A Robust Cooperative Behavior Control Architecture For Teams of Unmanned Systems

Abstract

This short White Paper briefly describes a method to allow teams of unmanned vehicles to communicate using an approach that has the potential to overcome common communications bandwidth limitations. This is accomplished through the development of an open ISO standard knowledge architecture in which knowledge instead of data is shared between the individual unmanned entities. Point of contact at BRC for this White Paper is Andy Bevilacqua (256) 882-6229, Ext 102, email: andyb@brc2.com.

1.0 IDENTIFICATION AND SIGNIFICANCE OF THE PROBLEM OR OPPORTUNITY

> ***Replacing single manned or unmanned systems with clusters or teams of unmanned systems will require the use of innovative processing techniques, which can effectively handle both data (information) and knowledge. Recent advances in the field of cognitive science show promise for providing a robust solution to this problem.***

Because unmanned systems are relatively "dumb" in terms of human intelligence, they tend to be data collectors (sensors), collecting information and transmitting that information to ground stations where it can be processed further with intervention by humans. The functions of collection and retransmission of data/information from a single collection source are relatively simple and can be handled quite easily by classic methods of programming. Teams of unmanned vehicles (UV) however, working cooperatively, must also communicate between themselves to assure that missions are completed in the most efficient manner. Therefore, once the single collector/transmitter platform is replaced with individual unmanned systems, the problem becomes much larger. Cooperation among the team is required to share the functions /tasks for which the team is assigned. What this means is that in order for a team to handle the normal data collecting functions of a single large platform, knowledge (fused or correlated data) must be shared in some way. Knowledge is required to allow the leader to task and control the individuals in the group, to respond to emergencies, to handle data or hardware problems, to fuse multiple heterogeneous data sources, and to perform other decision-making functions critical to team mission and functions. There are two basic ways to implement this knowledge sharing, either through a centralized leader-follower configuration in which the leader has full control over all followers and there is instantaneous sharing of all information or through a decentralized control approach where each individual UV is a leader and makes its own decisions through cooperation with other leaders. Each of these approaches has its good and bad points. The centralized leader-follower configuration keeps communications bandwith and the need for extensive knowledge handling low, however, it is susceptible to a single-point failure if the leader.

Decentralization of the control over all team members reduces the problem of single point failures but can significantly increase communications bandwidth if not handled properly. Realistically the teams will encounter partial information errors, bandwith restrictions, latencies in the data, dropouts and other problems while communicating. This drastically increases the complexity of the control algorithms (knowledge content) and therefore the amount of inter-communication that must take place.

In the past, efforts of this type would have been attacked using an algorithmic approach such as a rule-based expert system (or some variation of one). In such a system "rules" based upon the "IF"-"THEN" paradigm of logic would be used to examine the current situation and make a decision regarding the next course of action. Finite state machines (FSM) and approaches that mix genetic algorithms, neural networks, etc with rules or states are simply variations of the basic rule-based expert system paradigm and therefore tend to have the same basic strengths and weaknesses. The problem with these outdated approaches is that they tend to be extremely brittle, cannot adapt their logic dynamically, fill up available bandwidth quickly as they grow, become far too complex as cross-coupling between logic paths occur, and are far too slow to work for real-time applications such as those required on a moving sensor platform. What is needed for this application is a new approach to knowledge storage, sharing and retrieval that can work in real time, is non-brittle and that can adapt its knowledge base in real time in response to outside stimuli while still allowing its logic to be "bounded" within pre-set limits. Fortunately, recent advances in the field of cognitive science and the establishment of new standards for knowledge interchange that promote knowledge-sharing and reusability have led to the development of new algorithmic modeling methods that meet these challenges.

The perfect unmanned system team control system would be scalable, adaptable, maintainable, non-brittle, easy to use, and have an associated method for the rapid acquisition and modification of knowledge. Bevilacqua Research Corporation (BRC) has been working in the areas of knowledge acquisition and processing since its incorporation in June of 1992. Since that time, BRC has been investigating new approaches to hybrid Artificial Intelligence (AI) information processing using conceptual modeling techniques. The result of that research has led to the development of new methods of cognitive reasoning that show promising results in terms of the "perfect system" attributes described above. This white paper briefly describes this technology.

