12-2007

Linking Moving Object Databases with Ontologies

Kraig King

LINKING MOVING OBJECT DATABASES WITH ONTOLOGIES

By

Kraig King

B.S. University of Maine, 2000

A THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

(in Spatial Information Science and Engineering)

The Graduate School

The University of Maine

December, 2007

Advisory Committee:

Kathleen Stewart Hornsby, Assistant Professor, Department of Geography, University of Iowa, Advisor

Max Egenhofer, Professor of Spatial Information Science and Engineering

Silvia Nittel, Associate Professor of Spatial Information Science and Engineering

LIBRARY RIGHTS STATEMENT

Signature: ___________________________

Date: ________________________________

LINKING MOVING OBJECT DATABASES

WITH ONTOLOGIES

By Kraig King

Thesis Advisor: Dr. Kathleen Stewart Hornsby

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

This work investigates the supporting role of ontologies for supplementing the information contained in moving object databases. Details of the spatial representation as well as the sensed location of moving objects are frequently stored within a database schema. However, this knowledge lacks the semantic detail necessary for reasoning about characteristics that are specific to each object. Ontologies contribute semantic descriptions for moving objects and provide the foundation for discovering similarities between object types. These similarities can be drawn upon to extract additional details about the objects around us. The primary focus of the research is a framework for linking ontologies with databases. A major benefit gained from this kind of linking is the augmentation of database knowledge and multi-granular perspectives that are provided by ontologies through the process of generalization. Methods are presented for linking based on a military transportation scenario where data on vehicle position is collected from a sensor network and stored in a geosensor database. An ontology linking tool,

implemented as a stand alone application, is introduced. This application associates individual values from the geosensor database with classes from a *military transportation device* ontology and returns linked value-class pairs to the user as a set of equivalence relations (i.e., matches).

This research also formalizes a set of motion relations between two moving objects on a road network. It is demonstrated that the positional data collected from a geosensor network and stored in a spatio-temporal database, can provide a foundation for computing relations between moving objects. Configurations of moving objects, based on their spatial position, are described by *motion relations* that include *isBehind* and *inFrontOf*. These relations supply a user context about binary vehicle positions relative to a reference object. For example, the driver of a military supply truck may be interested in knowing what types of vehicles are in front of the truck. The types of objects that participate in these motion relations correspond to particular classes within the *military transportation device* ontology. This research reveals that *linking* a geosensor database to the *military transportation device* ontology will facilitate more abstract or higher-level perspectives of these moving objects, supporting inferences about moving objects over multiple levels of granularity. The details supplied by the *generalization* of geosensor data via linking, helps to interpret semantics and respond to user questions by extending the preliminary knowledge about the moving objects within these relations.

ACKNOWLEDGMENTS

This thesis represents a journey of accomplishment, friendship, and sacrifice. Being a grueling passage into the next level of my education, it has left a trail of caffeinated beverages, late nights, and several hours peering deep into my laptop. I owe an enormous gratitude to Kripa Joshi, Daniel Bowe, Maria Vasardani and all of my friends and co-workers who have provided me direction along the way. I truly would have been lost without you!

I must express a special thanks to my wife Nichole, for supporting me throughout this adventure. Her tolerance to my elevated stress levels and the inconveniences caused by deadlines and time constraints kept me on track to achieve success. Without her support and caring, I would have run out of steam a long time ago!

I would also like to thank my parents, John and Natalie, for making sure that I stayed focused on my studies throughout my elementary, high school, and undergrad years. Your careful guidance on how to maintain a balance between academics and fun, plus the patience you demonstrated in answering my numerous ‘why’ questions, provided the foundation needed to get me where I am today!

Finally, I express my gratitude towards my advisor Kathleen Stewart-Hornsby for nurturing the ideas that have blossomed into this research. Our long discussions and numerous revisions were instrumental in forming this thesis.

This research was supported by the National Geospatial-Intelligence Agency under grant number HM1582-05-1-2039.

TABLE OF CONTENTS

Chapter

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The integration of geosensor networks with common transportation devices, such as passenger and military vehicles, is fostering a nearly continual record of measured object positions with respect to time. For example, a network of sensors placed along an established roadway has the ability to measure the location of vehicles at fixed reference points. The resulting geosensor data streams by themselves, however, do not necessarily supply a context for the motion data that includes expressive semantics such as the class of vehicle or its attributes. Therefore, ontologies can be drawn upon to provide a multi-granular way for describing the characteristics of objects found in the world around us and the relationships between those objects.

This thesis presents a novel framework for linking databases and ontologies. The application for this study is a military transportation scenario. Geosensor data on military vehicle positions is collected from a sensor network and stored within a geospatial database. An ontology linking tool, implemented as a stand-along application, is introduced to associate individual values from the geosensor database with related classes from a *military transportation device* ontology. This mechanism allows for inferencing about moving objects by facilitating more abstract or higher-level perspectives of the moving objects. The *linking tool* returns *value-class* pairs to the user as a set of matching terms and is further expanded to support spatio-temporal matches between the database and ontology.

This process of linking database instances with ontological semantics provides additional details of the preliminary source data. Although this work is based on a military transportation scenario, the foundations of this research can be applied to broader applications such as a department of transportation (DOT) setting.

1.1 Motivation

Geosensor data streams provide a steady feed of measured values for geospatial phenomena. For example, at time 1/1/2007 14:56:24 an armored personnel carrier with identification number BGR534 may be sensed at position (44.80650, -68.78868). This information, by itself, provides little semantic context for the object (for instance color, weight, or purpose) and its detected movement. However, a more detailed understanding of dynamic geospatial domains that incorporates semantics can be derived by combining data streams with ontologies that contain such details.

Augmenting these positional measurements with the information contained by supporting ontologies will provide a basis for reasoning about the type of vehicles traveling on the road network (e.g., is the vehicle in front of me a military vehicle?) as well as semantic similarities between them (e.g., the supply truck and the personnel carrier behind my car are both a kind of military support vehicle). The data streams containing the sensed details of the moving objects and their associated positions are stored in a relational database, while generalizations or refinements of the classes of moving objects are supplied by related ontologies.

Ontologies have been a subject of interest for researchers in the geographic information science (GIScience) community for systems integration, interoperability and data sharing (Fonseca *et al.* 2002). Existing ontological models are being evaluated to determine what terminology, frameworks and methods may be available and applicable for the GIScience domain (Agarwal 2005). This thesis further extends this area of research by using ontologies to assist with the integration and recombination of data streams from different sources (e.g., images from satellites combined with data from fixed-location sensors) in order to provide the foundation for augmenting database knowledge. The ontologies supply semantic descriptions that enable humans and computerized devices to process, extend, and reuse these data streams by providing additional perspectives. This contribution of ontologies makes possible automated machine learning algorithms that facilitate linking mechanisms.

1.2 Goal and Hypothesis

Methods for combining data from geosensor networks by linking the databases storing sensor data with related ontologies are considered in this research. The topic of linking databases with ontologies is still relatively new. Existing techniques have typically mapped the database schema with elements of the ontology. This thesis investigates an alternative approach that links the instance-level data stored within the database to the class names in a related ontology such that more information can be derived. The resulting list of matches, referred to in this thesis as *equivalence relations*, that hold for related terms between the database and ontology aids in the semantic understanding of the moving objects being sensed.

- The goal of this thesis is: *To create a linking framework and implement it as a software tool*.
- The hypothesis of this thesis is: *Generalization based on the set of returned equivalence relations, extends existing database knowledge by adding additional information that is drawn from a related ontology.*

1.3 Approach

To derive attachment points between a moving object database and a related ontology, details of the sensed motion of these objects are stored and then evaluated by computers and/or domain experts. Geosensor data streams contain continuous location data that are sampled such that they meet a desired spatial or temporal granularity requirement. To manage this data, a framework for collecting and storing the details of these objects and their motion is introduced. In this thesis the moving objects are assumed to be land-based military vehicles that travel along road networks or predefined routes. A relational database is used to accumulate the sensed position of each moving object, as well as the object's unique identifier, type (e.g., supply truck) and length. Additional details of object movement such as the sensor identification number, traveled route (e.g., lane identification number), lane direction, and sensed time are also stored. Three relations, *SensorData*, *LaneData*, and *ObjData*, will provide the foundation for a geosensor database that stores this motion data.

Geosensor data streams often supply only the spatial representation and the sensed location of the moving objects. This data lacks semantic details (for example generalization of object classes in addition to class attributes) that are necessary for a comprehensive understanding of the state of objects within a dynamic domain. For that reason, the classes, attributes and instances (if available) within a related ontology will be leveraged to supply additional semantic descriptions for the moving object data. Additional semantic knowledge for these entities is derived from the Suggested Upper Merged Ontology (SUMO) knowledge base that has been developed for the IEEE (http://www.ontologyportal.com). The SUMO framework defines a hierarchy of classes, rules and relationships. This upper level ontology was developed as a base ontology that is used in a variety of computer systems and applications, for instance, eLearning ventures (Angelova *et al.* 2004) and the BioImage Database project (Shotton 2003).

This thesis utilizes a partial mid-level SUMO ontology that is based primarily on types of land-based military entities that move on a transportation network. The relevant classes in SUMO are related by taxonomic *is_a* relations to form an ontology of *military transportation devices*. These classes are derived from the CIA Word Fact Book 2002 (http://www.cia.gov/cia/publications/factbook/), as well as the Universal Joint Task List (http://www.dtic.mil/doctrine/jel/cjcsd/cjcsm/m3500_4b.pdf) and the on-line Glossary of Landform and Geologic Terms (http://www.statlab.iastate.edu/soils/nssh/629.htm).

The ontology of land-based military transportation entities is represented using Protégé, an open source ontology editor and knowledge base framework. The Protégé editor is a tool that enables the creation of OWL and XML translated ontologies by providing an interface to input values of classes, subclasses and relations. It was developed by the Stanford Medical Informatics at the Stanford University School of Medicine (http://protégé.stanford.edu/). Protégé has a clientele that includes the Defense Advance Research Projects Agency (DARPA), the National Institute of Standards and Technology (NIST) and the National Science Foundation (NSF).

To facilitate the integration of moving object databases with ontologies, a set of equivalent terms must first be derived by linking the sensor data with available elements of the *military transportation device* ontology that is modeled in Protégé. A mechanism for linking, implemented as stand-alone application, is introduced. This application will be used to associate data values from the geosensor database with classes from the *military transportation device* ontology. Linked terms (i.e., matches) will be returned to the application user as a set of equivalence relations.

In this thesis *specification*, *parsing*, *matching* and *granularity control* comprise the sequence of steps used for linking. During *specification* the source geosensor database, the target ontology, and the key elements of each that are to be connected, must be identified. Once these entities are specified the linking application iteratively selects each database value and begins *parsing* the ontology structure for potential matches. During the *matching* process, similar terms are returned as equivalence relations. Once a list of equivalence relations has been generated, one has the ability to choose the desired granularity of the results. *Granularity control* thereby provides a method for choosing a higher-level and more abstract understanding of the domain by augmenting equivalent terms through a process of generalization. Chapter five provides an evaluation of this linking framework's expressive power, by quantifying the amount additional knowledge gained through the linking and generalization of preliminary data.

1.4 Research Questions

This thesis investigates four primary topics associated with the linking of geosensor databases with related ontologies. These topics are encapsulated by the following research questions.

- Q1. What type of software device can be used to link instance level data within a geosensor database to the classes, attributes and instances of a related ontology?
- Q2. How does this linking mechanism facilitate augmentation of the preliminary geosensor database knowledge?
- Q3. What methods can be used to further derive generalizations of the preliminary knowledge about the sensed moving objects?
- Q4. Can generalization techniques enable an automated system to further evaluate spatio-temporal relationships between two moving entities?

1.5 Scope of Thesis

The primary focus of this thesis is connecting geosensor databases and ontologies. Furthermore, the term *linking* will be used to describe this process. The term *linking* is similar to *mapping* in that elements from independent databases and ontologies are matched in order to determine what equivalence relations exist, enabling information to be shared between them. During the process of linking, the structure and elements within each source remain unchanged.

In this research, the similarity between a set of ontology classes for moving objects and a set of related database values is computed by lexical pattern matching. This work could be extended in the future to consider semantic similarity based matching algorithms (Rodriguez *et al*. 1999; Nedas and Egenhofer 2003; Nedas 2006). Although these advanced matching techniques are beyond the scope of this study, the reader is familiarized with them through a brief discussion provided in chapter 5.

This thesis utilizes a mid-level ontology from the Suggested Upper Merged Ontology knowledgebase (http://sigma.ontologyportal.org) that is based primarily on the classes of land-based military entities commonly encountered moving on a transportation network. It is assumed that the moving entities are land vehicles that travel about on a transportation network that is composed of some sort of roadway or predefined land based route. Airplane- and waterway-related classes will not be considered because they are outside of this scope.

Tracking patterns of moving objects is a subfield of geosensor research, with one focus relating to the modeling of moving objects via sequences of location-time pairs that form *trajectories* (Pfoser *et al.* 2000; Wolfson *et al.* 2001; Stefanidis *et al.* 2003; Meka and Singh 2005; Pfoser and Jensen 2005; Chen *et al.* 2006). Although trajectories are fundamental for tracing past and current object movement as well as predicting future motion plans (Dillenburg *et al.* 2004), this thesis will only consider the sensed location of an object at a discrete time *t*.

1.6 Major Results

One major result of this research is a four step process for linking that consists of *specification*, *parsing*, *matching* and *granularity control*. This framework is used to intuitively connect geosensor network data with an ontology in order to increase or decrease information granularity. Combining such data streams with a related ontology, provides a foundation for deriving a real-time understanding of dynamic geospatial domains that incorporate semantics.

A second result of this thesis is the development of a software tool which provides a mechanism that augments a geosensor database by linking it with an ontology. This augmentation will also include a means for returning more generalized details of the preliminary database knowledge. It is expected that the functionality of this linking tool will enhance next-generation information systems by assisting in their understanding, modeling, and indexing of moving objects. It will be demonstrated that this linking tool can be leveraged to provide alternative perspectives of the types of vehicles traveling on a road network, as well as semantic similarities between them.

A third result of this thesis is the development of a method for distinguishing a set of motion relations that describe the position of a pair of vehicles relative to each other on a road network. These relations are derived from vehicle positional data that is collected from a geosensor network and then stored in a spatio-temporal database. This information provides additional user contexts for binary vehicle patterns relative to a reference object. For example, to query relations such as *"is that an armored personnel vehicle inFrontOf the supply truck I am driving?"*

1.7 Intended Audience

The intended audience of this thesis is any researcher, knowledgebase expert, or software developer interested in extending geospatial databases with data from related ontologies to acquire a deeper understanding of object similarities and relations. The thesis may also be of interest to other audiences such as: GIS professionals, the sensor community, computer scientists, and database researchers as it discusses a framework for augmenting database knowledge with details from supporting ontologies.

1.8 Organization of Remaining Chapters

Where data collected from geosensor networks are capable of providing an essentially continual stream of measured values with respect to geospatial phenomena, these data streams alone do not necessarily give a semantic context for the data that is collected. Combining such data streams with ontologies, however, provides a foundation for deriving a real-time understanding of dynamic geospatial domains that incorporate semantics. The details of the objects and their associated positions are derived from data streams and stored within databases, while generalizations or refinements of the moving objects are supplied by related ontologies.

To provide the groundwork for the linking concepts presented in this thesis, Chapter 2 introduces moving object databases, ontologies, and their integration. It discusses some of the most central moving object database topics for this thesis including point positioning, trajectories and real-time data considerations. Ontologies are highlighted as a structured way for describing the characteristics and relationships of objects that are found in the world around us. In addition, the application of ontologies in GIScience as well as the Suggested Upper Merged Ontology (SUMO) knowledge base is examined. This chapter concludes with a study of existing mechanisms for combining databases and ontologies.

Chapter 3 examines specific details of the framework utilized for collecting the moving object data that is used by the linking interface. The method employed for sensing positions of moving objects as well as the database structure used to store and analyze this data is discussed. In addition, ontologies for describing the semantics of these moving objects are examined. More specifically, an ontology of land-based *military transportation devices* is introduced.

A detailed description of the mechanism for linking moving object databases with ontologies is presented in Chapter 4. The four components of linking: *Specification*, *Parsing*, *Matching* and *Granularity Control* are discussed. The implementation of the linking framework as a stand-alone application is examined, as well as the motivation and benefits associated with using the linking mechanism to augment preliminary database knowledge further with semantic details from a related ontology. Chapter 4 concludes with a study of the linking tool's interface layout and structure.

Chapter 5 describes a practical application of the collection and storage framework discussed thus far, to describe the basic actions of two or more moving vehicles on a road or predefined route. These actions form the foundation for a typology that distinguishes a set of basic *motion relations* (i.e., an elementary set of relations between two moving objects). The basic relations introduced in this work, inFrontOf( ) and isBehind( ), correspond to the relative positions of vehicles with respect to each other. Ontologies in combination with geospatial data, such as a dataset of sensor-derived vehicle positions, become the basis for formally computing these motion relations. Chapter 5 concludes with a formal example and discussion of how the linking application provides additional semantic knowledge about the types of objects involved in motion relations through the process of generalization. The remainder of this thesis, Chapter 6, provides a summary along with conclusions and recommendations for future work on the topic of linking databases with ontologies.

CHAPTER 2

INTEGRATION OF MOVING OBJECT DATABASES AND ONTOLOGIES

Sensor derived positional data for moving objects often does not provide the semantic detail necessary for human decision making. It has been demonstrated that ontologies can play a supporting role in expanding the preliminary sensor knowledge that is stored within databases. Existing research in aligning databases with ontologies is drawn upon, and further enhanced, to aid in developing a tool that augments existing geospatial database knowledge for moving objects with semantic details from a related ontology. This chapter introduces areas of related work that support the theories, models and hypothesis presented by this research.

2.1 Moving Object Database Terminology and Structure

Developing data models that support the collection of moving object data has been a major topic in the computer science as well as the geographic information science communities. One focus of this research has been on moving object databases (Forlizzi *et al.* 2000; Wolfson *et al.* 2001; Güting and Schneider 2005; Rodriguez-Tastets 2005), where some of the themes include querying moving object databases (Güting *et al.* 2000;

Xie and Shibasaki 2005), indexing attributes (Pfoser and Jensen 2003), modeling moving objects over multiple granularities (Hornsby and Egenhofer 2002), modeling moving objects through the use of the geographic data technology maps (Vazirgiannis and Wolfson 2001) and modeling dynamic transportation networks (Ding and Guting 2004).

2.1.1 Modeling Moving Objects

Formalizations of moving objects using methods based on qualitative spatial reasoning include, the double cross calculus (Freska and Zimmerman 1992) and the qualitative trajectory calculus (QTC) (VanDeWeghe *et al.* 2005), which are used to describe an object's motion. The QTC extends the double cross calculus to consider the movement of two objects with respect to one another (Figure 2.1). The QTC framework also provides a language for differentiating between disjoint groups of moving objects. Further analysis of types of movement patterns has identified some basic types of relative motion within groups of moving objects, such as herds of deer or teams of soccer players (Laube and Imfeld 2002; Laube *et al.* 2005). Additional research has focused on the comparison and aggregation of moving object trajectories and continuous queries (DuMouza and Rigaux 2005) as well as hybrid representations for modeling moving objects, such as nonmaterialized trajectories, in an effort to overcome location imprecision due to sensor error (Cao and Wolfson 2005).

