Proceedings of the International Conference on Integration of Knowledge Intensive Multi-Agent Systems
KIMAS '05: Modeling, EVOLUTION and Engineering, April 18 - 21, 2005, Waltham, MA.

Entity Knowledge Model Supporting Intelligent Systems

Hui-Min Huang, James Albus, Elena Messina, Harry Scott, Maris Juberts

National Institute of Standards and Technology

Gaithersburg, MD 20899

Phone: 301 975 {3427, 3418, 3510, 3437, 3424}

Abstract—*Intelligent systems operate in uncertain and complex environments. In order to achieve their goals, these systems require rich and updated knowledge about the environment and about their own capabilities to enable proper decision making processes. A critical subset of the required knowledge is entity, which models the physical environment. This paper provides a generic and systematic model for the entities.*

1. INTRODUCTION

Within the context of this research effort, we define intelligent systems as real-time control systems that are able to operate in uncertain and complex environments and accomplish their goals. An intelligent, unmanned, military vehicle system may receive a mission such as to survey an unknown and potentially hostile area. The system may require, among other types of knowledge, maps overlaid with objects and geological features, updated in real-time. This knowledge base is critical to the system's decision making and execution capability and needs to be supported by capable sensory subsystems. The NIST 4D/RCS Reference Architecture [1], on which this research is based, provides a detailed description of this intelligent process.

The knowledge base of an intelligent system consists of data structures and the static and dynamic information that collectively form a model of the world. The knowledge includes the system's best estimate of the current state of the world plus parameters that define how the world state can be expected to evolve in the future under a variety of circumstances. Specifically, an intelligent system should possess the following types of knowledge:

- Spatial perception, in the forms of entities, images, and maps.
- Temporal perception, in the forms of time, states, and events.
- Logical knowledge: rules of physics, mathematics, and logic; knowledge of how to perform particular tasks and how to derive the task values with respect to mission requirements; and specifications of sensors and actuators.

Among this wide spectrum of issues, we focus on a critical subset of the knowledge in this paper, entity modeling.

Related work includes a representational framework for geographic modeling described by Smyth [2]. The framework's ontological components include entities, space, time, physics, and logic. Such constructs are consistent with our knowledge base, at a conceptual level. Our objective is to extend beyond these concepts for an implementation model. Megalou and Hadzilacos [3] pointed out the following two different abstractions in modeling information: conceptual and presentational. The former concerns issues like entities and relationships and the latter concerns issues like logical structure and temporal synchronization. While these concepts sound consistent with our general approach, we argue that the entity concept should be extended to multiple levels of abstraction to be useful for complex environment modeling.

ISO and Open GIS Consortium maintain a number of standard specifications in this area. The OpenGIS® Geography Markup Language (GML) Implementation Specification [4] involves geometrical modeling. Commonality among various geometry types is extracted and represented within GML with an objective of efficient models for general geometrical features. The model that we proposed in this paper could potentially provide an extra dimension to GML on how these geometrical types can be further related among themselves from a sensory data acquisition and assimilation perspective.

There are research results that complement our focus on knowledge organization. Ng and Han [5] extended the cluster analysis method to search for patterns in large data sets. Tomic, et al. [6] described a new approach to measure temporal consistency for derived objects in real-time database systems.

The issue of the intelligent system knowledge base has also been studied and implemented [7, 8, 9]. However, our focus is on establishing the overall framework for the knowledge base, as such a framework could facilitate rigorous development efforts for large-scale intelligent systems. We leave many detailed, specific data elements for some of the identified entities as future study issues.

2. REQUIREMENTS

Intelligent systems require rich and updated knowledge about the environment and about their own capabilities to advance along their courses of actions. Depending on the assigned tasks, the knowledge that is required varies. For a complex task such as conducting a battle, knowledge items such as maps, system capability, terrain features and their distribution, etc., may all be required. The system may also need to differentiate friend and foe. For the near-distance object avoidance task, spatial occupancy information alone may be sufficient and object classification may not be needed, nor may the system have the time to process the "What is the object?" question. The difference between these two situations highlights the existence of decision making at multiple levels of abstraction.

A common requirement for the knowledge base for intelligent systems is that the systems must know the environment that the system operates in and must be able to detect and respond, in time, to objects that are either already or predicted to affect the systems missions.