2.0 Technical Basis

2.1 METHODOLOGY: Conceptual Modeling of Autonomous Vehicle Behaviors Using An ISO Standard Agent Behavior Language

The heart of BRC's approach to autonomous vehicle behavior development is the use of an advanced conceptual modeling approach for development of human reasoning software called a Bounded Neural Network (BNN). This specialized intelligent agent architecture and its associated research-level toolset/editor, called CORE, were created by BRC as a quick and efficient way to add complex cognitive processing capabilities to software systems. CORE utilizes a new innovative hybrid AI approach in which neural networks (NN), fuzzy logic, Genetic Algorithms, or other intelligent agent constructs can be tied to information storage structures called conceptual graphs (CG). By integrating these two separate AI methods into a hybrid system, the user gains the ability to store expert information in a graphic (semantic) ANSI-standard format and to retrieve it very quickly without the common problems associated with rule-based expert systems approaches. The innovative nature of this approach is attested to by BRC's receipt of the 1998 Tibbett's award for excellence in innovation in the Small Business Innovation Research (SBIR) program where it was originally developed. The stability and utility of the approach is attested to by several successful applications of the approach on previous contracts for the Government. Benefits of the conceptual modeling approach for advanced behavioral modeling include:

- The architecture is not brittle like rule-based systems - *Stable*
- Employs the newest (draft) ANSI standard for knowledge exchange - *Standardized*
- The logic can be developed graphically - *Modifiable*
- Conceptual graphs fit well with object-oriented programming schemes - *Up to date*
- Used by many Government Agencies - *Accepted*
- The processing is parallel and not linear - *Fast*
- An extremely high density of knowledge can be stored in a small space - *Accurate*
- Processing logic can be changed without extensive user training - *Maintainable*

Unlike rule-based approaches to knowledge modeling, the unique structure of the CORE engine allows a great depth of information to be stored and quickly accessed through intelligent agents. In this way, complex data and semantic level information can be accessed and correlated quickly within any application. BRC intends to utilize the CORE engine, as the basis for creating an expert knowledge base that will provide the foundation of the correlation engine's knowledge of the groups missions, needs and behaviors.

2.2 LANGUAGE: What Are Conceptual Graphs?

The CORE engine uses a standardized conceptual modeling paradigm based upon extensions to conceptual graphs initially introduced by Sowa (Sowa 1984). Conceptual graphs are the most powerful, robust conceptual modeling language available today. They combine the best features of first-order predicate logic, and semantic networks. There are several excellent short introductions to conceptual graph theory, but this quick review of the key features will clarify the importance of using conceptual graphs to develop the conceptual models need for this effort.

A conceptual graph consists of concepts and relations; concepts denoted by a box and relations denoted by a circle or oval. Relations are connected to concepts via directed links; the direction of the arrow is decided arbitrarily but usually follows a linguistic convention; e.g., in Figure 1, we read the "used for" relationship is "The used-for of Tank 9847 is attack".

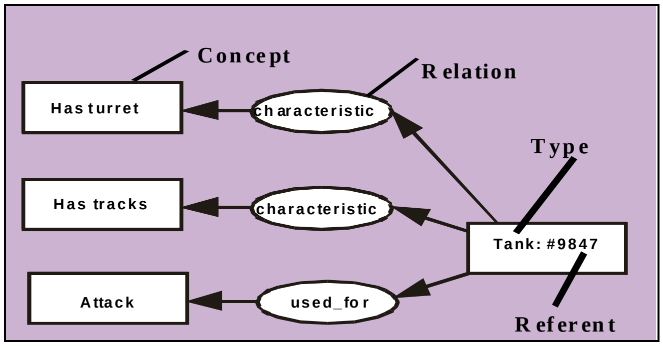

Figure 1: Parts Of A Conceptual Graph

A concept contains a type identifier (e.g., Tank) with an optional referent, which indicates a specific individual (or individuals) of that type (e.g., #9847). Associated with one or more conceptual graphs is a type hierarchy showing subtype/supertype relationships with multiple supertypes allowed, and a set of definitions (not shown here). There are several aspects of conceptual graph theory and implementation that can support modeling knowledge dependencies in a system. While other languages have some of these features, conceptual graphs have them all. These aspects are its type/relation support, dynamic logic features, inferencing capability, and graphical form.

