sensible specification could be made of the mapping, and even if it is the model classes of the ontologies that are preserved by the mapping, such a mapping reveals only "perfect matches" or full semantic interoperability. It ignores finer-grained issues, about the relationship between queries and the models of the ontologies. These finer-grained issues help explain certain relationships between model classes that govern query results (Chapter 3).

Other Approaches to Semantic Interoperability. Other approaches to defining and achieving semantic interoperability that merit discussion include Euzenat (2001) and Masolo (2000).

Euzenat (2001) takes what he calls a "principled approach" to semantic interoperability. He defines semantic interoperability as "the faculty of interpreting knowledge imported from other languages at the semantic level, i.e. to ascribe to each imported piece of knowledge the correct interpretation or set of models" (Euzenat, 2001, pg. 19). His work considers "several proposals for expressing semantic interoperability across different languages as it shall happen on the semantic web" (Euzenat, 2001, pg.22). In this thesis, by contrast, we address the more basic challenge of understanding semantic interoperability when ontologies use the *same* language but with different semantics for the primitive relation symbols.

The work by Masolo (2000) discusses set-theoretic relations between theories and models. In terms of theories, Masolo considers relations like "theory A is a subset of theory B" (where the theory of A in a language L is the set of all the sentences in L that are provable from the sentences in A). Masolo explicates his ideas using different kinds of order relations (e.g., partial orders and dense linear orders) and different kinds of mereological relations (e.g., $proper\_part\_of$). From a syntactical and proof-theoretic viewpoint, he shows how the set-theoretic relation between theories can be determined in certain special cases based on which axioms (e.g., reflexivity, transitivity) comprise the theory. So for instance, if theory A contains just the one axiom of reflexivity and theory B contains the two axioms of reflexivity and transitivity, then theory B will be a subset of theory A. That is, any L-formula that can be proved in theory B can also be proved in theory A. Masolo uses models in two ways. First, he uses the model-class relations of equals, subset, and superset to define, respectively, one theory being equal to, a subtheory of, or a supertheory of another (Masolo, 2000, pg. 123). Second, he uses models, along with a translation between the languages of two theories, to determine when two theories are "translationally equivalent" (my term, not his) (Masolo, 2000, pg. 128).

Yet, Masolo apparently does not take the further step of actually working with specific models of two theories, in order to determine the relation between the sets of models, or to comment on what might be the semantic interoperability between the two theories.

2.6 Is Spatial Special?

Is the treatment in this thesis of semantic interoperability of geospatial ontologies particularly 'spatial'? The answer is 'Yes and no,' and it points to the generalizability of this research.

On the one hand, in the actual specifications of the different conceptualizations of, for instance, the spatial relation 'in,' *spatial is special*, in that the spatial properties of the objects and relations under consideration are exactly what the human modelers who construct the specifications of their conceptualizations (i.e., their ontologies) are trying to understand, capture, and exploit. The same would be the case for other spatial relations, be they 'on,' 'through,' 'connected to,' 'between,' 'meets,' 'inside,' 'contains,' 'part,' or 'complement' (Stell, 2004), etc. The same kind of specialness would also apply to other kinds of properties, which, though not unique to spatial information, are often treated in spatial information science. Such properties include the granularity of the domain under question (Hobbs, 1985), or the vagueness, uncertainty, or imprecision often found in descriptions of geospatial phenomena (Burrough and Frank, 1996; Clementini and Di Felice, 1997).

The key principle is: whatever aspect of 'spatial' is being dealt with (e.g., granularity, processes, spatiotemporal events, topology, orientation, geometry, coordinates, vagueness, imprecision, uncertainty), if the concepts can be described in a logical language with a model-theoretic semantics, then the analysis used in this thesis applies at least in principle.

In some cases, the analysis might apply, but with appropriate limitations. Often, for example, axioms in mathematical theories are very straightforward (e.g., group theory), and the intuition of the implicitly specified models is somewhat clear. But when these axioms are used to derive logical consequences — i.e., to decide whether the mathematical structures so axiomatized have a given property — the arguments are made via proof theory rather than model theory. That is, one proves logical consequences via syntax, not semantic, and relies on the soundness of the logic involved, which says that if a result can proved syntactically, it is true semantically. Thus, where the emphasis is on proofs by syntax, the analysis used in this thesis is not particularly applicable.

In the special case of geospatial ontologies, though, if the domains can reasonably be assumed to be finite, the models can at least in principle be calculated, and the model classes of two ontologies can then in principle be compared. Note that the model-theoretic approach taken in this thesis depends for its computational results on the assumptions of finiteness of the conceptual domain and the related assumption that the number of constant symbols is the same as the number of entities in the conceptual domain. If these assumptions of finiteness are relaxed, so that the conceptual domain and the size of the non-logical vocabulary could be infinite, the *principles* of the model-theoretic analysis in this thesis remain unchanged, though the computations may be intractable. Note also that with the finiteness assumptions, a proof calculus for the logical language used could determine, *for a given query*, whether that query is a logical consequence of the ontology. If the assumptions of finiteness were relaxed, so that the language used is first-order logic with equality, the proof calculus would not in general be able to decide whether a query is a logical consequence of an ontology.

To consider the matter from another angle, suppose that *dimension* is a spatial property that varies between two geospatial ontologies. For example, suppose that in ontology A everything is being considered in two dimensions, whereas in ontology B everything is considered in three dimensions.

In this case, spatial *is* special, because the meanings of spatial relations like 'in' and 'between' can clearly be different, depending on whether one is considering a 2-dimensional or a 3-dimensional world.

On the other hand, spatial is *not* special in this thesis, because the analysis of semantic interoperability presented here does not directly depend on any differences in the spatial properties (e.g., dimension, topology, metrics, coordinatization, or orientation) of the geographic domain under consideration. Thus, even though the analysis of semantic interoperability in this thesis focuses on different conceptualizations of spatial relations—as these conceptualizations are specified in the data and axioms of the ontologies and interpreted in the models of the ontologies—the *method* of the analysis is independent of the spatial properties *per se* of the spatial relations under consideration.

That is, the fact that 'in' may capture certain topological or geometrical properties of spatial objects does not affect the method of analysis, which exploits the abstract logical framework of a formal ontology specified in a language that has a model-theoretic semantics. The relevant aspect of geospatial ontologies, as far as the analysis goes, is that they are axiomatizable in such a logical framework.

The method employed in this thesis is generalizable to other, non-spatial relations, such as 'is the parent of', or 'is employed by,' since the functioning of this method is tied to the formal specification in terms of data tuples and axioms, rather than to a given conceptual domain.

It can be the case, from the appropriate point of view, that what appear to be fundamentally spatial issues are seen to be simply particular domain-specific issues of a particular domain. That is, the axiomatization of the dimensionally dependent properties of the spatial relations ‘in’ and ‘through’ proceeds without special regard to the spatial domain. And the subsequent analysis of semantics in terms of models, and of semantic interoperability in terms of model classes (Chapter 3) would be the same for the entities and relations in the geospatial ontologies as it would be for other entities and relations in say, an employee ontology.

Chapter 3

SEMANTIC INTEROPERABILITY: MODEL CLASSES AND QUERIES

Recall from Chapter 1 that the research question is: "When two geospatial ontologies use the same language to describe the same domain, but differ in the model-theoretic semantics of their primitive spatial-relation symbols, in what sense and to what extent are the ontologies semantically interoperable?" Recall from Chapter 2 that although there is significant related research on the issue of semantic interoperability, there has been no detailed analysis of the semantic interoperability of two geospatial ontologies in which the semantics of the ontologies is specified by model-theoretic semantics. This chapter presents such an analysis.

3.1 Language, Queries, and Truth Values

As mentioned in Chapters 1 and 2, in order to specify a formal geospatial ontology one needs to specify both its axioms and its data. To specify its axioms, we use a logical language that

is described in the next section. To specify its data, we use tuples like $(Route2, Maine)$ to indicate that for a particular relation in question, say the spatial relation *in*, Route 2 is in Maine. These tuples will not be described further.

3.1.1 Logical Language

We use a language L that is a subset of the language of first-order logic (Ebbinghaus et al., 1994; Boolos et al., 2002). The vocabulary of L consists of two parts, a logical vocabulary and a non-logical vocabulary.

The *logical vocabulary* of L consists of a finite set of variables (e.g., x, y, z, etc.), the logical operators $\vee$ and $\wedge$ (corresponding roughly to natural-language 'and' and 'or', respectively), the universal quantifier ($\forall$) (read 'for all'), and the existential quantifier ($\exists$) (read 'there exists'), the negation symbol $\neg$, and the equality symbol =. Also included in L are the following grouping symbols: the comma, and the left and right parentheses.

The *non-logical vocabulary* of L consists of a finite set of constant symbols and a finite set of relation symbols (each with a finite arity). The non-logical vocabulary used in this thesis is special in the sense that the constant symbols are not only finite in number, but they equal in number the number of entities in the ontology under consideration. Further, every constant symbol in the non-logical vocabulary refers to a unique entity in the conceptual domain. This restriction on the constant symbols is made for two reasons. First, one supposes that creators of ontologies would use only as many names as there are entities under consideration. Second, by limiting the number of constant symbols to be the same as the number of entities in the conceptual domain, the number of models of a given ontology is also limited; thus, the computations comparing model classes of two ontologies can in some cases be tractable.

We make the explicit assumption that the size of the logical domain is the same size as the conceptual domain for each ontology. That is, the number of entities that can be considered in the language (via its variables, constants, and relation symbols) is the same as the number of real-world entities (e.g., Bangor, Maine, Route 2, and Orono) that exist in the domain under consideration. Thus, logical inferences are carried out on a finite domain, and this finite domain is assumed to be fixed in advance and to be the same across models within a given ontology, as well as across models between different ontologies.

The *syntactic rules* for L are the same as those for first-order logic, and they allow us to combine the symbols from the logical and non-logical vocabularies to create statements (well-formed formulas with no free variables) that are used both to describe the domain (as axioms in the ontologies) and to form queries about the domain, which will be evaluated against the ontologies.

The *semantics* of L are specified via model-theoretic semantics (Manzano, 1999; Hodges, 1997; Farrugia, 2003), which uses models (here, essentially, just sets with relations defined on them) and which assigns formal meanings to legitimate statements of L by interpreting them in set-theoretical structures (e.g., relations). See Section 3.1 for more details on the models of some sample geospatial ontologies.

For a *proof theory* associated with L, one may assume any of several equivalent proof theories of first-order logic. The proof theory of L is not directly relevant to this thesis and will not be dealt with in any detail. As will be mentioned in Chapter 6, for certain languages a proof theory could be used to calculate whether or not a given query, or a set of given queries, follows as a logical consequence of the statements in the ontology (Patel-Schneider, 2006). But this capability falls short of the kind of analysis that is needed to determine the level of

semantic interoperability between two ontologies, since that determination is made, in part, by considering whether *all* queries that are a logical consequence of one ontology are also a logical consequence of the other ontology. A proof theory for the language we use (with its restriction of finiteness on its non-logical vocabulary) can determine, for any particular query that we specify, whether that query is a logical consequence of the data and axioms of an ontology. But it cannot give us this determination for all queries that we might specify, which is what we need for our analysis.

In the examples of Chapter 1 (Figures 1.8 and 1.9), the non-logical vocabulary consists of four constant symbols ($Route2$, $Orono$, $Bangor$, and $Maine$, which denote the entities Route 2, Orono, Bangor, and Maine, respectively), and one binary spatial relation symbol in, which denotes the the binary relation 'in.' The geospatial nature of these ontologies is reflected by the facts that: (1) the entities are geographic entities; (2) the relation is a spatial relation; and (3) the axioms using the spatial relation describe certain plausible spatial properties of the relation 'in.'

The constant symbols and relations symbols of the non-logical vocabulary together comprise the *signature* of an ontology. (Whenever two ontologies are compared in this thesis, they both have the same signature; thus differences in signature are eliminated as potential sources of variability between the ontologies.) The signature, along with the syntactic rules for L, allow us to express claims (in the form of queries, i.e., statements) about the spatial configuration of these entities, e.g., which entities are in which other entities.

Certain of these claims are taken to be *axioms* of the ontologies, which are explicitly given statements that are True in all models of the ontology (see Section 2.2.2).

Other claims are put to the ontologies as *queries*, which are then evaluated against the models of the ontologies to see whether the claims are True in all, some, or no models of the ontologies.

Since the symbols of L and the queries formulated in L are interpreted via models, *any question about the meaning of the symbols of a given query is appropriately treated as a question about the models of that ontology*. Queries to these ontologies will be evaluated as True or False according to whether the state of affairs they describe does or does not hold in one or more models of the ontology. More details on queries and truth values are provided in the next section.

3.1.2 Queries and Truth Values

As noted above, statements in L can serve not only as axioms, but also as queries. For example, the statement $in(Route2, Orono)$ can be used to ask whether, according to a given ontology, Route 2 is in Orono.

In this thesis, the truth value of a query Q evaluated against an ontology O is determined by the model-theoretic semantics of L, which is essentially the same semantics used to determine whether a statement is True in first-order logic with equality (Ebbinghaus et al., 1994; Boolos et al., 2002), the only differences being due to the various assumptions of finiteness regarding L (Section 2.2.1). Thus, the constant symbols have a fixed semantics, where each constant symbol denotes the corresponding geographical object in the domain. The binary relation symbols are interpreted as binary relations on this domain. Finally, any variables in Q (which is finite and assumed to be the same size as the domain of quantification) receive variable assignments only from the elements in the geographic domain. The assumption of finiteness for

the conceptual domain (the elements in the geographic domain) makes sense, because geospatial ontologies are likely to deal with only a finite number of geospatial elements, assuming that there is no practical need, for instance, to consider time and space as domains of infinite size. The assumption that the logical domain (the domain over which variables can range) is the same size as the conceptual domain is made to remove from consideration those models containing elements that have no counterpart in the conceptual domain. Using this semantics, Q is interpreted as referring to a particular state of affairs that may or may not hold in one or more *models* of the ontology.

There are different ways to evaluate a query against an ontology, which depend among other things on the number and kinds of truth values available. A conventional database, for example, uses just the two truth values, True and False. So for instance, in the $DB1$ of Section 1.5.1, the query $in(Orono, Maine)$ (Is Orono in Maine?) would evaluate to True, since the tuple $(Orono, Maine)$ is in the database. On the other hand, the query $in(Route2, Orono)$ (Is Route 2 in Orono?) would evaluate to False, since the tuple $(Route2, Orono)$ is not in the database. This example illustrates the conventional use of the closed-world assumption (Reiter, 1978), which says that the truth value of a query that cannot be verified to be True is taken to be False. In this example, since the desired tuple is not in the database, the value of $in(Route2, Orono)$ cannot be verified to be True; thus, it is assigned a truth value of False.

On the other hand, an ontology of the kind discussed in this thesis needs, in effect, three truth values for queries, with a corresponding *open*-world assumption (Baader and Nutt, 2003): True, False, and a value that may be called 'Other.' The reason is that the situation with ontologies is complicated by the fact that, unlike databases, they generally will have more than

one model. For a query evaluated against an ontology, 'True' will mean that the query is satisfied by all models of the ontology. 'False' will mean that the query is satisfied by no models of the ontology. And 'Other' will mean that the query evaluates to True in some models of the ontology, and it evaluates to False in other models of the ontology. In all these cases, we are assuming that the ontology is consistent, i.e., that it has at least one model, so these three "values" suffice.

The model-theoretic semantics described in this thesis is of the latter type, and the conventions for naming the relevant truth values are as follows.

- $\mathcal{E}_O(Q) = (T/F)_S$ means the evaluation of query Q in ontology O yields a truth value of *True in some models of* O and a truth value of *False in other models of* O.
- $\mathcal{E}_O(Q) = T_A$ means the evaluation of query Q in ontology O yields a truth value of *True in all models of* O.
- $\mathcal{E}_O(Q) = F_A$ means the evaluation of query Q in ontology O yields a truth value of *False in all models of* O.

3.2 Similar Ontologies with Different Semantics

Figures 3.1 and 3.2 show two similar geospatial ontologies, O_1 and O_2, that deal with the spatial relation *in*. In these ontologies, the constant symbols $Route2$, $Orono$, $Bangor$, and $Maine$, refer to the real-world entities Route 2, Orono, Bangor, and Maine, respectively. The sole relation symbol is in.

Each of these two ontologies has the same two axioms. The first of these axioms says that each entity is in itself, and the second says that for any two distinct entities x and y, if x is in y, then y is not in x. The intended interpretation of the tuples are the obvious ones, e.g., $(Route2, Maine)$ is intended to mean that Route 2 is in Maine.

In addition to using the same axioms, O_1 and O_2 have three data tuples in common. Thus the two ontologies appear, at first glance, to be very similar, since they differ only in their fourth data tuple. Yet in terms of their semantics they are considerably different. The differences in semantics are revealed by the differences in the *models* of each ontology.

Axioms for in	Data Tuples for in
$\forall x, in(x, x)$ $\forall xy\ (x \neq y) \rightarrow (in(x, y) \rightarrow \neg in(y, x))$	$(Route2, Orono)$ $(Route2, Bangor)$ $(Route2, Maine)$ $(Bangor, Maine)$

Figure 3.1: Ontology O_1 for in

Axioms for in	Data Tuples for in
$\forall x, in(x, x)$ $\forall xy\ (x \neq y) \rightarrow (in(x, y) \rightarrow \neg in(y, x))$	$(Route2, Orono)$ $(Route2, Bangor)$ $(Route2, Maine)$ $(Orono, Maine)$

Figure 3.2: Ontology O_2 for in

The models of O_1 and O_2 are specified using the abbreviations of R2 for $Route2$, O for $Orono$, B for $Bangor$, and M for $Maine$. A *model* for one of these ontologies consists of (1) the individual elements of the domain, and (2) the tuples (ordered pairs in this case) that are in the binary relation in. As noted in Chapter 1, a convenient way to picture models is as graphs (e.g., Figure 3.3 and Figure 3.4).

For a given ontology, any such collection that contains the data tuples and that does not violate any of the constraints imposed by the axioms is a *model* of that ontology. We recall for the reader two key assumptions discussed in Chapter 2, Section 2.2.1. First, each ontology considers the same finite number of real-world entities. Second, the number of available referents for the constant symbols of the ontologies is the same as this finite number of real-world entities.

Any model of O_1 must contain the following elements:

- R2, O, B, and M (because these are elements of the domain)
- (R2,R2), (O,O), (B,B), (M,M) (because of axiom 1)
- (R2,O), (R2,B), (R2, M), and (B,M) (because these are tuples in the data)

Further, no model of O_1 can contain any of the following tuples: (O,R2), (B,R2), (M,R2), or (M,B), because axiom 2, along with the tuples in the data, eliminates these tuples as candidates in any model.

Finally, a model of O_1 can contain: either (O,B) or (B,O) but not both; and either (M,O) or (O,M) but not both.

The models of O_1 are essentially just a set with a single binary relation defined on them. The 9 models of O_1 are depicted as graphs in Figure 3.3.

Similarly, any model of O_2 must contain the following elements:

- R2, O, B, and M (because these are elements of the domain)
- (R2,R2), (O,O), (B,B), (M,M) (because of axiom 1)
- (R2,O), (R2,B), (R2, M), and (O,M) (because these are tuples in the data)

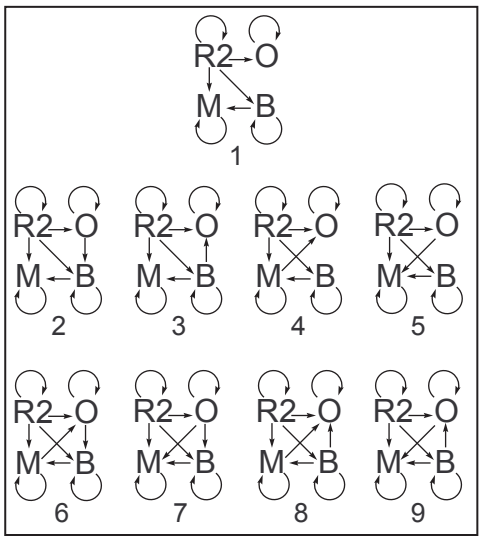

Figure 3.3: The 9 models of ontology O_1

Further, no model of O_2 can contain any of the following tuples: (O,R2), (B,R2), (M,R2), or (M,O), because axiom 2, along with the tuples in the data, eliminates these tuples as candidates in any model.