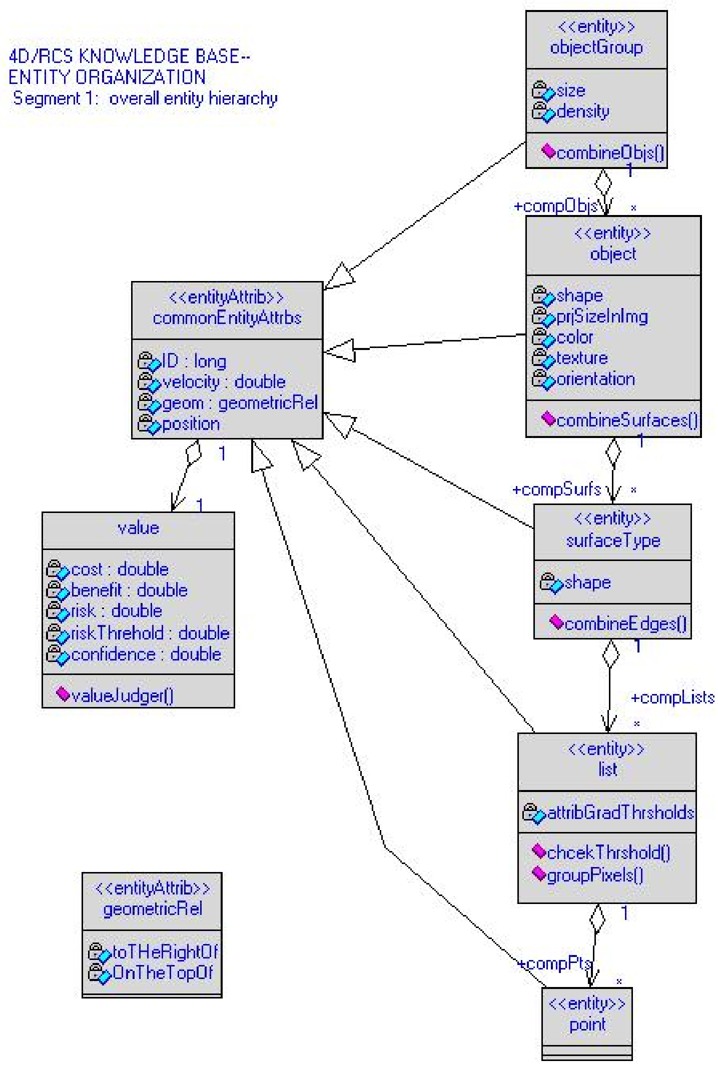

Figure 1 - the 4D/RCS entity knowledge hierarchy[1]

[1] The figures in this paper are developed with Rational Rose RealTime® software.

Due to the diverse nature of the knowledge requirements, it is, therefore, desirable to build the intelligent system knowledge bases with a structure that is scalable, efficient, and understandable. We propose that the knowledge base for an intelligent system should meet the following criteria:

- Support aggregating simple knowledge items to complex ones.
- Support multiple levels of abstraction of the knowledge, as this would facilitate multiple levels of the decision making processes within the system.
- Support classifying similar knowledge items.
- Support sharing common components and expand them to form specific knowledge items.

3. HIERARCHICAL ENTITY MODELING

We define an entity to be system internal perception of the spatial phenomena of the external world. These two, internal and external aspects, are distinctly different. Within the constraints of particular sensory data sets, a house may be perceived as a point and a hill may be perceived as a set of edges when they are located far enough. A small rock to human eyes may be perceived as endless wall to a micro-scale system.

The intelligent systems' entity knowledge structure is constructed from the viewpoint of acquisition and assimilation of sensory information. The complexity of the knowledge requirements warrants a hierarchical structure. Entity classes reside at each level and represent a particular level of abstraction. At the lowest level of the hierarchy are points, or pixels entity classes. This is because they correspond directly to output from various sensors that intelligent systems would employ.

The hierarchical model expands from points to lists, surfaces, objects, and to object groups, as seen in Figure 1, to support multiple levels of abstraction as required in intelligent systems. This is consistent with spatial geometry, expanding from the fundamental units to compound features. Figure 1 represents an overall hierarchy. Details will be described in the later diagrams.

We use UML, an industrial standard for software engineering modeling to describe the entity model. We define the following notations for use in all of the diagrams in the paper:

- A box represents a data structure and the necessary, associated manipulation methods, typically a C++ class or an enumerated data type.
- An <<entityAttrib>> stereotype models the common attributes for the entities.

