ARL-TN-1056 ● APR 2021

Artificial Intelligence for Mobility and Maneuver (AIMM) World Model Progress Report: Modeling Patterns

by Robert St. Amant

NOTICES

Disclaimers

ARL-TN-1056 ● APR 2021

Artificial Intelligence for Mobility and Maneuver (AIMM) World Model Progress Report: Modeling Patterns

Robert St. Amant
Computational and Information Sciences Directorate,
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* April 2021	**2. REPORT TYPE** Technical Note	**3. DATES COVERED (From - To)** January–April 2021

4. TITLE AND SUBTITLE
Artificial Intelligence for Mobility and Maneuver (AIMM) World Model Progress Report: Modeling Patterns

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Robert St. Amant

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DEVCOM Army Research Laboratory
ATTN: FCDD-RLC-IS
Aberdeen Proving Ground, MD 21005

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TN-1056

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID ID: Robert St. Amant, 0000-0003-1417-9278

14. ABSTRACT

This report summarizes progress to date in ontology development toward the goal of having a semantic world model in the US Army Combat Capabilities Development Command Army Research Laboratory autonomy architecture. Our focus is on a simple robot, an autonomous system capable of sensing and navigating through its environment, on a route reconnaissance mission. We provide a number of typical examples of the type of knowledge and reasoning we expect of a world model, and we identify a small set of modeling constructs to handle temporality, spatial location and other properties, expectations, sets, and more-specific domain concepts such as paths and blockage. The current status is summarized and directions for the future are outlined.

15. SUBJECT TERMS
world modeling, ontology, Web Ontology Language, OWL, image schemata, route reconnaissance

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Robert St. Amant
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UU	18	**19b. TELEPHONE NUMBER (Include area code)**
Unclassified	Unclassified	Unclassified			(410) 306-0073

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

1. Introduction

This report summarizes progress to date in ontology development toward the goal of having a semantic world model in the US Army Combat Capabilities Development Command Army Research Laboratory autonomy architecture. This is the first such progress report, so it will include some introductory material. We have adopted OWL 2, the Web Ontology Language, as a representation language for the world model.

> The W3C Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language such that knowledge expressed in OWL can be exploited by computer programs, e.g., to verify the consistency of that knowledge or to make implicit knowledge explicit. OWL documents, known as ontologies, can be published in the World Wide Web and may refer to or be referred from other OWL ontologies.[*]

OWL is targeted at the Semantic Web, which may seem conceptually distant from the physical environment in which a robot must operate. OWL is nevertheless a reasonable candidate for technical and strategic reasons.

- OWL has been used as the knowledge representation language in other robotic systems for tasks ranging from factory manipulation to household tasks, including navigation. In these systems, there is relatively little commitment to the specifics of the physical platform. An ontology is intended to capture a robot's action capabilities (e.g., requirements for a "pick" operation to be executable[1]) to describe a specific application domain,[2] or in general to store the range of knowledge useful to a robot, as in the KnowRob system.[3] We expect that an OWL ontology will be technically feasible in our domain as well.

- OWL is also one of the languages used as the basis for knowledge representation standardization. The Institute of Electrical and Electronics Engineers standard ontology for robotics and automation[4] relies on OWL. The US Army Operational Environment (OE) ontologies[†] (designed to capture knowledge about specific domains related to the OE to facilitate automated reasoning) are in OWL. In adopting OWL, we are hoping to build

---

[*] https://www.w3.org/OWL/

[†] https://github.com/TRADOC-G2/oe-ontology

on past work and to facilitate the use of our work by others—OWL supports easy adoption, interoperability, and incremental extensibility.

- With the growth of the World Wide Web, we have gained access to an enormous wealth of knowledge. The Semantic Web attempts to put this information in a form interpretable and usable by machines. For robots, the motivation is straightforward: If a robot is to acquire some information, and the robot can either learn the information or be told the information, we will prefer whichever means is more efficient.