Finally, a model of O_2 can contain: either (O,B) or (B,O) but not both; and either (M,B) or (B,M) but not both.

Thus, there are only 9 models of O_2, and they, too, can be depicted as graphs (Figure 3.4).

The *model class* of an ontology O, $\mathcal{MC}(O)$, is the set of models of that ontology.

We can list the nine models in the model class of O_1 as $\mathcal{MC}(O_1)$ = { $M1_{O_1}$, $M2_{O_1}$, $M3_{O_1}$, $M4_{O_1}$, $M5_{O_1}$, $M6_{O_1}$, $M7_{O_1}$ $M8_{O_1}$, $M9_{O_1}$ }.

Similarly, the model class of O_2 is $\mathcal{MC}(O_2)$ = { $M1_{O_2}$, $M2_{O_2}$, $M3_{O_2}$, $M4_{O_2}$, $M5_{O_2}$, $M6_{O_2}$, $M7_{O_2}$ $M8_{O_2}$, $M9_{O_2}$ }.

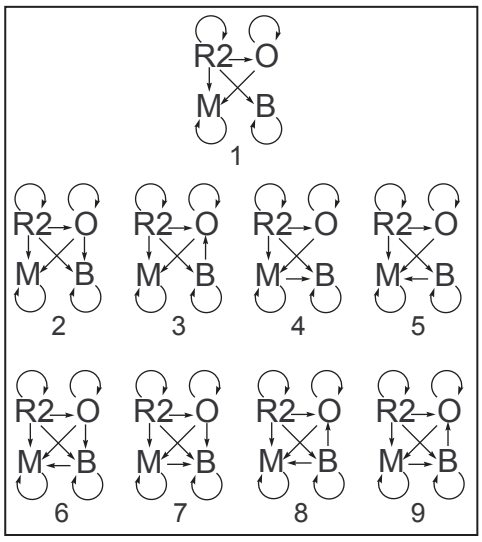

Figure 3.4: The 9 models of ontology O_2

The semantics of the ontology essentially *is* the model class of that ontology. In other words, if you accept the axioms and data tuples of an ontology, then the semantics that you are committing yourself to is revealed precisely in the model class of that ontology.

As Figures 3.3 and 3.4 demonstrate, an ontology can have more than one model. Thus, *when you commit yourself to a given ontology and its semantics (as revealed through its collection of models), you are generally committing yourself to more than one fixed possible way that the world might be.*

This fact helps explain why defining semantic interoperability of ontologies is so challenging. Since ontologies generally have multiple models, and there are different relationships that could hold between their model classes, the analysis of semantic interoperability must consider not only the multiple models in each model class but also the different possible relationships between model classes. Different possible relationships between model classes form the basis of our definition (Section 3.3) of different 'levels' of semantic interoperability.

Because the specification of ontologies O_1 and O_2 are very similar, one might expect that these ontologies are also 'very' semantically interoperable. After all, they refer to the same domain, they use the same symbols, they have the same axioms, and they even have three of their four data tuples in common. Yet, when one considers the differences between the ontologies in terms of their different semantics (as revealed through their model classes), one can see that the ontologies are not 'very' semantically interoperable, at all. A precise characterization, in terms of models and queries, of the semantic interoperability of these ontologies is given in Section 3.3.

Given two ontologies with similar specifications but possibly substantially different model classes, one wants to be able to assess the degree to which the ontologies are 'semantically interoperable.' Before doing this, though, we explain semantic interoperability in terms of model classes *and* the queries that we put to them. In the next section we explain this viewpoint, using ontologies O_1 and O_2 as examples.

3.3 Semantic Interoperability

The phrase 'semantic interoperability,' whatever other meanings it might take on, should, intuitively, have something to do with 'working between meanings' or 'working together with meanings,' or 'working together meaningfully.' One can extract from these intuitive notions the idea that if two ontologies are semantically interoperable, then their meanings somehow work together with, 'get along with,' or are compatible with each other. Since the semantic framework used in this thesis is model-theoretic semantics, and since 'semantic' in this framework practically refers to *model classes*, the 'semantic interoperability' of two ontologies must

have something to do with how the ontologies' *model classes* work together. We describe this 'working together' in terms of 'compatible' query results.

3.3.1 Intuition: Compatible Query Results

How well the model classes of two ontologies work together is determined by the results of queries to the ontologies, because the queries are the devices that probe the structures of the model classes for similarities and differences. The following analogy may be helpful.

This thesis analyzes pairs of ontologies with much in common: their domain, their logical language, their non-logical vocabulary, etc. These common elements are like the common tools and building materials that each ontology creator can use to build ontologies. The two different ontologies are thus at the outset perfectly compatible with each other *insofar as their tools and building materials are concerned*. Where they differ, and where they may be incompatible, is in the actual buildings (i.e., the models) that are constructed using the available tools. So, any potential semantic *in*compatibility between ontologies lies in the differences between the actual buildings built, not in the tools or materials used to build them. Or, looked at in a different way, the possible semantic *interoperability* between the ontologies lies in the 'compatibility' of the buildings (the models) of each ontology. Some way to define and assess this compatibility is needed; this is where queries come in.

Queries help define and assess the compatibility of the model classes of two ontologies. In a sense, a query is like a probe that is sent to a city of buildings (a model class) and sends an answer of True or False, depending on what it finds there. The answer returned by the probe may be True or False in some, none, or all of the buildings (Section 3.1.2). The query, or probe, has accesses to the entire state of affairs that are explicitly or implicitly specified by the

collection of models of an ontology, and the result of evaluating a query in this state of affairs may be True or False in some, none, or all of the models.

Thus, the question of to what extent two ontologies are semantically interoperable is in essence a question of the degree to which their model classes 'work together,' which in turn reduces to a question of whether the ontologies exhibit *compatible query results.* To explain exactly what we mean by 'compatible query results,' we need to discuss the different ways a query put to an ontology can evaluate to True or False.

3.3.2 Evaluating Queries in Ontologies

Recall from Section 3.1.1 that a 'query,' as used in this thesis, is statement in the logical language used to specify the ontology (i.e., a well-formed formula with no free variables). Further recall that the ontologies under consideration are assumed to be consistent (i.e., they each have at least one model).

When a query is evaluated against an ontology O, the query could evaluate to:

- True in some model(s) of O, and false in others O;
- True in all models of O;
- False in all models of O;

For convenience in discussing truth values we provide the following notation.

- $\mathcal{E}_O(Q) = (T/F)_S$ means that the evaluation of query Q against ontology O yields a truth value of *True in some models of* O *and False in other models* of O.

- $\mathcal{E}_O(Q) = T_A$ means that the evaluation of query Q against ontology O yields a truth value of *True in all models* of O.

- $\mathcal{E}_O(Q) = F_A$ means that the evaluation of query Q against ontology O yields a truth value of *False in all models* of O.

3.3.3 Possible Query Results

To begin to understand the notion of 'compatible query results,' and to gain some idea of the relationship among queries, models, and semantic interoperability, consider Figure 3.5 below, where the second column, $\mathcal{E}_{O_1}(Q_i)$, refers to the evaluation in Ontology 1 of query i, and the third column, $\mathcal{E}_{O_2}(Q_i)$, refers to the evaluation in Ontology 2 of query i.

Query	$\mathcal{E}_{O_1}(Q_i)$	$\mathcal{E}_{O_2}(Q_i)$	Compatibility Condition
Q_1: $\neg in(Orono, Bangor)$	$(T/F)_S$	$(T/F)_S$	
Q_2: $in(Orono, Maine)$	$(T/F)_S$	T_A	
Q_3: $\neg in(Orono, Maine)$	$(T/F)_S$	F_A	
Q_4: $in(Bangor, Maine)$	T_A	$(T/F)_S$	
Q_5: $in(Route2, Orono)$	T_A	T_A	
Q_6: ***	T_A	F_A	1
Q_7: $\neg(Bangor, Maine)$	F_A	$(T/F)_S$	
Q_8: ***	F_A	T_A	2
Q_9: $\neg in(Orono, Orono)$	F_A	F_A	

Figure 3.5: Sample Queries and Results for O_1 and O_2

This table shows some sample queries to the two example ontologies O_1 and O_2. A query put to either ontology will evaluate to one of the three truth values $(T/F)_S$, T_A, or F_A. Thus, the analysis of compatible query results will center on a consideration of the nine possible combinations of query results for a single query put to two ontologies.

For a given pair of ontologies, it may or may not be possible to find example queries for each of the nine possibilities. In the case of the ontologies from Figures 3.1 and 3.2, for

instance, the asterisks in the first column of Figure 3.5 indicate that one cannot find queries for a Q_6 or a Q_8 that would result in the indicated pairs of truth values. (This fact is proved in Section 3.4.2).

Though it may sound odd, it is precisely on the basis of certain *im*possible combinations of query results that different levels of *compatible* query results are defined. That is, the fact that one *cannot* find queries that correspond to the asterisked rows in Figure 3.5 actually demonstrates a certain minimum level of *compatibility* of query results. To see why, consider the following argument.

The asterisks in row 6 of Figure 3.5 indicate that no query evaluates to True in all models of O_1 but to False in all models of O_2. Similarly, the asterisks in row 8 indicate that no query evaluates to False in all models of O_1 but to True in all models of O_2.

Intuitively, one might say that it is never the case that, loosely speaking, 'white' in O_1 means 'black' in O_2, or vice versa. These two conditions, taken together, constitute the minimal requirements for compatible query results (and of semantically interoperable ontologies), because without them, it is possible to find some query that returns 'opposite' (i.e., contradictory) results for the two ontologies. Because no such query can be found for O_1 and O_2, these two compatibility conditions rule out the grossest kind of incompatibility of query results, where some query could generate diametrically opposite results in the two ontologies. In terms of the analogy of sending a probe to a city of buildings, we rule out the possibility that whatever the probe is looking for, it will find it in all buildings of one city and in no buildings of the other city.

The above result — that for a query Q, we cannot have $\mathcal{E}_{O_1}(Q) = T_A$ and $\mathcal{E}_{O_2}(Q) = F_A$, or $\mathcal{E}_{O_1}(Q) = F_A$ and $\mathcal{E}_{O_2}(Q) = T_A$ — holds for *any* query that can be formulated in the language

L (Section 3.1.1). As such, it guarantees a minimal level of compatible query results for *any* query expressible in L, not just for some queries we think we may be interested in at a particular time. The importance of this fact is that if we cannot in advance think of all the queries we may want to ask, we are nevertheless 'safe' if the two compatibility conditions hold, in the sense that there cannot be contradictory results from the same query to the two ontologies.

This minimal level of compatibility of query results, in turn, is the basis for the definition of *Level 1 Semantic Interoperability* between ontologies ($L1_{SI}$), and we refer to the two ontologies as being Level 1 Semantically Interoperable ($L1_{SI}$).

The $L1_{SI}$ compatibility conditions, along with similar conditions given below, function in sort of a 'double negative' sense: a certain degree of semantic *interoperability* is achieved by *removing from consideration* particular combinations of query results that would indicate a *lack* of such interoperability.

3.4 Levels of Semantic Interoperability

The above discussion describes the level of semantic interoperability between example ontologies O_1 and O_2 from Figures 3.1 and 3.2. This section gives explicit definitions of this level and 4 other levels of semantic interoperability defined in our framework. We present these 5 levels 'out of order,' to aid the exposition. A final subsection gives a summary of all five levels of semantic interoperability.

3.4.1 Level 1 Semantic Interoperability

Definition 1 An ontology O_i is *Level 1 Semantically Interoperable* ($L1_{SI}$) with another ontology O_j if and only if O_i and O_j meet the following two compatibility conditions:

- it is not possible to have a query that evaluates to True in all models of O_i but to False in all models of O_j (compatibility condition 1);
- it is not possible to have a query that evaluates to False in all models of O_i but to True in all models of O_j (compatibility condition 2).

The discussion in Section 3.3.3 shows that ontology O_1 of Figure 3.1 is $L1_{SI}$ with ontology O_2 of Figure 3.2. This result follows immediately from the above definition, which also shows that ontology O_2 is $L1_{SI}$ with ontology O_1 (reading the truth values from right to left instead of left to right). Thus, we have also shown that the relation of $L1_{SI}$ is symmetric. A detailed demonstration will be given Section 3.5 of why both conditions of the definition hold for each of the ontologies O_1 and O_2. This demonstration is based on the relationship between the model classes of the ontologies.

So, $L1_{SI}$ is the minimum level of semantic interoperability that we define. What is the maximum level?

3.4.2 Level 3 Semantic Interoperability

Consider ontologies O_3 and O_4 in Figures 3.6 and 3.7. These ontologies have just two entities, Bangor and Maine, and use one relation, in. Although the ontologies themselves differ in their syntactical specification (i.e., they do not have identical sets of axioms and data tuples), their model classes are the same set, which contains two models $M1$ and $M2$, where, using the

obvious abbreviations, $M1$ = $\{\{B, M\}, \{in(B, B), in(B, M), \neg in(M, B), in(M, M)\}\}$, and $M2$ = $\{\{B, M\}, \{in(B, B), in(B, M), in(M, B), in(M, M)\}\}$.

Axioms for in	Data Tuples for in
$in(x, x)$ $\forall xy\ (\neg(x = y) \wedge \neg in(x, y)) \rightarrow in(y, x)$	$(Bangor, Maine)$

Figure 3.6: Ontology O_3 for in

Axioms for in	Data Tuples for in
$\forall xy\ \neg in(x, y) \rightarrow in(y, x)$	$(Bangor, Maine)$

Figure 3.7: Ontology O_4 for in

Consider a table of possible query results for O_3 and O_4. Figure 3.8 shows that no queries can be found that correspond to what would be queries Q_2, Q_3, Q_4, Q_6, Q_7, or Q_8.

Query	$\mathcal{E}_{O_3}(Q_i)$	$\mathcal{E}_{O_4}(Q_i)$	Compatibility Condition
Q_1: $in(Maine, Bangor)$	$(T/F)_S$	$(T/F)_S$	—
Q_2: ***	$(T/F)_S$	T_A	5
Q_3: ***	$(T/F)_S$	F_A	6
Q_4: ***	T_A	$(T/F)_S$	3
Q_5: $in(Bangor, Bangor)$	T_A	T_A	—
Q_6: ***	T_A	F_A	1
Q_7: ***	F_A	$(T/F)_S$	4
Q_8: ***	F_A	T_A	2
Q_9: $\neg in(Maine, Maine)$	F_A	F_A	—

Figure 3.8: Sample Queries and Results for O_3 and O_4

For queries Q_1, Q_5, and Q_9, there is no compatibility condition given. The reason is that for each of these queries, it makes no sense to connect the impossibility of query results to a compatibility condition. Such a compatibility condition would say that no query corresponding to a Q_5 can evaluate to True in all models of O_3 and to True in all models of O_4. But this condition says nothing of positive value about semantic interoperability, since it says that no query is entailed by both ontologies.

On the other hand, consider compatibility condition 5. This condition says that no query can be found that evaluates to True in some models of O_3 and to False in other models of O_3, but that evaluates to True in all models of O_4. This makes sense, because, since O_3 and O_4 have the same model class, if a query evaluated to False in some models of O_3, it would have to evaluate to False in some models of O_4; that is, it could not evaluate to True in all models of O_4.

Similar reasoning shows that for O_3 and O_4, the six compatibility conditions are all met. This fact leads us to the next definition.

Definition 2 An ontology O_i is *Level 3 Semantically Interoperable* ($L3_{SI}$) with another ontology O_j if and only if O_i and O_j meet the following six compatibility conditions:

- it is not possible to have a query that evaluates to True in some models of O_i and to False in other models of O_i, but to True in all models of O_j (compatibility condition 5);
- it is not possible to have a query that evaluates to True in some models of O_i and False in other models of O_i, but to False in all models of O_j (compatibility condition 6);
- it is not possible to have a query that evaluates to True in all models of O_i but to True in some models of O_j and to False in other models of O_j) (compatibility condition 3);
- it is not possible to have a query that evaluates to True in all models of O_i but to False in all models of O_j (compatibility condition 1);
- it is not possible to have a query that evaluates to False in all models of O_i, but to True in some models of O_j and False in other models of O_j (compatibility condition 4).

- it is not possible to have a query that evaluates to False in all models of O_i but to True in all models of O_j (compatibility condition 2).

Intuitively, two ontologies are completely semantically interoperable when 'the same queries give the same results' for all queries and for all models. This requirement is fulfilled when the six compatibility conditions are met. These six compatibility conditions are met when the two ontologies have identical model classes (Section 3.4.2).

As is the case with $L1_{SI}$, the $L3_{SI}$ relation is also symmetric.

Three additional levels of semantic interoperability remain to be dealt with: two intermediate levels of semantic interoperability, as well as a level that indicates no semantic interoperability (in the framework we have established).

3.4.3 Level 2A Semantic Interoperability

Consider the ontologies O_5 and O_6 in Figures 3.9 and 3.10. Note the similarity with ontologies O_1 and O_2. Ontology O_5 is just ontology O_1 minus the data tuple (*Bangor*, *Maine*), while ontology O_6 is the same as ontology O_2. These similarities are exploited to ease the exposition. The critical feature of these ontologies is, as before, the relationship between their model classes.

Axioms for in	Data Tuples for in
$\forall x, in(x,x)$ $\forall xy\ (x \neq y) \rightarrow (in(x,y) \rightarrow \neg in(y,x))$	$(Route2, Orono)$ $(Route2, Bangor)$ $(Route2, Maine)$

Figure 3.9: Ontology O_5 for in

Axioms for in	Data Tuples for in
$\forall x,\ in(x,x)$ $\forall xy\ (x \neq y) \rightarrow (in(x,y) \rightarrow \neg in(y,x))$	$(Route2, Orono)$ $(Route2, Bangor)$ $(Route2, Maine)$ $(Orono, Maine)$

Figure 3.10: Ontology O_6 for in

The table of possible query results for these two ontologies is given in Figure 3.11. In addition to meeting compatibility conditions 1 and 2, ontologies O_5 and O_6 also meet compatibility conditions 3 and 4 (proved in Section 3.4.2). But O_5 and O_6 do not meet compatibility conditions 5 and 6. This latter fact is demonstrated by the queries Q_2 and Q_3 in Figure 3.11.

Query	$\mathcal{E}_{O_5}(Q_i)$	$\mathcal{E}_{O_6}(Q_i)$	Compatibility Condition
Q_1: $\neg in(Orono, Bangor)$	$(T/F)_S$	$(T/F)_S$	—
Q_2: $in(Orono, Maine)$	$(T/F)_S$	T_A	
Q_3: $\neg in(Orono, Maine)$	$(T/F)_S$	F_A	
Q_4: ***	T_A	$(T/F)_S$	3
Q_5: $in(Route2, Orono)$	T_A	T_A	—
Q_6: ***	T_A	F_A	1
Q_7: ***	F_A	$(T/F)_S$	4
Q_8: ***	F_A	T_A	2
Q_9: $\neg in(Orono, Orono)$	F_A	F_A	—

Figure 3.11: Sample Queries and Results for O_5 and O_6

It turns out that the model class of O_5 contains 27 models, and the model class of O_6 contains 9 models. What's more, the model class of O_5 contains the model class of O_6 as a proper subset.