- An <<entity>> stereotype models the entities.
- A line with a solid-white arrowhead represents a generalization relationship. The arrowhead points to the super class.
- A line with a diamond and an arrow on the ends represents an aggression relationship. The line is used to relate two classes. The class at the diamond end represents the aggregate. The numbers attached to the lines indicate the multiplicity of the involved classes. Note that, the aggregate relationship means, in C++ code, that the aggregate class contains only pointers to the component class and that it does not allocate that actual memory for the data structure of the component class. Note that, this is contrary to another "composition" relationship in UML which means that the super class does contain the actual data structure of the component class.

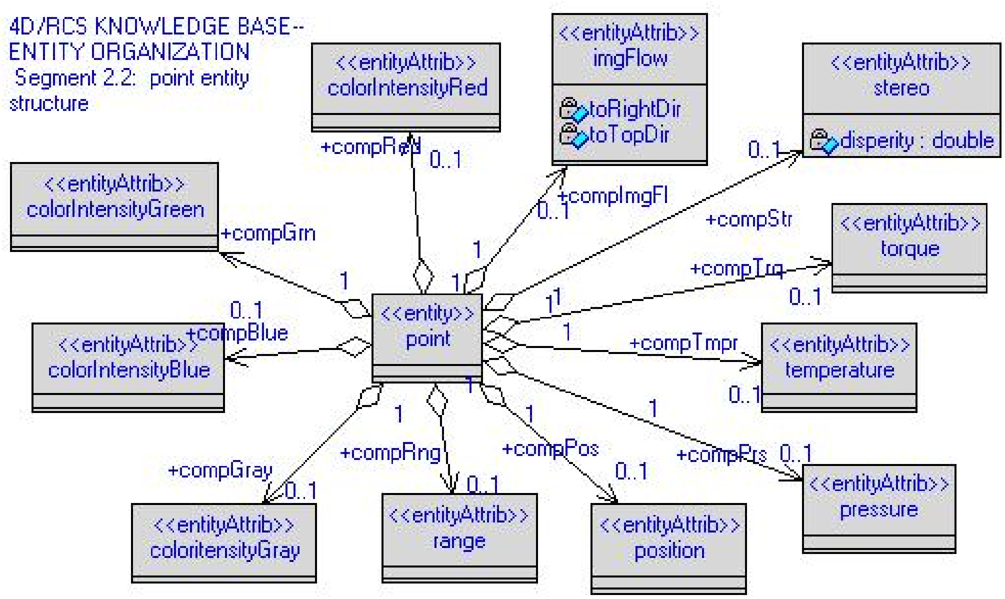

Figure 2 - point entity structure

3.1 Common Attributes

The top, left class in Figure 1 is called commonEntityAttribs, which contains the attributes that are to be a part of all the entities in the knowledge base. The attributes include:

- ID: for identifying an entity. Users can determine ranges and scopes for applying the ID. This attribute may, conceivably, be expanded to also allow the identification of particular instances of an entity class.
- Position and velocity of entity with respect to its predefined coordinate frame. The position and velocity of a multiple dimensional entity is with respect to its centroid.
- Geometrical relationship: Since entities model spatial properties, the relative spatial relationships among different entities are essential. We define a primitive class, geometricRel, for this purpose.

Another set of the common entity attributes is value, shown as the class at the middle, left of Figure 1. Each entity may be of particular values per particular system missions. Particular woods off the traveling course of an unmanned vehicle may be of little value until the vehicle is under fire and needs to traverse along the treeline. The attributes for value include:

- Cost and benefit: required resources, in terms of types and amounts, and the resulting rewards for handling the entity.
- Risks: how much danger there may be for the system to pursue this entity and how much risk is allowed per mission requirements. It may worth the risk for a military autonomous vehicle to be destroyed in order to retrieve a critical piece of information. It may not worth the risk for some marginally relevant information.
- Confidence or reliability level associated with the entity as a result of the system perception process.

Note that, although these attributes are specified as simple variables, in some situations, they might need to be modeled as functions or algorithms keyed on critical mission variables.

Detailed descriptions of the entity hierarchy follow.

3.2 Point Entity Classes

Point (or pixel) entity classes have attributes that can be measured by a single sensor at a single point in time and space, or that can be computed at a single point (or over a single pixel) in time and space. Point attributes may describe the properties of a single pixel in an image. The attributes may include intensity, color (red, green, blue), range, spatial and temporal gradients of intensity or range, flow direction and magnitude.

