Observations on the Use of Ontologies for Autonomous Vehicle Navigation Planning‡

Ron Provine, Mike Uschold, Scott Smith
Boeing Phantom Works

P.O. Box 3707, m/s 7L-43
Seattle, WA 98124-2207, USA
ronald.c.provine@boeing.com

Stephen Balakirsky, Craig Schlenoff
Nat. Inst. of Standards and Technology

100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899, USA
{stephen, craig.schlenof}@nist.gov

Abstract

This paper reports some initial observations as we explore the hypothesis that ontologies can be used to improve the capabilities and performance of on-board route planning for autonomous vehicles. We are in the early stages of an effort to evaluate the performance of ontologies in different components of our chosen infrastructure: the 4D/RCS system architecture developed at NIST. Our initial focus has been on simple roadway driving scenarios where the controlled vehicle encounters potential obstacles in its path. As reported elsewhere [7], our approach is to develop an ontology of objects in the environment, in conjunction with rules for estimating the damage that would be incurred by collisions with the different objects in different situations. Automated reasoning is used to estimate collision damage; this information is fed to the route planner to help it decide whether to avoid the object. We describe the issues and insights developed during the first phase of the project and discuss the changes to our approach that have resulted.

1 Introduction

a) Statement of the Problem

An autonomous vehicle is an embodied intelligent system that can operate independently from human supervision. The field of autonomous vehicles is continuing to gain traction with both researchers and practitioners. Funding for research is this area has continued to grow over the past few years, and recent high profile funding opportunities have started to push theoretical research efforts into practical use.

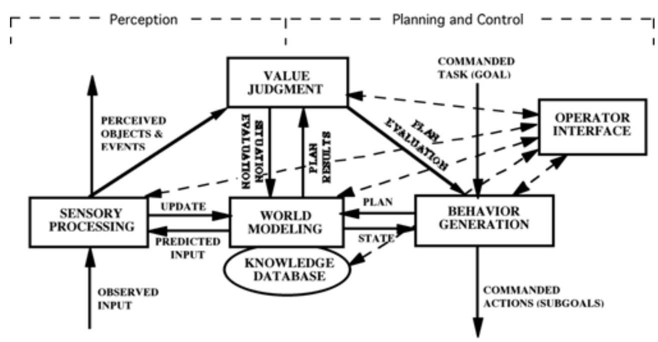

Figure 1: RCS Node

To behave appropriately in an uncertain environment, many researchers and practitioners believe that "the vehicle must have an internal representation (world model) of what it feels and experiences as it perceives entities, events, and situations in the world. It must have an internal model that captures the richness of what it knows and learns, and a mechanism for computing values and priorities that enables it to decide what it wishes to do." [3] p. 196. The inability to do this well hinders effective task planning and execution and thus the overall effectiveness of the vehicle. A major challenge in autonomous vehicles is the ability to accurately maintain this internal representation of pertinent information about the environment in which the vehicle operates.

b) The 4D/RCS Reference Model Architecture

For reasons discussed more fully in [7] we have selected the Real-Time Control System (4D/RCS) [1,2] as the architecture in which we will implement and evaluate the use of ontologies for autonomous vehicles. 4D/RCS is a hierarchical, distributed, real-time control system architecture that provides clear interfaces and roles for a variety of functional elements.

‡ Certain software tools are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the software tools identified are necessarily the best available for the purpose.

Under 4D/RCS, the functional elements of an intelligent system can be broadly considered to include: behavior generation (task decomposition and control), sensory processing (filtering, detection, recognition, grouping), world modeling (store and retrieve knowledge and predict future states), and value judgment (compute cost, benefit, importance, and uncertainty). These are supported by a knowledge database and a communication system that interconnects the functional elements and the knowledge database. This collection of modules and their interconnections make up a generic node in the 4D/RCS reference model architecture (see Figure 1) [3]. A generic node is defined as a part of the 4D/RCS system that processes sensory information, computes values, maintains a world model, generates predictions, formulates plans, and executes tasks. Each module in the node may have an operator interface.

c) Our Initial Focus

An ontology component promises to be helpful in many aspects of the 4D/RCS architecture. We decided to focus our initial efforts on the value judgment and behavior generation components, specifically in the area of assisting the planner in deciding upon the most cost-effective plan.

The value judgment component evaluates perceived and planned situations. It computes what is important (for attention), and what is rewarding or punishing (for learning). The value judgment component assigns priorities and computes the level of resources to be allocated to tasks. It assigns values and costs to recognized objects and events, and computes confidence factors for observed, estimated, and predicted attributes and states. [2] The outputs of the value judgment component are used by the behavior generation component to select and set priorities during route planning.