Definition 3 An ontology O_i is *Level 2A Semantically Interoperable* ($L2A_{SI}$) with another ontology O_j if and only if O_i and O_j meet the following four compatibility conditions:

- it is not possible to have a query that evaluates to True in all models of O_i, but to True in some models of O_j and to False in other models of O_j (compatibility condition 3);

- it is not possible to have a query that evaluates to True in all models of O_i but to False in all models of O_j (compatibility condition 1);
- it is not possible to have a query that evaluates to False in all models of O_i, but to True in some models of O_j and to False in other models of O_j (compatibility condition 4);
- it is not possible to have a query that evaluates to False in all models of O_i but to True in all models of O_j (compatibility condition 2).

The $L2A_{SI}$ relation between two ontologies is antisymmetric; that is, if $O_1 L2A_{SI} O_2$, then the only way that we can have $O_2 L2A_{SI} O_1$ is if the model classes of O_1 and O_2 are the same (based on results in Section 3.4.2).

A similar, but 'reversed' situation, is obtained by reversing the roles of ontologies O_5 and O_6.

3.4.4 Level 2B Semantic Interoperability

Consider the ontologies O_7 and O_8 in Figures 3.12 and 3.13.

Axioms for in	Data Tuples for in
$\forall x, in(x,x)$ $\forall xy\ (x \neq y) \rightarrow (in(x,y) \rightarrow \neg in(y,x))$	$(Route2, Orono)$ $(Route2, Bangor)$ $(Route2, Maine)$ $(Orono, Maine)$

Figure 3.12: Ontology O_7 for in

The table of possible query results for these two ontologies is given in Figure 3.14. In addition to meeting compatibility conditions 1 and 2, ontologies O_7 and O_8 also meet compatibility conditions 5 and 6. But O_7 and O_8 do not meet compatibility conditions 3 and 4. This latter fact is demonstrated by the existence of queries Q_4 and Q_7 in Figure 3.14.

Axioms for in	Data Tuples for in
$\forall x,\ in(x,x)$	$(Route2, Orono)$
$\forall xy\ (x \neq y) \rightarrow (in(x,y) \rightarrow \neg in(y,x))$	$(Route2, Bangor)$
	$(Route2, Maine)$

Figure 3.13: Ontology O_8 for *in*

Query	$\mathcal{E}_{O_7}(Q_i)$	$\mathcal{E}_{O_8}(Q_i)$	Compatibility Condition
Q_1: $\neg in(Orono, Bangor)$	$(T/F)_S$	$(T/F)_S$	—
Q_2: ***	$(T/F)_S$	T_A	5
Q_3: ***	$(T/F)_S$	F_A	6
Q_4: $in(Orono, Maine)$	T_A	$(T/F)_S$	
Q_5: $in(Route2, Orono)$	T_A	T_A	—
Q_6: ***	T_A	F_A	1
Q_7: $\neg in(Orono, Maine)$	F_A	$(T/F)_S$	
Q_8: ***	F_A	T_A	2
Q_9: $\neg in(Orono, Orono)$	F_A	F_A	—

Figure 3.14: Sample Queries and Results for O_7 and O_8

The model class of O_7 contains 9 models, and the model class of O_8 contains 27 models. In addition, the model class of O_7 is a proper subset of the model class of O_8.

Definition 4 An ontology O_i is *Level 2B Semantically Interoperable* ($L2B_{SI}$) with another ontology O_j if and only if O_i and O_j meet the following four compatibility conditions:

- it is not possible to have a query that evaluates to True in some models of O_i and to False in other models of O_i, but to True in all models of O_j (compatibility condition 5);
- it is not possible to have a query that evaluates to True in some models of O_i and to False in other models of O_i, but to False in all models of O_j (compatibility condition 6);
- it is not possible to have a query that evaluates to True in all models of O_i but to False in all models of O_j (compatibility condition 1);

- it is not possible to have a query that evaluates to False in all models of O_i but to True in all models of O_j (compatibility condition 2).

As the case with $L2A_{SI}$, the $L2B_{SI}$ relation between two ontologies is antisymmetric; that is, if $O_1 L2B_{SI} O_2$, then it is not the case that $O_2 L2B_{SI} O_1$, unless the model classes of the two ontologies are the same.

3.4.5 Level 0 Semantic Interoperability

Consider the sample ontologies O_9 and O_{10} and the table of possible query results in Figures 3.15 through 3.17. One can verify by inspection that the model classes for O_9 and O_{10} are disjoint, since all models of O_9 satisfy the axiom $in(x, x)$ (i.e., every entity is in itself), whereas all models of O_{10} satisfy $\neg in(x, x)$ (i.e., no entity is in itself). Thus no model of O_9 can be a model of O_{10}, and no model of O_{10} can be a model of O_9.

These two ontologies meet none of the 6 compatibility conditions, as can be seen by the fact (Figure 3.17) that there do exist queries corresponding to Q_2, Q_3, Q_4, Q_6, Q_7, and Q_8.

That is, for these two ontologies one cannot restrict any undesirable combinations of query results, which would in turn result in some level of semantic interoperability. This fact leads to the following definition.

Definition 5 An ontology O_i is *Level 0 Semantically Interoperable* ($L0_{SI}$) with another ontology O_j if and only if *none* of six compatibility conditions given in Definition 2 holds for O_i and O_j.

Axioms for in	Data Tuples for in
$\forall x,\ in(x,x)$	$(Route2, Orono)$
$\forall xy\ (x \neq y) \rightarrow (in(x,y) \rightarrow \neg in(y,x))$	$(Orono, Maine)$

Figure 3.15: Ontology O_9 for in

Axioms for in	Data Tuples for in
$\forall x,\ \neg in(x,x)$	$(Route2, Orono)$
$\forall xy\ (x \neq y) \rightarrow (in(x,y) \rightarrow \neg in(y,x))$	$(Orono, Maine)$

Figure 3.16: Ontology O_{10} for *in*

3.4.6 Summary: The 5 Levels of Semantic Interoperability

Figure 3.18 illustrates the 5 levels of semantic interoperability. Each node in the figure shows the level of semantic interoperability, the compatibility conditions met at that level, and the symmetry or antisymmetry of the relation. The figure shows clearly that while Level 1 is the lowest level and Level 3 is the highest level, there is no unique 'middle level' between these two levels. In fact, both Levels 2A and 2B are situated in some sense at a middle level. The figure also shows the relationships between levels in terms of compatibility conditions.

It may be asked why only six compatibility conditions have been defined. The answer has to do with the discriminatory power of the nine possible query results that are defined in our framework. The reason there are six, and not more, compatibility conditions is that the pairs of query results corresponding to queries of type Q_1, Q_5, and Q_9 provide no constraints on the relationship of the model class of the ontologies (see Section 3.4.2). Further, there are no fewer than six compatibility conditions, because each of the six conditions given provides some insight (because it provides a constraint on possibilities) into the relation that can hold between the model classes of the ontologies.

Query	$\mathcal{E}_{O_9}(Q_i)$	$\mathcal{E}_{O_{10}}(Q_i)$	Compatibility Condition
Q_1: $in(Route2, Maine)$	$(T/F)_S$	$(T/F)_S$	—
Q_2: $in(Bangor, Maine)$	$(T/F)_S$	T_A	
Q_3: $\neg in(Bangor, Maine)$	$(T/F)_S$	F_A	
Q_4: $in(Route2, Orono)$	T_A	$(T/F)_S$	
Q_5: $in(Orono, Maine)$	T_A	T_A	—
Q_6: $in(Orono, Orono)$	T_A	F_A	
Q_7: $\neg in(Route2, Orono)$	F_A	$(T/F)_S$	
Q_8: $\neg in(Orono, Orono)$	F_A	T_A	
Q_9: $\neg in(Orono, Maine)$	F_A	F_A	—

Figure 3.17: Sample Queries and Results for O_9 and O_{10}

It may also be asked what is special about the five levels of semantic interoperability shown in Figure 3.18. Certainly, one is free to declare as many or as few levels of semantic interoperability as one wishes. In the framework for this thesis, however, these five levels are special for three reasons. First, that they correspond exactly to the five relations that exist between the model-classes of the ontologies (Section 3.5). Second, these five model-class relations are pairwise disjoint and mutually exhaustive, as can be seen from their definitions in Section 3.5.1. Thus, the five levels of semantic interoperability correspond to a partition of the possible model-class relations. Third, the five levels of semantic interoperability correspond to the six compatibility conditions in a hierarchical way that reflects the underlying connections between query results and model-class relations (Figures 3.18 and 1.9).

3.5 Proving Semantic Interoperability

Section 3.3 made several claims about certain example ontologies satisfying particular compatibility conditions (Figure 3.19).

These claims rested on intuitive arguments about the impossibility of certain query results, given a particular relationship between the model classes of the ontologies in question. In

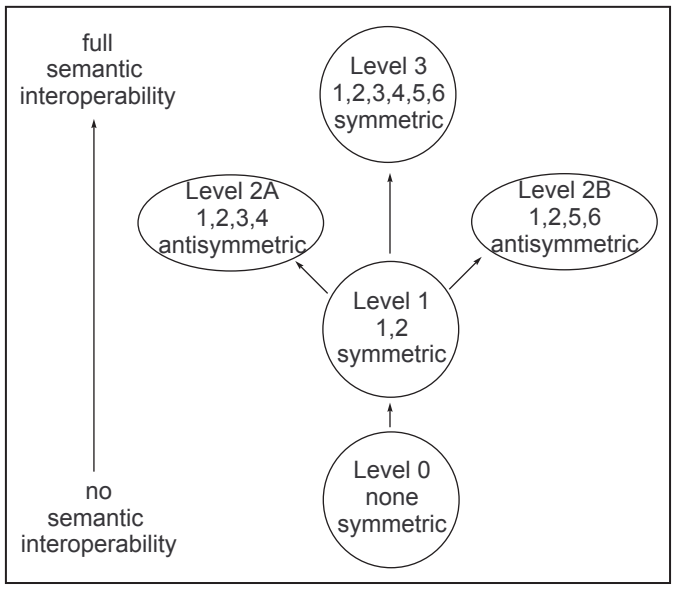

Figure 3.18: 5 levels of semantic interoperability, and compatibility conditions

Model Class Relation	Compatibility Conditions
Disjoint	
Overlap	1,2
Contains	1,2,3,4
Contained by	1,2,5,6
Equals	1,2,3,4,5,6

Figure 3.19: Relation Between Model Classes and Compatibility Conditions

this section, we state these claims more precisely, in their general form, and we prove each of them. Before proving these claims, we state more precisely what we mean by each of the five model-class relations.

3.5.1 Definitions

Disjoint. Two model classes $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$ are *Disjoint* if and only if $\mathcal{MC}_{O_i} \cap \mathcal{MC}_{O_j} = \emptyset$.

Overlap. Two model classes $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$ *Overlap* if and only if they meet the following three conditions: (1) they contain at least one model in common; (2) there is at

least one model in $\mathcal{MC}_{O_i}$ that is not in $\mathcal{MC}_{O_j}$; and (3) there is at least one model in $\mathcal{MC}_{O_j}$ that is not in $\mathcal{MC}_{O_j}$.

Contains. The model class $\mathcal{MC}_{O_i}$ *Contains* the model class $\mathcal{MC}_{O_j}$ if and only if $\mathcal{MC}_{O_j} \subset \mathcal{MC}_{O_i}$, that is, if and only if $\mathcal{MC}_{O_j}$ is a proper subset of $\mathcal{MC}_{O_i}$.

Equal. Two model classes $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$ are *Equal* if and only if $\mathcal{MC}_{O_i} = \mathcal{MC}_{O_j}$, that is, if and only if they are equal as sets.

Contained by. The model class $\mathcal{MC}_{O_i}$ is *Contained by* the model class $\mathcal{MC}_{O_j}$ if and only if $\mathcal{MC}_{O_i} \subset \mathcal{MC}_{O_j}$, that is, if and only if $\mathcal{MC}_{O_i}$ is a proper subset of $\mathcal{MC}_{O_j}$.

3.5.2 Proofs

All of the proofs below deal with ontologies O_i and O_j and with their respective model classes, $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$, which are assumed to be non-empty.

Claim 1: If $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$ *overlap*, then the ontologies O_i and O_j meet compatibility conditions 1 and 2.

Proof: Because $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$ overlap, they have at least one model in common. Therefore if a query Q evaluates to True in all models of O_i, it also evaluates to True in at least one model that is common to $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$. Thus it is not possible that Q evaluates to False in all models of O_j. Hence, ontologies O_i and O_j meet compatibility condition 1.

Similarly, if a query Q evaluates to False in all models of O_i, it also evaluates to False in at least one model that is common to $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$. Thus it is not possible that Q evaluates to True in all models of O_j. Hence, ontologies O_i and O_j meet compatibility condition 2.

Claim 2: If $\mathcal{MC}_{O_i}$ *contains* $\mathcal{MC}_{O_j}$, then the ontologies meet compatibility conditions 1, 2, 3, and 4.

Proof: Because $\mathcal{MC}_{O_i}$ contains $\mathcal{MC}_{O_j}$ there is at least one model in $\mathcal{MC}_{O_j}$ that is also in $\mathcal{MC}_{O_i}$. Thus, by the arguments used in Proof 1, the ontologies O_i and O_j meet compatibility conditions 1 and 2.

Suppose that a query Q evaluates to True in all models of $\mathcal{MC}_{O_i}$. Then, because $\mathcal{MC}_{O_j}$ is a subset of $\mathcal{MC}_{O_i}$, it is not possible that Q evaluates to False in any model of $\mathcal{MC}_{O_j}$. Thus, O_i and O_j meet compatibility condition 3.

Similarly, suppose that a query Q evaluates to False in all models of $\mathcal{MC}_{O_i}$. Then, because $\mathcal{MC}_{O_j}$ is a subset of $\mathcal{MC}_{O_i}$, it is not possible that Q evaluates to True in any model of $\mathcal{MC}_{O_j}$. Thus, O_i and O_j meet compatibility condition 4.

Claim 3: If $\mathcal{MC}_{O_i}$ is *contained by* $\mathcal{MC}_{O_j}$, then the ontologies O_i and O_j meet compatibility conditions 1, 2, 5, and 6.

Proof: Because $\mathcal{MC}_{O_j}$ contains $\mathcal{MC}_{O_i}$ there is at least model in $\mathcal{MC}_{O_i}$ that is also in $\mathcal{MC}O_j$. Thus, by the arguments used in Proof 1, the ontologies O_i and O_j meet compatibility conditions 1 and 2.

Suppose that a query Q evaluates to True in all models of $\mathcal{MC}_{O_j}$. Then, because $\mathcal{MC}_{O_i}$ is a subset of $\mathcal{MC}_{O_j}$, it is not possible that Q evaluates to False in any model of $\mathcal{MC}_{O_i}$. Thus, O_i and O_j meet compatibility condition 5.

Similarly, suppose that a query Q evaluates to False in all models of $\mathcal{MC}_{O_j}$. Then, because $\mathcal{MC}_{O_i}$ is a subset of $\mathcal{MC}_{O_j}$, it is not possible that Q evaluates to True in any model of $\mathcal{MC}_{O_i}$. Thus, O_i and O_j meet compatibility condition 6.

Claim 4: If $\mathcal{MC}_{O_i}$ *Equals* $\mathcal{MC}_{O_j}$ then the ontologies O_i and O_j meet compatibility conditions 1, 2, 3, 4, 5, and 6.

Proof: Because $\mathcal{MC}_{O_i}$ equals $\mathcal{MC}_{O_j}$ (and because we assume that each ontology has at least one model, i.e., the ontologies are consistent) there is at least one model in $\mathcal{MC}_{O_i}$ that is also in $\mathcal{MC}_{O_j}$. Thus, by the arguments used in Proof 1, the ontologies O_i and O_j meet compatibility conditions 1 and 2.

Because the models of O_i are the same as the models of O_j, it is not possible that a query Q evaluates to True in all models of O_i but to False in some model of O_j. Thus O_i and O_j meet compatibility condition 3.

Because the models of O_i are the same as the models of O_j, it is not possible that a query Q evaluates to False in all models of O_i but to True in some model of O_j. Thus O_i and O_j meet compatibility condition 4.

Because the models of O_i are the same as the models of O_j, it is not possible that a query Q evaluates to True in all models of O_j but to False in some model of O_i. Thus O_i and O_j meet compatibility condition 5.

Because the models of O_i are the same as the models of O_j, it is not possible that a query Q evaluates to False in all models of O_j but to False in some model of O_i. Thus O_i and O_j meet compatibility condition 6.

Claim 5: If $\mathcal{MC}_{O_i}$ is *Disjoint* from $\mathcal{MC}_{O_j}$, then the ontologies O_i and O_j meet none of compatibility conditions 1, 2, 3, 4, 5, or 6.

Proof: To show that O_i and O_j do not meet a particular compatibility condition, it suffices to construct a single counterexample. For instance, to show that if $\mathcal{MC}_{O_i}$ is *Disjoint* from $\mathcal{MC}_{O_j}$, then O_i and O_j do not meet compatibility condition 1, it suffices to give examples of two ontologies O_i and O_j such that there is some query Q that evaluates to True in all models of O_1 but to False in all models of O_2. Let O_i and O_j be ontologies with only 2 constant

symbols, *a* and *b*, and with a single binary relation symbol *R*. Suppose that the only data tuple in O_i is *(c,d)* and that the only data tuple in O_j is *(d,c)*. Suppose further that in O_i the relation *R* is axiomatized to be reflexive, but that in O_j the relation *R* is axiomatized to be irreflexive. Let Q be the query *R(a,a)*.

$\mathcal{MC}_{O_i}$ is disjoint from $\mathcal{MC}_{O_j}$, since every model in O_i contains the tuples *R(c,c)*, *R(d,d)* but no model in O_j contains these tuples. Further, Q evaluates to True in all models of O_i but to False in all models of O_j. Thus, O_i and O_j do not meet compatibility condition 1, because we have found a query Q that evaluates True in all models of O_i but to False in all models of O_j.

The creation of other counter-examples that show that O_i and O_j do not meet compatibility conditions 2-6 is left to the reader.

The next chapter discusses computational issues in calculating the relationship between model classes of two ontologies and determining their level of semantic interoperability.

Chapter 4

COMPUTING MODEL-CLASS RELATIONS

This chapter describes one exact method and one heuristic method for determining the relation between the model classes of two ontologies. The implementations of these methods are documented in Chapter 5.

From sections 3.4 and 3.5, it is assumed that all the models in the model classes of two ontologies O_i and O_j are available for computation and that the set-theoretic relation holding between the model classes (e.g., overlaps, contains) can be calculated. From such a calculation and the results in Sections 3.3 and 3.4, the level of semantic interoperability between O_i and O_j could then be determined.

However, even when the model classes of two ontologies are finite, they may have such a large number of models that it is not feasible to calculate all of them. Or, if all the models of each ontology can be calculated and stored on disk, it still may not be feasible, due to memory limitations, to determine the exact model-class relation that holds between the two ontologies.

In this latter case, a heuristic procedure can be used to narrow down the number of model-class relations that could possibly hold.

Section 4.3 discusses one such heuristic, which, since it is uses incomplete information (i.e., subsets of the model classes, rather than entire model classes), is not guaranteed to determine the single model-class relation that holds between the two ontologies. The use of this heuristic is therefore a trade-off: when computational costs make it impractical to compare the complete model classes directly and so obtain the exact model-class relation holding between the two ontologies, a heuristic can be used to do an incomplete comparison of model classes that will narrow down the possibilities of which model-class relation holds.

The heuristic discussed in Section 4.3 will always narrow down the outcome to 2 model-class relations that could hold, and it may narrow down to just one (Section 4.3).