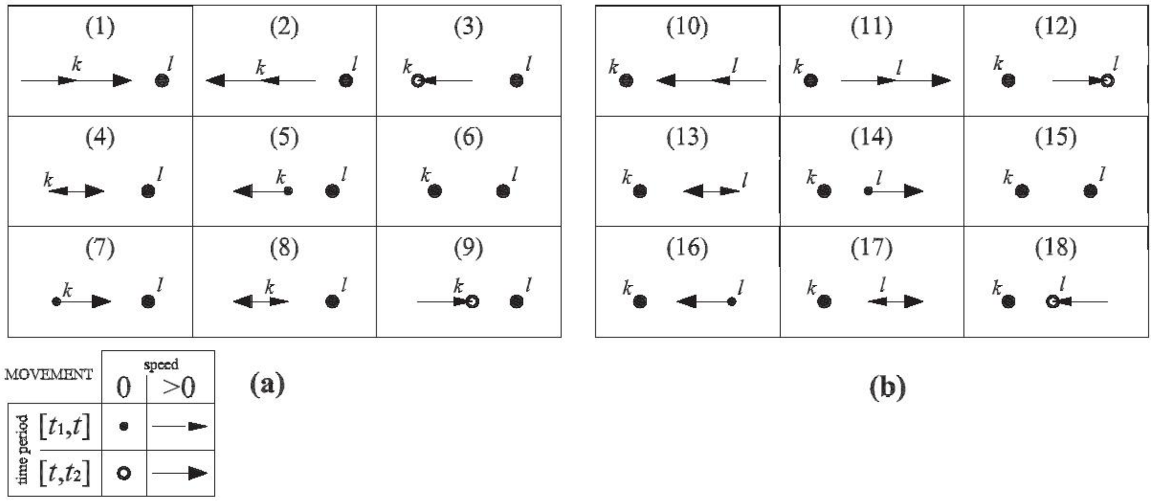


Figure 2.1. Different cases of motion in QTC for a basic one dimensional space.
(a) movement of object *k* with respect to the position of a second object *l* at time *t*
(b) movement of object *k* with respect to the position of a second object *l* at time *t*
(VanDeWeghe et al. 2006)

Tracking patterns of moving objects (i.e., vehicles) is a subfield of spatial databases and geosensor research, with one focus relating to the modeling of moving objects via sequences of location-time pairs that form *trajectories* (Pfoser *et al.* 2000; Wolfson *et al.* 2001; Stefanidis *et al.* 2003; Meka and Singh 2005; Pfoser and Jensen 2005; Cheng *et al.* 2006). Trajectories are fundamental for tracing past and current object movement as well as predicting future motion plans (Dillenburg *et al.* 2004). Additional research has focused on hybrid representations for modeling moving objects, such as nonmaterialized trajectories, in an effort to overcome location imprecision due to sensor error (Cao and Wolfson 2005).

2.1.2 Relations that Describe Moving Objects

Existing research in the area of sensing moving objects has defined relations such as *meet*, *cross,* or *leave* (Erwig and Schneider 2002) and *overtake* (VanDeWeghe *et al.*

2005). In these studies, the change in relative distance between two vehicles is considered, such as cases where one vehicle approaches and passes another, an *overtake* event is defined (VanDeWeghe *et al.* 2005). The work presented in this thesis is complementary to existing research in that it treats the temporal evolution of vehicle movement in order to provide a dynamic view of object movements. Two motion relations are introduced (i.e., *isBehind* and *inFrontOf)* in chapter 5 as they are critical for forward moving objects on road networks and are the most elemental relations associated with the relative positions of a pair of vehicles on a road (Figure 2.2). These relations are a key subset of a larger set of motion relations that may occur on road networks.

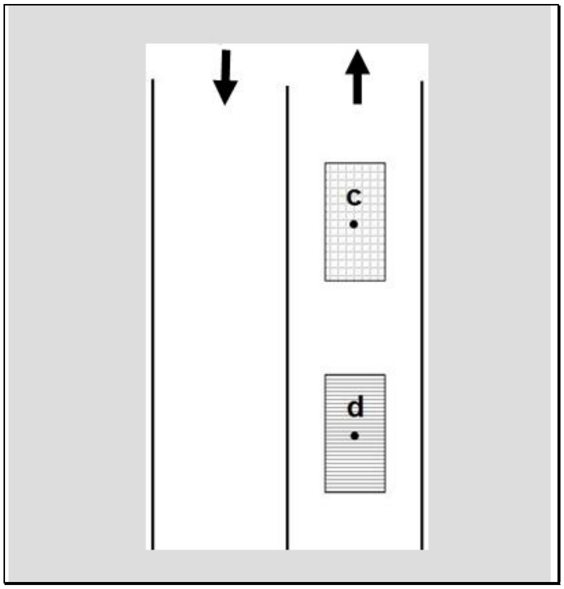


Figure 2.2. The motion relation "***d** isBehind **c***" and its converse *"**c** inFrontOf **d**"*

2.1.3 Data Collection and Storage for Moving Objects

The collection, organization, analysis and delivery of geospatial moving object information from distributed sensor networks is an active research area (Stefanidis and Nittel 2004). The real-time characteristic of moving objects introduces challenges for managing sensor derived data. Some of the challenges include determining update intervals, dealing with sensor imprecision, and handling uncertainty regarding an object's spatio-temporal location. Techniques such as dead-reckoning (Wolfson *et al.* 2001), point location management, and trajectory location management (Wolfson 2002) are some of the proposed solutions. These techniques have led to geosensor database approaches for tracking the movement of objects, for example vehicles, that focus on computing moving object trajectories (Wolfson *et al.* 1999; Pfoser *et al.* 2000; Stefanidis *et al.* 2003; Pfoser and Jensen 2005).

2.2 Ontologies for Moving Objects

In this thesis, we show how ontologies can play a role in the integration and combination of data streams from different sources (e.g., specific attributes broadcast by the vehicles themselves combined with data from fixed-location sensors along a road) by providing the foundation for discovering the similarities between the sensor data sources. The semantic descriptions supplied by ontologies enable computers to process these geosensor data streams, such that their data can be extended and reused. Ontologies contribute to this *interoperability* by providing additional perspectives that are intelligible by both computers and humans (Fonseca *et al.* 2002). This feature makes possible machine learning algorithms that support linking mechanisms for geosensor data.

Research into developing geospatial ontologies has explored the prime geospatial categories and concepts that underlie such ontologies, highlighting, for example, basic geographic features such as mountains, rivers, and lakes (Mark *et al.* 2001; Smith and Mark 2001; Agarwal 2005). These ontologies are especially useful for supporting geographic information integration in a seamless and flexible way (Fonseca *et al.* 2002). A recent focus has been to extend geospatial ontologies to include the treatment of dynamic happenings or *occurrents* (Grenon and Smith 2004) to enable the modeling of events and processes (Worboys and Hornsby 2004; Cole and Hornsby 2005; Galton and Worboys 2005; Worboys 2005). A better understanding of such semantics aids interoperability where the desired goal is to design systems and services that can communicate and exchange data easily including geospatial data (Kuhn 2005).

Incorporating semantics is also important for providing a continuous view of data at multiple levels of detail affording complex relationship discovery (Arpinar *et al.* 2006). For geospatial data, these relations may be based on topology, directional, or proximity associations. Within a transportation network, drivers use these relations to derive a user context and help them make decisions that influence future trajectories (Dillenburg *et al.* 2004). For example, a driver of one vehicle (e.g., a military supply truck) may speed up to maintain a close proximity to a specific class of vehicle that is in front of them (e.g., an armored vehicle).

2.3 Suggested Upper Merged Ontology (SUMO)

In this thesis, an upper level ontology is used to describe the most general classes of entities for a domain. One commonly used ontology by researchers is the suggested upper merged ontology known as SUMO (http://www.ontologyportal.com). Formally defined with over 20,000 terms and over 60,000 axioms, SUMO and its associated mid-level domain ontologies is recognized as the largest formal public ontology (Figure 2.3). Although SUMO is owned by the IEEE, it is within the free public domain. In addition, each of the mid-level domain ontologies it contains are released under the GNU general public license (http://www.gnu.org/copyleft/gpl.html).

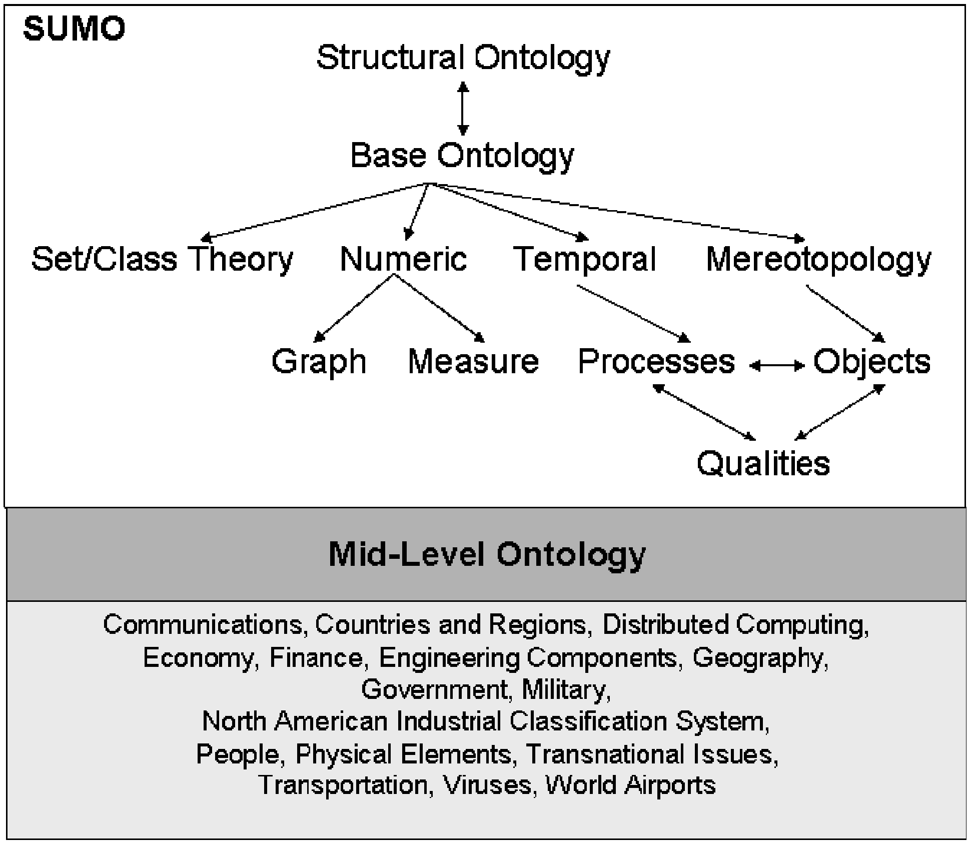

SUMO
Structural Ontology
Base Ontology
Set/Class Theory
Numeric
Temporal
Mereotopology
Graph
Measure
Processes
Objects
Qualities
Mid-Level Ontology
Communications, Countries and Regions, Distributed Computing,
Economy, Finance, Engineering Components, Geography,
Government, Military,
North American Industrial Classification System,
People, Physical Elements, Transnational Issues,
Transportation, Viruses, World Airports

Figure 2.3. The SUMO ontology and its mid-level domain ontologies.
(Adapted from http://www.onotologyportal.com)

As an upper level ontology, SUMO attempts to capture the most general and reusable terms and definitions. To aid in this generalization, the contents of SUMO have been mapped to the semantic lexicon library *Wordnet* that was developed by the Princeton University Cognitive Science Laboratory (http://wordnet.princeton.edu/). Independent uses of SUMO include: adaptivity in eLearning (Angelova *et al.* 2004), biomedical text understanding (Burhans *et al.* 2003), temporal cognitive reasoning (Moldovan *et al.* 2005), and semantic annotation of images (Hollink *et al.* 2003).

In addition to these applications of SUMO, there are many other contexts from which a geospatial domain is modeled. Each of these contexts has contributed to the development of an ontology that describes constituents of the domain. For example, in this thesis a *military transportation device* ontology is extracted from a mid-level domain of SUMO. The classes within this ontology are derived from the CIA Word Fact Book (http://www.cia.gov/cia/publications/factbook/), as well as the Universal Joint Task List (http://www.dtic.mil/doctrine/jel/cjcsd/cjcsm/m3500_4b.pdf) and the on-line Glossary of Landform and Geologic Terms (http://www.statlab.iastate.edu/soils/nssh/629.htm). In this thesis the *military transportation device* ontology is used as a prototypical ontology for augmenting preliminary database knowledge.

2.4 Aligning, Combining, Mapping, Merging, and Linking

Ongoing research relating to the supporting role of ontologies for geosensor data has been investigated, with a focus on methods for connecting geosensor network data with ontologies that are modeled using the Protégé ontology editor (Agarwal 2005). Various terms have been used to derive connecting either multiple ontologies or ontologies and databases (Klein 2001). These terms and associated definitions include: *aligning*: two or more ontologies are brought into mutual agreement so that they appear consistent and coherent; *combining* where two or more ontologies that have similar elements (e.g., classes, attributes) are used in such a way that they act like a single unit; *mapping* is the relating of similar elements from different sources with an equivalence relation such that they appear to be integrated virtually; and *merging* where creating a new ontology is created from two or more ontologies that contain overlapping elements.

Each of the terms discussed above conveys a semantic meaning that pertains to the relation between classes or elements of different ontologies. The first of these terms, *aligning* does not necessarily specify if a new ontology is created, nor if the original ones persist. *Combining*, on the other hand, recognizes the similarities between ontologies and specifies they are treated as a single unit, but again, it is not known if this is a new ontology. In contrast, *mapping* specifies that the resulting ontology appears to be integrated, perhaps implying that the parents persist while a new 'virtual ontology' is created. Another relation, *merging,* specifies that the parent ontologies are integrated to form a new, independent ontology.

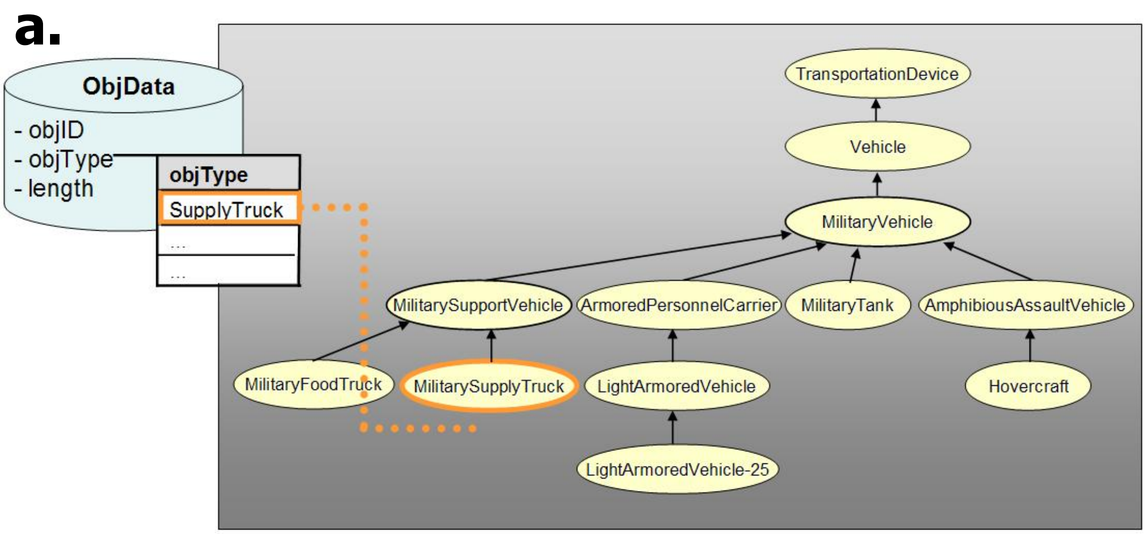


Figure 2.4. Linking between a database (a) and an ontology (b) to share information

In this thesis, we focus on connecting geosensor databases and ontologies, rather than pairs of ontologies. The term *linking* will be used to describe this process. *Linking* is similar to *mapping,* in that elements from independent databases and ontologies will be matched in order to determine what equivalence relations exist, enabling information to be shared between them (Figure 2.4). However, the term linking is used in order to highlight that the original data sources persist and only their similarities are returned. Therefore, during the process of integration, the structure and elements within each source remain unchanged.

The management and integration of databases and ontologies is a significant topic of interest for researchers. The diversity of computational resources that provide geosensor data, (e.g., sensors, satellites, embedded processors, or GPS) must be properly managed to create a fully collaborative system that offers transportation solutions such as autonomous real-time driving, routing and navigation (Dillenburg *et al.* 2004). This research contributes to designing next-generation transportation information architectures by proposing a method for relating these geosensor databases and ontologies.

2.5 Enhancing Database Knowledge with Ontologies

Additional semantic information about moving objects (for example, those modeled in motion relations) is obtained by augmenting the geospatial moving object database with details from a supporting ontology of transportation devices. Establishing common links between ontologies and database content is still a relatively new area of investigation. However, delivering content for semantic web applications is encouraging further research and automated methods for mapping between the database schema and ontologies are being explored (An *et al.* 2006; Konstantinou *et al.* 2006; Yuan *et al.* 2006). Additionally, a number of tools have been created to share knowledge between databases and ontologies by exploiting the similarities between them. For example, applications such as oMAP (Straccia and Troncy 2005), and PROMPT (Fridman and Musen 2000), each provide a semi-automated system for ontology alignment. Others, such as MAPONTO (An *et al.* 2006) and VisAVis (Konstantinou *et al.* 2006)*,* provide a mechanism for connecting relational database schemas with ontologies. The MAPONTO tool locates semantic matches between a database schema and an ontology and returns any plausible relationships as logical rules. The VisAVis process, on the other hand, locates mappings between a database and an ontology and then stores these mappings in a separate relational database that is accessible by the ontology.

The research presented in this thesis expands upon these ideas by considering methods for combining data from sensors on roads via linking the databases storing positional data with ontologies that describe the moving objects being sensed. It will be demonstrated in chapter five that this technique aids in providing more generalized views of these moving objects. We present a tool which permits the linking of complex data from a geosensor network, which is stored in a database, with an ontology such that the individual structure and content of each one still persists. By applying this tool, a list of potential attachment points that hold between the database and ontology is automatically generated by the system.

2.6 Summary

This chapter has reviewed a number of concepts that are critical for forming the foundation of this research. First, a brief introduction to work relating to moving object databases, terminology and structure is provided. Methods for modeling moving objects as well relations that describe moving objects are described in further detail in order to provide a basic understanding of the inherent challenges in collecting and storing moving object data.

The role ontologies play in modeling additional semantic attributes is also discussed. The upper merged ontology SUMO is described to provide insight into the derivation of the *military transportation device* ontology that is introduced as a prototypical ontology for augmenting preliminary database knowledge. Finally, we discuss the semantic meanings conveyed by the terms aligning, combining, mapping, merging and linking, in the context of relations between classes or elements of different ontologies. This topic is expanded further to introduce the focus of this thesis, that is, enhancing preliminary database knowledge by linking a database with an ontology. In the next chapter, a spatial framework for the collection and storage of moving object data is introduced. In addition, an ontology of military transportation devices is presented in order to supply the additional semantic details for sensed moving objects.

CHAPTER 3

A SPATIAL FRAMEWORK FOR COLLECTING MOVING OBJECT DATA

The integration of geosensor networks with common transportation devices (e.g., vehicles) necessitates a framework to store positional data describing entity movement. This research exploits a relational database to store the sensed position of vehicles with respect to either a road network or a predefined route. Ontologies are used in conjunction with this database to supply additional details about the moving objects such as specific vehicle attributes like type, color or purpose. Together, the database and ontology provide the components for a linking mechanism that supplies important semantic details about moving objects and augmentation of the geospatial database knowledge.

3.1 Sensing Positional Data

In this section, a framework for modeling moving entities and the relations between them is introduced. In addition an approach for collecting sensed data for object movement is described. Moving objects require that continuous data samples occur such that they meet a predefined granularity. For example, a spatial granularity requirement may write data to the database every meter in contrast to a temporal requirement that may process data every second. This characteristic often demands frequent positional updates to ensure that data is accurate and current. Positional updates are managed by leveraging the ID-Triggered Locations Update (ITLU) schema employed by the Moving Objects Dynamic Transportation Network (MODTN) framework (Ding and Guting 2004). Within this approach, moving objects are modeled as moving graph points that travel only within a predefined network. Location updates are triggered whenever an object transfers from one sensed location to another. The position of the moving object is then encoded as a location-time pair and stored within a database for future analysis using the *point-location management* approach (Wolfson 2001).

It is understood that the real-time data collection techniques employed by this moving object infrastructure will demand high computational requirements in order to satisfy transaction deadlines that ensure current data (Kao and Molina 1995). A large number of sensors coupled with the typical speed and number of vehicles on a road network will trigger frequent database updates. As techniques improve for collecting geospatial data with real-time data services (Ramamritham *et al.* 2004), the feasibility and scalability of this project are expected to improve over time.

It is beneficial to model the road network explicitly, describing object movement relative to the network, since this makes it easier to query relationships between the moving objects and their positions on the network (Güting *et al.* 2006). For this research, the point-location approach is used in conjunction with a fixed-length linear referencing model to represent object movement. Instead of positioning groups of sensors at route intersections (e.g., MODTN), spatial units are held constant by dividing the route into segments of a standardized length (Miller and Shaw 2001) such that sensed locations are uniformly distributed along the traveled route. Each sensor observes a specific number of fixed reference positions *p*. These reference positions are discretely numbered from +1 to +∝ along the route segment (Figure 3.1).