The next section fleshes out some of the relevant background about the domain of application. Section 3 then lays out a set of domain examples that can be represented in the world model at present. Finally, Section 4 moves to the modeling techniques and primitive constructs in use. The ontology we have built is not yet integrated into the DEVCOM Army Research Laboratory autonomy stack; some technical issues remain to be explored. The conclusion of this report summarizes our status and outlines our plans for making further progress.

2. Context

In this section we identify basic assumptions about our effort. Our focus is on a simple robot, an autonomous system capable of sensing and navigating through its environment. For the present, we presume the robot has no manipulation capability. The high-level purpose of the robot's movement is maneuver, achieving a position of tactical advantage. Within the scope of maneuver is reconnaissance, gaining information about an area or potential enemies in an area by observation. The robot maintains a map of the environment and its place on the map. It can perceive objects and features of the environment, associating them with symbolic labels. A human commander or operator specifies tasks, at varying levels of detail, for the robot to carry out.

The following extended quotation is taken from *Army Field Manual 7-92*[5] (*Army Techniques Publication No. ATP 3-20.98 Scout Platoon, and Army Techniques Publication No. ATP 3-34.81 Engineer Reconnaissance* are alternative sources that provide related descriptions of route reconnaissance; in future work, those perspectives will be incorporated into our analysis.):

> Route reconnaissance focuses on obtaining information on a specified route and all terrain from which the enemy could influence movement along that route. Route reconnaissance can be oriented on a road, a narrow axis (such as an infiltration lane), or a general direction of attack.

1) The battalion commander orders a route reconnaissance when he needs information on routes to his objective or to alternate or supplementary defense positions. Usually, an overlay is given to the platoon leader along with specific information requirements needed for that specific route. Possible information requirements are as follows:

a. The available space in which a force can maneuver without being forced to bunch up due to obstacles (reported in meters). The size of trees and the density of forests are reported due to the effect on vehicle movement.

b. The location and types of all obstacles and the location of any available bypass. Obstacles can consist of minefields, barriers, steep ravines, marshy areas, or NBC contamination.

c. The enemy forces that can influence movement along the route.

d. The observation and fields of fire along the route and adjacent terrain. This information will assist planners as a supplement to map information.

e. The locations along the route that provide good cover and concealment.

2) When conducting a route reconnaissance, the platoon leader organizes the platoon based on METT-T [Mission, Enemy, Terrain, Troops & Time Available]. Depending on the time available, he conducts a thorough map reconnaissance and plans a series of fans (Figure 4-9, page 4-16) along the route that provides detailed terrain information. Roads and trails intersecting or traversing the route must be reconnoitered to where they cross terrain. The enemy could influence friendly movement from adjacent terrain.

3) If all or part of the proposed route is a road, the platoon considers the road a danger area. It moves parallel to the road using a covered and concealed route. When required, reconnaissance and security teams move close to the road to reconnoiter key areas.

The robot's general external capabilities can be reduced to three categories, though with a rich potential set of actions within each category. (Because our approach to world modeling produces descriptions in terms of symbols and logic, there is a bias toward representation of discrete objects in the environment [e.g., "a built structure" rather than "grass"]. We are also constrained by what can be recognized by the robot. All this affects the examples we have developed.)

Let S be a geospatial location or region; P a physical object; E an entity that could be either. Assume some internal spatial reasoning capacity.

- **Go**: go to *E*; go toward *E*; go along *E* (e.g., a wall or a road); go away from *E*; go around *E* (e.g. an obstacle); go past/by *E*; go through *S* (e.g., a doorway or a clearing); go into/out of *S* (e.g., a building or a region); go between *E*1 and *E*2; go in some direction; . . .
- **Look**: look at *E*; look for *E*; look in some direction; look around in *S* (i.e., a visual search); look through *E* (e.g., a window or opening); . . .
- **Message** sending or receiving, to accept a command or to report on internal state. We are abstracting away the challenges of dialog and natural language understanding here, among other issues. As an example, imagine that the robot is asked to identify some object in its field of view; the response would be some transformation of its internal representation into human-understandable form.