We are exploring the use of ontologies as a mechanism to allow the planner in the behavior generation component to better understand the costs and consequences of colliding with other objects. By representing the factors that could impact a path's cost, an ontology can be used to reason over the information that is available to determine what the consequences of a collision would be. Further reasoning could then be performed to determine the cost of these consequences. This cost would then be fed back to the planner for consideration when deciding the "cheapest" plan for the system to execute.

The remainder of this paper will introduce our initial obstacle ontology, describe the issues and insights developed during the first phase of the project, and discuss the changes to our approach that have resulted.

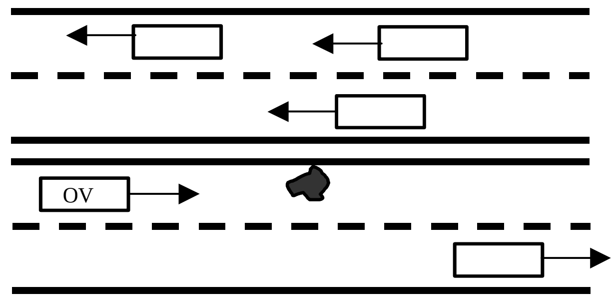

Figure 2: Simple Driving

2 Developing a Simple Ontology of Obstacles

In its full generality, the problem of automated vehicle navigation is extremely challenging. In developing the ontology, our current focus is navigation on a roadway. We start with the simple scenario illustrated in Figure 2. Our vehicle (labeled OV) is in the left lane of a four-lane two-way undivided highway. An object is detected in our lane. The goal is to formulate an optimal route plan that takes into account the potential damage from a collision with the object. The main role of the ontology component is (initially) to provide assessments of collision damage.

A number of parameters may be varied in this scenario. These include: the type of vehicle we are controlling, the speed at which we are traveling, the payload we are carrying, and type of object in our path that may be an obstacle. We wish to show how varying these parameters will change the plan that is ultimately executed. For example, if the object is a newspaper in the middle of the roadway, then:

- the ontology component will conclude that no damage will occur;
- the value judgment component should decide that colliding with it has no real cost;
- the behavior generation component will conclude that the best course of action is to maintain the current lane (because changing lanes always accumulates some additional risk over maintaining your lane).

However, if the object were a large cinder block, significant damage would be likely and the final route should be quite different.

a) Obstacles as Roles

An initial examination of the scenario led to the recognition that an obstacle is a role that an object plays in a certain situation. A person may not be an obstacle to a certain vehicle if s/he is sitting on a park bench, but if that same person walks into the middle of the road a short distance in front of that vehicle, they become an obstacle. A general theory of obstacles should define a set of conditions that must occur for a given object to take on the role of an obstacle to a certain vehicle. The determination of the extent to which an object is an obstacle (or not) depends on the *relationship* between the object and another entity (for us, an autonomous vehicle). If the relationship includes impeding the progress of the vehicle, or impeding the vehicle's ability to carry out its goals, then the object is an obstacle.

In Guarino and Welty's taxonomy of property types [6], being an obstacle is a role. This means two things. First, being an obstacle is an optional property for all its instances. This is an alternate way to express the fact that being an obstacle is not an inherent property of an object. Secondly, being a role means that the very existence of every instance of obstacle depends on the existence of some *other* instance (i.e. the thing being impeded). Something cannot be an obstacle unless there is something else that is being impeded in some way. Thus, our ontology must include the class **Object** and a relationship for **Obstacle** (some instance of Object) and **Vehicle** (the impeded object). We also introduce a generic notion called **Situation** for collecting information that is relevant for determining the extent to which an object is an obstacle to the vehicle. Associated with a situation are the vehicle, the object that is the potential obstacle, driving speed, road conditions, vehicle clearance, object height, weight and density etc.

b) Obstacles have a Cost

As mentioned previously, we are using the ontology in a component that assists the planner in determining the consequences of colliding with an object. Specifically, we want to provide inputs to the value judgment module for use in computing costs for plan segments. This led us immediately to the concept of obstacles as cost factors. In this formulation, a "major" obstacle (e.g. a large cinder block) is represented by a higher cost than a "minor" obstacle (e.g. a paper on the road). This **Cost** is a function of the relationship between the obstacle and the vehicle; hence, it is a property of the Situation.

This approach seemed both intuitive and easy to integrate into a value judgment calculation. However, examination of several simple scenarios led to the realization that constructing these costs would not be straightforward. Consider the following situations:

- You are driving by yourself in heavy traffic when you encounter a brick in your lane.
- You are carrying a piece of sensitive, fragile equipment when you encounter a brick in your lane.
- You are delivering urgent medical supplies when you encounter a brick in your lane.
- You are delivering urgent medical supplies when you encounter a child in your lane.