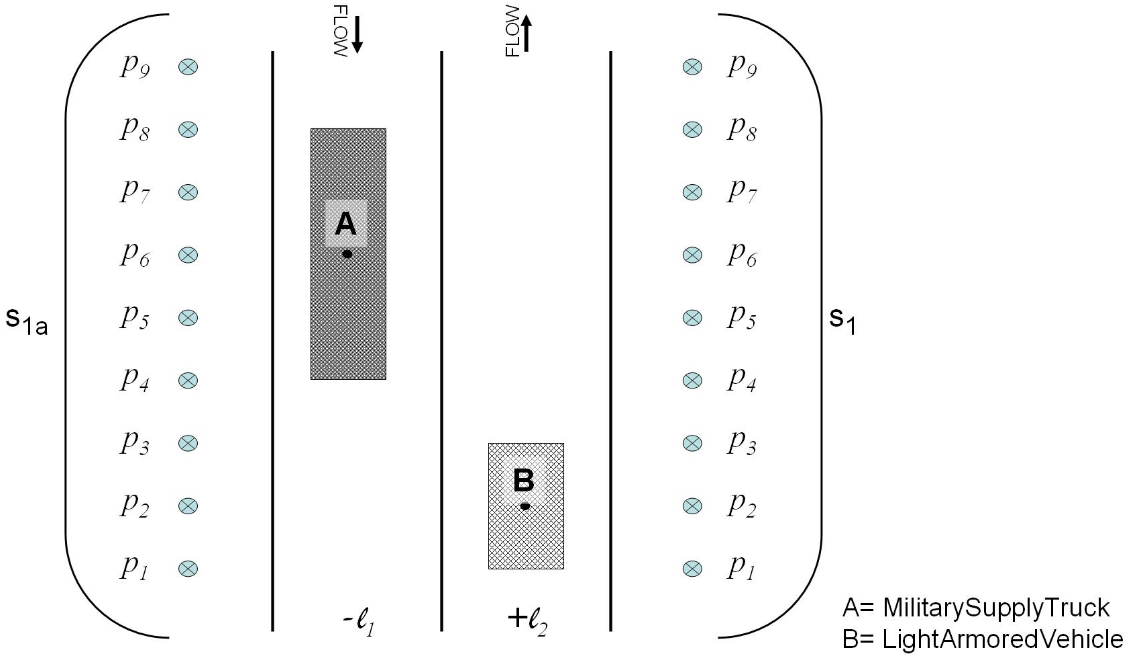


Figure 3.1. Infrastructure for collecting moving object data.

In this work, the moving objects are drawn from an ontology of land-based *military transportation devices*. A more detailed discussion of this ontology and the objects it encompasses is presented in section 3.3. It is assumed that the environment for these moving objects is a road network or predefined route. At time *T*, each object occupies a unique reference position or a set of unique reference positions for any given lane (e.g., one object can not be on top or beside another object in the same lane). For example, object A that is depicted in Figure 3.1 occupies the set of reference positions

{p_4, p_5, p_6, p_7, p_8} in lane ℓ_1. To reduce the degree of parallax introduced by line-of-sight techniques, it is assumed that pairs of sensors are placed on both sides of the route such that the first detects object movement in one direction and the second detects object movement in the opposite direction. To minimize storage requirements this research assumes that a single identifier references both sensors, for example, a pair of sensors {s_1, s_{1a}} is represented by the single identifier s_1.

Each lane (ℓ) is assigned a *laneID* and *direction* attribute. Roads or routes are divided into lanes and are numbered sequentially from 1 to $\propto$. This identification number serves as the unique identifier or *laneID.* This lane direction attribute is based on one of two possible values: flow either *follows* the sequencing of sensors or is *against* the sequencing of sensors. If by default, objects move in ascending sequence of position, then the lane direction is assigned '+1'. Conversely, if objects travel in the direction of descending reference position the lane direction is assigned a '-1'. For example, if a (forward-moving) vehicle transitions from position p_4 at time t to position p_3 at time (t+1) then the lane identifier is assumed to be prefixed with a '-' symbol indicating that movement is against the sequencing of sensor positions.

Data collection begins when object movement is detected. A timestamp is encoded with each triggered sensor reading to provide a temporal reference. From the initial point of movement, a series of sensor readings *r* are collected at a fixed time *t* from one another. Thus, if a supply truck triggers sensor s_1 a set of *n* readings are collected {r_1, r_2, r_3... r_n} with each reading occurring at a constant time *t* from the last. The position of each individual object's midpoint, within the range of the sensor (relative to a reference point *p*) are then collected for each reading. For example, during readings r_4 through r_8, the movement of the supply truck in lane ℓ_2 may be described by the following relationships, $\{ r_4(t_1, s_1, \ell_2, p_2)$, $r_5(t_2, s_1, \ell_2, p_3)$, $r_6(t_3, s_1, \ell_2, p_4)$, $r_7(t_4, s_1, \ell_2, p_5)$ and $r_8(t_5, s_1, \ell_2, p_6) \}$. During the same interval, a light armored vehicle may be detected in the opposite lane (e.g., lane ℓ_1) and its movement would be described as: $\{ r_4(t_1, s_1, \ell_1, p_{10})$, $r_5(t_2, s_1, \ell_1, p_8)$, $r_6(t_3, s_1, \ell_1, p_6)$, $r_7(t_4, s_1, \ell_1, p_4)$ and $r_8(t_5, s_1, \ell_1, p_2)\}$. In this way the sensed movements of vehicles are captured, and if this data is stored, it can be used for future analysis.

Special consideration must be given to the collection and storage of moving object data given that it is characterized by continuous positional changes over time. Conventional database management systems assume that data remains constant unless it is modified. In contrast, moving objects require that continuous data are sampled such that they meet a predefined granularity (Wolfson 2002). For example, a spatial granularity requirement may necessitate database updates every meter in contrast to a temporal requirement that may process updates every second (Bhattacharya and Das 1999). The dynamics of this granularity characteristic often demands frequent database updates to ensure that data is accurate and up-to-date (Saltenis *et al.* 2000; Pitoura and Samaras 2001). For example, this research requires that every time an object transfers from one sensed position to another, a location update will be triggered to store the object's measured location within the geosensor database.

The real-time data collection techniques employed by this moving object infrastructure will demand high computational requirements in order to satisfy transaction deadlines that ensure current data (Kao and Molina 1995). A large number of sensors coupled with the typical speed and number of vehicles on a road network will trigger frequent database updates. A traditional database will provide some of the functionality required by this moving object infrastructure, for example, concurrent transactions and shared data access. However, traditional database architectures are unable to enforce application timing constraints such as turn-around and latency. Research in the areas of real-time databases and real-time services is ongoing in an effort to improve the quality of service, data freshness, and timing constraints (Ramamritham *et al.* 2004). Although this research does not further investigate the real-time aspects of this project, its feasibility and scalability are expected to improve over time as techniques improve for collecting geospatial data with real-time data services.

3.2 Geospatial Database Relations for Storing Positional Data

Additional details of the sensed object and its associated movement are also stored within the geosensor database. In this research a data model describing the structure for capturing details of a mobile object in a moving objects database (MOD) is based on the structure proposed in (Wolfson *et al.* 1999). As part of the MOD implementation, attributes such as the object's unique identifier (ID), route, start location, start time, direction, speed and uncertainty are stored. In a similar fashion, the database representation used for this research to store the geospatial positional data depends on a relation *SensorData* that contains the attributes, *objID*, *sensorID*, *laneID*, *position*, and *time*. This relation stores location readings generated by the sensors within the network.

Details of the moving object are stored in the relation *ObjData* with attributes, *objID*, *objType* and *length*. With respect to the positional data that is stored within the DBMS, the sensors receive a unique object identifier that is broadcast by each vehicle. This object ID (*objID)* corresponds to the registration identification number (RegID) assigned to each vehicle during its registration. In addition to the object's unique ID, the sensors also receive its stored type classification (*objType)*. For example, a vehicle may broadcast that its *objID* is MS0405 and its *objType* is MilitarySupplyTruck. To address privacy concerns, such as location-based spam and intrusive inferences, a number of obfuscation techniques can be used to prevent the abuse of this data. Examples include negotiation (Duckham and Kulik 2005) and location cloaking (Cheng *et al.* 2005) to conceal an object's precise location by exploiting aspects of geospatial uncertainty such as location imprecision and inaccuracy.

A third relation, *LaneData* is based on a lookup table of road infrastructure attributes. It is used to provide the functionally dependent attributes *laneID* and *laneDir*, which describe a lane and corresponding direction for the sensed moving object. These three relations, *SensorData*, LaneData and *ObjData*, provide the foundation for a geospatial database that stores motion data captured within the sensor network.

3.3 A Geometric Representation for a Moving Object

In order to further reason about object movement, a representation of the length and position of the sensed object must be developed. Geometric representations that are commonly drawn upon are points, lines and regions (Forlizzi *et al.* 2000). The most primitive of these three types, a point, is used in this research to denote the midpoint of an object, as this representation helps to simplify the positional analysis. However, details of the object's surrounding volume are lost with this abstraction making it difficult to capture accurate relations between moving objects. To alleviate this problem, the linear extent of the object (i.e., *length*), in addition to its midpoint (i.e., *position*), are stored in the database (Figure 3.2). *Length* is calculated as a function of the spatial reference position interval within the sensor network (e.g., an object length of 1.5 is equivalent to 1.5 reference position intervals). This method allows straightforward positional encoding yet still provides a means to derive specific object relations based on the volume occupied by the object.

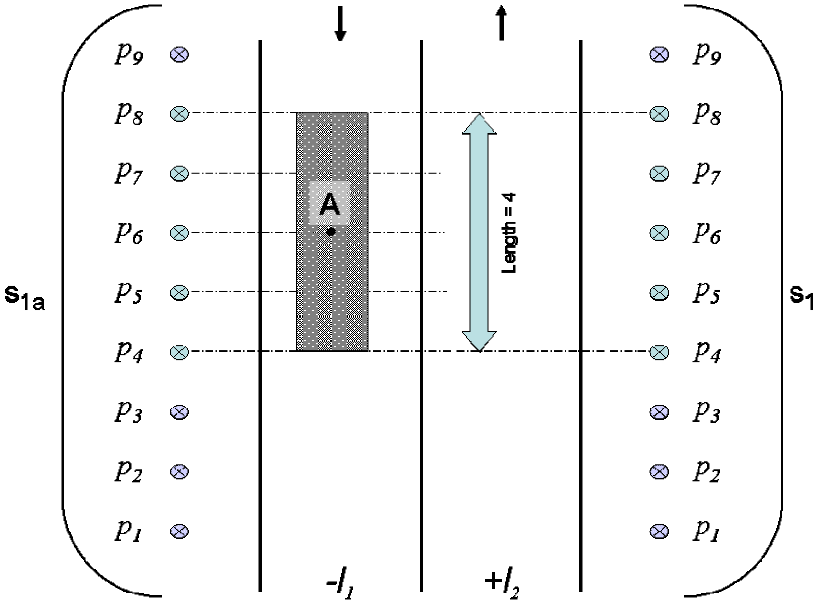


Figure 3.2. Object *A* with attributes *position* = p_6 and *length = 4*

A moving object's location is encoded by obtaining the corresponding reference *position* in the sensor network that is closest to the midpoint of that object. If necessary, another relation can be constructed that explicitly defines the location as a set of coordinates for each of these positions. In addition to the location, a timestamp for the sensed movement is stored in a general date/time format such as *mm/dd/yyyy hh:mm:ss.* Another characteristic of object movement, the lane in which the movement takes place, is encoded as a signed integer before being inserted into the database. Each lane is assigned an identification number (*laneID*) and direction (*laneDir*) attribute that is based on one of two possible values: flow either follows the sequencing of sensors or is against the sequencing of sensors. If by default, objects move in ascending sequence of position, then the lane direction is assigned '+1'. Conversely, if objects travel in the direction of descending reference position, the lane direction is assigned a '-1' (Figure 3.2).

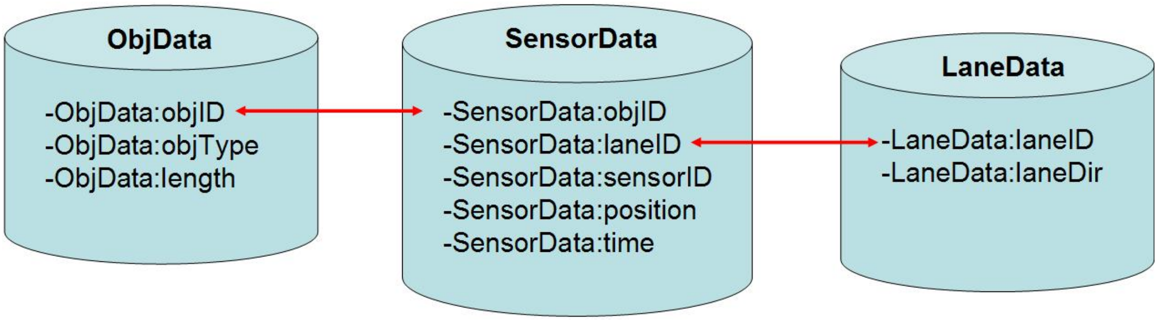


Figure 3.3. Database structure for storing moving object data.

To retrieve information about the moving objects, the relations *SensorData, LaneData* and *ObjData* are used to provide the basis for supporting queries that capture knowledge of moving object semantics (Figure 3.3). Based on the vehicle position data captured and stored within the spatio-temporal database, queries will return specific attributes that are used to inform a driver of, for example, the objects around their vehicle at a given time *T*. Such a query provides details of the objects by returning attributes such as the object identification number as well as the object type, position, and lane ID.

3.4 A Military Transportation Device Ontology

In order to provide a framework for the kinds of moving entities that are commonly encountered on a road network, an ontology of moving entities is introduced. In this thesis it is assumed that the moving entities are land-based military vehicles that travel about on a transportation network composed of roadways or predefined routes. Airplane- and waterway-related classes will not be considered as they are outside of this scope.

This research uses a partial set of moving entity classes that are derived from a mid-level ontology of military vehicles. This mid-level ontology is extracted from the Suggested Upper Merged Ontology (SUMO) transportation knowledge base developed for the IEEE (Niles and Pease 2001). These object classes have been derived from the CIA Word Fact Book (http://www.cia.gov/cia/publications/factbook/), as well as the Universal Joint Task List (http://www.dtic.mil/doctrine/jel/cjcsd/cjcsm/m3500_4b.pdf) and the web based Glossary of Landform and Geologic Terms (http://www.statlab.iastate.edu/soils/nssh/629.htm). It should be noted that although the focus of this work is the military transportation domain, this work is broadly applicable to any sensed moving object domain, for example, vehicle traffic movement or animal movement.

The classes of military land-based vehicles, derived from SUMO, are linked by *is_a* relations to form an ontology. The *is_a* relation represents a taxonomic relation that defines one class to be a *kind of* another class. For example, the *military transportation device* ontology demonstrates that a *MilitaryVehicle is_a* kind of *Vehicle* which *is_a* kind of *TransportationDevice*. Therefore, it is understood that the class *MilitaryVehicle* is a subclass of *Vehicle*.

One difference between this ontology and SUMO is that this work assumes no multiple inheritance within the *military transportation device* ontology. We follow the tradition of classification that avoids multiple inheritance, i.e., every subclass belongs to only one superclass (Smith *et al.* 2004). In this way, we avoid pitfalls of overloading is_a relations, and avoid complicating any possible future alignment of the ontology with other ontologies. This subsumption hierarchy embodies all of the relations between classes within this ontology.

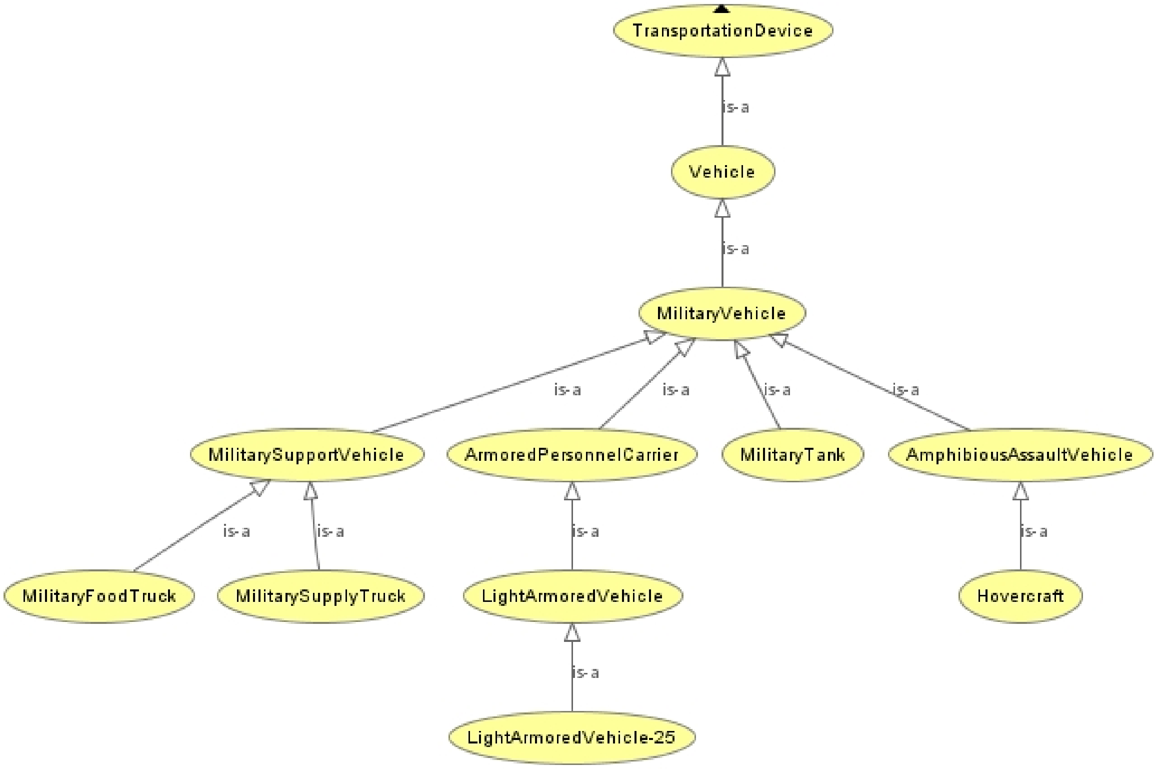


Figure 3.4. A *military transportation device* ontology based on the SUMO mid-level transportation ontology (http://www.ontologyportal.com).

The *military transportation device* ontology (Figure 3.4) considers only land-based vehicles and contains a class *MilitaryVehicle* that has four primary subclasses: *MilitarySupportVehicle*, *ArmoredPersonnelCarrier*, *MilitaryTank*, and

AmphibiousAssaultVehicle. The class *MilitarySupportVehicle* has subclasses *MilitaryFoodTruck* and *MilitarySupplyTruck*. The class *ArmoredPersonnelCarrier* has subclass *LightArmoredVehicle* which also encompasses a specialized eight wheeled personnel carrier denoted by the class name *LightArmoredVehicle-25*. The class *AmphibiousAssaultVehicle* subsumes the specialized class *Hovercraft*. It should be noted that *AmphibiousAssaultVehicle* and *Hovercraft* are included as land-based military vehicles due to the fact that SUMO defines their purpose to include, "inland objectives and to conduct mechanized operations and related combat support in subsequent mechanized operations ashore". Even though the SUMO framework treats these two classes as both water and land-based vehicles, this research models them exclusively as a land-based vehicle.

3.5 Summary

This chapter presents a framework for sensing the position of moving objects on a road network or predefined route. This framework draws upon the MODTN structure for sensing positional data. In addition, this chapter introduces a structure for a moving objects database that is based on three primary relations. The relation *SensorData* with attributes, *objID*, *sensorID*, *laneID*, *position*, and *time* is defined for storing location readings from sensors within the network. In conjunction with the sensor data, the moving objects broadcast specific details of themselves such as the attributes *objID*, *objType* and *length*. These attributes are stored within the relation *ObjData*. A third relation, *LaneData*, is based on a lookup table of road infrastructure attributes and provides the functionally dependant attributes *laneID* and *laneDir* Together, these three relations, *SensorData, ObjData, and LaneData* provide the foundation for a geosensor database that stores motion data captured within the sensor network.

Additional semantic details of the sensed moving objects are supplied by a related ontology. In this work, the moving objects are entities drawn from a *military transportation device* ontology that is derived from the SUMO knowledge base. These are land-based military entities such as a supply truck or armored vehicle that move on either a road network or predefined route. Each kind of moving entity is modeled as a class and these classes are related by is_a relations to form an ontology. The next chapter will demonstrate how these classes, attributes, and associated relations from the *military transportation device* ontology are possible candidates for *linking* with instance values from the moving objects database.