Figure 4.1: 5 possible relations between model classes $MC(O_1)$ and $MC(O_2)$

4.1 Review of Assumptions

The ontologies O_i and O_j are assumed each to be consistent (i.e., each has at least one model) and to have the following in common:

1. Both ontologies are specified with the same logical language L, which is a subset of the language of first-order logic.

2. Both ontologies refer to the same finite (conceptual) domain, and they use the same finite set of constant symbols to refer to elements in this domain. The number of constant symbols is assumed to be identical to the number of elements in the conceptual domain. Further, if a constant symbol c used in O_i denotes an object *X* in the domain, then the same constant symbol c used in O_j denotes the same object *X*.

3. No model contains more elements than the finite number of entities in the conceptual domain.

4. The ontologies are assumed to be non-empty (i.e., they each have at least one data tuple and at least one axiom).

5. The ontologies use the same relation symbols.

The ontologies are assumed to differ in:

1. their data tuples (i.e., at least one of the ontologies has at least one data tuple not in the other ontology), and

2. their axioms (i.e., each ontology has at least one axiom, and at least one of the ontologies includes an axiom that is not included in the other).

These differences will generally result in *different semantics* for the ontologies, i.e., different model classes for the ontologies.

The language L has the following characteristics.

1. L contains finitely many variables, finitely many constant symbols (one for each element of the finite domain), finitely many relation symbols (each of which has a finite arity), and no function symbols.

2. The syntax and semantics of L are that of first-order logic with the restrictions that result from the various assumptions of finiteness just given.

3. L is used both to specify the ontologies and to formulate queries that are put to the ontologies.

4. Each ontology is specified using L via finitely many axioms and finitely many data tuples.

Finally, from an implementation point of view, we assume that finitely many models of each (consistent) ontology can be generated by a software program. The particular program we use for the implementations in Chapter 5 is SEM (Zhang and Zhang, 1995), which is an efficient generator of models for relatively small specifications. The translation from an ontology in the language L (which is used for illustration in the first four chapters) to a specification in the format required for SEM is done manually. Further details on the use of SEM are given in Chapter 5.

4.2 Four Computational Scenarios

4.2.1 What They Are

Figure 4.2 describes the four computational scenarios discussed in this thesis, where M refers to the number of models of ontology O_i, and N refers to the number of models of ontology O_j. For simplicity, we do not treat the cases where it is practical to compute M but not N, or practical to compute N but not M.

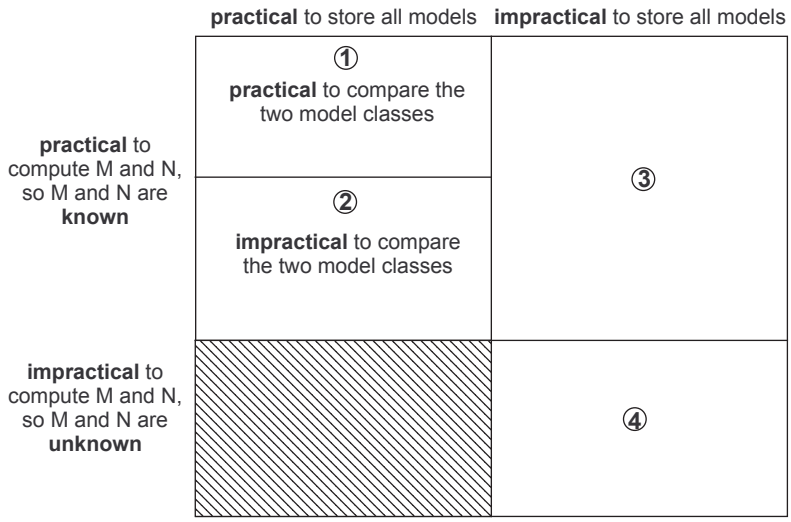

Figure 4.2: Feasibility of comparing M models of O_i and N models of O_j

1. *Scenario 1*: It is practical to compute both M and N and to store all models on disk. It is also practical to compare the two model classes completely and so determine the exact relation that holds between them.

2. *Scenario 2*: It is practical to compute both M and N and to store all models on disk, but it is impractical (e.g., due to memory or time limitations) to compare the two model classes completely in order to determine the unique relation that holds between them.

3. *Scenario 3*: It is practical to compute both M and N, but it is impractical to store all the models on disk or in memory. Thus, only certain subsets of models can be generated in this scenario. The models in these subsets will be generated in the order determined by the particular model-generating software that is used.

4. *Scenario 4*: It is impractical to compute M and N (e.g., due to memory or hard disk limitations), and it is impractical to store all models on disk or in memory. Thus, only certain subsets of models can be generated in this scenario. The models in these subsets would be generated in the order determined by the particular model-generating software that is used.

4.2.2 How They Are Treated

In Scenario 1, it is always possible *in practice* to compute all the models from each ontology, store each of these on disk, and algorithmically to determine which particular relationship (e.g., overlaps, contains) holds between the model classes. Thus, for Scenario 1 the exact level of semantic interoperability between the two ontologies can always be determined. Chapter 5 describes an implementation for Scenario 1 that uses the model-generator SEM (Zhang and Zhang, 1995) and the programming language Perl (Sections 5.1 and 5.2).

Scenario 2 is treated with a heuristic procedure (Section 4.3) that narrows down the possible relations from Figure 4.1 that could hold between the two model classes.

Chapter 5 presents implementations of an algorithm for Scenario 1 and an implementation of a heuristic for Scenario 2.

Scenarios 3 and 4 are not covered in this thesis, because it is impractical to store all the models in those scenarios: thus, these two scenarios cannot effectively illustrate the three central concepts used to define semantic interoperability: compatibility conditions (Sections 3.3 and 3.4), model-class relations (Section 3.5), and the connection between compatibility conditions, levels of semantic interoperability, and model-class relations (Section 3.5). Note, however, the following points concerning Scenario 3. First, since M and N are assumed known in this scenario, the possible model-class relations can be reduced from five to three, by the same arguments used in Section 4.3 (see Figure 4.3). Second, assuming that one could generate and store on disk at least *some* of the models of each ontology, certain comparisons could still be made. For instance, suppose that a million models of each ontology could be stored on disk. Then, if comparing those two subsets of models, it is found that there is at least one model in common, we can reduce the possible model-class relations to just two, regardless of the relative sizes of M and N. Of course the drawback to such an approach is that if find *no* models in common, that does not necessarily mean that the model classes actually have no models in common. They may simply not appear in our subsets. Thus, we are limited both by the manner in which models are generated by the model-generating software (in our case, SEM, which we treat as a black box), and by the fact that by looking at just subsets of the full model class, we cannot be sure, even if we detect no models in common, that the model classes really have no models in common.

Note that in some cases, with a decidable language, a proof calculus that is complete with respect to a given semantics of the language, and a prover such as RACER (http://www.sts.tu-harburg.de/∼r.f.moeller/racer/), it may be possible to determine, for *particular queries*, whether a query that evaluates to true in all models of O_i also evaluates to true in all models of O_j (Patel-Schneider, 2006). Such proof-theoretic means of getting at certain issues of compatible query results (and hence at certain issues of semantic interoperability), while interesting in and of themselves, lie outside the scope of this thesis.

4.3 Scenario 2, a Heuristic Method

The heuristic for Scenario 2 works by combining two pieces of information, one old and one new. The old piece of information, from the assumptions of Scenario 2, is that the sizes of both model classes are known. Given this information, the number of possible relations between the model classes of the ontologies is reduced immediately from five to three (Figure 4.3), where M is the number of models in $\mathcal{MC}_{O_i}$ and N is the number of models in $\mathcal{MC}_{O_j}$.

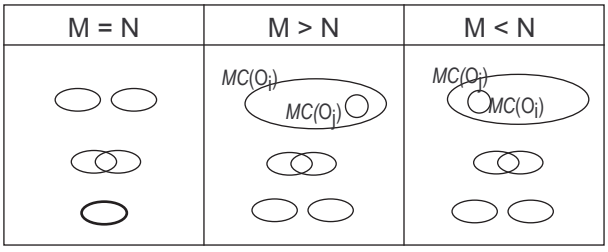

Figure 4.3: Constraints on model-class relations from sizes of M and N

The new piece of information is obtained by choosing a certain number of arbitrary models from one model class and testing whether those models are also in the other model class. For

instance, choose an arbitrary model p from $\mathcal{M}C_{O_i}$ and an arbitrary model q from $\mathcal{M}C_{O_j}$. Independent of the relative sizes of M and N, testing whether p is in $\mathcal{M}C_{O_j}$, lets us narrow down the number of possible model-class relations to four or three (Figure 4.4). Similar results apply after testing whether q is in $\mathcal{M}C_{O_i}$ (Figure 4.5). Combining both tests, and again independent of the relative sizes of M and N, the picture of possible model-class relations is as shown in Figure 4.6.

Combining both pieces of information—the relative sizes of M and N, along with the testing for membership of p in $\mathcal{M}C_{O_q}$ and of q in $\mathcal{M}C_{O_i}$—yields the possibilities depicted in Figure 4.7. This figure shows that after selecting just one arbitrary model from each ontology, the number of relations that could hold between the model classes of the ontologies is just two, though which two depends on the relative sizes of the model classes of the two ontologies.

The above observations lead to the following heuristic procedure, which for Scenario 2 will always narrow down the number of possible model-class relations to at most two.

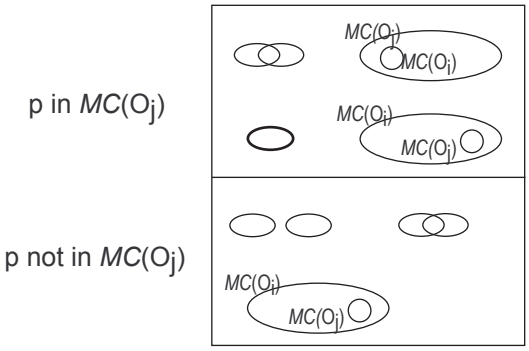

Figure 4.4: Possible model-class relations, checking p against O_j

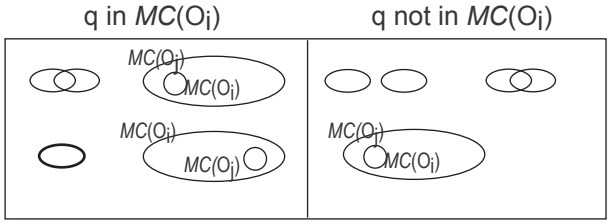

Figure 4.5: Possible model-class relations, checking q against O_i

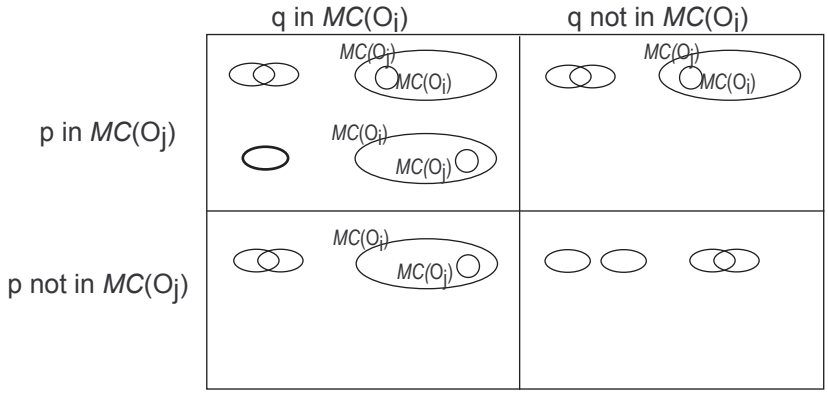

Figure 4.6: Possible relations, checking p against O_j and q against O_i

The steps of the heuristic are as follows.

1. Obtain the values of M (number of models in $\mathcal{MC}_{O_i}$) and N (number of models in $\mathcal{MC}_{O_j}$).

2. Pick an arbitrary model p_i from $\mathcal{MC}_{O_i}$ and test whether p_i is in $\mathcal{MC}_{O_j}$.

3. Pick an arbitrary model q_j from $\mathcal{MC}_{O_j}$ and test whether q_j is in $\mathcal{MC}_{O_i}$.

4. Use results of steps 1-3 to determine which two model class relations could hold.

5. Repeat steps 2-4 for a predetermined number of iterations.

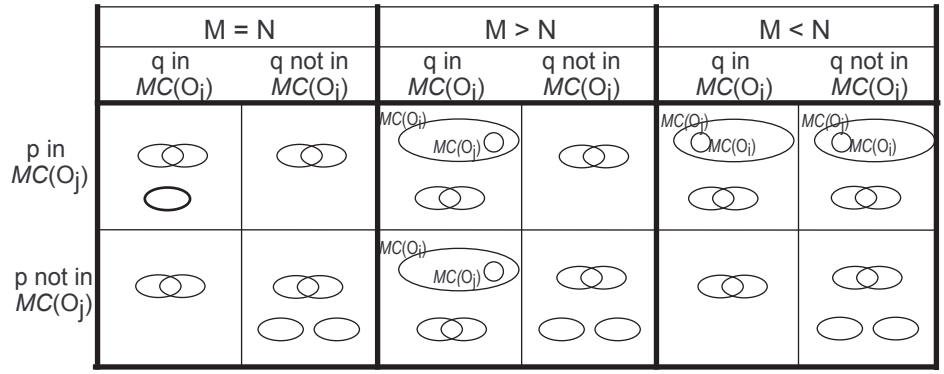

Figure 4.7: Checking p against O_j, q against O_i, and sizes of M and N

To see how this heuristic works, start at Step 1 and suppose that the result is $M < N$. Then, as can be seen from Figure 4.3, only three model-class relations could possibly hold: (1) $\mathcal{MC}_{O_i}$ is contained by $\mathcal{MC}_{O_j}$; (2) $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$ overlap; or (3) $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$ are disjoint.

Suppose that at Step 2 it is found that p_i is not in $\mathcal{MC}_{O_j}$. Then, as can be seen from Figure 4.4, only three model-class relations could possibly hold: (1) $\mathcal{MC}_{O_i}$ contains $\mathcal{MC}_{O_j}$; (2) $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$ overlap; or (3) $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$ are disjoint.

Now suppose that at Step 3 it is found that q_j is in $\mathcal{MC}_{O_i}$. Then, as can be seen from Figure 4.5, four of the five model-class relations could possibly hold: (1) $\mathcal{MC}_{O_i}$ contains $\mathcal{MC}_{O_j}$; (2) $\mathcal{MC}_{O_i}$ is contained by $\mathcal{MC}_{O_j}$; (3) $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$ overlap; or (4) $\mathcal{MC}_{O_i}$ and $\mathcal{MC}_{O_j}$ are identical.

In Step 4 we combine the information from Steps 1, 2, and 3, with the result that there is only one model-class relation that could hold: overlaps (Figure 4.7).

Note that although Step 1 needs to be done only once (since M and N do not change), nevertheless it may pay to iterate Steps 2 and 3, if the result at Step 4 is not a single model-class relation.

After each iteration of Steps 2 and 3, the end result will be that at most two model-classes relations could hold. It is possible that subsequent iterations can reduce the number of possible model classes to just one: Overlap. Consider, for instance, Figure 4.8, which shows different quadrants for each test of whether p_i is in $\mathcal{MC}_{O_j}$ and q_j is in $\mathcal{MC}_{O_i}$. Figure 4.9 shows which model-class relations could hold after a second iteration of testing p_i in $\mathcal{MC}_{O_j}$ and q_j in $\mathcal{MC}_{O_i}$.

For instance, the third row of Figure 4.10 shows that for all relative values of M and N, if the first iteration yields A (p_1 is in $\mathcal{MC}_{O_j}$ and q_1 is also in $\mathcal{MC}_{O_i}$) and the second iteration yields C (p_2 is not in $\mathcal{MC}_{O_j}$ but q_2 is in $\mathcal{MC}_{O_i}$), then the model-class relation must be Overlap.

In this case, with just two iterations, the heuristic has determined the unique relation that exists between the model classes of the ontologies, even though, by assumption, it is not feasible to use an algorithm to make this determination.

Note that this heuristic is not guaranteed to narrow the possibilities down to just one model-class relation. In fact, the heuristic will give this result only when the relation between model classes is Overlap.

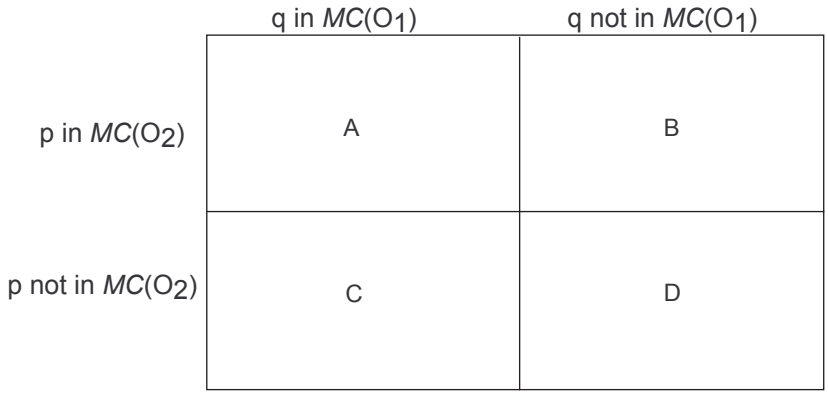

Figure 4.8: The four quadrants of possible results for each iteration

The different possible outcomes of using this heuristic are:

1. The heuristic narrows down the possible model-class relations to just one: Overlap.

2. The heuristic narrows down to just two possible model-class relations, which could be: 1) Overlap or Contains; 2) Overlap or Contained by; 3) Overlap or Disjoint; or 4) Overlap or Identical.

The heuristic always narrows down either to the single relation Overlap or to one of the 4 pairs of model-class relations just mentioned.

In terms of semantic interoperability, the heuristic will allow us to draw one of the following conclusions:

1. The two ontologies are semantically interoperable *exactly* at Level 1.

2. O_i is semantically interoperable with O_j at Levels 1 and 2A.

3. O_i is semantically interoperable with O_j at Levels 1 and 2B.

4. The two ontologies are semantically interoperable *at most* at Level 1.

5. The two ontologies are semantically interoperable at Level 1 or at Level 3.

Or, in terms of the compatibility conditions that underlie our definitions of semantic interoperability, calling our ontologies O_i and O_j,

1. it is not possible to have a query that evaluates to True in all models of O_i but to False in all models of O_j (compatibility condition 1); and it is not possible to have a query that evaluates to False in all models of O_i but to True in all models of O_j (compatibility condition 2).

2. it is not possible to have a query that evaluates to True in all models of O_i, but to True in some models of O_j and to False in other models of O_j (compatibility condition 3); and it is not possible to have a query that evaluates to True in all models of O_i but to False in all models of O_j (compatibility condition 1); and it is not possible to have a query that evaluates to False in all models of O_i, but to True in some models of O_j and to False in other models of O_j (compatibility condition 4); and it is not possible to have a query that evaluates to False in all models of O_i but to True in all models of O_j (compatibility condition 2).

3. it is not possible to have a query that evaluates to True in all models of O_j, but to True in some models of O_i and to False in other models of O_i (compatibility condition 5); and it is not possible to have a query that evaluates to True in all models of O_i but to False in all models of O_j (compatibility condition 1); and it is not possible to have a query that evaluates to False in all models of O_j, but to True in some models of O_i and to False

in other models of O_i (compatibility condition 6); and it is not possible to have a query that evaluates to False in all models of O_i but to True in all models of O_j (compatibility condition 2).

4. *Either* it is not possible to have a query that evaluates to True in all models of O_i but to False in all models of O_j (compatibility condition 1) and it is not possible to have a query that evaluates to False in all models of O_i but to True in all models of O_j (compatibility condition 2); *or* none of the six compatibility conditions given in Section 3.4.2, Definition 2 (for Level 3 Semantic Interoperability) holds between O_i and O_j.