Type and Relation Support: Conceptual graphs incorporate types as a basic feature. Every concept is of a particular type, where the types themselves are arranging in a lattice of subtypes. Changes to a type's definition are therefore immediately linked to all concepts of that type. Relations can also be arranged in a hierarchy, permitting the system to quickly identify similar relations. The hierarchy also gives us the capability of comparing dependencies in a measurable way, e.g., how far apart are related concepts in the hierarchy, etc. We have already explored using the hierarchy to determine similarity (Delugach 1993). This capability offers the potential for supporting a spectrum of dependencies, ranging from strongly dependent to weakly dependent that can be of tremendous value in a usable system.

Actors and Demons: Conceptual graphs possess the capability of representing direct or chained functional dependencies through the use of *actors*. (A CG actor is not to be confused with an intelligent entity in a simulation model.) In CGs, an actor is defined as a graph node whose input concepts directly determine its output concepts' identities. The actor's behavior can therefore be envisioned as a procedure, which "computes" the output concepts from the input concepts. As such, dynamic logic can be performed whenever input concepts' identities change during the course of knowledge base modifications. As mentioned before, it is this property of actors that would allow any type of behavior model representation (legacy, future development, etc.) to be used. There is a more powerful actor, called a *demon*, which was introduced by one of the principle investigators (Delugach 1991). This construct allows a conceptual graph to actually alter its own structure, asserting new concepts and relations, while retracting others. This means that a knowledge base can adapt its structure according to the demands of newly asserted knowledge or changing values.

Inferencing Capability: Some dependencies can be subtly indirect and must be deduced rather than stored directly. Conceptual graphs are well suited for logical inferencing, since they are based on Peirce's pioneering work in logic over a century ago. Indirect dependencies can be established as successive inference steps or links. These steps also serve to support tracking of dependencies, so that dependencies can be tracked over time, as well as over the knowledge space.

Graphical Form: Conceptual graphs are most naturally presented in visual form. They are therefore a very powerful aid to the understanding of complicated knowledge. Since we assume that any knowledge-based conceptual model is ultimately intended to support human beings in their activities, it is particularly useful to have a language with a formal semantics, yet easy to grasp visually. One promising aspect of our approach is to use conceptual graphs directly as the interface language to the knowledge base. This eliminates the need for an added operational burden of a separate interface language.

2.3 PROCESS / TOOLS: The Cognitive Reasoning (CORE) Toolkit

The Cognitive Reasoning Toolkit can be used as a way to quickly build knowledge bases to demonstrate the viability of this approach for a feasibility demonstration. The toolkit has been built and delivered to the Army STRICOM Threat Simulator Management Office (TSMO) by BRC and is Government-owned. Using the toolkit, users can turn expert information into a conceptual model format (Conceptual Graphs). Example displays from the NT-based toolkit are shown below in Figure 2.

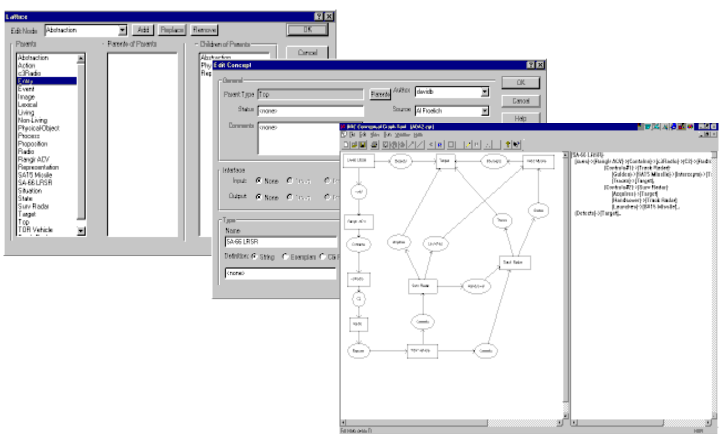

Figure 2: CORE Toolkit Displays

Graphs built using the toolkit editor are represented by an ISO Standard language (right side of figure 2) that is automatically generated as icons are pulled down onto the screen and connected by the user in logical sequences. A special parser built by BRC then turns this language into executable conceptual graphs that can be stored as classes and used by the current application. It will be extremely important for users to have access to a tool such as this, with a user-friendly interface so that cognitive control logic can be modified quickly and easily.