CHAPTER 4

MECHANISM FOR LINKING MOVING OBJECT DATABASES WITH ONTOLOGIES

To maximize the benefit of geosensor data streams, mechanisms need to be developed for combining and extending moving object data. Ontologies provide additional details for the objects referenced by the geosensor data and allow both humans and machines to perform reasoning and make inferences. These inferences can be either more generalized views of the sensed objects or, under special circumstances, more specialized views as is deemed necessary. In this chapter a software mechanism is presented to link a geospatial database with the ontology of land-based military transportation devices. This *linking mechanism* facilitates the extension of preliminary database knowledge by augmenting sensed details of the moving objects with information from the ontology.

4.1 Methods

A number of tools have been created to link independent ontologies by exploiting the similarities between them. For example, applications such as CRAVE (Gkoutos *et al.* 2004), OLA (Euzenat *et al.* 2004), oMAP (Straccia and Troncy 2005), and PROMPT

(Fridman and Musen 2000), each provide a semi-automated system for ontology alignment. These systems combine ontological data by locating common classes and attributes and then using them as attachment points between pairs of ontologies. Additional research has produced systems such as MAPONTO (An *et al.* 2006) and VisAVis (Konstantinou *et al.* 2006) that provide a mechanism for connecting relational database schemas with ontologies. For example, the MAPONTO tool assists users in discovering semantic relationships that exist between a database schema and a target ontology. The resulting output is a set of rules that express these semantic mappings. Similarly, VisAVis compares a database schema with an ontology by employing a graphical interface that is developed within the Protégé ontology editor. VisAVis identifies similarities (i.e., mappings) between these two knowledge bases, and outputs this information within a new ontology that contains SQL references to the database.

This research builds upon these existing techniques of mapping databases to ontologies. It introduces a method for linking that extends existing approaches by looking beyond the database schema and linking instance-level data contained within a geospatial database with the classes and attributes of a related ontology. This section will discuss in greater detail, a system architecture that is implemented as a standalone application, which employs a sequence of steps to link a geospatial database with the ontology of military transportation devices.

4.1.1 Specification

The first step in this linking process is to specify both the geospatial database as the *source* and the ontology as the *target* from which additional semantic details will be drawn. In addition to the data sources themselves, key elements that will be used for connecting these entities must be identified. The stored values within these elements become possible attachment points for linking. For example, key elements within the geospatial database may be specific *tuple values* or components of the database schema such as *attribute name*, *attribute datatype* and additional *attribute metadata*. Attribute metadata refers to supplementary data that further describes a relation (e.g., attribute definitions, search keys, etc). For an ontology, the key elements that serve as a basis for linking are *class name*, *attribute name* and possibly *instances of classes* if they have been defined.

Existing systems (e.g., MAPONTO and VisAVis) map database schemas with elements of an ontology via lexical matches. In this research we focus on an alternative method of linking, where individual attribute values from the geosensor database are linked with the classes, attributes and instances found within a related ontology (Table 4.1). This is a type of *deep linking* where the attachment points are the actual instance data contained within the geosensor database rather than the database schema.

To further refine the knowledge used as the input for the linking process, two possible options for filtering source data from the geospatial database are available. The first option considers the scenario where the database attributes used for linking are not specified. If no particular attribute names are specified, the linking application assumes that the values within *all attributes* are to be processed for potential matches. The second option leads to a reduction of the source data per the specification of desired attribute names. This is achieved by applying the relational projection operator, *π*, via an internal application query. This reduces the search space so that only values of interest from the geospatial database are parsed for semantic equivalence. For example, the *ObjData* relation discussed in section 3.3 contains three attributes: *objID*, *objType* and *length*. The linking application, by default, will process instances from all three attributes for corresponding ontology matches. However, the ability to declare that matches only be found for a specific attribute also exists. For instance, if the single attribute *objType* is chosen from the relation *ObjData*, this reduces the number of database instances to be processed by approximately sixty-six percent (one out of three attributes selected).

Table 4.1. Possible database and ontology elements for linking

		ONTOLOGY		
		class names	attribute names	instance values
DATABASE	tuple values	X	X	X
	attribute names			
	attribute datatypes			
	attribute metadata			

More advanced filtering can be achieved by specifying a SQL statement that further restricts the geospatial dataset that is used by the linking mechanism. For example, the following statement would only allow objects with a prefix *Military* to be used as an input stream to the linking mechanism.

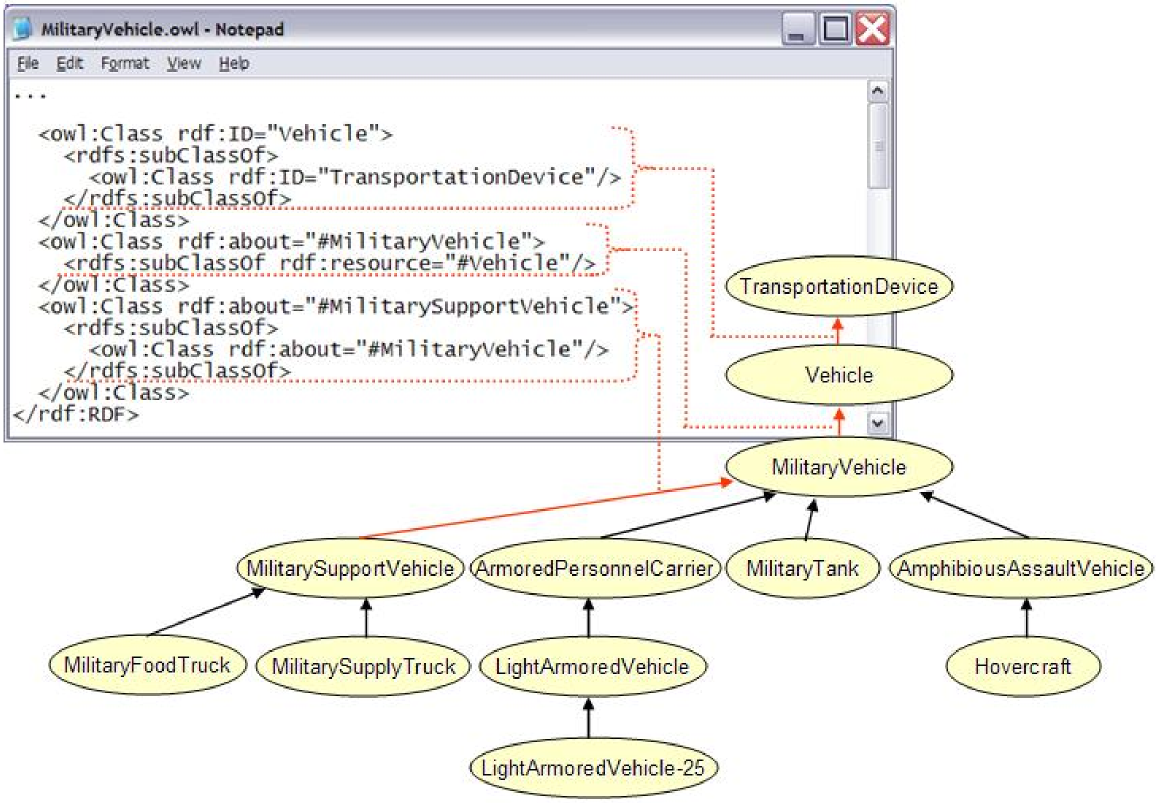
```
SELECT objType
FROM objData
WHERE objType like ‘Military*’;
```

It will be demonstrated in next chapter that the ability to specify the attributes and corresponding instance data, helps the linking mechanism locate potential attachment points between the database and ontology.

4.1.2 Parsing

To facilitate the linking process, a parsing algorithm iterates through the ontology structure, searching for the values of specified linking elements (e.g., class names, attribute names, and instance values). This parsing algorithm, *parseClasses()*, is used to manipulate ontology source files that have been stored in either the Web Ontology Language (OWL) or Resource Description Framework (RDF) formats, both of which can be generated from within the Protégé ontology editor (Figure 4.1). Currently, these are the only two formats recognized by this application.

The *parseClasses()* function identifies class names that are stored within the target ontology, by extracting classes from the ontology source file. A simple function is used to traverse each branch of the ontology until the entire class hierarchy has been searched. In addition to class names, this function can be modified to discover additional characteristics of each class such as attribute names and specific instances. For example, a subroutine *getSlots()* could be used to retrieve a collection of attributes for a given class (attributes are referred to as *slots* within Protégé). Additionally, a collection of available instances for each class can be returned with a *getInstances()* subroutine. Both of these subroutines would be included in the body of the *parseClasses()* function to retrieve additional information about each class as it is located.



```
...
  <owl:Class rdf:ID="Vehicle">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="TransportationDevice"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#MilitaryVehicle">
    <rdfs:subClassOf rdf:resource="#Vehicle"/>
  </owl:Class>
  <owl:Class rdf:about="#MilitarySupportVehicle">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#MilitaryVehicle"/>
    </rdfs:subClassOf>
  </owl:Class>
</rdf:RDF>
```

Figure 4.1. An RDF representation for ontology class hierarchy

Once a collection of class names and associated semantic attributes is manifested, additional logic is employed by the linking application to make use of this ontological knowledge. The next section will discuss how the parsed ontology data can be used to locate potential attachment points between the source database and target ontology.

4.1.3 Matching

The linking mechanism must determine if the identified elements from the ontology are common to the geosensor database, such that an equivalence relation is defined. The matching process begins with the database tuple value serving as the *pattern* (i.e., character sequence) that is being searched for. Possible *candidates* for a match are sought from the class names, attribute values and instances of the related ontology. For example, consider the relation *ObjData* that contains the attribute value *FoodTruck* (Figure 4.2). During the parsing process the linking mechanism locates all occurrences of the value *FoodTruck* by searching every class name of the related *military transportation device* ontology. As a result, pattern-candidate (P-C) pairs that include, for example, (*FoodTruck*, *TransportationDevice*), (*FoodTruck*, *Vehicle*), and (*FoodTruck*. *MilitaryVehicle*) would be generated during this comparison. Of these possible pairs, the only equivalence relation to be defined would be [*FoodTruck* | *MilitaryFoodTruck*].

To find an equivalence relation, each attribute value selected from the database is compared to all available elements of the ontology. Therefore, the total number of iterations of this algorithm will be dN*(oN), where dN is the number of tuple patterns and oN is the number of ontology candidates. The resulting sets of potential patterns and candidates are normalized by ensuring that their constituent characters are all lower case. Making the linking algorithm case insensitive ensures that the maximum number of possible matches is returned by the application. The algorithm processes each pattern and candidate as an independent pair to determine if the pair is a valid attachment point (i.e., link). The entire set of pattern-candidate (P-C) pairs is formally represented as: [P_1-C_1, P_1-C_2… P_1-C_{oN}, P_2-C_1, P_2-C_{oN}… P_{dN}-C_{oN}].

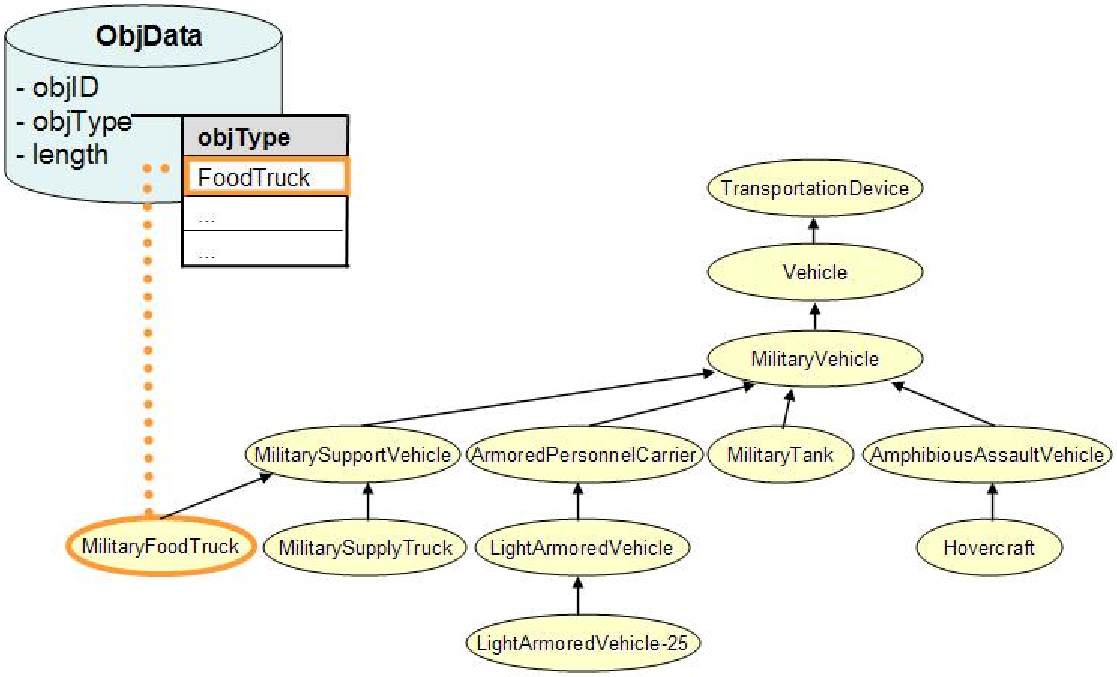
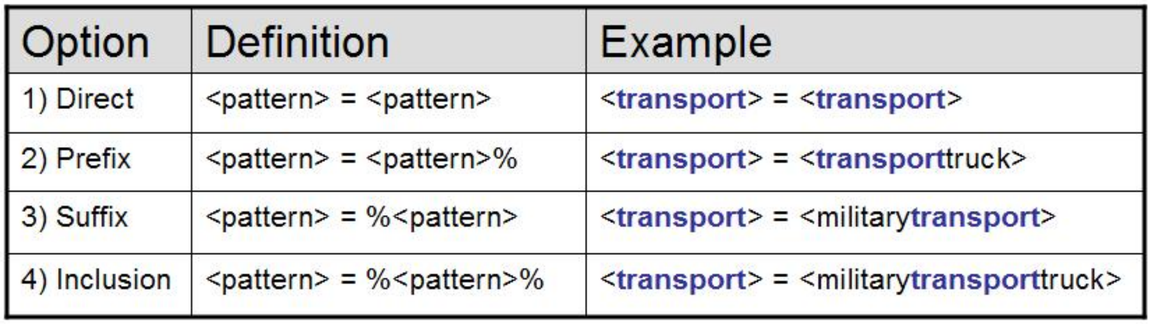


Figure 4.2. Linking a geospatial database with the *military transportation device* ontology.

A number of techniques exist for processing these P-C pairs for possible matches. One frameworks for dealing with similarity assessments is provided by research in the area of semantic similarity algorithms (Rodriguez *et al*. 1999; Nedas and Egenhofer 2003; Nedas 2006). These algorithms compute and analyze similarity coefficients based upon the quality of the potential match and different assumptions of similarity, for example using functions, parts, and attributes as the basis for matching. Although these similarity algorithms are robust, in this research we employ straightforward lexical pattern matching only to compare each potential pair of geosensor database and ontology terms. This allows us to focus on developing methods for deriving and delivering additional knowledge through generalization. The prototype for linking databases and ontologies can be extended in the future to support semantic similarity matching algorithms.

The application provides four pattern matching options that are used to influence the linking mechanism between the source geospatial database and the target ontology. These options are: *direct*, *prefix*, *suffix* and *inclusion* (Table 4.2). The first linking option, *direct*, utilizes a straightforward one-to-one comparison mechanism that is analogous to a logical equality operator. This is the most restrictive case since the exact value from the database must be matched identically with the ontology elements. An example of a successful direct comparison would be an equivalence relation between the database instance *transport* and the ontology class name *transport*.

Table 4.2. The four pattern matching options used by the linking mechanism.



Option	Definition	Example
1) Direct	<pattern> = <pattern>	<**transport**> = <**transport**>
2) Prefix	<pattern> = <pattern>%	<**transport**> = <**transport**truck>
3) Suffix	<pattern> = %<pattern>	<**transport**> = <military**transport**>
4) Inclusion	<pattern> = %<pattern>%	<**transport**> = <military**transport**truck>

The remaining three matching options employ a wildcard character to search for a range of candidate values. Similar to standard programming language syntax, the character * can be substituted for a string of zero or more characters within the candidate. When determining the semantics of these matching options, the position of the pattern within the candidate must be considered. For instance, the prefix option would produce an equivalence relation for any expression of the form [pattern] = [pattern]*. Thus, a prefix based match would be generated between the tuple instance ***transport*** and class name ***transport**truck*. In contrast, the suffix option produces equivalence relations for expressions of the form [pattern] = *[pattern]. The tuple instance ***transport*** and class name *military**transport*** would satisfy the requirements of a suffix-based match. Alternatively, inclusion is a concatenation of both the prefix and suffix matching options. An inclusive match searches for expressions that comply with the specification [pattern] = *[pattern]*. A relation between the instance ***transport*** and ontology class name *military**transport**truck* illustrates an inclusive match.

The cardinality of the matches is represented by one of three forms. Several matches may be generated for a single source entity, that is, a tuple value can be linked to multiple ontology elements (i.e., 1:n). The converse is also true; several source entities can be linked to a single target entity (i.e., n:1). However, it is not required that every tuple within the source database form an equivalence relation with some element of the related ontology. In such cases, no matching elements would be found (i.e., 1:0).

4.1.4 Granularity Control

Equivalence relations can be augmented further through a means of *granularity control*. This process can extend the linking results by coarsening the matching ontology class. That is, more generalized classes are returned by locating subsuming classes. For example, the class *MilitaryVehicle* is generalized further by locating its subsuming parent class, *Vehicle* (Figure 4.3). This process is known as *generalization* and can be iteratively repeated until the upper most ontology node is reached (e.g., *TransportationDevice*).

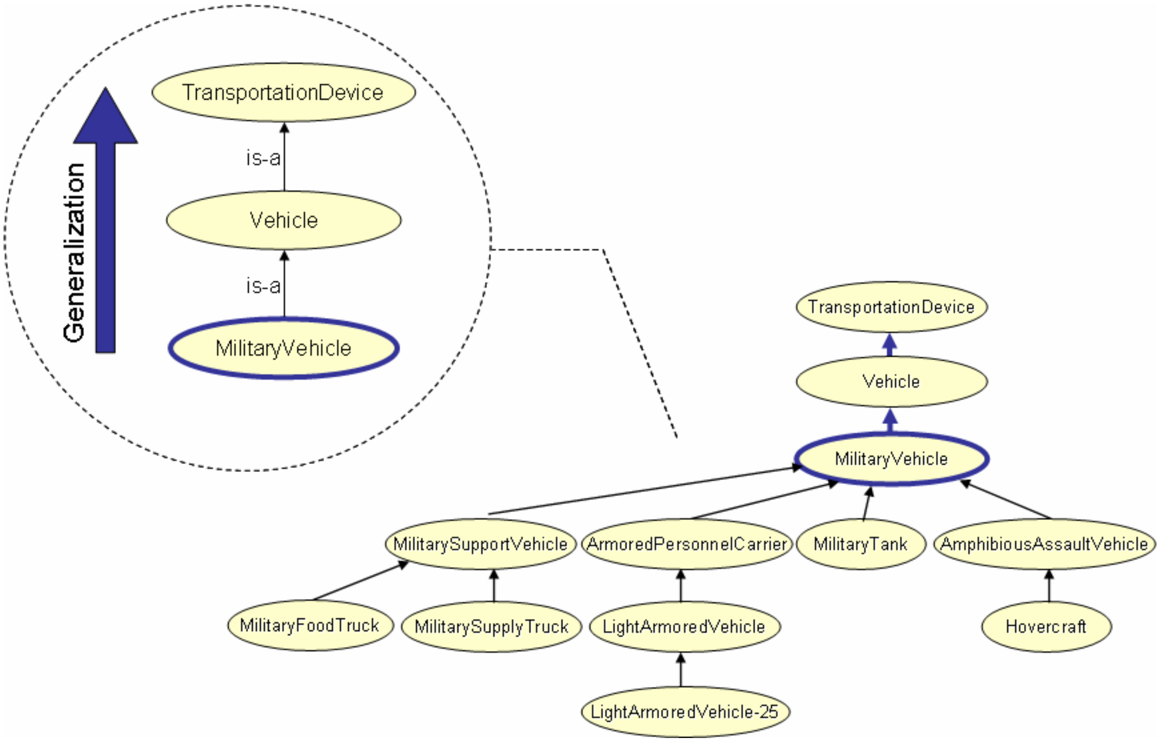


Figure 4.3. Equivalence relations made more generalized

The second and alternative means of granularity control is *specialization*, which is the converse of generalization. Whereas generalization coarsens the linking results, specialization refines the matching ontology classes (Figure 4.4). However, the process of specialization can only transpire if additional information is already known. For example, in order to specialize the ontology class *MilitaryVehicle*, additional information must be included such that the system knows to traverse the branch containing *MilitarySupportVehicle* rather than the ones containing *ArmoredPersonnelCarrier*, *MilitaryTank*, or *AmphibiousAssaultVehicle*. For this reason, in this thesis we will focus only on the process of generalization.