5. *Either* it is not possible to have a query that evaluates to True in all models of O_i but to False in all models of O_j (compatibility condition 1) and it is not possible to have a query that evaluates to False in all models of O_i but to True in all models of O_j (compatibility condition 2); *or* all six of the compatibility conditions given in Section 3.4.2, Definition 2 (for Level 3 Semantic Interoperability) hold between O_i and O_j.

4.4 How the Heuristic Performs

The discussion in the previous section of the workings of the heuristic showed that it will always, in a single pass through steps 1-4, reduce the number of possible model-class relations from five to at most two. One might still question, 'how well the heuristic performs,' in the sense of asking what is the likelihood of obtaining any of the particular outcomes listed in the previous section.

To answer this question, we consider basic empirical probabilities associated with steps 1 through 4 for the case where $M > N$. The other two cases ($M = N$ and $M < N$) can

be analyzed similarly. Assuming that $M > N$, we ask and answer the following questions related to the different possible outcomes of the heuristic.

If the model-class relation is overlap, what is the probability that one pass through steps 1 through 4 will show that:

- the model-class relation between O_i and O_j is exactly overlap?
- the model-class relation between O_i and O_j is either overlap or contains?
- the model-class relation between O_i and O_j either overlap or disjoint?

The pattern for asking and answering other questions of this type (e.g., 'If the model-class relation is disjoint, what is the probability that one pass through steps 1 through 4 will show that the model-class relation between O_i and O_j is either disjoint or overlaps?) is the same.

To take a simple numerical example, which is generalized below, let $M = 1000$ be the number of models in the model class of O_i, $N = 500$ be the number of models in the model class of O_j, and $K = 100$ be the number of models that the model classes have in common.

Then, the probability that, in one pass through steps 1 through 4, the heuristic shows that the model-class relation between O_i and O_j is exactly overlap (Figure 4.7) is the probability that a randomly selected model p from O_i *is* in $\mathcal{MC}_{O_j}$ and a randomly selected model q from O_j *is not* in $\mathcal{MC}_{O_i}$ (Figure 4.10). Call this probability P_1. Then P_1 = (100/1000)*(400/500) = (.1)(.8) = 0.08.

Similarly, the probability P_2 that in one pass through steps 1 through 4, the heuristic shows that the model-class relation between O_i and O_j is *either overlap or contains* (Figure 4.7) is the probability that a randomly selected model p from O_i *is* in $\mathcal{MC}_{O_j}$ and a randomly selected model q from O_j *is* in $\mathcal{MC}_{O_i}$, *plus* the probability that a randomly selected model p from O_i

is not in $\mathcal{MC}_{O_j}$ and a randomly selected model q from O_j *is* in $\mathcal{MC}_{O_i}$ (Figure 4.10). Then P_2 = (100/1000)*(100/500) + (900/1000)*(100/500) = (.02) + (.18) = 0.20.

Similarly, the probability that, in one pass through steps 1 through 4, the heuristic shows that the model-class relation between O_i and O_j is *either overlap or disjoint* (Figure 4.7) is the probability that a randomly selected model p from O_i *is not* in $\mathcal{MC}_{O_j}$ and a randomly selected model q from O_j *is not* in $\mathcal{MC}_{O_i}$ (Figure 4.10). Call this probability P_3. Then P_3 = (900/1000)*(400/500) = (.9)*(.8) = 0.72.

In general, given M, N, and K as above,

- P_1, the probability that in one pass through steps 1 through 4 the heuristic shows that the model-class relation between O_i and O_j is exactly overlap, is $((K/M) * ((N-K)/N))$.

- P_2, the probability that in one pass through steps 1 through 4 the heuristic shows that the model-class relation between O_i and O_j is *either overlap or contains*, is $((K/M) * (K/N)) + (((M-K)/M) * (K/N))$.

- P_3, the probability that in one pass through steps 1 through 4 the heuristic shows that the model-class relation between O_i and O_j is *either overlap or disjoint*, is $(((N-K)/N) * ((N-K)/N)$.

The above analysis of the performance of the heuristic made use of the values of M and N, which are assumed known in scenario 2, but also of the value K (the number of models in common to the ontologies' model classes), which in Scenario 2 is unknown. So the question may be asked, "How do you know K?" to which the answer is "We don't." Then the question becomes, "So of what use is the above analysis if you don't know K?" The answer is that the above analysis shows the constraints that operate when trying to assess the performance of the

heuristic. That the heuristic performs according to the probabilities given: in any instance of scenario 2, probabilities of the kinds given above are at the ones in play.

The above analysis shows that the performance of the heuristic depends on K, and it shows exactly how. What one can take away from this analysis is that the performance of the heuristic, for any given pair of ontologies that fall into Scenario 2, depends on the relative amount of overlap between the model classes (with the details spelled out by probabilities like the ones above).

4.5 Finite versus Tractable

One consequence of the assumptions given in Section 4.1 is that every model of each ontology is finite. A second consequence of these assumptions is that the model class of each ontology is also finite, i.e., there are only finitely many models in each model class. This fact follows from two other facts: (1) the signature of each ontology is finite, and therefore, since we also assume the size of the logical domain is equal to the number of entities in the conceptual domain, the number of "possible realizations" (Suppes, 2002, pg. 26) of this finite signature is finite; and (2) the model class of an ontology is a subset of the number of possible realizations of the signature of the ontology.

Even though each ontology's model class will be finite, these model classes may still be too large to be practically computable. To gain some idea of how large the size of a model class might be, consider the upper bound on the number of possible models of a given ontology.

This upper bound is obtained by calculating how many models could possibly be constructed based on the signature alone, disregarding any possible axioms or data tuples. For an

ontology O_i, call this number U_{O_i}. An actual ontology O_i will contain axioms and data tuples, which constrain the number of models in the model class of the ontology, so the number of possible models of O_i should generally be at least several orders of magnitude smaller than U_{O_i}. It is nevertheless instructive to see how rapidly U_{O_i} grows as the number of constant symbols and relation symbols increases.

Consider the ontology O_1 from Chapter 3 (Figure 3.1) that contains 4 constant symbols and 1 binary relation symbol. There are $4^2 = 16$ possible binary relations that could be in any realization of this signature. Since a given model could contain anywhere from 0 through 16 of these relations, the total number of models possible is the sum of all the different ways that a model could contain $0, 1, 2, \ldots 16$ of these relations. That sum is given by: $\sum_{i=0}^{16} C(16, i) = 65,536$, where $C(16, i)$ is the number of ways of choosing i objects from 16 objects.

Note that each of the ontologies in Figures 3.1 and 3.2 has only 9 models. Thus, the data and axioms of those ontologies constrain the model classes sharply, from a possible 65,536 models to just 9.

Consider now an ontology with 6 domain elements, 4 unary relations (classes), and 2 binary relations (e.g., any of the four sample ontologies in Appendix A).

Without any axioms or data tuples specified in the ontology, a model for such an ontology could contain $4 * 6 = 24$ 1-tuples for the 6 entities and the 4 unary relations (i.e., each of the six entities could be a town, a road, a river, or a state). In addition, considering the signature alone, a model of the ontology could contain $6 * 6 = 36$ 2-tuples for one of the binary relations (e.g., *in(Bangor,Maine)*) and $6 * 6 = 36$ 2-tuples for the other binary relation (e.g., *through(I95,Bangor)*). Taken together, then, any model for such an ontology could have from 0 to $4 * 6^1 + 2 * 6^2 = 96$ tuples. That is, the model class of such an ontology could contain

$\sum_{i=0}^{96} C(96, i)$ structures that could be models. In other words, the potential size of the model class of this ontology is $\sum_{i=0}^{96} C(96, i)$ or more than $79 * 10^{27}$.

In general, for an ontology O with d domain elements and j_i relations of arity $i > 0$, the number of tuples that could be in any model is $P = \sum_i j_i(d^i)$. Any model could have from 0 to P of these relations. Thus, the upper bound on the number of possible models (assuming the ontology has no axioms and no data tuples) is $U_O = \sum_{i=0}^{P} C(P, i)$.

Each of the four sample ontologies in Appendix A actually has many fewer models, by orders of magnitude. Specifically, the ontology with the most models is ch5O4.in, with $187,392$ models, is about 18 orders of magnitude smaller than $79 * 10^{27}$.

Yet, for ontologies that describe dozens or hundreds of entities, classes, and relations, even when these have many axioms and data tuples, the finite number of models in their model classes may still be too large to be practically computed. Thus, "finite" does not always mean "practically computable." So, in practice, it may not always be possible to compare the two model classes directly, in order to determine the model-class relation that holds between the ontologies. Whether or not it is possible depends on several factors, including the number of axioms and data tuples relative to the number of constant and relation symbols in the domain, the detail with which a given domain is modeled, and the computational resources available to compute the model classes. An investigation into these factors lies outside the scope of this thesis.

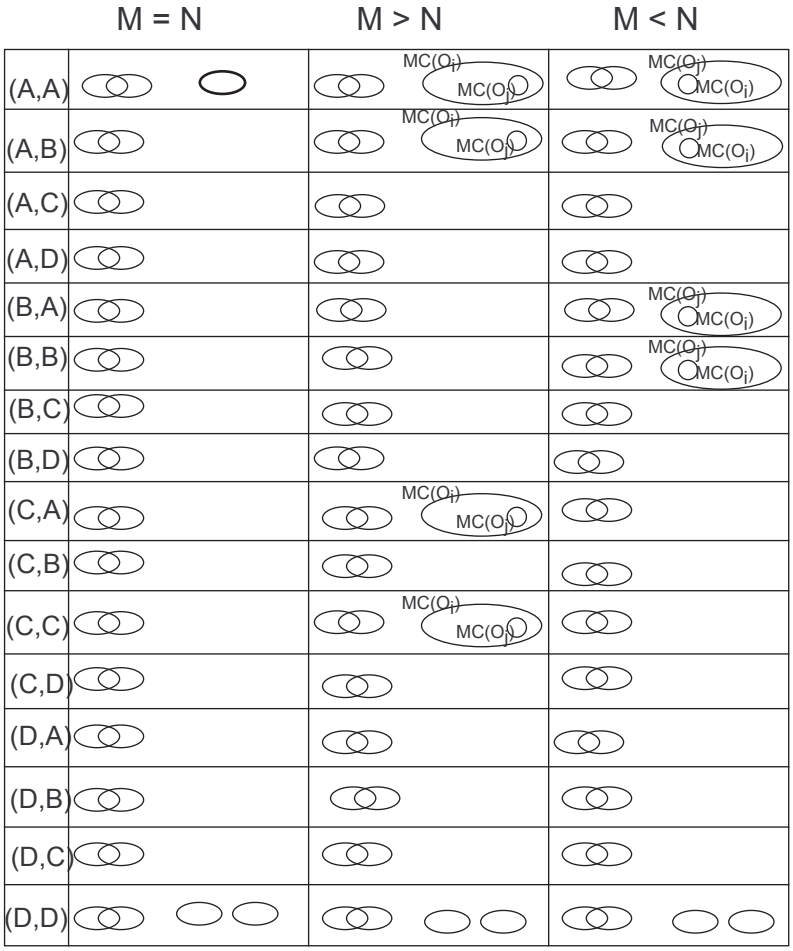

Figure 4.9: Possible model-class relations after 2 checks of p and q

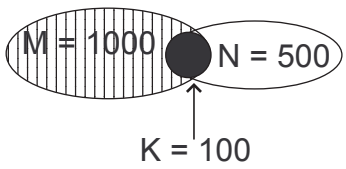

Figure 4.10: Example where model-class relation is Overlap

Chapter 5

IMPLEMENTATIONS

This chapter implements one algorithm and one heuristic that test for the model-class relation that exists between two example geospatial ontologies O_i and O_j. The algorithm determines exactly which one of the five model-class relations (Figure 4.1) holds, and based on the results of Sections 3.4 and 3.5, the level at which the two ontologies are semantically interoperable. The heuristic (Section 4.3) is not guaranteed to identify a single model-class relation that holds. This heuristic could identify a single relation if that relation is Disjoint, or it could identify two possible model-class relations (Section 4.3). Thus, the heuristic, though it is not guaranteed to produce a definitive level of semantic interoperability of the ontologies, will always reduce the number of possible model-class relations that could hold from five to at most two (Figure 4.7).

The models for all the ontologies in this chapter were generated using SEM (Zhang and Zhang, 1995), with source code at: *ftp://ftp.cs.uiowa.edu/pub/hzhang/sem/sem.tar.Z*. Some changes were needed to get the source code to compile. Permission was granted by the code's author (J. Zhang) to make these changes and to modify it as needed for research purposes.

5.1 Implementation for Scenario 1, Exact

In this Scenario, the full model classes of both ontologies can be calculated, stored on disk, and compared to determine the exact model-class relation that holds between them. From the results of Section 3.4, the exact level of semantic interoperability is determined.

In the two ontologies that follow, there are six entities (Route2, Orono, Bangor, Maine, I95, and river1), four classes (state, town, road, and river), and two binary spatial relations ('in' and 'through'). Each ontology contains a number of axioms and data tuples. Instead of presenting these ontologies in the style of previous chapters, we present English-language summary statements of the axioms and data used in the SEM files themselves, $ch5O1.in$ and $ch5O2.in$ (Appendix A). Note that these English-language summary statements are just approximations to the exact formulations given in the $.in$ files. They are provided for the reader's convenience.

The following statements provide an English-language version of the ontology that is specified in the file $ch5O1.in$, which is the input file of this ontology in the format that SEM requires (see Appendix A).

- IN is reflexive, antisymmetric, and transitive.
- THROUGH is antisymmetric and transitive.
- Towns, roads, rivers, and states are all pairwise disjoint.
- if x and y are different roads/rivers/towns/states, then x is not IN y
- if x is a town/state/river and y is a road, then x is NOT IN y
- if x is a road/town/state and y is a river, then x is NOT IN y
- if x is a state and y is a town, then x is NOT IN y
- if x is a state and y is a town, then x does NOT go THROUGH y
- if x is a town/state/river and y is a road, then x does NOT go THROUGH y
- if x is a town/state and y is a river, then x does NOT go THROUGH y
- if x is a town/state and y is a town, then x does NOT go THROUGH y
- if x goes THROUGH y, then y is NOT IN x
- Orono and Bangor are towns.
- Route2 and I95 are roads.
- Maine is a state.

- River1 is a river.
- Route2 is IN Orono and IN Bangor.
- Orono, Bangor, and River1 are IN Maine.
- River1 goes THROUGH Bangor, Orono, and Maine.
- I95 goes THROUGH, Bangor, Orono, and Maine.
- Route2 goes THROUGH Bangor, Orono, and Maine.

The following statements provide an English-language version of the ontology that is specified in the file $ch5O2.in$, which is the input file of this ontology in the format that SEM requires (see Appendix A).

- IN is reflexive, antisymmetric, and transitive.
- THROUGH is antisymmetric and transitive.
- Towns, roads, rivers, and states are all pairwise disjoint.
- If x and y are different roads/rivers/towns/states, then x is NOT IN y.
- If x is a town/state/river and y is a road, then x is NOT IN y
- If x is a road/town/state and y is a river, then x is NOT IN y.
- If x is a state and y is a town, then x is NOT IN y.
- If x is a state and y is a town, then x does NOT go THROUGH y.
- If x is a town/state/ and y is a road, then x does NOT go THROUGH y.
- If x is a town/state/road and y is a river, then x does NOT go THROUGH y.
- If x is a town/state and y is a town, then x does NOT go THROUGH y.
- If x goes THROUGH y, then y is not IN x.
- Orono and Bangor are towns.
- Route2 and I95 are roads.
- Maine is a state.

- River1 is a river.
- Orono and Bangor are IN Maine.
- River1 is IN Maine.
- River1 goes THROUGH Bangor, Orono, and Maine.
- I95 goes THROUGH, Bangor, Orono, and Maine.
- Route2 goes THROUGH Bangor, Orono, and Maine.
- Orono does NOT go THROUGH Maine.
- Bangor does NOT go THROUGH Maine.

5.2 Discussion: Scenario 1, Exact

The two ontologies specified in $ch5O1.in$ and $ch5O2.in$ are very similar, differing only in a few axioms and data tuples. There are 800 models of the ontology $ch5O1.in$, and 1250 models of the ontology in $ch5O2.in$.

The SEM program was used to generate all the models (having six entities, four classes, and two binary relations) for the ontologies in $ch501.in$ and $ch5O2.in$. The resulting SEM output files (see Appendix A for what these SEM output files contain) were processed by a Perl program (Appendix B) to determine exactly which model-class relation holds between the two ontologies.

The output from this Perl program was:

```
number of models in ontology 1 is: 800
number of models in ontology 2 is: 1250
number of models in ontology 2 but not in ontology 1: 1190
number of models in ontology 1 but not in ontology 2: 740
number of models in both ontology 1 and ontology 2 is: 60
```

Thus, because there are some models in the intersection of the two model classes, and because each model class also contains models that are not in the other model class, the unique model-class relation holding between the model classes of the two ontologies is *overlap*. Therefore, based on the arguments in Sections 3.3.1 and 3.4, the ontology in $ch5O1.in$ is *Level-1 semantically interoperable* with the ontology in $ch5O2.in$. What this means in terms of the compatibility conditions that underlie our definitions of different levels of semantic interoperability (Section 3.4.1, Definition 1) is that

- it is not possible to have a query that evaluates to True in all models of $ch5O1.in$ but to False in all models of $ch5O2.in$ (compatibility condition 1);

- it is not possible to have a query that evaluates to False in all models of $ch5O1.in$ but to True in all models of $ch5O2.in$ (compatibility condition 2).

5.3 Implementation for Scenario 2, Heuristic

In this scenario, we use two ontologies in the files $ch5O3.in$ and $ch5O4.in$ (Appendix A). Each of these ontologies uses the same six entities, four classes, and two binary spatial relations as the two ontologies in the previous section. Recall (Section 4.2.1) that by assumption in Scenario 2: (1) the model classes of the ontologies can be computed and stored on disk, (2) the number of models in each model class is known, and (3) the computing resources (e.g., time, memory) are not available to implement an algorithm that determines exactly which model-class relation holds. Thus for Scenario 2 we implement a heuristic (Section 4.3) that narrows down the possibilities of which model-class relations could hold.

The following statements provide an English-language version of the ontology that is specified in the file $ch5O3.in$, which is the input file of this ontology in the format that SEM requires (see Appendix A).

- IN is reflexive, antisymmetric, and transitive.
- THROUGH is antisymmetric.
- Towns, roads, rivers, and states are all pairwise disjoint.
- If x and y are different roads/rivers/towns/states, then x is NOT IN y.
- If x is a town/state/river and y is a road, then x is NOT IN y
- If x is a road/town/state and y is a river, then x is NOT IN y.
- If x is a state and y is a town, then x is NOT IN y.
- If x is a state and y is a town, then x does NOT go THROUGH y.
- If x is a town/state/river and y is a road, then x does NOT go THROUGH y.
- If x is a town/state/road and y is a river, then x does NOT go THROUGH y.
- If x is a town/state and y is a town, then x does NOT go THROUGH y.
- Orono and Bangor are towns.
- Route2 and I95 are roads.
- Maine is a state.
- River1 is a river.

- Orono and river1 are IN Maine.
- River1 goes THROUGH Bangor and Orono.
- I95 goes THROUGH, Bangor and Orono.
- Route2 goes THROUGH Bangor and Orono.
- Bangor does NOT go THROUGH Maine.
- Orono does NOT go THROUGH Maine.

The following statements provide an English-language version of the ontology that is specified in the file $ch5O4.in$, which is the input file of this ontology in the format that SEM requires (see Appendix A).

