Developing a distributed ontological world model for autonomous multi vehicle operations
Joel Cartwright, Pedro Patrón, Jonathan Evans, David Lane
jjc5@hw.ac.uk
Ocean Systems Laboratory, Heriot-Watt University
Edinburgh, Scotland, UK, EH14 4AS

Abstract
This study considers the problem of constructing a decentralised, distributed world model to enable more effective cooperation of multiple autonomous vehicles, whilst retaining the capability for isolated operation. A brief literature review highlighted the immature status of this area of the field, and a new architecture was proposed employing semantic web ontologies. This significantly improves on the sophistication and capabilities of the models used in existing works, whilst remaining generic and portable. A prototype implementation is now proceeding, and will be evaluated using simulated and real assets of the Ocean Systems Laboratory in cooperative underwater vehicle missions, such as mine counter measures.

Keywords: multi-agents, world model, ontology, semantic web, autonomy, distributed operations, vehicle force multiplier, multi vehicle communications, machine awareness

Introduction
Formations of multiple unmanned vehicles are raising interest in the military domain by providing new capabilities while reducing time and costs of operations. In order to provide these capabilities, it is necessary to maintain some common understanding of the environment between participants. The reliable distribution of the platform world model will be the key enabler of multi vehicle technologies.

A world model is a representation of the environment in which an agent operates. In the multi vehicle context, this also includes a representation of other vehicles within the collective. In the context of this paper, an agent is a vehicle. Within a distributed world model system, each agent maintains an independent model of the world, but information is exchanged between agents to improve their respective models. If this is performed in a peer-to-peer to fashion, a decentralised system is formed.

Such a distributed, decentralised system is advantageous because there is no single point of failure, and the agents do not need to constantly query a central information store. This reduces communication bandwidth requirements, potentially giving the agents more independence, and in consequence more autonomy.

This paper describes a distributed world model system based on ontologies. In computer science terms, an ontology is a formal specification of concepts in a domain, and the relationships between those concepts. There are widely available and proven tools for building ontologies in languages such as OWL [11]. Ontology reasoning engines are able to perform logical consistency checking of ontologies, and infer new information based on ontology axioms and rules.

An ontology may serve as a Rosetta Stone for inter-agent communication, with agents exchanging values with reference to ontological concepts. The main requirement is for the agents to both hold
the portions of ontology structure about which data is to be exchanged.

The next section gives a brief review of recent literature in the field of multi-agent world modelling. Following that are sections detailing the potential advantages of such systems, and the requirements identified for this study. The issues surrounding the implementation of a distributed world model are described in the subsequent section. Finally, a conclusion is given in summary.

Background

The area of distributed world modelling for autonomous robots is still relatively immature. In [3], two trash collecting robots are shown to use communication of position information to allow them to cooperate. This could be considered a very simple distributed world model. By sharing information about their environment, these agents are able to make more informed decisions than if they independently rely on their own partially observed state of the world.

A more recent research application is for autonomous RoboCup agents [4], where teams of Sony Aibo robots compete in a game of soccer. In this paper, each agent maintains two world models, an individual world model and a shared world model, and information is merged between these models as necessary. As in [3], these RoboCup agents benefited from having a more complete world view. However, theirs’ too is a very simple model, with the world represented by a vector of positions and headings.

In [5], ontologies for individual vehicles are used to coordinate task execution. In their system, data is sent to a central coordinator which then performs the allocation of tasks. This is contrary to the decentralised architecture that this study is concerned with.

The papers cited above show that there are some benefits to using a distributed world model in multi vehicle operations. In the following section, these advantages are discussed further.

Advantages

Here we iterate the main advantages of the decentralised, distributed world model under study.

1. Robust extraction of meta-data (low bandwidth) from raw sensor data (high bandwidth)

This allows the transmission of only critical data to the operator or between vehicles, and to the highest levels of autonomous decision making.

2. No central point of failure

By virtue of the decentralised architecture, no one node is critical to the operation of the vehicle collective.

3. Vehicles may still operate in isolation

Without the need for communications with a central node, vehicle can continue to operate even when isolated, with reference only to its local world model.