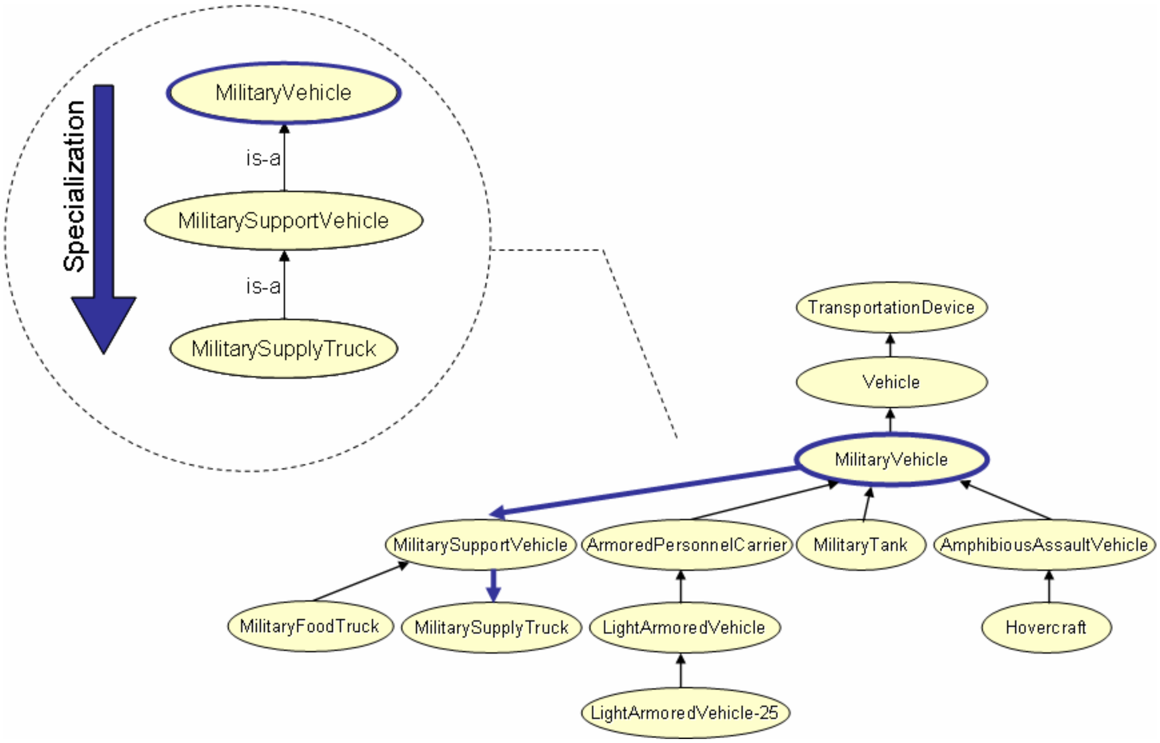


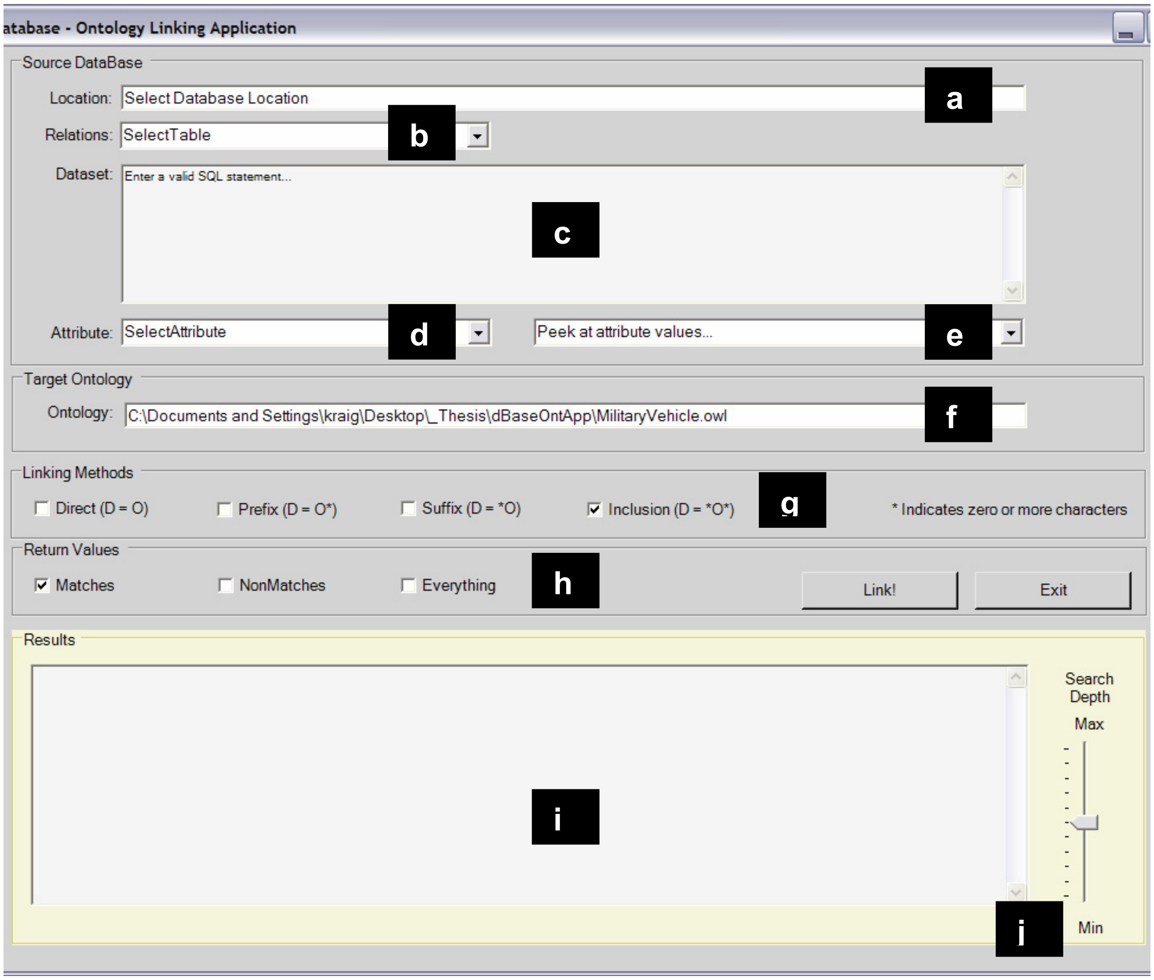
Figure 4.4. Equivalence relations made more specialized

4.2 Designing a Linking Tool

The details of moving objects and their associated positions are derived from geosensor data streams and stored in databases, while generalizations or refinements of the moving objects are supplied by ontologies. So far in this work we have considered methods for combining data from these geosensor data streams by linking the database storing moving object data with an ontology that describes the objects moving within a transportation domain. In order to facilitate this linking the following sections introduce a mechanism, implemented as a stand alone application, that will be used to intuitively connect geosensor network data with an ontology to increase or decrease information granularity. An application that demonstrates the linking mechanism is presented in chapter 5.

4.3 Interface Layout and Structure

The tool created for linking consists of a single pane that provides several objects with which the user must interact (Figure 4.5). These objects are the core components of the linking interface that provide intuitive mechanisms for specifying, viewing, and manipulating the equivalence relations identified by linking a database with an ontology.



atabase - Ontology Linking Application

Source DataBase

Location: Select Database Location

Relations: SelectTable

Dataset: Enter a valid SQL statement...

Attribute: SelectAttribute

Peek at attribute values...

Target Ontology

Ontology: C:\Documents and Settings\kraig\Desktop_Thesis\dBaseOntApp\MilitaryVehicle.owl

Linking Methods

□ Direct (D = O) □ Prefix (D = O*) □ Suffix (D = *O) ☑ Inclusion (D = *O*) * Indicates zero or more characters

Return Values

☑ Matches □ NonMatches □ Everything

Figure 4.5. A tool for database - ontology linking
(a) database location, (b) relations, (c) dataset definition, (d) attributes,
(e) attribute values, (f) ontology location, (g) linking methods,
(h) return values, (i) linking results, and (j) search depth slider

Once the linking tool has been launched, the user must specify a range of parameters that define the database and the associated source data for linking. The first of these parameters is the location of database itself (Figure 4.5a). This can either be the physical location of the database file (e.g., MS Access) or the name of an ODBC connection that has already been configured on the local PC. After the database location has been defined, the linking tool verifies if a connection can be established. Upon verification of a successful database connection, the interface populates a combo box with a list of available relations (Figure 4.5b). These attributes provide the user with a dictionary of relations that contain potential attributes for linking.

After the database connection is established the user must next specify the set of instances that will be used as the seed for linking process. The interface includes a text box that accepts any valid SQL expression that could used to define the dataset (Figure 4.5c). For example, any SQL statement of the form *SELECT * FROM table WHERE condition*. During the execution of this SQL expression, any errors in syntax or interpretation (for example, the database server indicates that a specified attribute is not found) are handled and returned by the interface. This feedback allows the user to correct the problem so the parameter specification for the linking process can resume.

Upon successful execution of the SQL expression, attribute names from the resulting seed dataset are displayed in a second combo box (Figure 4.5d). The user may select any combination of these attributes that are to be used for linking. If desired, a third list box is populated with sample instance values for the chosen attributes (Figure 4.5e). This preview is helpful for ensuring that the correct instance values will be linked with the target ontology.

The location of the target ontology must be specified next (Figure 4.5f). As mentioned earlier in section 4.1.2, this ontology file must be in a recognizable format that adheres to either the OWL or RDF specification. After the file has been specified the linking tool checks the file to make sure it is accessible and valid. Any problems are immediately reported to the user in order to facilitate a resolution.

At this point the source database and target ontology have been fully specified and tested for validity. However, before the linking process can begin two additional parameters must be specified. The first of these is the desired linking method that is used to control the logic for matching database instances and ontology class names. One of four linking methods must be specified by selecting the desired checkbox (Figure 4.5g). These methods are *direct*, *prefix*, *suffix* and *inclusion* as discussed in section 4.1.3. For example, if inclusion is elected, each database pattern is matched with corresponding ontology elements of the form **pattern**.

The final parameter that must be specified prior to linking is the type of results that should be returned by the tool. Check boxes are provided (Figure 4.5h) such that one can specify if the linking tool should return *matches* (i.e., equivalence relations), *non-matches* or *everything* (both matches and non-matches). It is anticipated that the *matches* option would be used most frequently as this option will extend the geosensor data stream knowledge. However, the *non-matches* option may also be useful in situations where one wants to create a list of all database terms that do not have a corresponding equivalence relation within the target ontology. Such a comparison may be performed, for example, by a domain expert that is looking to extend the knowledge contained within the *transportation device* ontology. In rare cases, one may want to view the comprehensive set of database instances and any additional corresponding equivalence relations that are found. This list could be analyzed to perform further data analysis between the matching and non-matching terms.

After all required linking parameters have been specified, the output generated by the linking tool is displayed to the user within the *results* window (Figure 4.5i). Each equivalence relation is shown in the form <*dbase*> *::* <*ontology*>. If the user requests to see non-matches, only the database value is returned. In addition to displaying the equivalence relations, a statistical analysis of the linking results is provided as well. The statistics function simply compares the number of database values processed to the number of database entries that had at least one corresponding ontology match. It should be noted that this statistical analysis is influenced by the type of results that are returned. The statistical calculation may be based upon the number of matches returned, the number of non-matches returned or the number of matches *and* non-matches (always 100%). This value provides the user with a simple way to quantify the volume of additional data returned by the linking algorithm.

In addition to displaying the matching terms of a geosensor database and an ontology, a feature has been added that allows a user to exploit the ontology further by coarsening the desired granularity of the equivalence relations. The *search depth slider* is the mechanism used to augment the linking results (Figure 4.5j). Decreasing the search depth returns equivalence relations that are more generalized by locating the subsuming classes. The *maximum depth* for the search is the ontology class specified in the initial equivalence relation, this is known as the *origin*. The *minimum depth* for the search is the upper most node of the ontology. These new generalized elements of the ontology are displayed to the user in addition to the initial equivalence relations.

4.4 Summary

This section describes a process for linking a geospatial database with the *military transportation device* ontology. This research is unique from other approaches, for example MAPONTO and VisaVis, because this mechanism links the instance-level data contained in a geospatial database with the classes and attributes of a related ontology. It provides a mechanism for combining independent data sources such that information can be shared between them. Thus, in contrast to other methods such as merging, the linking process introduced by this research minimizes storage requirements because a new database or ontology need not be created. Instead, a set of database-ontology attachment points are generated during the comparison of attribute values within a database with the classes and attributes of a related ontology. These attachment point pairs are returned as a set of *equivalence relations*.

This linking mechanism is composed of a sequence of four steps. These steps are: *specification*, *parsing*, *matching,* and *granularity control*. First, the user must *specify* the source geosensor database, the target ontology and the key elements of each that are to be connected. Once these entities are identified, the linking application iteratively selects each database value and *parses* the ontology structure for a potential *match*. If a *match* is found, an equivalence relation is generated. Once a list of equivalence relations has been created, the user has the ability to choose the desired *granularity* of the results, thereby choosing a higher-level and more abstract understanding of their domain, or a more refined view depending upon their needs.

The remainder of this chapter describes a tool, in the form of a standalone application, which employs the linking mechanism that has been introduced. This tool provides a number of objects that are the core parameters used for specifying, viewing, and manipulating the derived equivalence relations that are identified by linking a database with an ontology. In addition to returning a set of equivalence relations, the linking tool provides a *search depth slider* that allows the user to exploit the ontology further be coarsening the granularity of the equivalence relations. It will be shown in chapter five that this generalization increases the volume of knowledge obtained through linking.

CHAPTER 5

APPLYING THE LINKING MECHANISM TO MOVING OBJECT RELATIONS

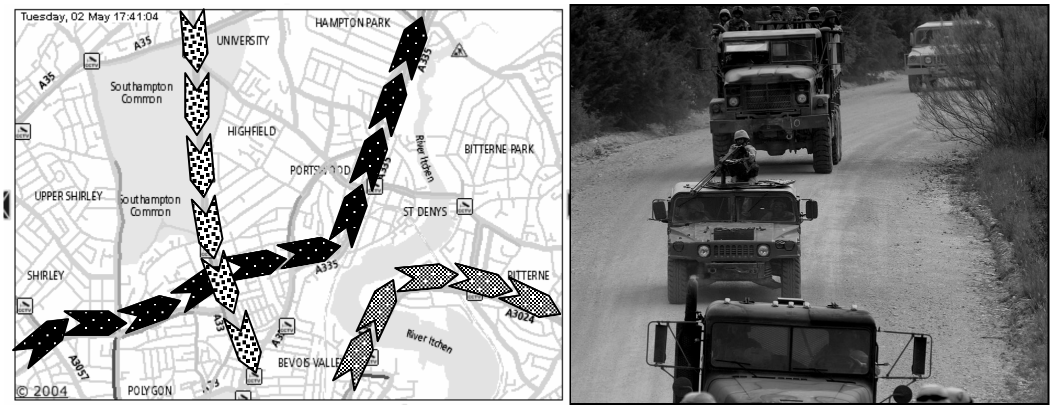
Positional data collected from a geosensor network and stored in a spatio-temporal database, provides a foundation for computing relations between moving objects. Particular configurations of moving objects, based on their spatial position, give rise to a number of *motion relations* such as *isBehind*, and *inFrontOf*. These relations supply a user context about binary vehicle positions relative to a reference object. For example, the driver of a military supply truck may be interested in knowing what kinds of vehicles are in front of the truck.

Utilizing the linking mechanism introduced in Chapter 4, the spatio-temporal database can be augmented with details from a related ontology to extend this motion relation information. It will be demonstrated later in this chapter that linking facilitates multi-granular perspectives of the moving objects (and their corresponding motion relations) by providing abstractions or higher-level perspectives of preliminary spatio-temporal data.

5.1 Introducing Motion Relations

Military support vehicles, tanks, armored personnel carriers, and amphibious assault vehicles are just some of the moving entities that may be encountered on a battlefield. In this chapter, we formalize binary configurations of vehicles that are commonly experienced by drivers on a road network or predefined path. A better understanding of these configurations enables next-generation information systems to represent patterns of movement more fully, in turn advancing vehicle navigation. For example, an information system may alert drivers or other active participants on the road to the different kinds of vehicles that are positioned around them (e.g., a supply truck is behind you, or an armored personnel carrier drives beside your vehicle). These configurations are formalized as *motion relations* that capture the position of a pair of vehicles relative to each other on the road network. These relations are derived from vehicle positional data which is collected from a geosensor network and stored in a spatio-temporal database.

Typically, details of vehicle movement are represented as flow lines on a map (Figure 5.1a). This is an abstraction of the actual movements which captures a high-level view, for example, of slow, moderate, or free-flowing traffic patterns. The relations presented in this work, however, model the relative positions of two vehicles with respect to each other. These are the kinds of vehicle movements that a person experiences while driving, and correspond to relations that can be extracted from imagery captured by cameras (Figure 5.1b) or datasets of sensor-derived vehicle movements. This research offers additional perspectives, that are complementary to ongoing research on other topics relating to moving objects, such as, computing trajectories of moving objects where the primary focus relates to modeling the *path* of a moving object supporting queries such as *where should I turn next?* or *how much farther is it to the target objective?* (Wolfson *et al.* 1998; Güting *et al.* 2000; Pfoser *et al.* 2000).



(a) (b)

Figure 5.1. Traffic flow (a) map representation and (b) imagery showing the position of vehicles relative to each other.

A driver or other active participant on the road network also needs to know about the types of vehicles that are in their immediate vicinity (e.g., an armored personnel carrier is behind them). These semantics are important for understanding and modeling the behavior of moving entities traveling on a road network or predefined route. Motion relations such as *isBehind* and *inFrontOf* assist in providing a *user context* for the kinds of moving objects that are around the driver of a subject vehicle (Hage *et al.* 2003). They can also be used in monitoring travel scenarios, such as convoy patterns, where it is important to understand the relative position of each vehicle involved.

Semantic modeling frequently involves ontologies that describe the entities and relations known for a domain. In this work, the ontology of *military transportation devices* provides a typology for the different classes of moving objects represented in a motion relation. The principal locations of two or more moving vehicles on a road are used to distinguish a set of basic *motion relations* (i.e., an elementary set of relations between a pair of moving objects) that describe a pattern of movement. The basic relations introduced in this section correspond to the relative positions of vehicles with respect to each other within a road network. Other kinds of movement, such as movements that result in changes in the orientation of vehicles (e.g., rolling or spinning) are possible, but these types of movement are outside of the scope of this work.

Two motion relations are introduced that help aid next-generation information systems: *isBehind* and *inFrontOf*. These motion relations are expressed in the form relation(*X,Y,T*) where *X* and *Y* are variable terms that refer to either a *reference object* (e.g. my car) or a *target object* (e.g., the vehicle behind my car) moving on the network at variable time *T*. Specific instances of objects are represented using constants, and are indicated by lower case letters (e.g., *relation(x,y,t)*). Classes of vehicles that are commonly found on roadways can be used to populate the *isBehind* and *inFrontOf* relations, and to systematically derive possible combinations of moving objects (e.g., *isBehind(supportVehicle,militaryTank,T)*).

These relations are based on the linear ordering imposed on traffic by the design of transportation networks (i.e., lanes of traffic). The relation *isBehind(Targ,Ref,T)* and its converse relation, *inFrontOf(Ref,Targ,T)*, describe the relative spatial relation between two moving objects (e.g., different military transportation devices) in the same lane of traffic at time *T*, such that no other object is between them (Figure 5.2). Although these two relations are understood to be a subset of a broader possible set of relations, the sections that follow show the challenges inherent in formalizing these relations and the benefits realized by linking the preliminary geospatial database knowledge with a related ontology. It will be demonstrated that ontologies in combination with the sensor-derived positional information, allow for higher-level reasoning about the kinds of vehicles near to a driver at variable time *T*.