- IN is reflexive, antisymmetric, and transitive.
- THROUGH is antisymmetric.
- Towns, roads, rivers, and states are all pairwise disjoint.
- If x and y are different roads/rivers/towns/states, then x is NOT IN y.
- If x is a town/state/river and y is a road, then x is NOT IN y
- If x is a road/town/state and y is a river, then x is NOT IN y.
- If x is a state and y is a town, then x is NOT IN y.
- If x is a state and y is a town, then x does NOT go THROUGH y.
- If x is a town/state/river and y is a road, then x does NOT go THROUGH y.
- If x is a town/state and y is a river, then x does NOT go THROUGH y.
- If x is a town/state and y is a town, then x does NOT go THROUGH y.
- Orono and Bangor are towns.
- Route2 and I95 are roads.
- Maine is a state.
- River1 is a river.

- Orono and River1 are IN Maine.
- River1 goes THROUGH Bangor and Orono.
- I95 goes THROUGH, Bangor and Maine.
- Route2 goes THROUGH Bangor, Orono, and Maine.
- Bangor does NOT go THROUGH Maine.
- Orono does NOT go THROUGH Maine.

5.4 Discussion: Scenario 2, Heuristic

The ontologies specified in $ch5O3.in$ and $ch5O4.in$ are very similar to each other, differing only in a few of their axioms and data tuples. There are 93,696 models of the ontology specified in $c5O3.in$ and 187,392 models of the ontology specified in $c5O4.in$.

The SEM program was used to generate all the models (having six entities, four classes, and two binary relations) for the ontologies in $ch503.in$ and $ch5O4.in$. The resulting SEM output files, $c53.out$ and $c54.out$, were processed by a Perl program (Appendix B) to determine which model-class relation(s) between the two ontologies could hold.

The output from one run of this Perl program was:

```
p1 is 25015; p2 is 22703
q1 is 12731; q2 is 154842
model 25015 in c53.out found as model 47829 in c54.out
model 22703 in c53.out not found in c54.out
model 12731 in c54.out not found in c53.out
model 154842 in c54.out not found in c53.out
```

The heuristic finds that model 25015 of $c53.out$ occurs in $c54.out$ (as model 47829), but that model 12731 of $c54.out$ does not occur in $c53.out$. Thus, in the language of Section 4.3, when testing p_1 for membership in O_j and q_1 for membership in O_i, we are in the upper-left quadrant of Figure 4.8, the quadrant labeled A.

The above results also show that the heuristic finds that model 22703 of $c53.out$ does not occur in $c54.out$, and that model 154842 of $c54.out$ does not occur in $c53.out$. Thus, in the language of Chapter 4, when testing p_2 for membership in O_j and q_2 for membership in O_i, we are in the lower-right quadrant of Figure 4.8, the quadrant labeled D.

Putting together this information we find that the appropriate row and column of Figure 4.8 is the row for the pair (A, D) and the column for $M < N$. Thus, according to Figure 4.9, we have determined that the unique model-class relation between the model classes of the two ontologies is *overlap*.

Therefore, based on the arguments in Sections 3.3.1 and 3.4, the ontology in $ch5O3.in$ is *Level-1 semantically interoperable* with the ontology in $ch5O4.in$.

What this means in terms of the compatibility conditions that underlie our definitions of different levels of semantic interoperability (Section 3.4.1, Definition 1) is that

- it is not possible to have a query that evaluates to True in all models of $ch5O3.in$ but to False in all models of $ch5O4.in$ (compatibility condition 1);

- it is not possible to have a query that evaluates to False in all models of $ch5O3.in$ but to True in all models of $ch5O4.in$ (compatibility condition 2).

Chapter 6

SUMMARY, CONCLUSIONS, AND FUTURE WORK

6.1 Thematic Summary

Chapter 1 began with a simple example (Sections 1.1 and 1.3) that provided the intuition and motivation for the treatment of semantic interoperability in Chapters 3-5. That example (Figures 1.3 and 1.4) highlighted how people can disagree about what certain agreed-upon statements actually *mean*. It was suggested that at the root of such disagreements are often different implicit assumptions people have about what the agreed-upon statements *entail*, i.e., what does or does not follow logically from those statements.

Ontologies can help get to the bottom of such disagreements, because they allow people to specify their conceptualizations of a given domain in a machine-processable way. In particular, when formal ontologies are used in conjunction with reasoning software that can compute inferences, queries can be put to these ontologies to test whether certain statements follow as

logical consequences from the ontologies. Using ontologies as specifications in this way will not eliminate any disagreements that the people who write them may have about meanings, but it will allow, under appropriate assumptions, machines to probe the implicit information in these specifications and so be able to make explicit the entailments (logical consequences) that are implicit in the specifications.

In particular, if the ontologies are specified using a logical language whose semantics is given by models (sets with a certain structure), it is possible in some cases to compute all the models of a given ontology and so get a handle on its semantics. In these cases, questions about whether a given statement follows as a logical consequence of the (statements of the) ontology can then be analyzed directly in terms of models. A statement is a logical consequence of (the statements of) an ontology if it is true in all models of the ontology.

In cases where not all the models of the ontologies can be computed (e.g., for infinite domains, or for finite domains but intractable model classes), an alternate way to analyze questions of logical consequence can be found through proof theory. With suitable assumptions — dealing with the decidability of the logical language, the computational properties of the proof calculus employed, and the connection between the results of syntax and semantics (i.e., completeness) for the given proof calculus — a computer can determine via proof theory whether a given statement is a logical consequence (syntactic and, by soundness, semantic) of the statements in the ontology.

For questions dealing with semantics, though, the arguments based on models (and semantic logical consequence) are often more intuitive than the arguments based on proofs (and syntactic logical consequence). For this reason, in this thesis, questions about semantic interoperability were asked and answered in terms of models, not proofs. That is, in explaining

what the explicit statements of an ontology collectively 'mean,' models were used as the intuitive background that could accommodate notions of implicit meanings, as well as the multiple possible 'worlds' that are consistent with a given ontology.

This model-based analysis of semantics connects the motivating example in Chapter 1 to the more detailed analyses in Chapters 3-5, via the following steps.

1. Geospatial ontologies are used to specify two plausible conceptualizations of a given geographic domain.

2. Certain assumptions of finiteness are imposed on the ontologies, and all the models of each ontology are computed.

3. A query to an ontology is a statement (written in the language of the ontology specification) that evaluates to true or false (see Section 3.1.2), according to whether or not it follows as a logical consequence of the statements of the ontology.

4. A query follows as a logical consequence of an ontology if it is true in all models of the ontology.

5. The test for semantic interoperability employs the notion of 'compatible query results,' which considers whether a query that is entailed by (follows as a logical consequence of) one ontology is also entailed by another ontology.

6. Different kinds of compatible query results are defined, indicating greater or lesser degrees of interoperability between two ontologies.

7. Different degrees of semantic interoperability are *defined* in terms of different kinds of compatible query results.

8. Mathematical results connect the different kinds of compatible query results to the set-theoretic relationship (e.g., Overlaps) between the model classes of the ontologies.

9. These mathematical results are used in conjunction with the calculated models of each ontology to *compute* the degree of semantic interoperability of the two ontologies in terms of the set-theoretic relation that holds between their model classes.

10. In some cases, when the model classes are extremely large (on the order of billions or more models in each class), the exact level of semantic interoperability may not be able to be determined.

6.2 Answer to the Research Question

The research question posed in Chapter 1, Section 1.7.1 is: *When two geospatial ontologies use the same language to describe the same domain, but differ in the model-theoretic semantics of their primitive spatial-relation symbols, in what sense and to what extent are the ontologies semantically interoperable?*

Chapter 3 created the framework (formal ontologies with their semantics specified by model-theoretic semantics, Sections 3.1 and 3.2) and the new constructs (compatibility conditions, Section 3.3.1) to answer this research question.

The answer to the research question is as follows:

When two geospatial ontologies use the same language to describe the same domain, but differ in the model-theoretic semantics of their primitive spatial-relation symbols, they are semantically interoperable *in the sense that* the evaluations of arbitrary queries put to each ontology are ‘compatible’ as determined by the six compatibility conditions (Figure 3.19).

When two geospatial ontologies use the same language to describe the same domain, but differ in the model-theoretic semantics of their primitive spatial-relation symbols, they are semantically interoperable *to the extent that* they meet the different groups of compatibility conditions that correspond to the five model-class relations between ontologies (Figures 3.19 and 3.18).

6.3 Results Summary

In answering this research question, this thesis has met its research goal set forth in Section 1.7.2: "to create a method to assess the extent of semantic interoperability between two geospatial ontologies" (based on the requirements in Section 1.6), which are to take into account both the models of the ontologies and the queries that can be put to the ontologies.

In meeting this research goal, this thesis has:

1. investigated the notion of semantic interoperability between two geospatial ontologies when these ontologies differ only in the semantics of their primitive spatial relations (Figure 2.1). Two geospatial ontologies are *different* "in the semantics of their primitive spatial relations" when the models of the ontologies differ, i.e., when the sets of models of the two ontologies are not equal as sets.

2. defined different degrees of *partial semantic interoperability* of geospatial ontologies in terms of *compatibility conditions*, which in turn are defined in terms of models and queries.

3. demonstrated that the different levels of semantic interoperability that exist between two ontologies can be determined based on the set-theoretic relation (e.g., disjoint, overlaps) that holds between the sets of models of the two ontologies.

4. implemented two procedures for determining the relation that holds between the sets of models of the ontologies. The first procedure calculates exactly which relation holds. The second procedure uses weaker assumptions and is guaranteed only to narrow down the possibilities of which relations could hold from five to two.

What was called in Section 3.4.1 'Level 1 semantic interoperability' between ontologies is the most fundamental level of semantic interoperability because it guarantees that for no query Q will Q be entailed by one ontology while $\neg Q$ is entailed by the other ontology. This condition thus represents a minimum 'safety' condition between ontologies. Given that one of the goals of using ontologies is to have machines process meanings without input from people, it is essential to know when the ontologies might give different results to the same query. Level 1 semantic interoperability insures a basic degree of compatibility between query results, without which it is unlikely that people would want to trust the interpretations of meanings to machines.

The definitions in Chapter 3 of five different levels of semantic interoperability make sense even when certain finiteness assumptions (Section 4.1) do not hold, though the actual computation of the level of semantic interoperability may not be possible in these cases. In this sense, this thesis has provided a platform upon which further discussions of semantic interoperability can be based.

This thesis also showed that for the kind of geospatial ontologies considered here, the question of whether there is anything "special about spatial" has two answers: No and Yes. No, in the sense that the method of analysis of semantic interoperability between ontologies remains the same whether spatial properties are present in the ontologies or not. Yes, in the sense that one supposes that in *geospatial* ontologies, particular spatial properties will be specified in the ontologies.

6.4 Significance of the Research Contributions

As was shown in Chapters 4 and 5, given certain finiteness assumptions, the levels of semantic interoperability defined in Chapter 3 can be computed. However, as was noted in Section 4.5, when the total number of entities, unary relations, and binary relations exceeds more than a few dozen, the number of possible models becomes very large: even with appropriate axioms and data tuples constraining the space of possible models, the computations — at least given the algorithms used in SEM — are quickly rendered intractable.

Therefore, one might ask, "What is the value of this research if, for realistic ontologies of hundreds or thousands of entities, classes, and binary relations, it cannot guarantee an effective mechanism to compute the actual level of semantic interoperability between two ontologies?"

To answer this question, recall the research contributions mentioned in Section 1.9.2: the novel analysis of semantic interoperability in terms of models and queries; the conceptual clarity achieved by narrowing the focus to one small kind of difference between ontologies; and a foundation for further study.

The research presented in this thesis is significant for two major reasons that are independent of whether the actual level of semantic interoperability can be computed in a given case.

The first reason is that the novel analysis of semantic interoperability in terms of models and queries can serve as a touchstone to assess claims about semantic interoperability, in two ways.

First, if a claim of semantic interoperability is being made in the context of formal ontologies with model-theoretic semantics, this thesis provides a basis for evaluating whether such a claim has standing. One is now in a position to challenge any such claims: "Since for your ontologies, semantics has something to do with models, can you explain how your definition of semantic interoperability relates to the models of the ontologies?"

Second, if a claim of semantic interoperability is being made in a context that is *not* concerned with formal ontologies and model-theoretic semantics, this thesis provides the basis to question the underlying notion of semantic interoperability. One is now in a position to ask: "Can you explain in detail exactly what you mean by 'semantics' and in exactly what sense your two ontologies are semantically interoperable according to your understanding of semantics?"

The research presented here also makes a contribution in that that readers of this research are equipped ask probing, detailed questions about semantic interoperability that *they might never before have been able to, or thought to, ask.* Thus, the research presented here serves not only as a touchstone for assessing claims of semantic interoperability, but also as *a springboard for launching new research questions* about exactly what it means for two ontologies to be semantically interoperable.

6.5 Conclusions

When two geospatial ontologies are formulated in a sublanguage of the language of first-order logic, and certain assumptions can be made about the finiteness of the individual models of the ontologies and of their model classes, then it is possible to compute the level of semantic interoperability between these ontologies. Regardless of whether or not the level of semantic interoperability between ontologies can be computed, the definitions of these different levels in terms of compatibility conditions between queries and models provides information and insights that have not appeared previously in the literature.

This study is foundational in that it provides base-level results of semantic interoperability for the case where two ontologies are the same in everything but the model-theoretic semantics of their primitive relation symbols. These base-level results are the sharpest that can be obtained, in the sense that relaxing any other assumptions of similarity—the number or names of elements and relations on the domain, the logical languages used, etc.—would increase (or at least not reduce) the variability between the ontologies, and so would make conclusions about semantic interoperability harder to draw.

Thus, any analysis of semantic interoperability that considers models and model classes would need to take the results of this thesis into account.

6.6 Future Work

Among several promising areas for future work are the following.

6.6.1 Incorporating Additional Heterogeneities

This thesis focused in detail on one particular difference between ontologies (a difference in the semantics of primitive relations). We have seen that the framework by Klein (2001) is closest in spirit to the analysis of this thesis. It is likely that further insights into semantic interoperability can be gained by considering other heterogeneities described by Klein, not just the differing semantics of primitives.

6.6.2 Explicating the Semantics in Methods for Integrating Ontologies

Much of the existing research on semantic heterogeneity and semantic interoperability is carried out so that different operations like integration or merging (Gomez-Perez et al., 2004; Kalfoglou and Schorlemmmer, 2003) can be accomplished, with 'good' results as far as the semantics goes.

Yet, in these operations, the most that is usually discussed are some kind of term-by-term equivalences, or inclusion relations. It would be interesting to consider the connections between:

1. the operations of integration and merging;

2. term-by-term equivalences and inclusion relations in carrying out these operations;

3. what happens to the semantics of the individual ontologies when they are merged using these equivalences and inclusion relations; and

4. a model-theoretic analysis of semantic interoperability.

6.6.3 Considering Infinite Rather than Finite Domains

Earlier chapters argued for the appropriateness of the assumptions of finiteness (of the conceptual domain and of the corresponding non-logical vocabulary) used in this thesis. It was also noted that these assumptions can render tractable for certain small ontologies the model-theoretic computations described in Chapter 5. Further, analysis of individual queries by proof-theoretic means is also decidable under these assumptions. That is, a terminating algorithm can always be found to decide the question of whether a query is a logical consequence of the ontology. It would interesting to investigate whether *tractable* algorithms could be developed for certain special cases if the finiteness assumptions are relaxed.

6.6.4 Considering Finer Distinctions Between Model Classes

There are more subtle kinds of analyses that one could make if one has available all models of each model class. For instance, even with disjoint model classes, one could consider the *tuples* that are common to both model classes. In this way, one could perhaps say something worthwhile about certain queries and their satisfaction in some or all models of the ontologies, even though such claims would not extend to *all* queries. Such an approach focuses directly on the models and the tuples that make them up.

A slightly different perspective would change the initial focus from the models to the ontologies themselves, focusing on partitioning the ontologies and considering the kinds of semantic interoperability that result from the partitioning. From this perspective one might ask whether a partition of the ontologies into subgroups of entities, relations, and axioms (e.g., just those entities 'Bangor,' 'Orono,' and 'Maine,' just the relation 'in,' and just the axioms of transitivity) results in a different level of semantic interoperability than that which holds for the full ontologies.

6.6.5 Exploring Relation to Translational Approach

It would be useful and informative to carefully trace out the different possible translational approaches of semantic interoperability that are based on notions like that of isomorphism. One focus of such an investigation could be on whether Level 3 semantic interoperability, as discussed in this thesis, is related to certain translational approaches. On the surface, Level 3 (or full) semantic interoperability would appear to be exactly what any translational approach to semantic interoperability seeks to find. Detailed examples need to be developed and worked out in order to verify whether or not this is indeed the case. Further, distinctions need to be drawn between the analysis in this thesis, which seeks to ascertain a given level of semantic interoperability, and the use of the translational approach, which seeks to create conditions on languages and their interpretations *so that* a certain kind of semantic interoperability is guaranteed to exist. Additionally, one might study under what circumstances (i.e., given what kinds of logical systems) such guarantees of semantic interoperability could be made.

6.6.6 Exploring Relation to More Abstract Approaches

It would be useful and informative to relate the rather focused analysis of this thesis to the broader and more abstract notions of semantic interoperability and ontology alignment presented in other works (e.g., Hitzler et al. (2006)).

6.6.7 Connecting to Ongoing Developments

The potential revival of OWL Space (Hobbs, 2006) could provide fundamental spatial constructs that could be broadly used in the geospatial community. In particular, it might be able to be employed in whatever ontology or ontologies NGA decides to create. If so, the notions of model classes and semantic interoperability will have role to play.

It would also be useful to explore how the ideas of semantic interoperability explicated in this thesis correspond to the ideas of semantic interoperability intimated in Kuhn (2005b), where the focus is on the semantics of service interfaces. In this vein, a useful stepping-off point could be the research begun in Farrugia (2002).

REFERENCES

Adams, D. (2006). Geospatial Ontologies - A View from NGA. *http://colab.cim3.net/file/work/SICoP/2006-06-20/2006-06-21/DAdams06212006.ppt*.

Agarwal, P. (2005). Ontological considerations in giscience. *International Journal of Geographical Information Science*, 19(5):501–536.

Baader, F. and Nutt, W. (2003). Basic description logics. In Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. F., editors, *The Description Logic Handbook: Theory, Implementation and Applications*, pages 43–95. Cambridge.

Beard-Tisdale, M. K. (2006). *Personal Communication*.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, May:34–43.

Bishr, Y. (1998). Overcoming the semantic and other barriers to gis interoperability. *International Journal of Geographic Information Science*, 12(4):299–314.

Boolos, G., Burgess, J., and Jeffrey, R. (2002). *Computability and Logic*. Cambridge, 4th edition.

Burrough, P. and Frank, A. (1996). *Geographic Objects with Indeterminate Boundaries*. Taylor and Francis.

Carnielli, W. and D'Ottaviano, I. (1997). Translations between logical systems: a manifesto. *Logique & Analyse*, 157:67–81.

Casati, R., Smith, B., and Varzi, A. (1998). *Ontological Tools for Geographic Representation*, pages 77–85. IOS Press.

Clementini, E. and Di Felice, P. (1997). Approximate topological relations. *International Journal of Approximate Reasoning*, 16:173–204.

Cohn, A. G. and Varzi, A. (2003). Mereotopological connection. *Journal of Philosophical Logic*, 32:357–390.

Cohn, A. M. and Hazarika, S. M. (2001). Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, 46(1-2):1–29.

Cox, S., Daisey, P., Lek, R., Portele, C., and Whiteside, A. (2003). OpenGIS Geography Markup Language GML Implementation Specification, Version 3.0. *http://portal.opengeospatial.org/files/*.