Discussion of these scenarios led us to recognize that a Situation must relate not just an Obstacle and a Vehicle, but also the vehicle's payload. In other words, in evaluating a potential collision we must take into account the amount of damage incurred by the payload, hence **Payload** information must be included in the Situation description.

c) Obstacle costs must be accumulated

Further consideration of the issue of payload damage led to the realization that damage is both relative and cumulative. For instance, a vehicle might have a wheel that is close to falling off, so that a collision that would be inconsequential for a new vehicle would be significant for this one. One can also envision scenarios (e.g. driving on a fresh gravel road) where repeated minor collisions eventually result in major damage. As a result, we added the concept of object integrity to ontology.

Integrity is a property of Object, and qualitatively describes the condition of that object with respect to the amount of damage it has accumulated. Currently we're using the set of values (None, Minor, Moderate, Severe and Catastrophic). We then introduced rules mapping from the current integrity value to the new value after a collision for each of the Situation participants – Vehicle, Obstacle and Payload. Thus, for the scenario where a vehicle collides with a melon, the vehicle and payload integrity would be unchanged, but the melon's integrity would change to "Catastrophic"; it would be destroyed.

d) The Situation must consider the Mission

These scenarios also led to the identification of the **Mission** as an important factor in determining cost. This is expressed in terms of the relative importance placed on maintaining or restricting the values for the integrity of the vehicle, the payload and the obstacle.

If the mission is to get from point A to point B as quickly as possible, regardless of the resulting condition of the vehicle or the payload, one can visualize the typical Hollywood car chase – nothing is an obstacle. In our formalization, for this mission all obstacles would have zero cost.

However, in a more typical mission, such as commuting to work, there is a desire to minimize damage to the vehicle and the payload (the driver and passengers). In this situation, most obstacles would have a significant cost.

We have not yet begun exploring approaches for representing the mission requirements in our ontology.

e) Not all Situations are Created Equal

In working through the qualitative physics that influence driving decisions (and determine the integrity transformations), we realized an initial evaluation is performed to determine which of three subclasses of “collisions” will occur:

- the vehicle can avoid the obstacle, swerving around it while remaining in its lane,
- the vehicle can pass over it, adjusting its path so that the obstacle passes cleanly underneath, or
- the vehicle will run into the obstacle.

Thus, a brick at the edge of the lane need not be considered as an obstacle (i.e., it has zero cost). Similarly, a long board laying across the road will have some non-zero cost, while the same board laying along the lane will have a zero (or nearly zero) cost, because it is easily straddled.

These considerations require that we represent the dimensions of obstacles in our ontology, as well as some of the basic specifications of the vehicle (e.g. wheel base and ground clearance) and the relative orientation and position of the vehicle and obstacle. It also requires that the ontology tool have the capability for performing mathematical operations and comparisons.

Pursing the development of the qualitative physics, we found that the closing velocity and the relative masses of the Situation participants, both real-valued quantities, were necessary to produce reasonable cost values.

At this point, we have encountered enough issues with our original choice of ontology tool that we have stopped development of the ontology while we reexamine our requirements. These issues will be discussed in the next section.

3 System Issues

The nature of the problem that we have selected places a number of requirements on the ontology and the tool in which it is implemented.

- Our implementation platform – an autonomous vehicle - clearly requires real-time performance by the nodes in order to exhibit acceptable behavior. Since the 4D/RCS architecture is hierarchical, the definition of real-time will change with level. At the level at which we are focusing planning decisions must be made within a few seconds.
- In the 4D/RCS architecture, the behavior generation component makes calls to the value judgment component for each node in the plan graph. As there may be thousands of nodes in a plan, any ontology tool operating within the value judgment module must support high transaction rates.
- As discussed above, there are concepts in the obstacle ontology that express relationships between measured properties, e.g. the closing velocity of the vehicle and object and the ratio of their masses. This necessitates the ability to represent and reason about (i.e. do computation with) real-, continuous-valued variables.
- The nature of the problem is such that the system performance must degrade smoothly. Conceptually this means that the ontology must include general (default) reasoning in addition to support for specific situations.

For our initial experiments, we have constructed a small ontology using OilEd [5]. Using a description logic [4] tool provided two advantages for us. First, the classifier detects logical errors in the ontology, which greatly increases confidence that the ontology is correct. Second, it is very fast at doing inference. This is important because the planner needs to query the ontology component up to a few thousand times a second to get damage estimates for the many nodes being explored in the search space.