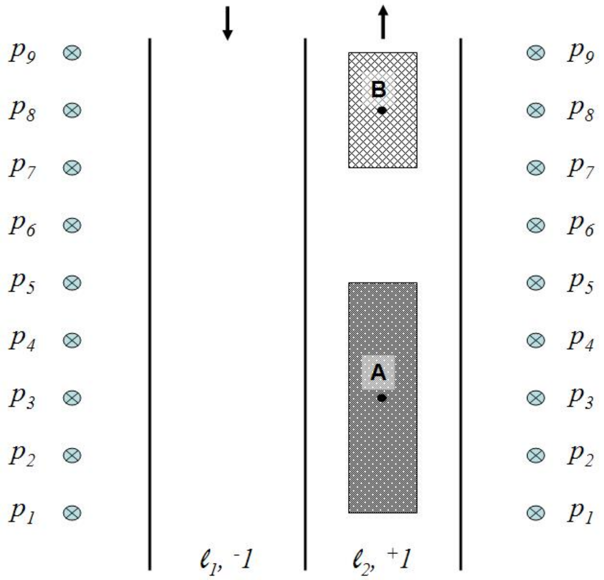


Figure 5.2. Modeling vehicle movement at time t, where *isBehind(a,b,T)* and *inFronfOf(b,a,T)*

5.2 Modeling the *IsBehind* Relation

To retrieve motion relations between two objects, the three database relations *SensorData*, *ObjData,* and *LaneData* are used to provide the basis for formalizing supporting queries that capture specific motion semantics. Based on the vehicle data captured for vehicle position and stored within the spatio-temporal database, a motion relation query will return the vehicle identification number as well as the object type, position, and time. These attributes are used to inform the driver of the objects around their vehicle, *the reference*, at a given time *T*.

To query, for example, *what moving object is currently behind my supply truck?* (i.e., *isBehind(Targ,ref,T)*), two datasets (*senRef* and *senTarg*) are created from the *SensorData* relation. The first of these, *senRef*, contains data corresponding to the object whose perspective is being considered (i.e., '*supplytruck*' the reference object with object id 'ME0692'). The second dataset, *senTarg*, contains all the sensed objects that are in the same lane as the reference object. Furthermore, the data is filtered further to include only sensor readings that were taken at the requested time *CurrentTime*, for example, '22-Jan-07 10:31:09'. To facilitate the query logic, the *senRef* and *senTarg* datasets are each joined with the *ObjData* and *LaneData* relations to provide access to the object length and lane direction attributes. The structured query language (SQL) expression for this query is composed of the set of statements shown in Figure 5.3.

inFrontOf(Target, ME0692, CurrentTime):

```
SELECT  senTarg.objID, objTarg.objType, senTarg.pos, senTarg.time, senRef.objID, objRef.objType, senRef.pos

FROM    (sensorData AS senRef INNER JOIN objData AS objRef ON senRef.objID = objRef.objID) INNER JOIN
        laneData AS laneRef ON senRef.laneID = laneRef.laneID,

        (sensorData AS senTarg INNER JOIN objData AS objTarg ON senTarg.objID = objTarg.objID) INNER
        JOIN laneData AS laneTarg ON senTarg.laneID = laneTarg.laneID

WHERE  senRef.objID = "ME0692" AND
        senRef.laneID = senTarg.laneID AND
        senRef.time = #CurrentTime# AND
        senTarg.time = #CurrentTIme# AND
        ( (senRef.pos+laneRef.laneDir*.5*(objRef.length)) - (senTarg.pos-laneTarg.laneDir*.5*(objTarg.length)))
        BETWEEN 0 AND (-3*laneRef.laneDir);
```

Figure 5.3. SQL definition for isBehind (Target,"ME0692",CurrentTime)

The last complete clause of this expression stipulates that only tuples should be returned that satisfy the condition that the object being searched for is a maximum of three positions behind the reference object. This threshold is based on the assumption that three vehicle lengths constitutes being behind another vehicle. For *inFrontOf*, we similarly assume three vehicle lengths in front of the reference vehicle. This criterion can be modified as necessary, in order to satisfy the constraints of other domains.

The SQL code first reconstructs the spatial region occupied by each vehicle and then locates any moving objects around it that satisfy the *isBehind* constraint relative to the reference object. The spatial region occupied by each vehicle is calculated using its centroid (i.e., sensed position) and its length (represented as a fixed number of positional units). The rear position of the reference object is calculated by either adding or subtracting half of its length from its midpoint. The decision to add or subtract half the length is based on the positive or negative notation of lane direction. This approach allows the motion relation between two objects to be determined regardless of the lane direction. Thus, the position of the reference object's rear extent is expressed as senRef.pos – (laneRef.laneDir * 0.5 * objRef.Length). The position of the object following the reference (i.e., target) is calculated in a similar way, however, for this car the most forward position is now of interest; i.e., senTarg.pos + (laneTarg.laneDir * 0.5 * objTarg.Length).

For example, in Figure 5.4, the rear of reference *B* would extend to location [8 – (+1 * 0.5 * 2) = 7] and the front of target object A is at [3 + (+1 * 0.5 * 4) = 5]. The most forward position of object *A* is subtracted from the most rearward position of the reference vehicle to determine the difference between them. If this value falls between zero and three times the lane direction, then the definition of *isBehind()* has been satisfied. In the current example, the difference is 2, so the query returns that object *A* is behind the reference (i.e., object *B*). Formally, the query would return values for the target attributes *senTarg.objID*, *senTarg.objType*, *senTarg.pos*, *senTarg.time*, and the reference attributes *senRef.objID, senRef.objType, senRef.pos*.

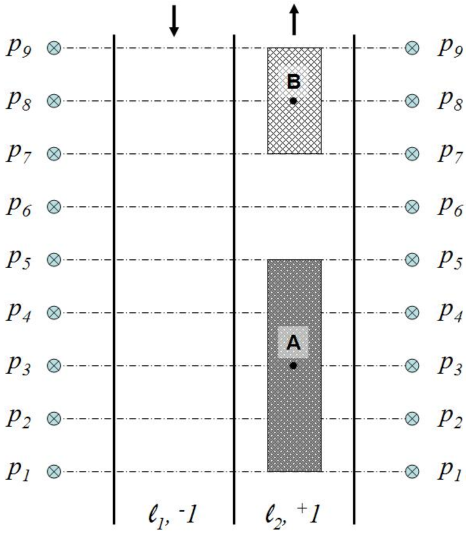


Figure 5.4. Vehicle *A* is behind vehicle *B* with laneDir = +1

5.3 Modeling the *InFrontOf* Relation

A similar formalization can be used for the motion relation *inFrontOf(Ref,Targ,T)* given that it is the converse of *isBehind(Targ,Ref,T)*. The only modification to the SQL expression necessary is that the last complete clause must be changed such that it satisfies the condition that the object being searched for is a maximum of three positions *ahead* of the reference object (Figure 5.5).

Verifying the *inFrontOf()* constraint is accomplished by taking the rear extent of the target object (i.e., senTarg.pos-laneTarg.laneDir*.5*(objTarg.length)) believed to be in front of the reference, and then calculating its difference from the reference's most forward position (i.e., senRef.pos+laneRef.laneDir*.5*(objRef.length)). If this value falls between zero and negative three times the lane direction, which is the assumed distance threshold, then the criteria for the *inFrontOf* relation is satisfied.

```
inFrontOf(Target, ME0692, CurrentTime):
SELECT  senTarg.objID, objTarg.objType, senTarg.pos, senTarg.time, senRef.objID, objRef.objType, senRef.pos

FROM    (sensorData AS senRef INNER JOIN objData AS objRef ON senRef.objID = objRef.objID) INNER JOIN
        laneData AS laneRef ON senRef.laneID = laneRef.laneID,

        (sensorData AS senTarg INNER JOIN objData AS objTarg ON senTarg.objID = objTarg.objID) INNER
        JOIN laneData AS laneTarg ON senTarg.laneID = laneTarg.laneID

WHERE  senRef.objID = "ME0692" AND
        senRef.laneID = senTarg.laneID AND
        senRef.time = #CurrentTime# AND
        senTarg.time = #CurrentTIme# AND
        ( (senRef.pos+laneRef.laneDir*.5*(objRef.length)) - (senTarg.pos-laneTarg.laneDir*.5*(objTarg.length)))
        BETWEEN 0 AND (-3*laneRef.laneDir);
```

Figure 5.5. SQL definition for inFrontOf (Target,“ME0692”,CurrentTime)

5.4 Augmenting the Moving Object Database with Data from an Ontology

The analysis of motion relations between pairs of moving objects, with respect to a particular reference object, is valuable knowledge for next-generation transportation systems. For example, drivers may be provided with a user context that is based on the kinds of moving objects around them. Derived from the vehicle position data that has been captured and stored within the spatio-temporal database, one of the attributes returned by the motion relation queries is *objType*. This attribute may not necessarily be meaningful to drivers trying to comprehend the vehicles around them at a given time.

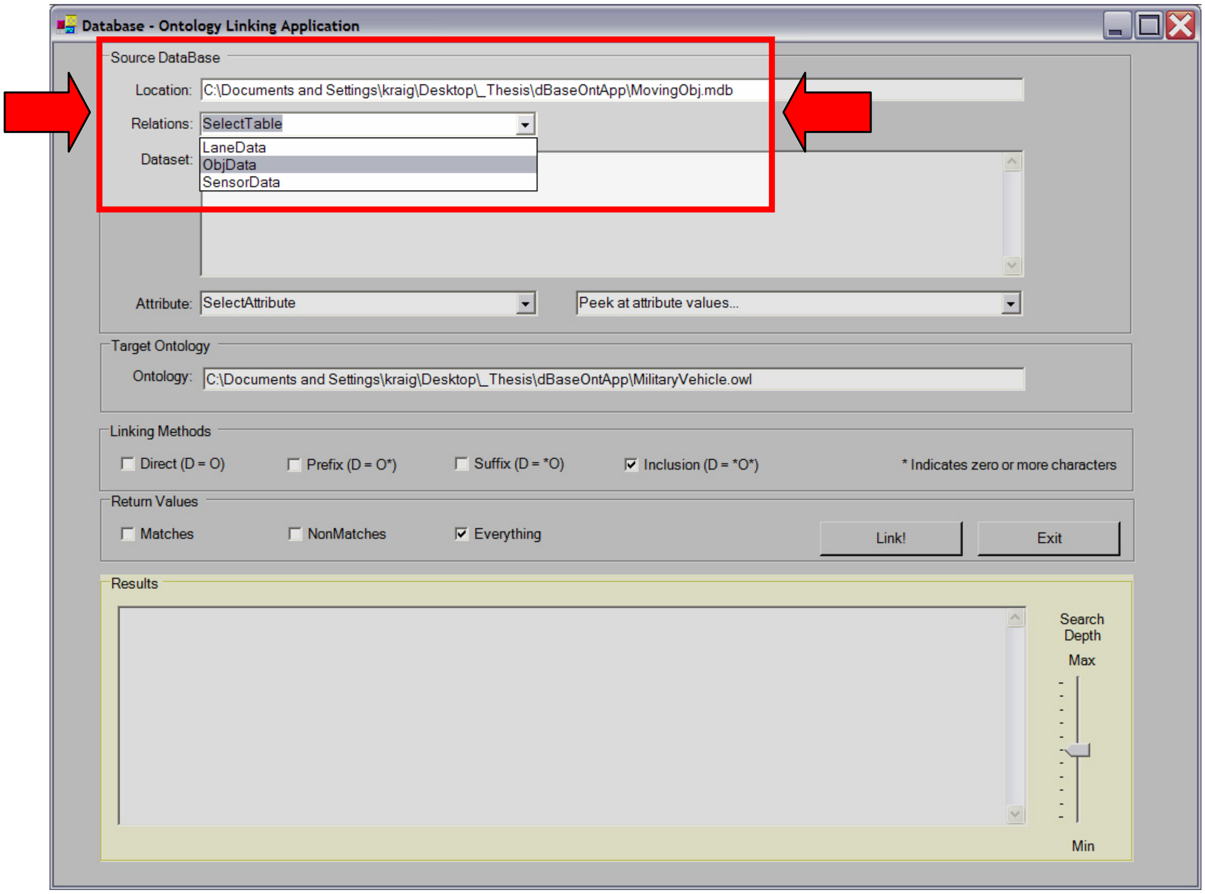
Such sensor data often lacks the semantic detail necessary to formulate an understanding of the objects involved in the motion relation. To improve upon this situation, knowledge from the spatio-temporal *moving object database* can be augmented by linking it with the ontology of *military transportation devices* to provide higher-level or more abstract views of the vehicles within the relation. By modifying the semantic granularity of the database knowledge, additional classifications about the moving objects are extracted and the benefits of the geosensor data stream are maximized (Hornsby and King 2007).

Consider the scenario where a military supply truck is cautiously traveling down a winding mountain road at night and the driver of the truck wants to know, *if the next vehicle in front of the truck is also a military vehicle.* A query of the sensor database searches for any tuples that would satisfy the requirements of the *inFrontOf()* formalization. Suppose, for example, that a tuple [LAV117 | LightArmoredVehicle | 734 | 4/10/07_14:23:56] is found and returned. The attributes comprising the tuple (i.e., *objID*, *objType*, *pos* and *time*) lack the semantic meaning required to evaluate if the object in front of the truck is a kind of military vehicle. However, additional knowledge about the classifications of moving object types can be exploited by linking the *military transportation device* ontology with the information supplied by this tuple.

Existing applications such as MAPONTO (An *et al.* 2006) and VisAVis (Konstantinou *et al.* 2006) have demonstrated methods to map database schemas with elements of a related ontology. However, more information may be discovered by linking the instance-level data stored within the database to specific class names and related attributes from an ontology (Hornsby and King 2007). The framework discussed for linking a geosensor database with an ontology in chapter 4, is drawn upon in the remainder of this chapter to extend the preliminary geosensor database knowledge of the moving objects.

5.4.1 Applying the Linking Application

The core components of the linking interface are a set of objects that specify the database, ontology, matching and linking parameters. Proper navigation of these parameters is facilitated by the organization and layout of the graphical user interface. The user must first identify the data source name for the geosensor database containing the moving object data, (it is assumed that an ODBC connection to this database has already been created on the host PC).



Database - Ontology Linking Application

Source DataBase

Location: C:\Documents and Settings\kraig\Desktop_Thesis\dBaseOntApp\MovingObj.mdb

Relations: SelectTable

- LaneData
- ObjData
- SensorData

Dataset:

Attribute: SelectAttribute

Peek at attribute values...

Target Ontology

Ontology: C:\Documents and Settings\kraig\Desktop_Thesis\dBaseOntApp\MilitaryVehicle.owl

Linking Methods

☐ Direct (D = O) ☐ Prefix (D = O*) ☐ Suffix (D = *O) ☑ Inclusion (D = *O*)

* Indicates zero or more characters

Return Values

☐ Matches ☐ NonMatches ☑ Everything

Figure 5.6. Relations available for linking from the source database

A subroutine within the linking application verifies the validity of the database connection and then populates a list box object with the name of each relation that is available for linking, such as, *ObjData*, *SensorData,* and *LaneData* (Figure 5.6). This information is used in the creation of a SQL query, entered into a list box within the linking interface, which defines the dataset that contains the preliminary moving object knowledge. For example, a dispatcher may inquire about what kind of moving object satisfies the *inFrontOf* relation, at time *'9/22/2007 12:42:00 PM',* where the reference object is a supply truck with id '*ST330'*. From the previous discussion of the *inFrontOf( )* motion relation, it is understood that this query requires positional data from the *SensorDat* relation, vehicle data from the *ObjData* relation, and roadway infrastructure data from the *LaneData* relation (Figure 5.7).

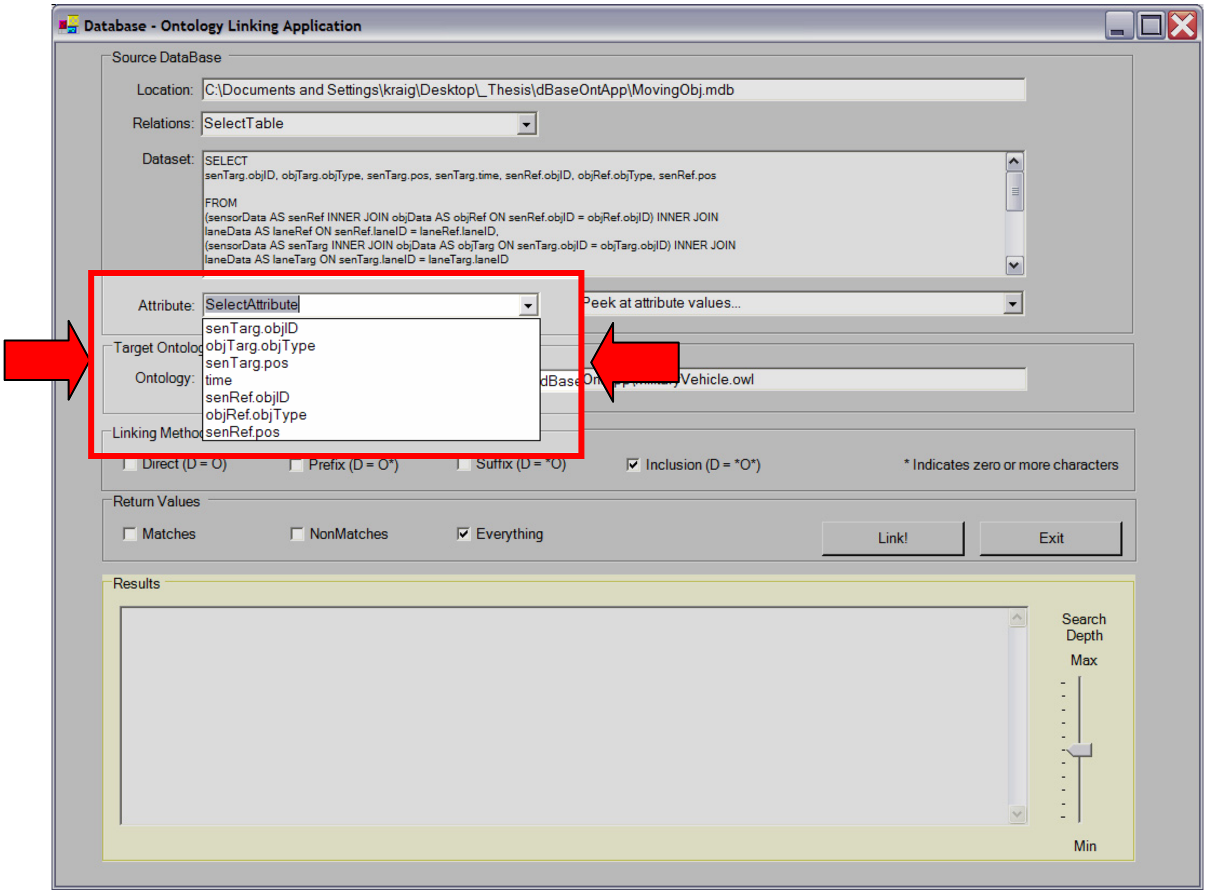
```
SELECT
senTarg.objID, objTarg.objType, senTarg.pos, senTarg.time, senRef.objID, objRef.objType, senRef.pos

FROM
(sensorData AS senRef INNER JOIN objData AS objRef ON senRef.objID = objRef.objID) INNER JOIN
laneData AS laneRef ON senRef.laneID = laneRef.laneID,
(sensorData AS senTarg INNER JOIN objData AS objTarg ON senTarg.objID = objTarg.objID) INNER JOIN
laneData AS laneTarg ON senTarg.laneID = laneTarg.laneID

WHERE
objRef.objID = “ST330” AND
senRef.laneID = senTarg.laneID AND
senRef.time = #9/22/07 12:42:00 PM# AND
senTarg.time = #9/22/07 12:42:00 PM# AND
((senRef.pos+laneRef.laneDir*.5*(objRef.length)) - (senTarg.pos-laneTarg.laneDir*.5*(objTarg.length)))
BETWEEN 0 AND (-3*laneRef.laneDir);
```

Figure 5.7. SQL definition for inFrontOf (Target, ST330, 9/22/07 12:42:00 PM)

Following the specification of a desired dataset with valid SQL statements, a second list-box gets populated with available attributes (Figure 5.8). For this scenario, the set of attributes available for selection would be: *senTarg.objID*, *objTarg.objType*, *senTarg.pos*, *senTarg.time, senRef.objID*, *objRef.objType*, and *senRef.pos*. When one of these attributes is selected for linking, its corresponding instance values can be previewed by scrolling through a third list-box. This enables the user to verify the desired dataset has been selected prior to linking the preliminary geosensor database knowledge with a related ontology. For example, if the attribute *objRef.objType* is selected, this list-box displays the instance value *ArmoredVehicle-25*.



Database - Ontology Linking Application

Source DataBase

Location: C:\Documents and Settings\kraig\Desktop_Thesis\dBaseOntApp\MovingObj.mdb

Relations: SelectTable

Dataset:

```
SELECT
senTarg.objID, objTarg.objType, senTarg.pos, senTarg.time, senRef.objID, objRef.objType, senRef.pos

FROM
(sensorData AS senRef INNER JOIN objData AS objRef ON senRef.objID = objRef.objID) INNER JOIN
laneData AS laneRef ON senRef.laneID = laneRef.laneID,
(sensorData AS senTarg INNER JOIN objData AS objTarg ON senTarg.objID = objTarg.objID) INNER JOIN
laneData AS laneTarg ON senTarg.laneID = laneTarg.laneID
```

Attribute: SelectAttribute

senTarg.objID
objTarg.objType
senTarg.pos
time
senRef.objID
objRef.objType
senRef.pos

Peek at attribute values...