Ebbinghaus, H.-D., Flum, J., and Thomas, W. (1994). *Mathematical Logic*. Springer, 2nd edition.

Egenhofer, M. (1999). Introduction: Theories and concepts. In Goodchild, M., Egenhofer, M., Fegeas, R., and Kottman, C., editors, *Interoperating geographic information systems*, pages 1–4. Kluwer.

Egenhofer, M. (2003). Towards the geospatial semantic web. *ACM Transactions on GIS*, pages 1–4.

Egenhofer, M. J. (2004). Ontological foundations for geographic information science. In McMaster, R. and Usery, E., editors, *A Research Agenda for Geographic Information Science*. Taylor & Francis.

Egenhofer, M. J. and Franzosa, R. (1991). Point-set topological spatial relations. *International Journal of Geographical Information Systems*, 5(2):161–174.

Etchemendy, J. (1990). *The Concept of Logical Consequence*. Harvard University Press.

Euzenat, J. (2001). Towards a principled approach to semantic interoperability. *Workshop on Ontologies and Information Sharing, IJCAI 2001*, pages 19–25.

Farrugia, J. (2002). Logical systems: Towards protocols for web-based meaning negotiation. *AAAI 2002 Workshop on Meaning Negotiation July, 2002*, pages 56–59.

Farrugia, J. (2003). Model-theoretic semantics for the web. *Proceedings of the Twelfth International Conference on World Wide Web*, pages 29–38.

Fonseca, F. (2000). Ontology-driven geographic information systems. *Ph.D. Dissertation, University of Maine, Orono, Maine, USA.*

Fonseca, F., Egenhofer, M., Agouris, P., and Câmara, G. (2002). Using ontologies for integrated geographic information systems. *Transactions in GIS*, 6(3):231–257.

Goguen, J. (2005). Three perspectives on information integration. In Kalfoglou, Y., Schorlemmer, M., Sheth, A., Staab, S., and Uschold, M., editors, *Semantic Interoperability and Integration*, number 04391 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.

Gomez-Perez, A., Fernandez-Lopez, M., and Corcho, O. (2004). *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer.

Goodchild, M., Egenhofer, M., Fegeas, R., and Kottman, C. (1999). *Interoperating geographic information systems*. Kluwer.

Gruber, T. (1993). A translation approach to portable ontology specification. *Knowledge Acquisition*, 5(2):199–200.

Grüninger (2004). Model-theoretic approaches to semantic integration. *Dagstuhl Seminar N 04391 on Semantic Interoperability and Integration (http://www.dagstuhl.de/04391/Materials/)*.

Grüninger, M. and Kopena, J. (2005). Semantic integration through invariants. *AI Magazine*, 26(1):11–24.

Guarino, N. (1998). Formal ontology and information systems. *Formal Ontology in Information Systems*, pages 2–15.

Guarino, N. and Giaretta, P. (1995). Ontologies and knowledge bases: Towards a terminological clarification. In Mars, N., editor, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, pages 25–32. IOS Press.

Hakimpour, F. and Geppert, A. (2001). Resolving semantic heterogeneity in schema integration: an ontology based approach. *Formal Ontology in Information Systems, FOIS'O1*, pages 297–308.

Hakimpour, F. and Geppert, A. (2002). Global schema generation using formal ontologies. In *Proceedings of the 21st International Conference on Conceptual Modeling ER2002*, pages 307–321. Springer.

Hakimpour, F. and Timpf, S. (2002). A step toward geodata integration using formal ontologies. *5th Agile Conference on Geographic Information Systems*.

Hayes, P. (2004). RDF Semantics. *http://www.w3.org/TR/rdf-mt/*.

Heflin, J. and Hendler, J. (2000). Semantic interoperability on the web. *Proceedings of Extreme Markup Languages 2000*, pages 111–120.

Hitzler, P., Euzenat, J., Krotzsch, M., Serafini, L., Stuckenschmidt, H., Wache, H., and Zimmermann, A. (2006). D2.2.5 integrated view and comparison of alignment semantics. *Knowledgeweb: realizing the semantic web*, http://www.inrialpes.fr/exmo/cooperation/kweb/heterogeneity/deli/kweb-225.pdf:1–36.

Hobbs, J. (1985). Granularity. *Proceedings, Ninth International Joint Conference on Artificial Intelligence*.

Hobbs, J. (2006). Interoperability among geospatial ontologies. *http://colab.cim3.net/file/work/SICoP/2006-06-20/JHobbs06202006.ppt*.

Hodges, W. (1997). *A Shorter Model Theory*. Cambridge.

Kalfoglou, Y. and Schorlemmmer, M. (2003). Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1-2):1–31.

Klein, M. (2001). Combining and relating ontologies: an analysis of problems and solutions. *Workshop on Ontologies and Information Sharing, IJCAI 2001*, pages 53–62.

Klyne, G. and Carroll, J. (2004). Resource Description Framework (RDF): Concepts and Abstract Syntax. *http://www.w3.org/TR/rdf-concepts/*.

Kuhn, W. (2005a). Geospatial semantics: Why, of what, and how? *Journal on Data Semantics, Special Issue on Semantic-based Geographical Information Systems, LNCS 3534*, Lecture Notes in Computer Science(4):1–24.

Kuhn, W. (2005b). Semantics of what? In Goodchild, M., Kyriakidis, P., Rice, M., and Schneider, P., editors, *Report of the NCGIA Specialist Meeting on Spatial Web, Santa Barbara, December 2-4, 2004*, pages 48–53.

Lassila, O. and McGuinness, D. (2001). The role of frame-based representation on the semantic web. *KSL Tech Report Number KSL-01-02, Knowledge Systems Laboratory, Stanford University*.

Lipski, W. (1981). On databases with incomplete information. *Journal of the ACM*, 28(1):41–70.

Manzano, M. (1999). *Model Theory*. Oxford.

Mark, D., Smith, B., and Tversky, B. (1999). *Ontology and Geographic Objects: An Empirical Study of Cognitive Categorization*, pages 283–298. Berlin.

Masolo, C. (2000). *Criteri di confronto e costruzione di teorie assiomatiche per la rappresentazione della conoscenza: ontologie dello spazio e del tempo*. PhD thesis, Univ. di Padova, Padova, Italy.

McGuinness, D. (2002). Ontologies come of age. In Fensel, D., Hendler, J., Lieberman, H., and Wahlster, W., editors, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, pages xxx–yyy. MIT Press.

Meseguer, J. (1998). Formal interoperability. In *Proceedings of the 1998 Conference on Mathematics in Artificial Intelligence, Florida, Jan. 1998*.

Obrst, L. (2003). Ontologies for semantically interoperable systems. *Proceedings of the twelfth international conference on Information and knowledge management*, pages 366–369.

Patel-Schneider, P. F. (2006). *Personal Communication*.

Patel-Schneider, P. F., Hayes, P., and Horrocks, I. (2004). OWL Web Ontology Language Semantics and Abstract Syntax. *http://www.w3.org/TR/owl-semantics/*.

Rahm, E. and Bernstein, P. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350.

Randell, D. A., Cohn, A. G., and Cui, Z. (1992). A spatial logic based on regions and connection. *Proc. 3rd Int. Conf. on Knowledge Representation and Reasoning*, pages 165–176.

Reiter, R. (1978). On closed world data bases. In Gallaire, H. and Minker, J., editors, *Logic and data bases*, pages 55–76. Plenum Press.

Renz, J. (2002). *Qualitative Spatial Reasoning with Topological Information*. Springer.

Rigaux, P., Scholl, M., and Voisard, A. (2002). *Spatial Databases with Application to GIS*. Morgan Kaufmann.

Schorlemmer, M. and Kalfoglou, Y. (2004). Formal support for representing and automating semantic interoperability. In *The Semantic Web: Research and Applications. First European Semantic Web Symposium, ESWS 2004. Heraklion, Crete, Greece, May 10-12, 2004. Proceedings*, pages 45–60.

Shaboldt, N., Hall, W., and Berners-Lee, T. (2006). The semantic web revisited. *IEEE Intelligent Systems*, May/June 2006:96–101.

Shekar, S. and Chawla, S. (2002). *Spatial Databases: A Tour*. Prentice Hall.

Sheth, A. (1999). Changing focus on interoperability from system, syntax, structure to semantics. In Goodchild, M., Egenhofer, M., Fegeas, R., and Kottman, C., editors, *Interoperating geographic information systems*, pages 5–29. Kluwer.

Sheth, A. and Larson, J. (1990). Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236.

Smith, B. (1996). Mereotopology: A theory of parts and boundaries. *Data and Knowledge Engineering*, 20:287–303.

Smith, B. and Mark, D. (1998). *Ontology and Geographic Kinds*, pages 308–320. International Geographical Union.

Smith, B. and Welty, C. (2001). Ontology: Towards a new synthesis. In Welty, C. and B, S., editors, *Formal Ontologies in Information Systems: FOIS '01*, pages iii–ix. ACM.

Sondheim, M., Gardels, K., and Buehler, K. (1999). *GIS Interoperability*, pages 347–358. Wiley.

Sowa, J. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole.

Stell, J. (2004). Part and complement: Fundamental concepts in spatial relations. *Annals of Artificial Intelligence and Mathematics*, 41:1–18.

Suppes, P. (2002). *Representation and Invariance of Scientific Structures*. CSLI.

Taylor, A. G. (2004). *The organization of information*. Libraries Unlimited, 2nd edition.

Vianu, V. (1997). Databases and finite-model theory. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 31:97–148.

Visser, P., Jones, D., Bench-Capon, T., and Shave, M. (1998). Assessing heterogeneity by classifying ontology mismatches. *Formal Ontology in Information Systems*, pages 148–162.

Visser, U., Stuckenschmidt, H., Schuster, G., and Voegele, T. (2002). Ontologies for geographic information processing. *Computers in Geosciences*, 28(1):103–117.

Zhang, J. and Zhang, H. (1995). Sem: a system for enumerating models. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montreal, Quebec, Canada, August 20-25 1995*, pages 298–303.

APPENDICES

Appendix A

SAMPLE ONTOLOGIES AND PROGRAM OUTPUT

A.1 ch5O1.in

The SEM file for $ch5O1.in$ follows.

```
%% Chapter 5 Ontology 1


%% Sorts
( thing : Route2, Orono, Bangor, Maine, I95, river1 )


%% Functions

{ state : thing -> BOOL }
{ town  : thing -> BOOL }
{ road  : thing -> BOOL }
{ river : thing -> BOOL }

{ in       : thing thing -> BOOL }
{ through  : thing thing -> BOOL }


%% Clauses (Axioms)


% IN is reflexive, antisymmetric, and transitive

[ in(x,x) ]
[ x=y | -in(x,y) | -in(y,x) ]
[ -in(x,y) | -in(y,z) | in(x,z)]



% THROUGH is antisymmetric and transitive

[ x=y | -through(x,y) | -through(y,x) ]
[ -through(x,y) | -through(y,z) | through(x,z)]
```

```
% towns, roads, rivers, and states are all
% pairwise disjoint

[ -town(x)  | -road(x) ]
[ -town(x)  | -river(x) ]
[ -town(x)  | -state(x) ]
[ -road(x)  | -river(x) ]
[ -road(x)  | -state(x) ]
[ -river(x) | -state(x) ]



% if x and y are different roads/rivers/towns/states,
% then x can't be IN y

[ x=y | -road(x)  | -road(y)  | -in(x,y) ]
[ x=y | -road(x)  | -road(y)  | -in(y,x) ]

[ x=y | -river(x) | -river(y) | -in(x,y) ]
[ x=y | -river(x) | -river(y) | -in(y,x) ]

[ x=y | -town(x)  | -town(y)  | -in(x,y) ]
[ x=y | -town(x)  | -town(y)  | -in(y,x) ]

[ x=y | -state(x) | -state(y) | -in(x,y) ]
[ x=y | -state(x) | -state(y) | -in(y,x) ]



% if x is a town/state/river and y is a road,
% then x is NOT IN y

[ -town(x)  | -road(y) | -in(x,y) ]
[ -state(x) | -road(y) | -in(x,y) ]
[ -river(x) | -road(y) | -in(x,y) ]



% if x is a road/town/state and y is a river,
% then x is NOT IN y

[ -road(x)  | -river(y) | -in(x,y) ]
```

```
[ -town(x)  | -river(y) | -in(x,y) ]
[ -state(x) | -river(y) | -in(x,y) ]



% if x is a state and y is a town,
% then x is NOT IN y

[ -state(x)  | -town(y) | -in(x,y) ]



% if x is a state and y is a town,
% then x does NOT go THROUGH y

[ -state(x)  | -town(y) | -through(x,y) ]



% if x is a town/state/river and y is a road,
% then x does NOT go THROUGH y

[ -town(x)  | -road(y) | -through(x,y) ]
[ -state(x) | -road(y) | -through(x,y) ]
[ -river(x) | -road(y) | -through(x,y) ]



% if x is a town/state and y is a river,
% then x does NOT go THROUGH y

[ -town(x)  | -river(y) | -through(x,y) ]
[ -state(x) | -river(y) | -through(x,y) ]



% if x is a town/state and y is a town,
% then x does NOT go THROUGH y

[ -town(x)  | -town(y) | -through(x,y) ]
[ -state(x) | -town(y) | -through(x,y) ]


% if x goes THROUGH y, then y is NOT IN x
```

```
 [ -through(x,y) | -in(y,x) ]



% data

[ town(Orono) ]
[ town(Bangor) ]
[ road(Route2) ]
[ road(I95) ]
[ state(Maine) ]
[ river(river1) ]

[ in(Route2, Orono) ]
[ in(Route2, Bangor) ]
[ in(Orono, Maine) ]
[ in(Bangor, Maine) ]
[ in(river1, Maine) ]

[ through(river1, Bangor) ]
[ through(river1, Orono) ]
[ through(river1, Maine) ]

[ through(I95, Bangor) ]
[ through(I95, Orono) ]
[ through(I95, Maine) ]

[ through(Route2, Bangor) ]
[ through(Route2, Orono) ]
[ through(Route2, Maine) ]
```

There are 800 models of $ch5O1.in$. A model for $ch5O1.in$ as it is produced by SEM looks like the following, which is the first model in SEM's output for $ch5O1.in$.

```
******** Model 1 ********

 state(Route2) = $F
 state(Orono) = $F
 state(Bangor) = $F
 state(Maine) = $T
 state(I95) = $F
 state(river1) = $F

 town(Route2) = $F
 town(Orono) = $T
```

```
town(Bangor) = $T
town(Maine) = $F
town(I95) = $F
town(river1) = $F

road(Route2) = $T
road(Orono) = $F
road(Bangor) = $F
road(Maine) = $F
road(I95) = $T
road(river1) = $F

river(Route2) = $F
river(Orono) = $F
river(Bangor) = $F
river(Maine) = $F
river(I95) = $F
river(river1) = $T

in(Route2,Route2) = $T
in(Route2,Orono) = $T
in(Route2,Bangor) = $T
in(Route2,Maine) = $T
in(Route2,I95) = $F
in(Route2,river1) = $F
in(Orono,Route2) = $F
in(Orono,Orono) = $T
in(Orono,Bangor) = $F
in(Orono,Maine) = $T
in(Orono,I95) = $F
in(Orono,river1) = $F
in(Bangor,Route2) = $F
in(Bangor,Orono) = $F
in(Bangor,Bangor) = $T
in(Bangor,Maine) = $T
in(Bangor,I95) = $F
in(Bangor,river1) = $F
in(Maine,Route2) = $F
in(Maine,Orono) = $F
in(Maine,Bangor) = $F
in(Maine,Maine) = $T
in(Maine,I95) = $F
in(Maine,river1) = $F
in(I95,Route2) = $F
in(I95,Orono) = $F
```

```
in(I95,Bangor) = $F
in(I95,Maine) = $F
in(I95,I95) = $T
in(I95,river1) = $F
in(river1,Route2) = $F
in(river1,Orono) = $F
in(river1,Bangor) = $F
in(river1,Maine) = $T
in(river1,I95) = $F
in(river1,river1) = $T

through(Route2,Route2) = $F
through(Route2,Orono) = $T
through(Route2,Bangor) = $T
through(Route2,Maine) = $T
through(Route2,I95) = $F
through(Route2,river1) = $F
through(Orono,Route2) = $F
through(Orono,Orono) = $F
through(Orono,Bangor) = $F
through(Orono,Maine) = $F
through(Orono,I95) = $F
through(Orono,river1) = $F
through(Bangor,Route2) = $F
through(Bangor,Orono) = $F
through(Bangor,Bangor) = $F
through(Bangor,Maine) = $F
through(Bangor,I95) = $F
through(Bangor,river1) = $F
through(Maine,Route2) = $F
through(Maine,Orono) = $F
through(Maine,Bangor) = $F
through(Maine,Maine) = $F
through(Maine,I95) = $F
through(Maine,river1) = $F
through(I95,Route2) = $F
through(I95,Orono) = $T
through(I95,Bangor) = $T
through(I95,Maine) = $T
through(I95,I95) = $F
through(I95,river1) = $F
through(river1,Route2) = $F
through(river1,Orono) = $T
through(river1,Bangor) = $T
through(river1,Maine) = $T
```

```
through(river1,I95) = $F
through(river1,river1) = $F
```

Subsequent models in this section have a similar form. The only differences between models are the values of $\$T$ (for True) and $\$F$ (for False) assigned to the relations.

A.2 ch5O2.in

The SEM file for $ch5O2.in$ follows.

```
%% Chapter 5 Ontology 2


%% Sorts
( thing : Route2, Orono, Bangor, Maine, I95, river1 )


%% Functions

{ state : thing -> BOOL }
{ town  : thing -> BOOL }
{ road  : thing -> BOOL }
{ river : thing -> BOOL }

{ in       : thing thing -> BOOL }
{ through  : thing thing -> BOOL }


%% Clauses (Axioms)


% IN is reflexive, antisymmetric, and transitive

[ in(x,x) ]
[ x=y | -in(x,y) | -in(y,x) ]
[ -in(x,y) | -in(y,z) | in(x,z)]



% THROUGH is antisymmetric and transitive

[ x=y | -through(x,y) | -through(y,x) ]
[ -through(x,y) | -through(y,z) | through(x,z)]
```

```
% towns, roads, rivers, and states are
% all pairwise disjoint

[ -town(x)  | -road(x) ]
[ -town(x)  | -river(x) ]
[ -town(x)  | -state(x) ]
[ -road(x)  | -river(x) ]
[ -road(x)  | -state(x) ]
[ -river(x) | -state(x) ]



% if x and y are different roads/rivers/towns/states,
% then x is NOT IN y

[ x=y | -road(x)  | -road(y) | -in(x,y) ]
[ x=y | -road(x)  | -road(y) | -in(y,x) ]

[ x=y | -river(x) | -river(y) | -in(x,y) ]
[ x=y | -river(x) | -river(y) | -in(y,x) ]

[ x=y | -town(x)  | -town(y) | -in(x,y) ]
[ x=y | -town(x)  | -town(y) | -in(y,x) ]

[ x=y | -state(x) | -state(y) | -in(x,y) ]
[ x=y | -state(x) | -state(y) | -in(y,x) ]



% if x is a town/state/river and y is a road,
% then x is NOT IN y

[ -town(x)  | -road(y) | -in(x,y) ]
[ -state(x) | -road(y) | -in(x,y) ]
[ -river(x) | -road(y) | -in(x,y) ]



% if x is a road/town/state and y is a river,
% then x is NOT IN y

[ -road(x)  | -river(y) | -in(x,y) ]
```

```
[ -town(x)  | -river(y) | -in(x,y) ]
[ -state(x) | -river(y) | -in(x,y) ]



% if x is a state and y is a town,
% then x is NOT IN y

[ -state(x)  | -town(y) | -in(x,y) ]



% if x is a state and y is a town,
% then x does NOT go THROUGH y

[ -state(x)  | -town(y) | -through(x,y) ]



% if x is a town/state and y is a road,
% then x does NOT go THROUGH y

[ -town(x)  | -road(y) | -through(x,y) ]
[ -state(x) | -road(y) | -through(x,y) ]



% if x is a town/state/road and y is a river,
% then x does NOT go THROUGH y

[ -town(x)  | -river(y) | -through(x,y) ]
[ -state(x) | -river(y) | -through(x,y) ]
[ -road(x)  | -river(y) | -through(x,y) ]



% if x is a town/state and y is a town,
% then x does NOT go THROUGH y

[ -town(x)  | -town(y) | -through(x,y) ]
[ -state(x) | -town(y) | -through(x,y) ]


% if x goes THROUGH y, then y is NOT IN x
```

```
 [ -through(x,y) | -in(y,x) ]



% data

[ town(Orono) ]
[ town(Bangor) ]
[ road(Route2) ]
[ road(I95) ]
[ state(Maine) ]
[ river(river1) ]

[ in(Orono, Maine) ]
[ in(Bangor, Maine) ]

[ through(river1, Bangor) ]
[ through(river1, Orono) ]
[ through(river1, Maine) ]

[ through(I95, Bangor) ]
[ through(I95, Orono) ]
[ through(I95, Maine) ]

[ through(Route2, Bangor) ]
[ through(Route2, Orono) ]
[ through(Route2, Maine) ]

[ -through(Orono, Maine) ]
[ -through(Bangor, Maine) ]
```

There are 1250 models of $ch5O2.in$.