The colorIntensityRed, ColorIntensityGreen, and ColorIntensityBlue attributes are applied when a system employs a color camera to sense the color intensity at the pixel. ColorIntensityGrey is used when a system employs a black-and-white camera to measure the light brightness.

Additional attributes include range, used when, for example, LADAR is used in a system. Point attributes may also describe the output of individual sensors, such as a position, velocity, torque, or temperature sensor, at a point in time. In an acoustic sensor, intensity typically refers to amplitude. There is also an image flow attribute that may be computed from the color intensity attributes. The right hand side of Figure 2 shows such an effect.

3.3 List entity Classes

A list entity is composed of sets of point entities that satisfy certain grouping hypotheses over space and/or time. List entity classes can be sub classified into edge, vertex, and surface patch classes, as shown in Figure 3.

- An edge may consist of a set of contiguous pixels for which the first or second derivatives of intensity and/or range exceed threshold and are similar in direction. Edge entity attributes may include the orientation, length, and curvature of the edge, the sharpness or magnitude of the discontinuity at the edge, as well as the centroid of the group of points that make up the edge.
- A vertex may consist of two or more edges that intersect. Vertex attributes may describe the relationship between the set of edges that make up the vertex, including the type (e.g. X, V, T, or Y), the orientation, and the angles between lines forming the vertex.
- A surface patch may consist of a set of contiguous pixels with similar first or second derivatives of intensity and/or range and similar image flow vectors. Surface patch attributes may include the collective properties of the set of connected points that make up the patch. For example, surface patch attributes may define the position, and velocity of the surface patch, the texture, and the orientation of the surface patch relative to the viewing point.

Note that redundant aggregation relationships, at both the generic class side (list to surface) and the derived class side (surface patch to surface) are allowed since they do not consume actual data structure memory of the component class until the relationships are actually needed at the implementation stage.

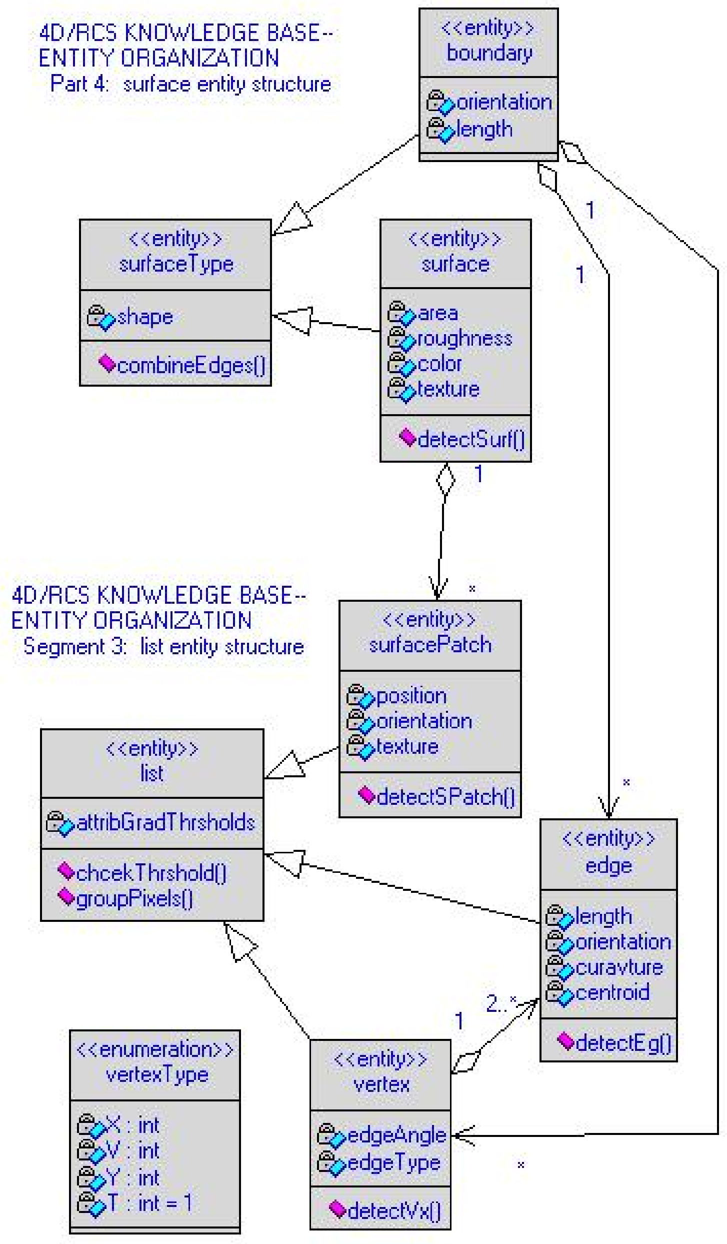

Figure 3 - list and surface entity model

3.4 Surface Entity Classes

Surface entity classes can be subclassified into surface and boundary entities, as seen at the top portion of Figure 3.