Target Ontolog

Ontology: dBaseOn Vehicle.owl

Linking Metho

□ Direct (D = O) □ Prefix (D = O*) □ Suffix (D = *O) ☑ Inclusion (D = *O*)

* Indicates zero or more characters

Return Values

□ Matches □ NonMatches ☑ Everything

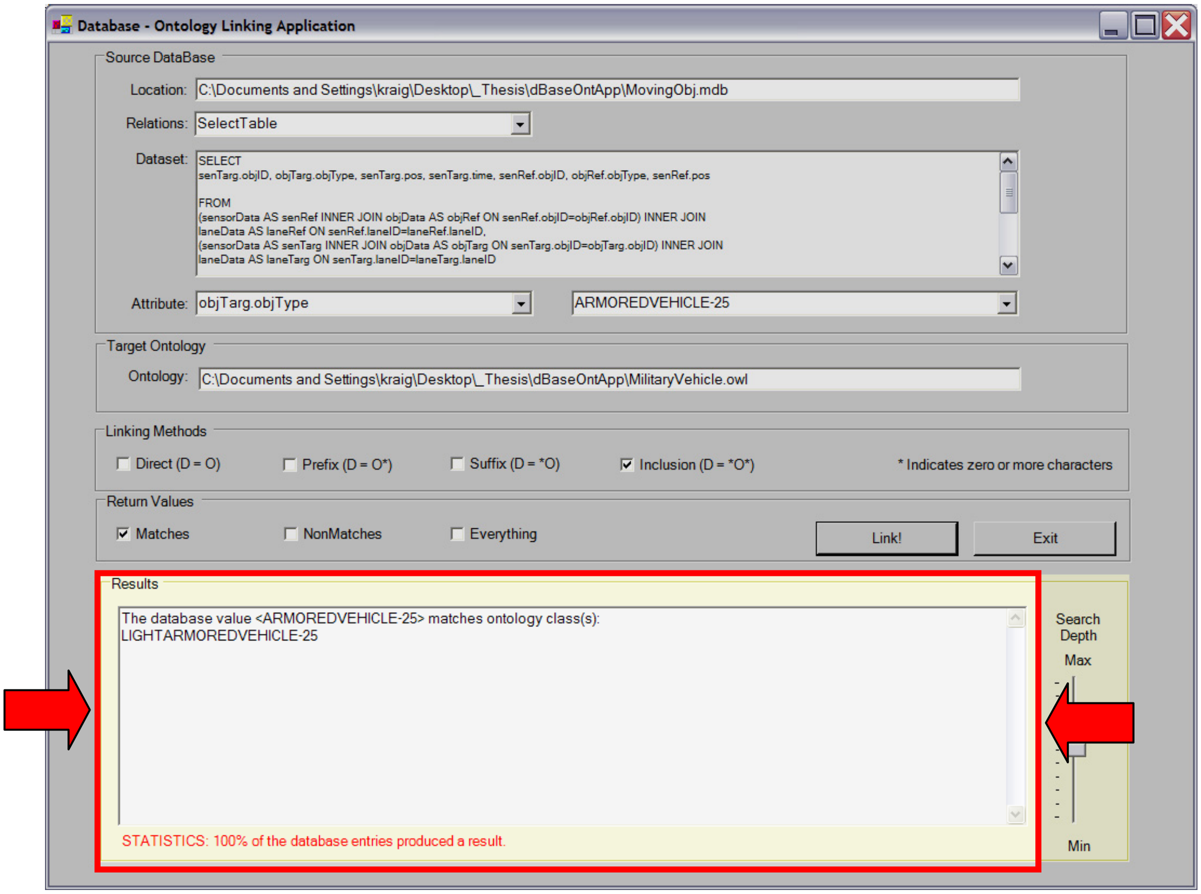
Figure 5.8. Available attributes from the user specified dataset

After the instance-level values are selected, the user can proceed to specify the ontology that is expected to supply additional semantic details for the preliminary geosensor database knowledge. Currently, the linking application requires that this ontology file must be saved in native OWL/RDF format.

Next, the matching options that control the logic used by the application for determining similarity between database instances and ontology class names must be specified. One of four linking methods can be specified within the interface. These methods are *direct*, *prefix*, *suffix* and *inclusion* as presented in chapter 4. For this example, we elect inclusion, where each database pattern is matched with corresponding ontology elements of the form **pattern**. Inclusion yields the largest number of database-ontology matches, thus maximizing the amount of knowledge extracted by linking.

The decision to return *matches* (i.e., equivalence relations), *non-matches* or all comparisons (both matches and non-matches) is the next feature of the linking interface that must be specified. It is anticipated that the *matches* option would be used most frequently as this option will extend the geosensor data stream knowledge. Thus, the *matches* option will be used in this example to extend knowledge about the moving objects. It should be noted that the *non-matches* option can also be useful, especially in situations where one wants to create a list of all database terms that do not have a corresponding equivalence relation within the target ontology. Such a comparison may be performed, for example, by a domain expert that is looking to extend the knowledge contained within the *military transportation device* ontology.

Once all required parameters have been specified, the output generated by the linking application is displayed to the user within the *results* window. For each equivalence relation found, the database *reference* term and matching *target* classes from the ontology are returned. (If the user had requested non-matches, only database values would have been returned.) The set of equivalence relations generated from the linking of the *ObjData* relation and the *military transportation device* ontology is [*lightarmoredvehicle-25* | *lightarmoredvehicle-25*]. For this scenario, a single equivalence relation was returned (Figure 5.9) due to pruning of the initial dataset to specify which vehicle is *inFrontOf* the supplytruck with ID *ST330* at time *'9/22/2007 12:42:00 PM'*.



Database - Ontology Linking Application

Source DataBase

Location: C:\Documents and Settings\kraig\Desktop_Thesis\dBaseOntApp\MovingObj.mdb

Relations: SelectTable

Dataset:

```
SELECT
senTarg.objID, objTarg.objType, senTarg.pos, senTarg.time, senRef.objID, objRef.objType, senRef.pos

FROM
(sensorData AS senRef INNER JOIN objData AS objRef ON senRef.objID=objRef.objID) INNER JOIN
laneData AS laneRef ON senRef.laneID=laneRef.laneID,
(sensorData AS senTarg INNER JOIN objData AS objTarg ON senTarg.objID=objTarg.objID) INNER JOIN
laneData AS laneTarg ON senTarg.laneID=laneTarg.laneID
```

Attribute: objTarg.objType ARMOREDVEHICLE-25

Target Ontology

Ontology: C:\Documents and Settings\kraig\Desktop_Thesis\dBaseOntApp\MilitaryVehicle.owl

Linking Methods

☐ Direct (D = O) ☐ Prefix (D = O*) ☐ Suffix (D = *O) ☑ Inclusion (D = *O*) * Indicates zero or more characters

Return Values

☑ Matches ☐ NonMatches ☐ Everything

Results

```
The database value <ARMOREDVEHICLE-25> matches ontology class(s):
LIGHTARMOREDVEHICLE-25
```

STATISTICS: 100% of the database entries produced a result.

Figure 5.9. Equivalence relations returned by the linking application

In addition to displaying the matching terms of a geosensor database and ontology, a feature has been added to the linking interface that allows a user to exploit the ontology further by coarsening the desired granularity of the equivalence relations. The *search depth slider* is the mechanism used to augment the linking results. The initial level that serves as the foundation for the depth, is the current ontological class immediately derived from the equivalence relation. In this scenario, *lightarmoredvehicle-25* is the starting point (i.e., origin) for the search depth slider and the range of coarsening is defined as the distance to the root of the ontology. Decreasing the search depth augments the equivalence relations by returning ontology classes that are more generalized by locating subsuming classes. For example, starting with an equivalence relation match [*lightarmoredvehicle-25* | *lightarmoredvehicle-25*], using the search depth slider to perform a first order generalization, the application returns [*lightarmoredvehicle-25* | *lightarmoredvehicle*] where *lightarmoredvehicle* is coarser than *lightarmoredvehicle-25* (Figure 5.10).

If the search depth slider is adjusted by two increments from the origin, the resulting output becomes [*lightarmoredvehicle-25* | *armoredpersonnelcarrier*], where class *lightarmoredvehicle* is a subclass of *armoredpersonnelcarrier*, and similarly the class *lightarmoredvehicle-25* (i.e., the origin) is a subclass of *lightarmoredvehicle*. The search depth slider can be applied iteratively until the uppermost node of the ontology (e.g., *transportationdevice*) is reached. These newly discovered generalizations of preliminary ontology classes are displayed to the user, as well as the original equivalence relations derived via linking.

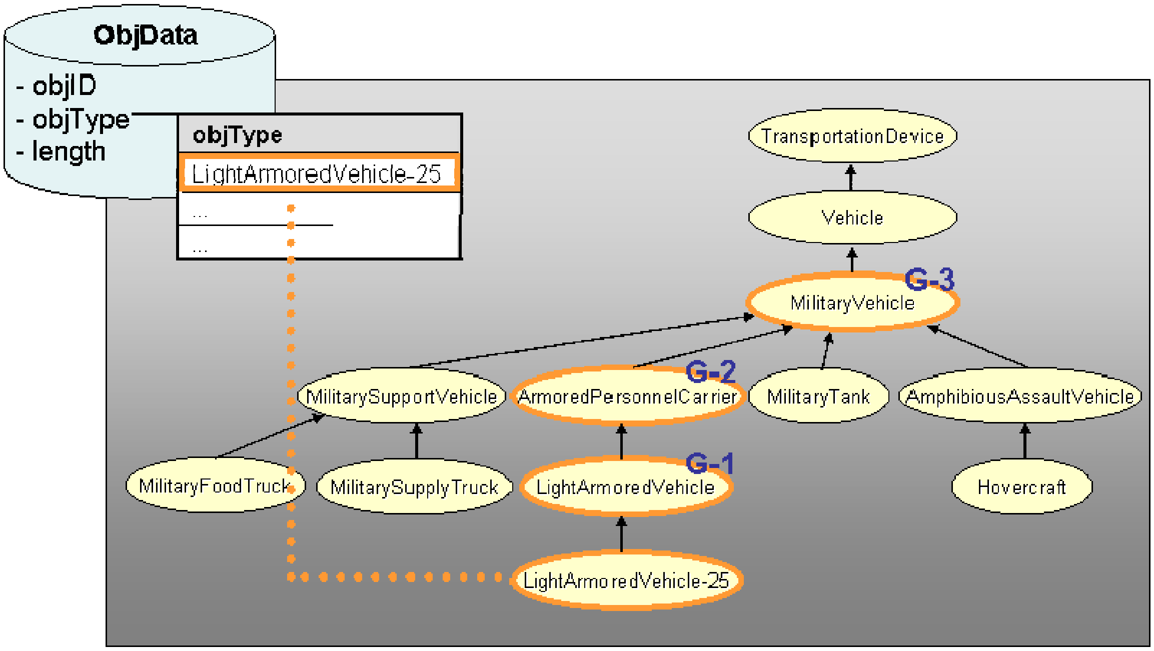


Figure 5.10. Generalization of the ontology class *lightarmoredvehicle-25* where the 1st generalization = **G-1**, 2nd generalization = **G-2** and 3rd generalization = **G-3.**

5.4.2 Analyzing Linking Results

By linking the moving object database with the *military transportation device* ontology and then augmenting the results through generalization, additional knowledge beyond what is known from the initial geosensor data stream is obtained. For example, the previous scenario demonstrated how the linking tool found an equivalence relation between the database and ontology, for the sensed object of type *lightarmoredvehicle-25*. This initial knowledge is then augmented further by applying two iterations of generalization using the search depth slider, which reveals that the sensed moving object of type *lightarmoredvehicle-25* can be generalized to class *armoredpersonnelcarrier*. The success of the linking tool at identifying equivalence relations, and the resulting knowledge gained from these relations, provides alternative semantic descriptions that facilitate higher level inferencing.

A quantitative analysis of the linking results provides a measure of the additional knowledge acquired through the use of the linking application. This analysis assumes that all database-ontology matches are meaningful and applicable, although in general practice some matches may be returned that are either erroneous (e.g. 'civilian' **personnelcarrier** matched with *armoredpersonnelcarrier*), or duplicate knowledge (*lightarmoredvehicle-25* matched with *lightarmoredvehicle-25)*. It is difficult to test for cases of erroneous data because the criterion is subjective to human interpretation. However, the latter case of duplication is dealt with by ignoring identical matches when quantifying the number of equivalence relations that contributed to linking. Although equivalence relations that return duplicate data are ignored in the statistical analysis, these matches are still valid and must be returned by the interface as they may be the origin for generalization that supplies supplementary knowledge.

At this point in this thesis we are able to support the hypothesis presented in chapter one. The hypothesis stated, "*Generalization based on the set of returned equivalence relations, extends existing database knowledge by adding additional attribute information that is drawn from a related ontology".* As a means for supporting this hypothesis, it is demonstrated that the range of attribute values is increased based on the number of generalizations that are possible.

The amount of knowledge gained through generalization is $\{K(g), g \in O\}$, where g is the set of generalizations that are produced from an ontology O. The function K(g) is formally defined as: $K(g) = \sum_{i=0}^{m} \sum_{j=0}^{r_i} (1 + g_j - d_j)_i$. Within this equation, m represents the number of databases instances i available for linking, and r_i represents the number of equivalence relations found for each database instance i. The variable d is the number of matches that produce duplicate knowledge, and this value is removed from the total knowledge gained through generalization. It should be noted that $(0 \leq d \leq g)$, therefore, the number of duplicates can only be within a range of 0 (i.e., no duplicates) to g (i.e., all generalizations produce duplicate knowledge). In practice it is unlikely that the value of d will ever exceed 1, as long as the ontology does not contain duplicate information.

This function K(g) can be used to analyze the previous scenario in section 5.4.1 that considered the motion relation *InFrontOf(Target, ST330)*. The motion relation query retuned a single database instance ($m = 1$) having the value *lightarmoredvehicle-25*. Through linking, one corresponding equivalence relation ($r_i = 1$) was located between this database instance and a related ontology. This equivalence relation, [*lightarmoredvehicle-25* | *lightarmoredvehicle-25*], can be augmented via generalization with repeated iterations of the search depth slider. Assuming the maximum number of iterations (i.e, the uppermost node of the ontology is reached), this equivalence relation is generalized five times ($g = 5$) returning: [*lightarmoredvehicle-25* | *lightarmoredvehicle*], [*lightarmoredvehicle-25* | *armoredpersonnelcarrier*], [*lightarmoredvehicle-25* | *military-vehicle*], [*lightarmoredvehicle-25* | *vehicle*], [*lightarmoredvehicle-25* | *transportation-device*]. Of all the equivalence relations returned only the initial one produced duplicate knowledge [*lightarmoredvehicle-25* | *lightarmoredvehicle-25*], therefore, $d = 1$. The resulting additional knowledge obtained by generalization is therefore K(g) = (1+5-1) = 4. Therefore, for the single database instance *lightarmoredvehicle-25* is determined that the number of additional attributes is equal to four, confirming our hypothesis that generalization based on the set of returned equivalence relations does in fact extend the initial database knowledge.

5.5 Summary

This chapter provided a foundation for describing *motion relations* between vehicles moving on a road network. Relations such as *isBehind* and *inFrontOf* capture movement semantics from the viewpoint of one of the vehicles involved in the relation (i.e., the reference). Assuming an underlying sensor framework that captures positional data about moving objects, a formalization of these relations in a database representation demonstrates how queries over these relations may be formulated. We show how relational queries using SQL are used to derive the relations, returning information about pairs of moving objects and their relative positions.

The results returned by the motion relation queries are augmented further by extracting additional details from a related ontology using the linking application. Ontologies play an important role in this work as they further describe the entities and their relations. The kinds of objects that participate in these motion relations correspond to classes within the *military transportation device* ontology that is derived from the SUMO framework. Linking the database to the ontology facilitates more abstract or higher-level perspectives of moving objects, supporting inferences about moving objects over multiple levels of granularity. The details supplied by the generalization of geosensor data via linking, helps to interpret semantics and respond to user questions by extending the preliminary knowledge about the moving objects within the relations. It is anticipated that deriving motion relations from geosensor data and further augmenting these results by linking with an ontology, will assist next generation transportation systems in better understanding and modeling moving objects.

Near the end of this chapter, a function K(g) was introduced to quantify the amount of additional knowledge gained through generalization. This function was then used to analyze the results of the linking application as applied to a motion relation query *InFrontOf(Target, ST330)*. This analysis demonstrated that generalization produced four additional (non-duplicate) attributes for a single database instance, thus confirming our hypothesis that generalization based on the set of returned equivalence relations extends initial database knowledge.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This thesis investigates methods for combining data from geosensor networks by linking the databases storing sensor data with related ontologies. By linking the instance-level data stored within the database to the class and attribute names in a related ontology, additional knowledge of the objects can be derived. This new information is augmented further through the process of generalization.

The first part of this chapter follows the structure of the thesis and provides a summary of the research, as well as highlighting major results. The remainder of the chapter discusses future research topics, including additions and enhancements to the linking application that will extend its usefulness.

6.1 Thesis Summary

The main goal of this thesis is to investigate and develop a framework and software tool for linking a geosensor moving objects database with a related ontology, such that more information about the preliminary sensor data can be derived. The resulting list of matches that hold between the database and ontology aids in the semantic understanding of the moving objects being sensed. The development and testing of the linking tool provides evidence that *generalization based on the set of returned equivalence relations, extends existing database knowledge by adding additional information that is drawn from a related ontology.*

This thesis considers four primary topics associated with the linking of geosensor databases with related ontologies. These topics are summarized with the following research questions.

- Q1. What type of software device can be used to link instance level data within a geosensor database to the classes, attributes and instances of a related ontology?
- Q2. How does this linking mechanism facilitate augmentation of the preliminary geosensor database knowledge?
- Q3. What methods can be used to further derive generalizations of the preliminary knowledge about the sensed moving objects?
- Q4. Can generalization techniques enable an automated system to further evaluate spatio-temporal relationships between two moving entities?

A framework for sensing and storing the position of moving objects that are traveling on a road network or predefined route is presented in this thesis. This storage structure is based on three primary database relations. The relation *SensorData* with attributes *objID*, *sensorID*, *laneID*, *position*, and *time* stores the location readings from sensors within the network. A second relation, *LaneData*, is based on a lookup table of road infrastructure attributes and provides the functionally dependant attributes *laneID* and *laneDir*. In conjunction with this sensor and lane data, specific details of the moving objects such as *objID*, *objType* and *length* are stored within the relation *ObjData*. These three relations, *SensorData, ObjData, and LaneData* provide the foundation for a geosensor database that stores motion data captured within the sensor network.

In this work, semantic details of the sensed moving objects are supplied by a *military transportation device* ontology that is derived from the SUMO knowledge base. The classes within this ontology are land-based military entities such as a supply truck or armored vehicle that move on either a road network or predefined route. These classes are related by is_a relations to form an ontology, and are possible candidates for *linking* with instance values from the moving objects database.

A four-step process for linking instance values from the geospatial moving objects database with object classes from the *military transportation device* ontology is introduced. This linking process combines independent data sources such that information is shared between them, by generating a set of database-ontology attachment points known as *equivalence relations*. Thus, in contrast to other methods such as merging, linking minimizes storage requirements for this new knowledge because a new database or ontology is not created.

As a result of this research, a standalone application that employs the linking framework is fully developed. This tool offers the ability to specify, view, and manipulate equivalence relations that are derived by linking a database with a related ontology. The linking tool also provides an object, known as a *search depth slider*, which allows a user to exploit the ontology further by coarsening the granularity of the equivalence relations. These increasingly abstract views facilitate additional semantic user contexts for the preliminary database knowledge.

This research also provides a foundation for describing *motion relations* between binary pairs of vehicles moving on a road network. Two relations are introduced, *isBehind* and *inFrontOf*, that capture movement semantics from the viewpoint of one of the vehicles involved in the relation (i.e., the reference). It is shown that these motion relations can be formalized in a database representation with SQL queries, which return information about the pairs of moving objects and their relative positions.

The results returned by the motion relation queries are augmented further by extracting additional details from a related ontology via the linking tool (i.e., search depth slider). Linking the resulting motion relations to the *military transportation device* ontology facilitates additional perspectives of the moving objects involved. Additional details, supplied by generalization, help interpret semantics and answer user questions by further extending the preliminary knowledge about the moving objects within the relations. Finally, it is demonstrated that this generalization process increases the volume of new knowledge obtained through linking.