A.3 Program Output

The first Perl program in AppendixB produced the following output:

```
number of models in ontology 1 is: 800
number of models in ontology 2 is: 1250
number of models in ontology 2 but not in ontology 1: 1190
number of models in ontology 1 but not in ontology 2: 740
number of models in both ontology 1 and ontology 2 is: 60
```

A.4 ch5O3.in

The SEM file for $ch5O3.in$ follows.

```
%% Chapter 5 Ontology 3


%% Sorts
( thing : Route2, Orono, Bangor, Maine, I95, river1 )


%% Functions

{ state : thing -> BOOL }
{ town  : thing -> BOOL }
{ road  : thing -> BOOL }
{ river : thing -> BOOL }

{ in       : thing thing -> BOOL }
{ through  : thing thing -> BOOL }


%% Clauses (Axioms)


% IN is reflexive, antisymmetric, transitive

[ in(x,x) ]
[ x=y | -in(x,y) | -in(y,x) ]
[ -in(x,y) | -in(y,z) | in(x,z)]



% THROUGH is antisymmetric

[ x=y | -through(x,y) | -through(y,x) ]



% towns, roads, rivers, and states are
% all pairwise disjoint

[ -town(x)  | -road(x) ]
[ -town(x)  | -river(x) ]
[ -town(x)  | -state(x) ]
[ -road(x)  | -river(x) ]
```

```
[ -road(x)  | -state(x) ]
[ -river(x) | -state(x) ]



% if x and y are different roads/rivers/towns/states,
% then x is NOT IN y

[ x=y | -road(x)  | -road(y) | -in(x,y) ]
[ x=y | -road(x)  | -road(y) | -in(y,x) ]

[ x=y | -river(x) | -river(y) | -in(x,y) ]
[ x=y | -river(x) | -river(y) | -in(y,x) ]

[ x=y | -town(x)  | -town(y) | -in(x,y) ]
[ x=y | -town(x)  | -town(y) | -in(y,x) ]

[ x=y | -state(x) | -state(y) | -in(x,y) ]
[ x=y | -state(x) | -state(y) | -in(y,x) ]



% if x is a town/state/river and y is a road,
% then x is NOT IN y

[ -town(x)  | -road(y) | -in(x,y) ]
[ -state(x) | -road(y) | -in(x,y) ]
[ -river(x) | -road(y) | -in(x,y) ]



% if x is a road/town/state and y is a river,
% then x is NOT IN y

[ -road(x)  | -river(y) | -in(x,y) ]
[ -town(x)  | -river(y) | -in(x,y) ]
[ -state(x) | -river(y) | -in(x,y) ]



% if x is a state and y is a town,
% then x doesn’t be IN y

[ -state(x)  | -town(y) | -in(x,y) ]
```

```
% if x is a state and y is a town,
% then x does NOT go THROUGH y

[ -state(x)  | -town(y) | -through(x,y) ]



% if x is a town/state/river and y is a road,
% then x does NOT go THROUGH y

[ -town(x)  | -road(y) | -through(x,y) ]
[ -state(x) | -road(y) | -through(x,y) ]
[ -river(x) | -road(y) | -through(x,y) ]



% if x is a town/state/road and y is a river,
% then x does NOT go THROUGH y

[ -town(x)  | -river(y) | -through(x,y) ]
[ -state(x) | -river(y) | -through(x,y) ]
[ -road(x)  | -river(y) | -through(x,y) ]



% if x is a town/state and y is a town,
% then x does NOT go THROUGH y

[ -town(x)  | -town(y)  | -through(x,y) ]
[ -state(x) | -town(y)  | -through(x,y) ]



% data

[ town(Orono) ]
[ town(Bangor) ]
[ road(Route2) ]
[ road(I95) ]
[ state(Maine) ]
[ river(river1) ]

[ in(Orono, Maine) ]
```

```
[ in(river1, Maine) ]

[ through(river1, Bangor) ]
[ through(river1, Orono) ]
%[ through(river1, Maine) ]

[ through(I95, Bangor) ]
[ through(I95, Orono) ]
%[ through(I95, Maine) ]

[ through(Route2, Bangor) ]
[ through(Route2, Orono) ]
%[ through(Route2, Maine) ]

[ -through(Bangor, Maine) ]
[ -through(Orono, Maine) ]
```

There are 93,696 models of $ch5O3.in$.

A.5 ch5O4.in

The SEM file for $ch5O4.in$ follows.

```
%% Chapter 5 Ontology 4


%% Sorts
( thing : Route2, Orono, Bangor, Maine, I95, river1 )


%% Functions

{ state : thing -> BOOL }
{ town  : thing -> BOOL }
{ road  : thing -> BOOL }
{ river : thing -> BOOL }

{ in       : thing thing -> BOOL }
{ through  : thing thing -> BOOL }


%% Clauses (Axioms)
```

```
% IN is reflexive, antisymmetric, transitive

[ in(x,x) ]
[ x=y | -in(x,y) | -in(y,x) ]
[ -in(x,y) | -in(y,z) | in(x,z)]



% THROUGH is antisymmetric

[ x=y | -through(x,y) | -through(y,x) ]



% towns, roads, rivers, and states are
% all pairwise disjoint

[ -town(x)  | -road(x) ]
[ -town(x)  | -river(x) ]
[ -town(x)  | -state(x) ]
[ -road(x)  | -river(x) ]
[ -road(x)  | -state(x) ]
[ -river(x) | -state(x) ]



% if x and y are different roads/rivers/towns/states,
% then x is NOT IN y

[ x=y | -road(x)  | -road(y) | -in(x,y) ]
[ x=y | -road(x)  | -road(y) | -in(y,x) ]

[ x=y | -river(x) | -river(y) | -in(x,y) ]
[ x=y | -river(x) | -river(y) | -in(y,x) ]

[ x=y | -town(x)  | -town(y) | -in(x,y) ]
[ x=y | -town(x)  | -town(y) | -in(y,x) ]

[ x=y | -state(x) | -state(y) | -in(x,y) ]
[ x=y | -state(x) | -state(y) | -in(y,x) ]



% if x is a town/state/river and y is a road,
```

```
% then x is NOT IN y

[ -town(x)  | -road(y) | -in(x,y) ]
[ -state(x) | -road(y) | -in(x,y) ]
[ -river(x) | -road(y) | -in(x,y) ]



% if x is a road/town/state and y is a river,
% then x is NOT IN y

[ -road(x)  | -river(y) | -in(x,y) ]
[ -town(x)  | -river(y) | -in(x,y) ]
[ -state(x) | -river(y) | -in(x,y) ]



% if x is a state and y is a town,
% then x doesn't be IN y

[ -state(x)  | -town(y) | -in(x,y) ]



% if x is a state and y is a town,
% then x does NOT go THROUGH y

[ -state(x)  | -town(y) | -through(x,y) ]



% if x is a town/state/river and y is a road,
% then x does NOT go THROUGH y

[ -town(x)  | -road(y) | -through(x,y) ]
[ -state(x) | -road(y) | -through(x,y) ]
[ -river(x) | -road(y) | -through(x,y) ]



% if x is a town/state and y is a river,
% then x does NOT go THROUGH y

[ -town(x)  | -river(y) | -through(x,y) ]
[ -state(x) | -river(y) | -through(x,y) ]
```

```
%[ -road(x)  | -river(y) | -through(x,y) ]



% if x is a town/state and y is a town,
% then x does NOT go THROUGH y

[ -town(x)  | -town(y)  | -through(x,y) ]
[ -state(x) | -town(y)  | -through(x,y) ]



% data

[ town(Orono) ]
[ town(Bangor) ]
[ road(Route2) ]
[ road(I95) ]
[ state(Maine) ]
[ river(river1) ]

[ in(Orono, Maine) ]
[ in(river1, Maine) ]

[ through(river1, Bangor) ]
[ through(river1, Orono) ]
%[ through(river1, Maine) ]

[ through(I95, Bangor) ]
%[ through(I95, Orono) ]
[ through(I95, Maine) ]

[ through(Route2, Bangor) ]
[ through(Route2, Orono) ]
[ through(Route2, Maine) ]

[ -through(Bangor, Maine) ]
[ -through(Orono, Maine) ]
```

There are 187,392 models of $ch5O4.in$.

A.6 Program Output

Below is the output of the second Perl program (Appendix B), used to implement the heuristic for Scenario 2.

```
p1 is 25015; p2 is 22703
q1 is 12731; q2 is 154842
model 25015 in c53.out found as model 47829 in c54.out
model 22703 in c53.out not found in c54.out
model 12731 in c54.out not found in c53.out
model 154842 in c54.out not found in c53.out
```

Appendix B

PERL CODE FOR SCENARIOS 1 AND 2

B.1 Perl Code for Scenario 1, Exact

```
#!/usr/bin/perl -w
use strict;
###########
# ------------------------------------------------------
#
# Purpose: Find the model-class relation (overlaps,
#          contains, contained_by, disjoint, or
#          identical) between two geospatial
#          ontologies, using SEM output files for
#          similar ontologies -- ontologies with the
#          same signature, but different data or axioms.
#
# Method:
#         Parse SEM files and store models as arrays
#         @models1 and @models2, where elements of
#         the arrays are strings of 1's and 0's
#         corresponding to truth values of SEM output.
#         This can be done easily, because SEM outputs
#         the truth values in so that if two ontologies
#         have the same signature, then truth values of
#         the models of one ontologies can be read in
#         the same way as the truth values for the
#         models of the other ontology.
# ------------------------------------------------------
###########


my $i;
my $line = '';
my $model = '';

my $file1="c51.out";
my $file2="c52.out";
```

```
my @models1 = ();
my @models2 = ();;

my $numModels1;
my $numModels2;
my $inOnt1Only;
my $inOnt2Only;
my $inBothOnts;

my $junk;
my %seen;
my $item;
my @keys;
my $key;

# my $elt;
# $elt = $array[ rand @array ];



# ------------------------------------------------------
#
# build @models1, each element a string of 1's and 0's
#
# ------------------------------------------------------


open(INFILE1, "$file1") or die "Cant' open $file1 $!\n";

while ($line = <INFILE1>) {

  # when hit one of these lines, will have full
  # model, except for 1st time which will be
  # null, so shift it away after the loop

  if ( ($line =~ (/\* Model /)) ||
       ($line =~ /Number of models/) ) {

      $model =~ s/.*(\$F)/0/g;
      $model =~ s/.*(\$T)/1/g;
      $model =~ s/\n//g;
      push(@models1,$model);
      $line = '';
  }
```

```
  if ( $line =~ (/^ \w+/) ) {
    $model .= $line;
  }

} # end while ($line = <INFILE1>)

close(INFILE1);


shift @models1; # since first element is null

$model = '';
$line = '';


# ------------------------------------------------------
#
# build @models2, each element a string of 1's and 0's
#
# ------------------------------------------------------

open(INFILE2, "$file2") or die "Cant' open $file2 $!\n";

while ($line = <INFILE2>) {

  if ( ($line =~ (/\* Model /)) ||
     ($line =~ /Number of models/) ) {

    $model =~ s/.*(\$F)/0/g;
    $model =~ s/.*(\$T)/1/g;
    $model =~ s/\n//g;
    push(@models2,$model);
    $line = '';
  }

  if ( $line =~ (/^ \w+/) ) {
    $model .= $line;
  }

} # end while ($line = <INFILE1>)

close(INFILE1);
```

```
shift @models2; # since first element is just null


$numModels1 = $#models1 + 1;
$numModels2 = $#models2 + 1;
print "number of models in ontology 1 is: $numModels1\n";
print "number of models in ontology 2 is: $numModels2\n";



# ------------------------------------------------------
# find those items in @Models2 but not in @Models1
# ------------------------------------------------------


%seen = ();
my @models2Only = ();
my $j = 0;

foreach $model (@models1) {$seen{$model} = 1; }

foreach $model (@models2) {
  unless ($seen{$model}) {
    push(@models2Only,$model);
    $j++;
  }
}

$inOnt2Only = $#models2Only + 1;
print "number of models in ontology 2";
print " but not in ontology 1: $inOnt2Only\n";



#-------------------------------------------
$model = '';
%seen = ();
my @models1Only = ();
$j = 0;
#-------------------------------------------


foreach $model (@models2) {$seen{$model} = 1; }
```

```
foreach $model (@models1) {
  unless ($seen{$model}) {
    push(@models1Only,$model);
    $j++;
  }
}

$inOnt1Only = $#models1Only + 1;
print "number of models in ontology 1";
print " but not in ontology 2: $inOnt1Only\n";


my @union = ();
my @intersection = ();
my @diff = ();
my %union = ();
my %intersection = ();
my %count = ();

foreach $model (@models1) {$union{$model} = 1; }

foreach $model (@models2) {
  if ( $union{$model} ) { $intersection{$model} = 1;}
  $union{$model} = 1;
}

@intersection = keys %intersection;

$inBothOnts = $#intersection + 1;
print "number of models in both ontology 1";
print " and ontology 2 is: $inBothOnts\n";

#-----------------------------------------------
```

B.2 Perl Code for Scenario 2, Heuristic

The Perl code for implementing the heuristic in Scenario 2 is:

```
#!/usr/bin/perl -w
use strict;
use locale;

#########
#-------------------------------------------------------
#
# Purpose: test whether random models from one ontology
#          are in the model class of another ontology
#
#
#
# Method:  hard-code values $M and $N, the values
#
#
#-------------------------------------------------------
#########


my $i;
my $line = '';
my $model = '';

my $file1="c53.out";
my $file2="c54.out";
#my $file1="c51.out";
#my $file2="c52.out";

my $M = 93696;
my $N = 187392;

#my $M = 800;
#my $N = 1250;

my @M = (1..$M);
my @N = (1..$N);

my @p;
my $p;

my $randomp1  = $M[ rand @M ];
my $randomp2  = $M[ rand @M ];
```

```
push(@p, $randomp1, $randomp2);

my $pInOj = 0;
my $qInOi = 0;

my @q;
my $q;

my $randomq1  = $N[ rand @N ];
my $randomq2  = $N[ rand @N ];
push(@q, $randomq1, $randomq2);

#@p = (195,189);
#@q = (1239,1245);

print "p1 is $p[0]; p2 is $p[1]\n";
print "q1 is $q[0]; q2 is $q[1]\n";

my $next;


 foreach $p (@p) {

  $next = $p + 1;

  open(INFILE1, "$file1") or die "Can't open $file1\n";
  my $foundFlag = 0;


  while ($line = <INFILE1>) {


    # leaves last model hanging,
    # to be picked up outside while loop

    if ( $line =~ /\* Model $p/ ) {

      #print "\nCURRENT is $p\n";
      $foundFlag = 1;
      $line  = '';
      $model  = '';
      next;
    }

    if ( ($line =~  /\* Model $next/)  ||
```

```
     ($line =~ /^\*{10}/) )                {
    $foundFlag = 2;
    #print "\n\ncurrent is $p \n$model\n\n";
    last;
  }

  if ($foundFlag == 1) {
    $model .= $line;
    $model =~ s/\n//g;
  }


} # end while ($line = <INFILE1>)


my $modelInHand = $model;
close(INFILE1);


#print "\n\nmodel in hand is $p\n";




$line = '';
$model = '';

$i = 0;
open(INFILE2, "$file2") or die "Can't open $file2\n";

while ($line = <INFILE2>) {

  if ( $line =~ /\* Model / ) {

    if ($model eq $modelInHand) {
      print "model $p in $file1 found as";
      print" model $i in $file2\n";
      $pInOj = 1;
      last;
    }

    if ($model ne '') {
      # print "********** $i **********\n$model\n";
    }
```

```
      $i++;
      $line  = '';
      $model  = '';
      next;
    }
    if ( $line =~ /^ \w+/ ) {
      $model .= $line;
      $model =~ s/\n//g;
    }


  } # end while ($line = <INFILE2>)


  close(INFILE2);

  if (!$pInOj) {
    print "model $p in $file1 not found in $file2 \n";
  }

  $pInOj = 0;

} # end foreach


# now test to see if model q in $file2 is in $file1

 foreach $q (@q) {

  $next = $q + 1;

  open(INFILE2, "$file2") or die "Can't open $file2\n";
  my $foundFlag = 0;


  while ($line = <INFILE2>) {


    # leaves last model hanging,
    # to be picked up outside while loop

    if ( $line =~ /\* Model $q/ ) {

      #print "\nCURRENT is $q\n";
      $foundFlag = 1;
```

```
    $line  = '';
    $model  = '';
    next;
  }

  if ( ($line =~  /\* Model $next/)  ||
     ($line =~ /^\*{10}/) )                  {
    $foundFlag = 2;
    #print "\n\ncurrent is $q \n$model\n\n";
    last;
  }

  if ($foundFlag == 1) {
    $model .= $line;
    $model =~ s/\n//g;
  }


} # end while ($line = <INFILE2>)


my $modelInHand = $model;
close(INFILE2);


# print "\n\nmodel in hand $q is \n\n $modelInHand\n\n";




$line = '';
$model = '';

$i = 0;
open(INFILE1, "$file1") or die "Can't open $file1\n";

while ($line = <INFILE1>) {

  if ( $line =~ /\* Model / ) {

    if ($model eq $modelInHand) {
      print "model $q in $file2 found as";
      print" model $i in $file1\n";
      $qInOi = 1;
      last;
```

```
      }

      if ($model ne '') {
        # print "********** $i ***********\n$model\n";
      }

      $i++;
      $line  = '';
      $model  = '';
      next;
    }
    if ( $line =~ /^ \w+/ ) {
      $model .= $line;
      $model =~ s/\n//g;
    }


  } # end while ($line = <INFILE1>)


  close(INFILE1);

  if (!$qInOi) {
    print "model $q in $file2 not found in $file1 \n";
  }

  $qInOi = 0;

} # end foreach
```

BIOGRAPHY OF THE AUTHOR

James A. Farrugia graduated in 1986 from East Stroudsburg University with a Bachelor's degree in Mathematics. In 1994 he graduated from The University of North Carolina at Chapel Hill with a Master's degree in Library Science. James A. Farrugia is a candidate for the Doctor of Philosopy degree in Spatial Information Science and Engineering from The University of Maine in May, 2007.