We created a simple taxonomy of physical objects including various types of vehicles, and other objects such as bricks, newspapers etc. that may be in the vehicle’s environment. These objects have characteristics such as weight, speed, density, etc. that are important in determining the damage category. Initially we created some qualitative categories for measuring these characteristics, such as low, medium and high for weight, or density.

Finally, we created some axioms that specify how to classify a given situation in terms of the resulting damage categories.

From our initial tests, it is clear that there are limits to using a description logic reasoner for our task. For example, we cannot directly express rules that conclude that Integrity of the Vehicle and the Payload is unchanged, or that it should be incremented by one level in the damage severity scale. There are also limitations with respect to concrete domains that may be needed to reason with numbers. Similar problems were discovered in an attempt to use description logic classification to implement a semantic publish and subscribe system [8].

We are exploring alternatives for solving this problem. Various workarounds are possible. Also, it may be that description logic classification will only play a limited role in reasoning about obstacles. Techniques that are more general may be required, such as production rules, probability, or fuzzy logic.

4 Future Work and Conclusion

The overall goal of this work is to apply ontologies to enhance the capabilities and performance of autonomous vehicles, particularly in the area of navigation planning. In order to do this, we are initially using an ontology to determine the damage resulting from collisions between autonomous vehicles and different types of objects that could be encountered during on-road driving.

Our intent was to implement the scenarios presented in this paper in a simulated environment by December 2003. In this evaluation system, we expected to include multiple types of vehicles, objects, and speeds of collision. We also hoped to show how the plan that is generated based on different combinations of inputs, changed as a function of the expected damage due to collision. Because of the issues mentioned in the previous section, we were not able to meet this goal. As a result, we are in the process of reassessing the available ontology tools against our requirements.

As part of our reassessment, we are also attempting to address a question that we posed for ourselves in [7]:

- To what extent can a general theory of obstacles be adapted to a wide variety of autonomous vehicle applications? Can we have a single ontology for multiple types of vehicles and contexts? How much will they have to be tailored? This is analogous to the long-time question about standard upper ontologies (SUO), but within a limited domain. Can there be a SUO of obstacles?

Guarino [6] suggests the concept of defining different kinds of ontologies according to their level of generality. We have developed a suggested decomposition of the obstacle ontology into top-level, domain, task and application ontologies using this approach, as follows:

- Top-Level Ontologies
    - Physical Objects
- Domain Ontology
    - Vehicles
    - Obstacles
    - Payloads
    - Road Segments (future)
    - Rules of the road (future)
- Task Ontology
    - Classification
    - Mission
- Application Ontology
    - Vehicle Integrity Transformations
    - Obstacle Integrity Transformations
    - Payload Transformations
    - Situation Classification

From this decomposition, we suggest that the top-level, domain and task ontologies can be adapted to a wide variety of autonomous vehicle applications. Demonstration of this result will depend, however, on successfully resolving the issues that we have identified with our current implementation.

References

1. Albus, J., "Outline for a Theory of Intelligence," *IEEE Transactions on Systems Man and Cybernetics*, Vol. 21, 1991, pp. 473-509.

2. Albus, J. and et.al., "4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems," NISTIR 6910, National Institute of Standards and Technology, Gaithersburg, MD, 2002.

3. Albus, J. and Meystel, A., *Engineering of Mind,* John Wiley & Sons, Inc. 2001.

4. Baader, F., McGuinness, D., Nardi, D., and Patel-Schnedier, F., *Description Logic Handbook: Theory, Implementation and Application*, Cambridge University Press 2002.

5. Bechhofer, S., Horrocks, I., Goble, C., and Stevens, R., "OilEd: a reason-able ontology for the semantic web," *Proc. of the Joint German Austrian Conference on AI, number 2174 in Lecture Notes In Artificial Intelligence*, Springer-Verlag, 2001, pp. 396-408.

6. Guarino, N. and Welty, C., "A Formal Ontology of Properties," *LADSEB/CNR Technical Report 01/2000*, 2000.

7. Schlenoff, C., Balakirsky, S., Uschold, M., Provine, R., and Smith, S., "Using Ontologies to Aid in Navigation Plannig in Autonomous Vehicles," *to appear in the Special Issue on Ontologies and Distributed Systems in the Knowledge Engineering Review*, 2004.

8. Uschold, M., Clark, P., Dickey, F., Fung, C., Smith, S., Uczekaj S., Wilke, M., Bechhofer, S., and Horrocks, I., "A semantic infosphere," *Proc. of the 2003 International Semantic Web Conference (ISWC 2003)*, 2003.