6.2 Major Results

The first major result of this research is the introduction of a multi-step process for linking ontologies and databases. This *linking framework* is comprised of four distinct steps: *specification*, *parsing*, *matching* and *granularity control*. First, the user must *specify* the source geosensor database, the target ontology and the key elements of each that are to be connected. Once these entities are identified, the linking application iteratively selects each database value and *parses* the ontology structure for a potential *match*. If a *match* is found, an equivalence relation is generated. Once a list of equivalence relations has been created, the user has the ability to choose the desired *granularity* of the results, thereby choosing a higher-level and more abstract understanding of their domain, or a more refined view depending upon their needs. This linking framework provides a mechanism for intuitively connecting geosensor network data with a related ontology in order to discover additional object semantics and decrease information granularity through generalization. In addition, it provides a foundation for deriving a real-time understanding of dynamic geospatial domains that incorporate semantics. This research is unique from other approaches, for example MAPONTO and VisaVis, because it links the instance-level data contained in a geospatial database with the classes and attributes of a related ontology.

A second result of this thesis is the development of a software tool that provides the ability to augment a geosensor database by linking it with an ontology. This linking tool can be leveraged to provide alternative perspectives of the types of vehicles traveling on a road network, as well as semantic similarities between them. In addition, this tool includes an instrument known as a *search depth slider* that aids in the discovery of more generalized views of the preliminary database knowledge. A function K(g), is used to quantify the amount of additional knowledge gained through this generalization. By analyzing the results of the linking application as applied to a motion relation query, it is demonstrated that generalization produces a significant number of additional (non-duplicate) attributes for a single database instance. This confirms the hypothesis that generalization, based on the set of returned equivalence relations, extends initial database knowledge.

A third result of this thesis is the development of a method for distinguishing a set of motion relations that describe the position of a pair of vehicles relative to each other on a road network. Specifically, the two relations *inFrontOf* and *isBehind* are derived from vehicle positional data that is collected from a geosensor network and then stored in a spatio-temporal database. It is demonstrated that this information provides additional user contexts for binary vehicle patterns relative to a reference object. In addition, formalization for each of these motion relations is provided as a set of SQL statements.

6.3 Future Work

One possible topic for future research is relaxing the elements required for linking a geospatial database and an ontology. Currently, the linking tool only provides a mechanism for linking database instance values with class names from a related ontology. Additional database candidates for linking include attribute names, data types and metadata (Table 4.1). Similarly, one may find it beneficial to exploit the ontology further by linking any of these database elements to the class attributes and instance values from a related ontology. These additional combinations of linking elements have the potential to amplify the expressive power of the linking tool, by increasing the volume of new knowledge returned.

Another open question for future research is the extension of the spatial linking component. Geospatial knowledge can be exploited further by moving beyond lexical matching and instead investigating spatial properties such as object boundaries. For example, a focus on spatial pattern matching where an ontology is extended to include geometric details can be used to augment a positional database. This type of linking can facilitate the generalization of geometric shapes such that the complexity of their spatial representation is decreased. Such work has the potential to simplify query processing by decreasing computational overhead, and also to assist spatial pattern matching algorithms.

A third topic for further work focuses on extending the set of relations to distinguish additional types of movement such as those that involve going around a stationary vehicle. In addition to the two basic relations that have already been discussed (i.e., *inFrontOf* and *isBehind*), sequences of individual motion relations as well as associative and distributive combinations can be further explored. Additionally, the role of an object's speed and how it affects these motion relations is a related area for further examination. A rich set of motion relations can be used in conjunction with the linking tool to facilitate the discovery of additional motion patterns that could occur. One may be able to predict patterns of movement, based on the existing motion relation between two vehicles and their associated attributes from an ontology. For example, a system may be able to infer that if a tank (known to travel relatively fast) is behind a military supply truck (known to travel significantly slower) then the tank is likely to go around the truck.

In conclusion, this research has provided a linking framework and a tool that aids in the extension of preliminary geosensor database knowledge about moving objects. The application of this tool provides additional details of the sensed objects, for example information about the kinds of vehicles involved in specific motion relations. However, this research has much broader application in other disciplines such as biology, medicine and intelligence where one can benefit from the additional semantic detail gained by linking a database and an ontology.

REFERENCES

Agarwal, P. (2005). Ontological considerations in GIScience. *International Journal of Geographical Information Science* 19(5): 36.

An, Y., J. Mylopoulos, and A. Borgida (2006). Building semantic mappings from databases to ontologies. *Proceedings of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*. American Association for Artificial Intelligence (AAAI Press) / The MIT Press: pp 450-455.

Angelova, G., O. Kalaydjiev and A. Strupchanska (2004). Domain ontology as a resource providing adaptivity in eLearning. *On the Move to Meaningful Internet Systems 2004*, Cyprus. *Lecture Notes in Computer Science*, Springer: pp. 700-712.

Arpinar, B., A. Sheth, C. Ramadkrishnan, L. Usery, M. Azami, and M.-P. Kwan (2006). Geospatial ontology development and semantic analytics. *Transactions in GIS* 10(4): 551–575.

Bhattacharya, A. and S. Das (1999). An information-theoretic approach to track mobile users in PCS networks. *Proceedings of the fifth ACM/IEEE International Conference on MobileComputing and Network (MOBICOM99)*, Seattle, WA: pp. 1-12.

Burhans, D., A. Campbel, and G. Skuse (2003). Exploring the role of knowledge representation and reasoning in biomedical text understanding. *Working Notes of the SIGIR Workshop on Text Analysis and Search for Bioinformatics*, Toronto, Canada.

Cao, H. and O. Wolfson (2005). Nonmaterialized motion information in transport networks. *10th International Conference on Database Theory (ICDT'05)*, Edinburgh, UK: pp. 173-188.

Chen, J., X. Meng, Y. Guo, S. Grumbach and H. Sun (2006). Modeling and Predicting Future Trajectories of Moving Objects in a Constrained Network. *Proceedings of the 7th International Conference on Mobile Data Management, MDM'06, IEEE Computer Society*: pp. 156.

Cheng, R., Y. Zhang, E. Bertino and S. Prabhakar (2005). Querying private data in moving-object environments. *Symposium of the Center for Education and Research in Information Assurance and Security (CERIAS) Tech Report*, Purdue University.

Cole, S. and K. Hornsby (2005). Modeling noteworthy events in a geospatial domain. *Proceedings of the First International Conference on Geospatial Semantics (GeoS 2005)*, Berlin. *Lecture Notes in Computer Science*, 3799, Springer Verlag: pp. 78-92.

Dillenburg, J., P. Nelson, O. Wolfson, O. Yu, A. Sistla, S. McNeil, A. Ouksel, B. Xu, and J. BenArie (2004). Applications of a transportation information architecture. *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC),* Taipei, Taiwan: pp. 480-485.

Ding, Z. and R. Guting (2004). Modeling temporally variable transporation networks. Database Systems for Advanced Applications, *Lecture Notes in Computer Scienc**e***, 2973, Springer: pp. 154-168.

Duckham, M. and L. Kulik (2005). A formal model of obfuscation and negotiation for location privacy. *Pervasive Computing, Third International Conference*, Munich, Germany. *Lecture Notes in Computer Science*, 3468, Springer: pp. 152-170.

DuMouza, C. and P. Rigaux (2005). Mobility patterns. *GeoInformatica* 9(4): 297-319.

Erwig, M. and M. Schneider (2002). Spatio-temporal predicates. *IEEE Transactions on Knowledge and Data Engineering* 14(4): 881-901.

Euzenat, J., D. Loup, M. Touzani, and P. Valtchev (2004). Ontology alignment with OLA. *Third EON Workshop, Third International Semantic Web Conference* 4(3): 333-337.

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

Forlizzi, L., R. Güting, E. Nardelli, and M. Schneider (2000). A data model and data structures for moving objects databases. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, TX. ACM: pp. 319-330.

Freska, C. and K. Zimmerman (1992). On the utilization of spatial structures for cognitively plausible and efficient reasoning. *Proceedings of the Conference on Systems, Man, and Cybernetics*, Chicago, IL: pp. 261-266.

Fridman, N. and M. Musen (2000). PROMPT: algorithm and tool for automated ontology merging and alignment. *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artifical Intelligence*. AAAI Press / The MIT Press: pp. 450-455.

Galton, A. and M. Worboys (2005). Processes and events in dynamic geo-networks. *Proceedings of the First International Conference on Geospatial Semantics (GeoS 2005)*, Mexico City, Mexico. *Lecture Notes in Computer Science,* 3799, Springer, Verlag: pp. 45-49.

Gkoutos, G., E. Green, S. Greenaway, A. Blake, A. Mallon, and J. Hancock (2004). CRAVE: a database, middleware and visualization system for phenotype ontologies. *Bioinformatics* 21(7).

Grenon, P. and B. Smith (2004). SNAP and SPAN: Towards a dynamic spatial ontology. *Spatial Cognition and Computation* 4: pp. 69-104.

Güting, R., V. D. Almeida, and Z. Ding (2006). Modeling and querying moving objects in networks. *VLDB Journal* 15(2): 165-190.

Güting, R., M. Bohlen, M. Erwig, C. Jensen, N. Lorentzos, M. Schneider, and M. Vazirgiannis (2000). A foundation for representing and querying moving objects. *ACM Transactions on Database System* 25: 1-42.

Güting, R. and M. Schneider (2005). *Moving Objects Databases*. Morgan Kaufmann Publishers.

Hage, C., C. Jensen, B. P. L. Speicys, and I. Timko (2003). Integrated data management for mobile services in the real world. *29th International Conference On Very Large Data Bases (VLDB) Conference*, Berlin, Germany: pp. 1019-1030.

Hollink, L., G. Schreiber, J. Wielemaker, and B. Wielinga (2003). Semantic annotation of image collections. In S. Handschuh, M. Koivunen, R. Dieng, and S. Staab, editors. *Proceedings of Knowledge Markup and Semantic Annotation Workshop (KCAP'03)*, Florida, USA: pp. 41-48.

Hornsby, K. and M. Egenhofer (2002). Modeling moving objects over multiple granularities. *Annals of Mathematics and Artificial Intelligence* 36: 177-194.

Hornsby, K. and K. King (2007). Linking geosensor network data and ontologies to support transportation modeling. In S. Nittel, A. Labrinidis, A. Stefanidis, editors. *Advances in Geosensor Networks. Lecture Notes in Computer Science*, 4530, Springer.

http://protege.stanford.edu/. "Protégé Ontology Editor and Knowledge Base Framework."

http://wordnet.princeton.edu/. "Wordnet." *Princeton University Cognitive Science Library*.

http://www.cia.gov/cia/publications/factbook/. "CIA Word Fact Book 2002."

http://www.dtic.mil/doctrine/jel/cjcsd/cjcsm/m3500_4b.pdf. "Universal Joint Task List."

http://www.gnu.org/copyleft/gpl.html "GNU General Public License."

http://www.ontologyportal.com. "Suggested Upper Merged Ontology (SUMO)."

http://www.statlab.iastate.edu/soils/nssh/629.htm "Glossary of Landform and Geologic Terms."

Kao, B. and H. Garcia-Molina (1995). An overview of real-time database systems. *Advances in Real-Time Systems*. Prentice-Hall, Inc.: pp. 463-486.

Klein, M. (2001). Combining and relating ontologies: an analysis of problems and solutions. In A. Gomez-Perez, M. Gruninger, H. Stuckenschmidt, and M. Uschold, editors. *Workshop on Ontologies and Information Sharing*, Seattle, WA: pp. 53-62.

Konstantinou, N., D.-E. Spanos, M. Chalas, E. Solidakis and N. Mitrou (2006). VisaVis: an approach to an intermediate layer between ontologies and relational database contents. In F. Frasincar, G.-J. Houben and P. Thiran., editors, *Proceedings of the International Workshop on Web Information Systems Modeling (WISM)*, *The 18th Conference on Advanced Information Systems Engineering*. Luxemburg. 239**:** pp. 36-47.

Kuhn, W. (2005). Geospatial semantics: why, of what, and how? *Journal on Data Semantics,* 3534: 1-24.

Laube, P. and S. Imfeld (2002). Analyzing relative motion within groups of trackable moving point objects. In M. Egenhofer and D. Mark, editors, *Proceedings of GIScience, Lecture Notes in Computer Science*, 2478, Springer Berlin: pp. 132-144.

Laube, P., S. Imfeld, and R. Weibel (2005). Discovering relative motion patterns in groups of moving point objects. *International Journal of Geographical Information Science* 19(6): 639-668.

Mark, D., A. Skupin, and B. Smith (2001). Features, objects, and other things: ontological distinctions in the geographic domain. *Proceedings of COSIT 2001, Lecture Notes in Computer Science*, 2205: pp. 489-502.

Meka, A. and A. Singh (2005). DIST: a distributed spatio-temporal index structure for sensor networks. *Proceedings of the 14th ACM International Conference on Information and Knowledge Management, CIKM 2005*. Bremen, Germany. ACM Press**:** pp 139-146.

Miller, H. and S. Shaw (2001). Geographic information systems for transportation: principles and applications. Oxford Press.

Moldovan, D., C. Clark and S. Harabagiu (2005). Temporal context representation and reasoning. In L. Kaelbling and A. Saffioti, editors, *Proceedings of the International Joint Conference on Artifical Intelligence (IJCAI)*, Edinburgh, Scotland, UK. Professional Book Center: pp. 1099-1104.

Nedas, K. (2006). Semantic similarity of spatial scenes. Ph.D. Dissertation, *Department of Spatial Information Science and Engineering,* University of Maine, http://www.library.umaine.edu/theses/pdf/NedasKA2006.pdf.

Nedas, K. and M. Egenhofer (2003). Spatial similarity queries with logical operators. In T. Hadzilacos, Y. Manolopoulos, J. Roddick, and Y. Theodoridis, editors, *Advances in Spatial and Temporal Databases*, *8th International Symposium (SSTD 2003)*, Santorini, Greece, *Lecture Notes in Computer Science*, 2750, Springer, Berlin: pp. 430-448.

Niles, L. and A. Pease (2001). Towards a standard upper ontology. In C. Welty and B. Smith, editors, *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, Ogunquit, Maine: pp. 2-9.

Pfoser, D. and C. Jensen (2003). Indexing of network constrained moving objects. *Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems (GIS)*, New Orleans, LA. ACM Press: pp. 25-32.

Pfoser, D. and C. Jensen (2005). Trajectory indexing using movement constraints. *GeoInformatica* 9(2): 93-115.

Pfoser, D., C. Jensen and Y. Theodoridis (2000). Novel approaches in query processing for moving object trajectories. *26th International Conference on Very Large Data Base (VLDB)*, Cario, Egypt. Morgan Kaufmann: pp. 395-406.

Pitoura, E. and G. Samaras (2001). Locating objects in mobile computing. *IEEE Transactions on Knowledge and Data Engineering* 13(4): 571-592.

Ramamritham, K., S. Son, and L. Dipippo (2004). Real-time databases and data services. *Real-Time Systems* 28(2-3): 179-215.

Rodriguez A., M. Egenhofer, and R. Rugg (1999). Semantic similarities among geospatial feature class definitions. *Interoperating Geographic Information Systems, Second International Conference (Interop '99)*, Zurich, Switzerland, In A. Vckovski, K. Brassel, and H.-J. Schek (eds.), *Lecture Notes in Computer Science*, 1580, Springer-Verlag, pp. 189-202.

Rodriguez-Tastets, M. (2005). Moving objects databases. In L. Rivero, J. Doorn, and V. Ferraggine, editors, *Encyclopedia of Database Technologies and Applications*. Idea Group: pp. 377-385.

Saltenis, S., C. Jenson, S. Leutenegger, and M. Lopez (2000). Indexing the positions of continuously moving objects. In W. Chen, J. Naughton, and P. Bernstein, editors, *Proceedings of the 19th ACM-SIGMOD International Conference on Management of Data*, Dallas, Texas: pp. 331-342.

Shotton, D. (2003). BioImage: from image database to grid service. *Proceedings of the 2nd e-BioSci / ORIEL Summer Conference*, Villa Monastero, Varenna, Italy.

Smith, B., J. Köhler, and A. Kumar (2004). On the application of formal principles to life science data: a case study in the gene ontology. In E. Rahm, editor, *Proceedings of Data Integration in the Life Sciences (DILS), First International Workshop*, Leipzig, Germany. Springer: pp. 79-94.

Smith, B. and D. Mark (2001). Geographical categories: an ontological investigation. *International Journal of Geographical Information Science* 15(7): 591-612.

Stefanidis, A., K. Eickhorst, P. Agouris, and P. Partsinevelos (2003). Modeling and comparing change using spatiotemporal helixes. *Proceedings of the Eleventh ACM International Symposium on Advances in Geographic Information Systems*, New Orleans, Louisiana. ACM Press: pp. 86-93.

Stefanidis, A. and S. Nittel (2004). *GeoSensor Networks*. CRC Press.

Straccia, U. and R. Troncy (2005). oMAP: combining classifiers for aligning automatically OWL ontologies. *Web Information Systems Engineering (WISE), 6th International Conference on Web Information Systems Engineering*, New York, NY. Springer: pp. 133-147.

VanDeWeghe, N., A. Cohn, G. DeTre, and P. DeMaeyer (2006). A qualitative trajectory calculus as a basis for representing moving objects in geographical information systems. *Control and Cybernetics* 35(1): 97-119.

VanDeWeghe, N., A. Cohn, P. D. Maeyer, and F. Witlox (2005). Representing moving objects in computer-based expert systems: the overtake event example. *Expert Systems with Applications*, 29(24): 977-983.

VanDeWeghe, N., B. Kuijpers, P. Bogaert and P. D. Maeyer (2005). A qualitative trajectory calculus and the composition of its relations. In M. Rodriguez, I. Cruz, M. Egenhofer, and S. Levashkin, editors, *Geospatial Semantics, First International Conference (GeoS 2005),* Mexico City, Mexico. *Lecture Notes in Computer Science*, 3799, Springer: pp. 60-76.

Vazirgiannis, M. and O. Wolfson (2001). A spatiotemporal model and language for moving objects on road networks. In C. Jenson, M. Schneider, B. Seeger, and V. Tsotras, editors, *Proceedings of the 7th international Symposium on Advances in Spatial and Temporal Databases*, Redondo Beach, CA. *Lecture Notes in Computer Science*, 2121, Springer: pp. 20-35.

Wolfson, O. (2002). Moving objects information management: the database challenge. *Proceedings of the Next Generation Information Technologies and Systems, (NGITS 2002), 5th International Workshop*, Caesarea, Israel. Springer: pp. 75-89.

Wolfson, O., S. Chamberlain, K. Kalpakis, and Y. Yesha (2001). Modeling moving objects for location based services. In B. Konig-Ries, K. Makki, S. Makki, N. Pissinou, and P. Scheuermann, editors, *Developing an Infrastructure for Mobile and Wireless Systems (NFS) Workshop, IMWS 2001*, Scottsdale, AZ. Springer: pp. 46-58.

Wolfson, O., L. Jiang, A. Sistla, S. Chamberlain, N. Rishe, and M. Deng (1999). Databases for tracking mobile units in real time. In C. Beeri and P. Buneman, editors, *Proceedings of the 7th International Conference on Database Theory (ICDT)*, Jerusalem, Israel. Springer: pp. 169-186.

Worboys, M. (2005). Event-oriented approaches to geographic phenomena. *International Journal of Geographical Information Science* 19(1): 1-28.

Worboys, M. and K. Hornsby (2004). From objects to events: GEM, the geospatial event model. In M. Egenhofer, C. Freska, and H. Miller, editors, *Proceedings of Geographic Information Science (GIScience), Third International Conference,* Adelphi, MD. *Lecture Notes in Computer Science,* 3234, Springer: pp. 327-344.

Xie, R. and R. Shibasaki (2005). A unified spatiotemporal schema for representing and querying moving features. *SIGMOD Record* 34(1): 45-50.

BIOGRAPHY OF THE AUTHOR

Kraig King was born in Caribou, Maine on January 22nd, 1977. He was raised in Woodland, Maine and graduated from Caribou High School in the spring of 1995. He attended the University of Maine and graduated in the fall of 2000 with a Bachelor of Science degree in Computer Science. He is a member of the computer science honor society Epsilon Pi Epsilon and the Francis Crowe Honor society for engineers. During the last year of his undergrad he started working for GE Energy as a Programmer, and currently holds a position as a Manufacturing Engineer. After a short time off from academia, he returned to the University of Maine and entered the Spatial Information Science and Engineering program in the fall of 2005.

Kraig is a candidate for the Master of Science degree in Spatial Information Science and Engineering from The University of Maine in December, 2007.