3. Domain Examples

The following are typical examples of what we expect a world model to support reasoning about. The list is very far from comprehensive, even for our limited robot.

Objects, types, and relationships: Given a labeled object, the robot can trace its type upward through an IS-A ("is a") hierarchy. An awning is a canopy is a protective covering is a covering, and so on. This means that in communication with a human commander, if the command references any one of these types to refer to the object, the object is known to be included as a candidate. Some types and objects are also known to be related to each other semantically. These relationships do not directly contribute to the "going" or "seeing" capabilities, at least not currently, but can provide robustness in message exchanges.

Temporality: Entities (objects, situations, etc.) exist over time but may change. For example, at different points in time the robot may be in different locations. Objects and environment features may have different states (e.g., a door in a public building may be open only during specific hours; a region may be free from surveillance for limited time). Some of these changes can be identified with specific events.

Spatial location: A physical object has a geospatial location. The location is represented as an entity in its own right. When the robot, for example, is at a specific location, a semantic relationship exists between the two entities. Some properties of a location propagate to entities in that location. For example, if a location is contained in a building (i.e., it is an inside location), then everything at that location is also contained by the building. This is an approximation that has other dependencies (e.g., what is meant by being "at" a location), but it works in the simple examples we have.

Collocation: An extended locality relationship exists between the robot and any other entities in the same location: The robot is said to be "at" those entities. For example, if a door is in a geographical location and the robot is in the same location, it can be inferred that "The robot is at the door".

Specialized object properties: Some physical objects and features have properties that affect interaction with them; these properties may change or be changed. For example, a door may be open or closed, which affects the ability of the robot to pass through the doorway.

Path following and blockage: The robot is given a path to follow as a sequence of waypoints (unique geospatial locations). The robot's starting location is its current state; the end location its "goal" state. The robot represents its progress along the path as a temporally ordered set of states. In each state, the robot is either at a waypoint or on a trajectory between two waypoints.

It is possible for a path to be blocked by an obstacle. Blockage can be of the path in general or of one of its waypoints or trajectories. If part of a path is blocked, the entire path is blocked.

Set formation: It may be useful to reason about sets of objects in a given state. For example, suppose the robot perceives some number of objects, and some are recognized and others are not. The robot can deal with those recognized objects for which it can identify relevant associated semantic properties. The robot can go further, though. After all, it "knows" *something* about every object, whether it is recognized or not. Elements of the unrecognized set might be treated differently rather than be ignored. The robot might draw attention for communication or in its own processing to the unrecognized objects (e.g., asking, "What is that?", or moving around an unknown object to gain a different viewpoint.)

Expectations: Information that the robot is given may be contingent. "Go to the traffic cone behind the building," for example, has the implicit assumption that such an object exists but this may not be immediately verifiable. The robot can record an assertion and label it as an expectation, recording it in a future state yet to be visited and the expectation verified.

4. Modeling Constructs

OWL is a language for representing knowledge, but not everything we know can be expressed in OWL. Within the constraints of expressiveness, however, we have a great deal of freedom. What should be represented?

At the most abstract level, independent of whatever domain is under consideration, a knowledge representation commits to some set of entities and the relationships between them—the kinds of things that exist. In the field of applied ontology, this means committing to an upper-level ontology such as the Basic Formal Ontology,[6] which includes such high-level entities as occurrents and continuants, processes and temporal regions, and so forth. An upper-level ontology is augmented by a mid-level ontology and then by an ontology that captures knowledge about a specific domain. In the rest of this section, we look at the domain level.

4.1 Schemata

We adopt concepts from the field of cognitive linguistics, specifically the concept of an *image schema*. Developed by Lakoff[7] and Johnson[8], image schemata are structured, general patterns intended to capture experience "at the level of our bodily movements through space, our manipulations of objects, and our perceptual interactions". Our representation focuses mainly on two schemata appropriate for our domain.

Containers have the characteristic operations of containing something, being entered, or being exited. Structurally, a container has

- a boundary,
- an interior and exterior, and
- openings, possibly, which may be passable.