2.4 Conceptual Architecture For The Unmanned Systems Team Demonstration "Testbed"

Rather than creating a full simulation environment for this program BRC proposes creating a simulation in which a scenario is run that involves communication between 3 individual unmanned systems performing mission functions over a specified time period. Output from the runs would be those data that are necessary to investigate the robustness of the architecture and algorithms implemented.

The Graphical user interface (GUI) would be built using Windows menu widgets that can be quickly and easily put together. Major functions of the GUI would be to allow access to the CORE editor, allow the user to specify the degree of user interaction, i.e. man-in-the-loop through autonomous, accessing output files and statistics and start and stop the simulation. The CORE editor already exists and would allow users to quickly build and compile ISO standard knowledge (logic) bases and intelligent agents for platform control logic and information fusion functions.

2.4.1 Data Collection

The data collection process is one of the major areas where BRC will have to work closely with the customer. Our experience on other programs shows that access to the necessary data, knowledge, SME expertise, etc. is always a difficult challenge. Many times, experts are busy, or security concerns make data hard to access or unavailable. BRC would work closely with the customer early in the program to solve the issues related to data/knowledge collection during this program. SME knowledge concerning the missions and functions to be simulated will be required to create the conceptual knowledge bases and perform a feasibility demonstration.

2.4.2 Data Preparation and Storage:

The algorithm development process within the CORE editor consists of a set of process steps, which prepare the knowledge, manipulate the knowledge and create the graphical form of the conceptual graphs. There are several reasons why we selected the conceptual graph format:

- It is graphical - easy to work with
- It is an ISO Standard – Standardized for sharing of knowledge and data
- Editors exist that allow the quick and easy formatting of the data
- Set rules apply to the organization of conceptual graphs. For example concepts cannot connect to concepts, only to relationships. Therefore, the validity of the organization of the data (in the CG format) can easily be checked by automated means.
- Conceptual graph actors can encapsulate legacy control algorithms, making the architecture developed during phase I a true "testbed".

2.4.3 The Cognitive Reasoning Process

The previous sections briefly described the important functions of collecting and preparing data and information into a standard form. This will not only allow the reasoning engine to work in a standard format it will allow the engine to gain an understanding of the context of the information it is presented with. The correlation / fusion process within the BRC-designed cognitive processing engine works much like the decision processes within the human brain. Human decision-making is not a linear process, i.e. a human does not think in terms of rules; rather humans associate pieces of data to form an "image" of the problem space (situation). They then relate that problem space to the "best" solution from a solution set that fits the desired outcome. Behavioral Psychologists have used conceptual graphs for years to model the human decision-making process because they, like the brain, work in terms of semantic level concepts or images and the relationships between those concepts. In the CORE processing engine semantic level information is graphically depicted and related through a paradigm that uses concepts and relationships to form a complete first-order predicate logic system. The BRC approach breaks up the problem domain into a large number of small pieces that relate to a separable part of the problem. For example, sensed data such as geography, weather, cultural features, etc. would be stored as one conceptual graph. The attributes of each missions functions, i.e. communications, reconnaissance, navigation, etc. as another. During a mission each conceptual graph starts out in a particular state, when changes occur, such as the team leader changing assignments (communicating), the CG's for the unmanned systems entity and the current scenario/situation are joined through a boolean conceptual graph operation. Once joined, the combined graphs (semantic network) achieve a steady state condition that relates the "image" of their interaction and each entity's function is in effect, instantly reconfigured. AI constructs such as neural networks, fuzzy logic, etc. in the form of intelligent agents are used to sense the state of the combined graphs and relate that "image" to a pre-selected possibility set.

This process has been shown to work extremely well in previous programs involving decision aids and intelligent semi-automated forces (SAF) model entities.

3.0 Summary

This short White Paper briefly describes a method to allow teams of unmanned vehicles to communicate using an approach that has the potential to overcome common communications bandwidth limitations. This is accomplished through the development of an open ISO standard knowledge architecture in which knowledge instead of data is shared between the individual unmanned entities. Point of contact at BRC for this White Paper is Andy Bevilacqua (256) 882-6229, Ext 102, email: andyb@brc2.com.