4. Peer to peer communication reduces total bandwidth requirements

As information does not have to be relayed via a central point, broadcast communications methods can be exploited to reduce the amount of information that needs to be transmitted between the individual vehicles.

5. The generic, modular world model subsystem is loosely decoupled from the rest of the vehicle’s subsystems and the other vehicles
Extension of the world model to future platforms should be possible with the minimum of modification.

Requirements

The following requirements were determined with reference to relevant vignettes and concepts of operations in documents produced for the SEAS DTC [6][7] and the US DoD master plans [8][9][10].

1. Temporal context

For a physically instantiated agent, temporal context is an important component of the world model. When observing and recording the state of the world, it is usually necessary to note the time at which those observations were made. Knowing the position of another mobile vehicle is of little use without knowledge of when that position was valid.

2. Unattended operation for long periods

The world model must provide capabilities to the vehicles for operating without human intervention and for long periods of time. Thus the system should be self-aware of its limitations and the quantities of data stored must not grow without bound, which would cause memory and/or disk storage to become full.

3. Tight resource constraints

Embedded hardware processing resources and limited inter-vehicle communications bandwidth, i.e. underwater acoustic communications, must be accounted for during system development.

4. Hard deadlines

Vehicle systems that utilise the world model are constrained by hard time limits, thus these will also apply to queries of the world model.

5. Limited observability

Broadcasting and full explicit communications are very expensive and inefficient, in addition to the contention between vehicles for underwater acoustic communications. Every data transmission performed between vehicles has to clearly add value to the mission to be justified for inclusion. Covert missions may also prohibit explicit communications. Consequently, information extraction from passive sensing and knowledge induction are desirable components. These improve system performance and capability for dealing with partially knowing the status of the other platforms and environment.

6. Uncertainty

It is often assumed that the sensors on vehicles are "perfect" sensors. In practise the variability of the environment and the observability of this variability from the sensors provide high levels of uncertainty. A distributed world model for multi vehicle operations should be able to deal with this.

7. Scalability and extensibility

By its definition, the distributed world model architecture should also be portable, extendable and applicable to different platforms in different domains.

The following section describes an approach that addresses these newly defined requirements.

World Model

The most common storage method for a large corpus of knowledge on a computer system is a database, which is built from tables of values tied together with simple relations. The Multi-Dimensional Dynamic World Modelling study developed under the MoD BAUUV contract [1] utilised such a relational database. This showed the power of a loosely decoupled world model.
with multiple levels of data refinement within a single vehicle. The current study will build on this to create a multi vehicle architecture, and utilise recent ontological technologies to add powerful semantic information to the model.

Ontologies allow richer data structure than databases, with the description of hierarchies of classes that can possess both data values, and relations to other classes. Thus they are the obvious choice for providing structure in a complex world model.

**Ontology languages and editors**

The Semantic Web ontology language, OWL [11], has emerged as the recent de-facto ontology language. With no compelling reasons to break from the status quo, OWL will be used in the ontologies for this study.

OWL has three sublanguages: OWL-Lite, OWL-DL and OWL-Full; each with increasing levels of expressiveness. OWL-Lite allows a classification hierarchy and simple constraints. OWL-DL is a superset of OWL-Lite, allowing more expressive ontologies, whilst still remaining decidable (all computations take a finite amount of time). OWL-Full is the most expressive dialect, but does not allow complete, decidable reasoning. Thus most applications of OWL utilise the OWL-DL sublanguage; we will follow suit, and any reference to OWL from this point on shall be taken to mean OWL-DL.

The Semantic Web rule language (SWRL) [12] complements OWL, allowing inference over ontologies using Horn-like rules. A simple example of such a rule:

\[ \text{Parent}(?x, ?y) \land \text{Male}(?y) \rightarrow \text{Son}(?y, ?x) \]

SWRL includes a substantial number of built in functions, including those to perform numerical operations. Thus it would be theoretically possible to perform even simple data fusion using appropriate SWRL rules. The inference capabilities afforded by SWRL may prove to be useful, particularly with the higher level concepts, such as those used in planning.