An example of a container is a building with a door acting as an opening. Important specialized types of containers are the following:

- Locations. One can say that something is in a given location, where the "in" of location is a close relative to the "in" of containment. In general, there may be an unbounded number of ways of ingress or egress; containment is ephemeral.
- 2-D regions. A robot might be in a clearing, for example.
- 3-D volumes. A robot might be in a building, for example, or in an area that is visible from some remote vantage point. Our focus is on ground robots, which limits the use of volumes, but aerial robots will require a richer representation.

Paths have the characteristic operation of being traversed, generally by change of location. Structurally, a path has

- a start and an end, which are locations;
- internal waypoints, which are also locations, disjoint with start and end; and
- trajectories, interpolated between pairs of locations (start, end, and internal waypoints).

An example of a path is a route that a robot is directed to follow. Paths are implicitly directional, from start to end. Paths have the following important related schemata:

- Line of sight, which can be conceptualized as a path with only a start and end and a straight-line trajectory. A line-of-sight path may be useful for an agent seeking to be seen or to avoid being seen. This is more specific than regions to be in or out of, in that it is directional.
- Blockage, which may be of a waypoint or a trajectory or the path itself. The blockage may be an object or part of the terrain itself. There is a relationship between a blockage and a barrier, the details of which have not been worked out (they may be identical). If a door is closed, it is not a passable opening, meaning that in its closed state it functions as a barrier.

The phrase "characteristic operation", used earlier, is used in connection with the path and container schemata.[9] We can characterize these schemata and some others by what they are for. For example, a path is for traversing, a container for containing or moving in or out of. The examples in the previous section are straightforward to conceptualize in terms of these schemata and their characteristic operations.

What is special about these schemata? Lakoff[7] and Johnson[8] argue that such schematic representations are foundational to human understanding of literal physical situations and of metaphorical analogies. I might tell you, for example, of the path I took to reach my current job, starting from when I was in college. Or, adopting a different metaphor, I could say that in my previous job I felt that I was kept in a box, and finding a new job was the only way I could find to break out. Metaphorical applications of path-following and containment are ubiquitous, and (as a side research topic) we are exploring ways to map these schematic structures onto a wider range of experiences.

4.2 Representational Nuts and Bolts

For conciseness, this section assumes that readers are familiar with OWL, including limitations on expressiveness (e.g., the open-world assumption, monotonicity). OWL classes are capitalized Roman; individuals are italicized class names; and properties that relate individuals are in single quotes. Subscripts are used for distinguishing individuals of the same class if/when needed. The Artificial Intelligence for Mobility and Maneuver (AIMM) ontology imports the Army OE ontology, including its representation of objects, temporal intervals, and so forth. The AIMM ontology adds a *Robot* class (a subclass of the OE Agent) and classes for the forgoing schemata as well as classes for the following representation.

Temporality: We adopt a state-based representation (also used in bioinformatics ontologies to capture qualitative change over time[10]). Individuals of the *State* class are associated with the *Robot*. A *State* 'containsInState' entities that are relevant, typically objects in the environment. Change in the environment, including the *Robot*'s actions and results, is represented as a set of temporally ordered states. As an example, consider the *Robot* moving between locations. A *State* is created for each location or trajectory that the *Robot* will visit or be on. A *State* 'happensDuring' a *TemporalRegion*; pairs of *TemporalRegion*s have the property 'interval is before'. The *Robot* 'hasState' for every *State* that it will visit or has visited or is in presently.

Spatial location and other properties: Because OWL is monotonic, the *Robot*'s location (a *GeospatialLocation*) cannot be related to it directly as a property because that relationship may change. What is asserted in an OWL ontology is true for all time, so that if we asserted that the *Robot* is in a given location, we could not say it was anywhere else without introducing inconsistency into the ontology. Instead, the *Robot*'s movement over time is characterized as visiting different *State*s, each of which includes a different location per *State*. This convention is followed for all properties that are nonstatic, nonpermanent.