- Surface entities consist of sets of contiguous list entities that satisfy certain grouping hypotheses. For example, a surface may consist of a set of contiguous surface patches that have similar range, orientation, texture, and color. Surface attributes may describe the properties of the surface, such as its area, shape, roughness, texture, color, and position and velocity of the centroid.

- A boundary may consist of a set of edge entities that are contiguous along their orientation. For example, boundary attributes may define the shape of the boundary, its orientation, its length, its position, and velocity, which side each surface lies on, etc.

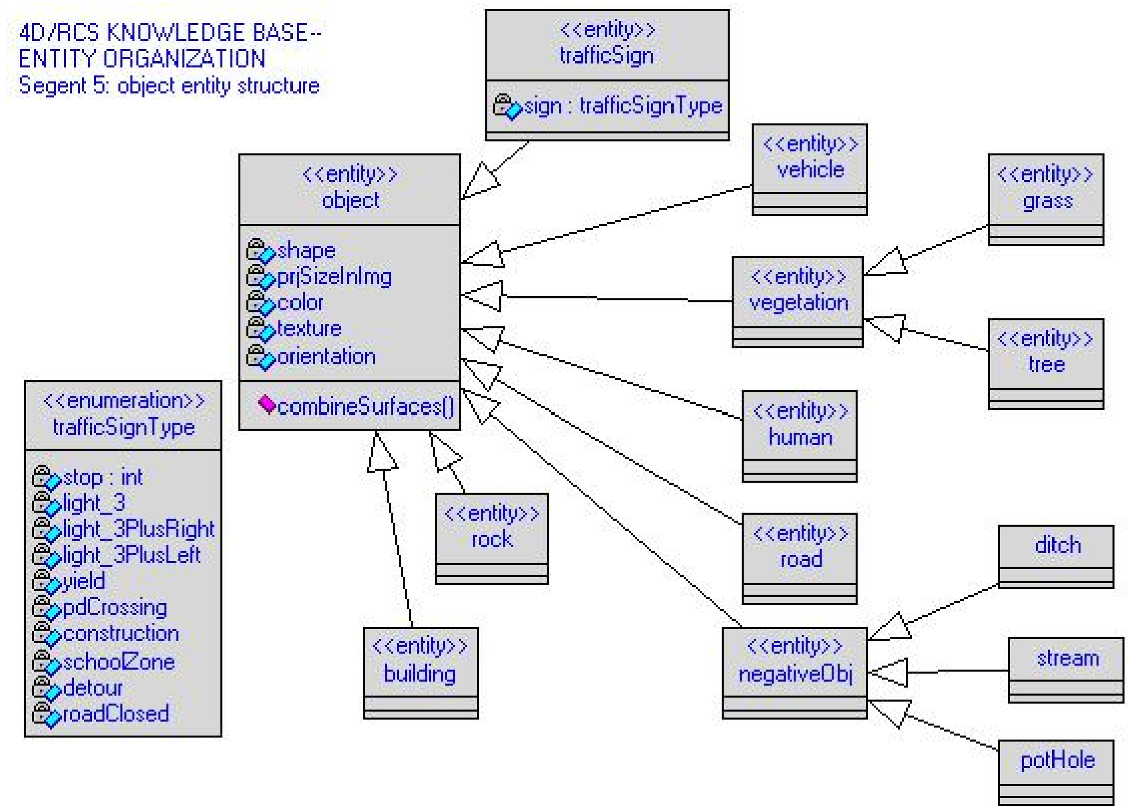

Figure 4 - object entity model

3.5 Object Entity Classes

Object entities consist of sets of contiguous surface and boundary entities that satisfy certain grouping hypotheses, as seen in Figure 4. For example, an object may consist of a set of surfaces that have roughly the same range and velocity, and are contiguous along their shared boundaries. Object attributes are computed over the entire set of points that are included within the object. Object attributes may describe the properties of an object, such as its volume, shape, projected size in the image, color, texture, position and velocity of centroid, orientation and rotation about the centroid.

Object entity classes can be further classified to form more detailed knowledge to support particular mission requirements. Figure 4 shows that the subclasses of traffic signs, vehicles, vegetation, humans, roads, negative objects, rocks, and buildings. They can be sub classified even further as needed.

3.6 Group Entity Classes

Group entity classes consist of sets of objects that have similar attributes, such as proximity, color, texture, or common motion, as seen in Figure 5. Group attributes are computed over the entire set of objects that are included within the group. Group attributes may describe the properties of a group, such as the number of its members, size, density, position, velocity, average direction of motion, and variance from the mean.

Figure 5 shows the sub classification of a generic object group into hill, village, woods, and a scout vehicle company.

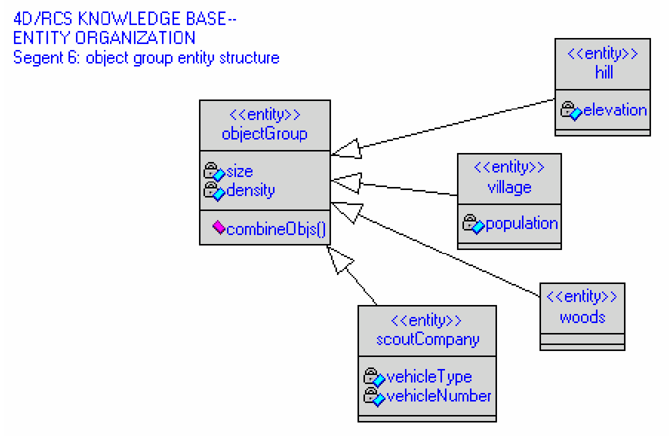

Figure 5 - object group entity model

4. SUMMARY AND FUTURE WORK

We described a hierarchical entity knowledge model, which contains multiple levels of abstraction and, therefore, supports multiple levels of intelligent system decision making requirements.

We anticipate expanding the model with further details, including specific details of attributes, subtypes and classifications of objects, and their detection methods. We also plan to describe how this entity knowledge model can be integrated into an intelligent system architecture and illustrate how the model can be applied.

We also intend to take operational aspects into consideration, such as:

- Real-time updates, tracking, and possibly versioning the perceived entities.
- Storing the knowledge efficiently in short through long term memories based on the access timing requirements.

PRODUCT/COMPANY DISCLAIMER:

Certain commercial products or company names are identified in this paper to describe our study adequately. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products or names identified are necessarily the best available for the purpose.

REFERENCES

[1] James Albus, et al., 4D/RCS: A Reference Model Architecture For Unmanned Vehicle Systems, Version 2.0, NISTIR 6910, Gaithersburg, MD, 2002

[2] Smyth, C. Stephen, "A Rrepresentational Framework for Geographic Modeling," Book Chapter in SPATIAL and TEMPORAL REASONING in GEOGRAPHIC INFORMATION SYSTEMS, Egenhofer M. and Colledge R., Eds., Oxford University Press, New York, NY, 1998.

[3] Megalou, E. and Hadzilacos, T., "Semantic Abstractions in the Multiple Media Domain," IEEE Transactions on Knowledge and Data Engineering, Vol 15, No. 1, January/Feuruary 2003.

[4] Open GIS Consortium, OpenGIS® Geography Markup Language (GML) Implementation Specification, Version: 3.00, OGC 02-023r4, http://www.opengis.org/, 2003.

[5] Ng, R.T. and Han, J., "CLARANS: A Method for Clustering Objects for Spatial Data Mining," IEEE Transactions on Knowledge and Data Engineering, Vol 14, No. 5, 2002

[6] Tomic, S. et al., "A new measure of temporal consistency for derived objects in real-time database systems," Information Science 124 (2000), pp. 139-152, Elsevier, New York, NY, 2000.

[7] Balakirsky, S.B., et al., "A Hierarchical World Model for an Autonomous Scout Vehicle," Proceedings of the SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, Orlando, FL, April 1-5, 2002.

[8] Chang, T., et al., "Concealment and Obstacle Detection for Autonomous Driving," International Association of Science & Technology for Development - Robotics & Applications99 Conference, Santa Barbara, CA, October 28-30, 1999,

[9] Hong, T.H., et al., Obstacle Detection and Mapping System, NISTIR 6213, National Institute of Standards and Technology, Gaithersburg, MD, August 1998