**Ontology reasoners and programming frameworks**

The two most popular ontological reasoning engines for OWL are RacerPro and Pellet. RacerPro is a closed source commercial product written in C++, with free academic licenses available. It is stated to provide support for reasoning over OWL ontologies, including SWRL rules. Pellet is open source software written in Java, and is able to reason over OWL ontologies. At this point in time, reasoning over SWRL rules is not supported by Pellet, but is planned in a future release.

Jena [17] is an open source Java framework that allows access to data stored in OWL ontologies. Queries are performed in SPARQL [12], which is effectively the equivalent of SQL for ontologies. Whilst it includes a built in rule-based inference engine, Jena also supports the application of external reasoners. Jena allows ontologies to be loaded into a database system (e.g. MySQL or PostgreSQL), for persistent storage and access. Information is stored as triples within the database.

The Pellet reasoner integrates well with Jena and the database storage backend. Unfortunately RacerPro does not have the same convenient interface to Jena, perhaps because it is written in C++ instead of Java.

KAON2 [17] appears to be the only suitable alternative to Jena, also written in Java. It is free for academic use, but commercial use is subject to a paid license, as with Racer. KAON2 does not enjoy as much community support as Jena, but more importantly does not share the same level
of integration with relational databases for persistent data storage.

For the above reasons, the tools chosen for this study are the Jena framework for interfacing to the ontologies, and Pellet for reasoning. This decision may be revisited at a later stage in this development, in light of the need for SWRL rules and Pellet’s support of such at that time. During thorough evaluation of the chosen tools, the hard deadline requirement must be noted.

**Time and uncertainty**

A good discussion of adding temporal context to ontologies is given in [19]. Their proposed solution is to create an instance of a *PropertyValue* class that contains the value (e.g. the sensor reading), and associate that with both the parameter the value refers to (e.g. vehicle position), and start and end events for the value, which give time bounds to its validity.

Additionally, to cope with uncertainty, they include a certainty parameter in their *PropertyValue* class. This could be a probabilistic measure or a variance, to be taken into account when using the value, such as in a sensor fusion process. The intention is to use a similar scheme for this study, in order to meet the uncertainty requirement (requirement 6).

**Data processing**

As in the BAUUV project [1], a key component of the world model will be modules to answer queries and process the information into other forms as needed. The robust extraction of low bandwidth meta-data from raw sensor data is an important feature of the system.

This processing may be performed pre-emptively for some sensor data, such as detecting features in raw sonar data. Other queries, such as for an interpolated vehicle position, could be serviced as needed to prevent excessive growth in the size of the data store.

This links back to the unattended operation requirement (requirement 2). A caretaker module will be required, whose task is to retire old data from the store as necessary to preserve long term system performance, either archiving or deleting it as appropriate.

**Data exchange**

To ensure inter-vehicle communications are properly controlled, a *data exchange manager* module is required on each vehicle, which will orchestrate the exchange of information between the local and remote world models. When sharing information between different vehicles, there are a number of issues to consider. These are discussed below.

1. **Access control of broadcast medium**

   In most multi vehicle environments, wireless communications will be used, such as radio waves through the air, or acoustic signals underwater. These are broadcast media, so a mechanism for media access control (MAC) is required to reduce transmission collisions. For the purposes of this research it is assumed that a MAC facility is in place, such as in [20].

2. **Bandwidth available per vehicle**

   If each vehicle is to be given an equal portion of bandwidth, on average the bandwidth available will be inversely proportional to the number of operating vehicles. Communications bandwidth is especially limited in the underwater domain that this study focuses on; typical throughput is of the order of a few kilobytes per second.
3. Data encoding

By using a shared ontology as a reference point, it will be possible to make the data exchange framework reasonably efficient whilst remaining generic. For example, if each class and relation in the shared ontology were given a unique ID number, it would be sufficient to merely prefix data values with these ID values for the data to be correctly understood by the recipient. Further size reduction might be possible using techniques based on standard compression mechanisms.

4. Level and quantity of information to be shared

The level of information to be exchanged will depend both on the needs of the recipient, and the capabilities of the communications channel. Another factor is the stealth status of the vehicles. If performing a covert mission, very little or no explicit communication may be possible. It is intended that the needs, capabilities and status of virtual links to all the peers in the collective be explicitly included in the local world model. The data exchange manager will then utilise and update this information during data exchange.