Because we will sometimes want to associate information with properties, we adopt another level of indirection beyond the state representation (again following a suggestion in the bioinformatics literature[10]). *State*s do not directly reference the properties of objects; an additional indirection is via a *Quality*. A *Quality* can be thought of as a reified property. Instead of saying that the *Robot* is 'located in' some *GeospatialLocation*, we say that the *State* of the *Robot* 'hasLocationQuality' some *Quality*, and that *Quality* in turn 'pointsTo' a specific *GeospatialLocation*.

We associate information with properties to manage (in a limited way) the status of other objects in the environment with respect to the *Robot*. For example, suppose the *Robot* is in a *State* such that it 'hasLocationQuality' some $\mathit{Quality}_R$ that 'pointsTo' a $\mathit{GeospatialLocation}_R$. Another object in this example is relevant to the *Robot*, a *TrafficCone*; its relevance means it is included in the same *State*. The *TrafficCone*'s location may change. Thus it 'hasLocationQuality' some $\mathit{Quality}_T$ that 'pointsTo' a $\mathit{GeospatialLocation}_R$. This is the same location as the *Robot*. The *State* now has 'hasLocationQuality' both $\mathit{Quality}_R$ and $\mathit{Quality}_T$. They are distinguishable, however, in that *Quality*$_T$ has a 'QualityOfEntity' property relationship with the *TrafficCone*.

Finally, the example used the "insideness" property of a location to propagate to the object in that location. This is managed by a chain property: If an entity 'hasLocationQuality' some *Quality* that 'pointsTo' a *GeospatialLocation* that 'isContainedBy' some containing entity, then whatever is in that *GeospatialLocation* also 'isContainedBy' the entity.

$$hasLocationQuality \circ pointsTo \circ isContainedBy :: SubPropertyOf\ isContainedBy$$

For example, if a tool is in a toolbox and the toolbox is in the shed, then it is inferrable that the tool is in the shed. If a person is inside a room, then the person is inside the building that contains the room.

Expectations: The example given was of a traffic cone behind a building to be treated as an expectation. There are different ways in which this can be cast in terms of expectation: Is it an expectation that the object itself exists? That it is of type TrafficCone? That it is in the specified location or region? All of these are plausible. The simplest solution (with generality remaining to be evaluated) is to create a *TrafficCone* (even if it may not exist in reality). It has an appropriate *Quality* that 'pointsTo' a *GeospatialLocation* behind the building. We then specify that the *Quality* is also of type Expectation.

Collocation: This is another simple chained property relating an entity at a location to the agent at the same location (e.g., the *Robot* 'isAtEntity' *Door*, or "The robot is at the door").

$$hasLocationQuality \circ isQualityOfEntity :: SubPropertyOf\ isAtEntity$$

Path following: Each part of the *Path* (start, end, waypoints, and trajectories) is inferred to be 'part of' the *Path*, as an inverse operation to 'has part'. The inverse property 'interval is after' of 'interval is before' is computed between temporal entities. Transitive closure of both properties is also computed.

Blockage: An object has the same *GeospatialLocation* as one of the waypoints of the *Path*. That location 'isBlockedBy' the obstacle. (It would be possible as an alternative to represent the obstacle as blocking the waypoint in a specific state—the direct representation means that the blockage cannot be retracted.) Because the waypoint is 'part of' the *Path*, a chain property

$$has\ part \circ isBlockedBy :: SubPropertyOf\ isBlockedBy$$

propagates the blockage to the *Path* as a whole.

Set formation: The example dealt with recognized versus unrecognized objects. In OWL, a suggestive construct is the object aggregation, with members being 'part of' the aggregation. This is useful in some situations, but the monotonic nature of OWL means that using aggregates directly would not meet our needs. For recognized and unrecognized objects, we would tie aggregates to states, because unrecognized objects might later be recognized or identified, rendering the original set partition incorrect.

Instead, we can use Semantic Query-enhanced Web Rule Language (SQWRL) queries, with built-ins that support set operations, even to the extent of simulating the closed-world assumption. SQWRL queries are analogous to database queries, with criteria that reflect OWL statements. This is a query that collects *RecognizedEntity* s in a given *State*.

states:containsInState(?s, ?x) ^ states:RecognizedEntity(?x) ->
sqwrl:select(?x)

Unrecognized entities are the complement of this set, though it is more complex to express in the absence of the closed-world assumption. Form a set ?s1 of objects in the *State*; form another set ?s2 of *RecognizedEntity*s in the *State*; create a third set ?s3 of the set difference of ?s1 and ?s2; collect elements of ?s3.

states:containsInState(?s0, ?x) ^ states:containsInState(?s0, ?y) ^

obo:BFO 0000030(?x) ∘ sqwrl:makeSet(?s1, ?x) ^ states:RecognizedEntity(?y) ^
sqwrl:makeSet(?s2, ?y) ∘ sqwrl:difference(?s3, ?s1, ?s2) ^

sqwrl:element(?e, ?s3) -> sqwrl:select(?e)

5. Conclusion

All of the modeling constructs discussed in Section 4 have been implemented and tested in prototype form. That is, OWL ontologies exist that define the structure of the image schemata and related classes such as states. A set of unit tests has been developed to exercise the ontologies over a range of scenarios (e.g., the generation of paths of varying length; each path waypoint and trajectory associated with its own state; each state in turn associated with qualities). A number of issues remain, however. The ontologies need to be made accessible through a programming interface, and the OWL application programming interface is the approach we are currently pursuing. The ontologies need to be aligned with perception and language modules in the ARL autonomy stack, an ongoing process. Finally, the ontologies need to be tested in practice in a real environment.

6. References

1. Crosby M, Petrick R, Rovida F, Krueger V. Integrating mission and task planning in an industrial robotics framework. Proceedings of the International Conference on Automated Planning and Scheduling. 2017;27.

2. Proctor F, Balakirsky S, Kootbally Z, Kramer T, Schlenoff C, Shackleford W. The canonical robot command language (CRCL). Ind Rob. 2016;43(5).

3. Tenorth M, Beetz M. Representations for robot knowledge in the KnowRob framework. Artif Intel. 2017;247:151–169.

4. Schlenoff C, Prestes E, Madhavan R, Goncalves P, Li H, Balakirsky S, Kramer T, Miguelanez E. An IEEE standard ontology for robotics and automation. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems; 2012. p. 1337–1342.

5. Headquarters, Department of the Army. Army field manual 7-92. The infantry reconnaissance platoon and squad (airborne, air assault, light infantry). 2001 Dec 13. p. 4-14–4-16.

6. Arp R, Smith B, Spear AD. Building ontologies with basic formal ontology. MIT Press; 2015.

7. Lakoff G. Women, fire, and dangerous things. University of Chicago Press; 1987.

8. Johnson M. The body in the mind: the bodily basis of meaning, imagination, and reason. University of Chicago Press; 1987.

9. St. Amant R, Fields M, Kaukeinen B, Robison C. Lightweight schematic explanations of robot navigation. Proceedings of the 2019 International Conference on Cognitive Modeling (ICCM). https://iccm-conference.neocities.org/2019/proceedings/papers/ICCM2019_paper _14.pdf.

10. Burek P, Scherf N, Herre H. Ontology patterns for the representation of quality changes of cells in time. J Biomed Semantics. 2019;10(1):1–18.

List of Symbols, Abbreviations, and Acronyms

2-D	two-dimensional
3-D	three-dimensional
AIMM	Artificial Intelligence for Mobility and Maneuver
ARL	Army Research Laboratory
DEVCOM	US Army Combat Capabilities Development Command
IS-A	“is a”
OE	Operational Environment
OWL	Web Ontology Language
SQWRL	Semantic Query-enhanced Web Rule Language

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLD DCI TECH LIB
1 (PDF)	DEVCOM ARL FCDD RLS IS R ST AMANT