5. Reliability of data exchange

Wireless communication systems usually have some degree of loss; nowhere is this truer than of acoustic underwater communications. Data exchange between local world models must therefore be robust to this. Where information such as a regular status update is sent, it may be that data loss is merely tolerated. In the case of critical non-repeating transmissions, loss must be detected and remedied. This could be achieved with some form of acknowledgement and retransmission system, perhaps using a sequence number as in TCP, for efficiency.

6. Merging of received information into local world model

This is one of the more difficult problems to address. The simplest method would involve merely replacing the relevant parameters with the received values, e.g. the classification of a potential mine. However, this process in itself would be a form of data fusion. How information is merged into the local world model is a current area of research.

In the next section we consider the overall system architecture of a vehicle in the collective.

Architecture

Figure 1 shows how the world model fits into the overall cooperative planning vehicle architecture for this DTC project. OceanSHELL [21] is the lightweight UDP based protocol used by the Ocean Systems Laboratory.

The world model contains data at various levels of refinement/abstraction, from raw sensor data up to fused data and metadata about the vehicles and their status. Interfacing with the world model system are modules that simply deposit raw data, those that process information into higher forms, and others that query the existing data. As mentioned in the previous section, some queries may trigger data processing on the fly, such as requesting an interpolated vehicle position.

The data exchange manager discussed in the previous section is executing as part of the world model to manage sharing of data with other vehicles. A caretaker module is also running to maintain the data storage inside the hardware platform limitations,
Figure 1: World model within overall vehicle architecture

The overall aim with the world model is to make a generic, modular system loosely decoupled from the rest of the vehicle’s subsystems and the other vehicles. This will allow reuse on future platforms with the minimum of effort.

Experiments and evaluation

The world model will be tested as part of the wider architecture pictured in Figure 1, within a multi-vehicle collective. The following evaluation metrics will be used:

1. Autonomous control levels (ACL) [9]
2. Minimal bandwidth required for cooperation
3. Highest level of information (most symbolic, smallest in size) that still allows cooperation
4. Level of data communications loss that may be tolerated whilst maintaining cooperation
5. Ability to resynchronise world models and continue cooperation after an extended communications blackout
6. Communications requirements as a function of number of vehicles
7. Processing requirements as a function of number of vehicles

The system will be evaluated in an embedded architecture for autonomous systems. The evaluation will concentrate on the underwater environment; this is considered one of the most demanding settings, requiring high levels of autonomy due to the lack of communications bandwidth. Evaluation will take place on some of the platforms of the Ocean Systems Laboratory, such as the RAUVER AUV (Figure 1) and the REMUS AUV. Integration with the outcomes of other SEAS-DTC projects developed at the Ocean Systems Laboratory will provide complementary benefits in these trials.

Figure 2: RAUVER, the Ocean Systems Laboratory’s hover capable AUV

Before physical trials are performed, tests will be performed in simulation using the OSL’s Synthetic Environment (OSL-SE) (Figure 3). Initially the study will use the multi vehicle MCM (Mine Counter
Measures) mission concepts as a test-bed. The aim will be to show true cooperation between the vehicles, illustrating the advantages that communications through the distributed world model can provide, and evaluating with the metrics given above.

Once a prototype system is complete, both simulated and real assets of the Ocean Systems Laboratory will be used to prove the worth of cooperative multi vehicle systems incorporating this world model technology.

**Conclusion**

Past works in the field of world modelling for mobile vehicles were reviewed, with the literature found to be lacking a system of the sophistication that this study considers. This was in terms of both the world models used, and the peer-to-peer communications performed.

Advantages and requirements of a distributed multi vehicle world model were discussed, and possible solutions analysed. An ontology-based architecture was then proposed. This is to serve as a core enabling technology for cooperative multi vehicle operations.

A review of potential semantic web tools and technologies was performed, and from this an initial selection was made. Development of the world model system is now to proceed using Protégé to create OWL-DL ontologies, with the Jena framework used to interface to the world model, and Pellet to perform reasoning.

**References**


Acknowledgements

The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence.