Improving Multi-AUV Coordination with Hierarchical Blackboard-Based Plan Representation

Chris C. Sotzing and **Nick Johnson** and **David M.Lane**
Ocean Systems Laboratory
Heriot-Watt University
Edinburgh
EH14 4AS
Email: ccs1@hw.ac.uk

Abstract

The current state of the art in autonomous underwater vehicle (AUV) mission representation is sequential script-based plans. Though simple and easy to understand by an experienced user, this way of representing a mission is limited to static goal order and cannot easily handle unforeseen events. This limitation is amplified in multi-vehicle missions where actions need to be coordinated. This research aims to improve upon the current state of the art by designing and implementing a dynamic, hierarchical mission representation system based around the principles of blackboard systems and specifically designed to facilitate multi-vehicle coordination. The functionality of the system is tested in a simulated multi-AUV mine countermeasures mission and efficiency is compared to the state of the art. Simulated results are then validated in real world trials with two AUVs.

Introduction

Autonomous underwater vehicles (AUVs) are becoming invaluable tools in marine environments around the world. Despite this rapidly growing usage the state of the art in mission representation is limited in its ability to handle the challenges of working in the ocean. Missions are currently written in the form of scripts which are executed sequentially by the vehicle. Though they are the simplest solution, these plan representations result in rigid mission execution and an inability to handle unforeseen events. This becomes even more limiting when multiple vehicles work together since scripts have to be communicated and synchronised (stoplight systems). This is a major challenge in marine environments where communication is limited to low bandwidth acoustic transmission.

This research aims to create a mission representation that embraces the challenges of coordinating multiple vehicles in communication poor environments. BIIMAPS (Blackboard Integrated Implicit Multi-Agent Planning Strategy) is a hierarchical mission representation system that has been designed to both dynamically enhance mission execution as well as facilitate multi-vehicle coordination. In order to show its benefits, the BIIMAPS system was implemented in a multi-AUV control architecture and compared to the state of the art in script-based multi-vehicle coordination in both simulated and real world experiments.

Background and Related Work

If-Then-Else Scripts

Script based mission plans are the current state of the art in both single and multiple AUV operations. They represent the mission as a list of sequential commands. In such plans, the script represents the solution to the problem from a temporal, procedural standpoint, and subsequently there is little or no room for plan modification during execution. An example of script-based AUV control can be seen in (Allen et al. 1997).

Some systems enable plans to be extended to contain multiple scripts. In these instances execution can move from the current script to another given a certain event, and then return to the original when it is completed. An example of this type of mission plan can be found in (Rowe and Stentz 1997) where a script based mission controller is used to control an autonomous robot excavator.

Despite the application of these types of mission plans, script based systems are still very rigid and require all execution decisions to be made in advance; a major limitation for intelligent mission execution. Furthermore, though extending the system with additional scripts leads to increased flexibility, the system is still unable to cope with unforeseen events.

Hierarchical Task Networks

Hierarchical Task Networks (Erol 1995; Tate 1977) represent plans as a tree of goals. These may be decomposed into more specific sub-goals, or options to provide different methods to complete the task (for example if the goal is *travel to Paris*, possible methods could be *drive to Paris* or *fly to Paris*). Due to the plan existing in the form of a tree, the order of the tasks can be left unconstrained. Hierarchical Task Networks have been used in many areas since their creation over twenty years ago, but in recent years their inherent flexibility has proved useful in fields ranging from interactive narrative (Charles and Cavazza 2004) to mobile robot task planning (Goldman et al. 2002; Belker, Hammel, and Hertzberg 2003).

Blackboards

A blackboard system (Erman et al. 1980; Corkill 1991) is a specialised database (*blackboard*) and a number of agents

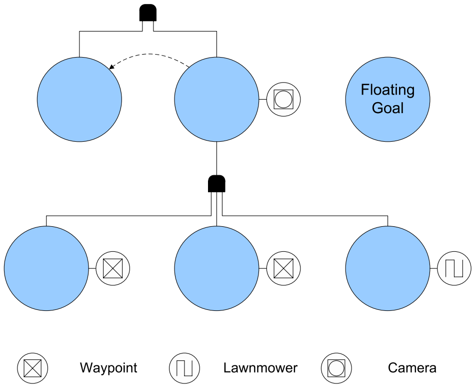


Figure 1: BIIMAPS mission modelling system diagram. Large circles represent goals; white circles contain operations (waypoint, lawnmower, camera, etc.). Dependency is represented by a dotted arrow.

(*knowledge sources*) that post information and use posted information to make further decisions. In this type of system if any data submission is deemed at any point to be invalid, all posts that were made because of it are in turn also made invalid. In this way when faced with a conflict, the blackboard can effectively roll back to the last valid point. This results in a powerful system for coordinating information between multiple agents.

Blackboard systems can be further enhanced by distributing them between multiple vehicles. In these systems blackboard data is spread over different vehicles and functions in the same manner as a normal blackboard system by synchronising data. In this way agents can communicate despite physical separation.

System Design

The BIIMAPS system resembles a Hierarchical Task Network in the way the information is arranged and structured, but with some differences and modifications. Similar to the systems described in (Goldman et al. 2002) and (Parker 1998), missions can be represented and stored in such a way to allow for more dynamic execution. In addition the BIIMAPS system utilises distributed blackboard functionality to help handle the challenges of working in communication poor environments. This section attempts to describe the system by breaking it down into its component parts and discussing each in turn. A graphical representation of the BIIMAPS system can be seen in Figure 1.

Goals and Sub-Goals

Goals are the basic building blocks of the BIIMAPS system. Represented by unique IDs they represent a specific objective in the mission and when necessary may be divided into more specific sub-goals. *Leaf* (or *atomic*) goals are those which have no sub-goals and may be considered to be discrete. Goals can be in one of three states: *complete*, *ready* (an agent may attempt to complete the goal), and *locked* (no agent may attempt to complete the goal). The state a goal is in is based upon its conditions, dependencies, and constraints, all of which are described later in this section.

A further sub-state a goal may occupy is *current* which means that that goal is currently being executed, either by the local agent or another working in the system. The current goal for any agent should always be a leaf goal, but all parents of this goal are also considered to be current (thus the root goal is always current whenever the plan is being executed).

Operations

In BIIMAPS, all leaf goals have operations associated with them. An operation specifies the behaviour required of the agent when executing a given goal. For instance if a goal is to navigate to a waypoint, the operation consists of the coordinates. Operations can also be given to non-leaf goals which again specify behaviour. In this case it is a behaviour which should be combined with those of the currently running sub-goal(s). For instance, if a goal is to navigate to a number of waypoints (notated as sub-goals), the super-goal could have an operation that calls for a video recording to be taken throughout the sub-goal operations.

Conditions

Each goal in the BIIMAPS system contains a condition that is used to determine when it has finished. For a leaf goal, the condition could be as simple as the completion of its operation. A condition may also specify the receipt of a message from another module. For example, this could be a message from a computer-aided classification/detection (CADCAC) program indicating that a particular object has been detected.

In the case of non-leaf goals, the condition should be the logical combination of its sub-goals. This can be an *and* relationship to specify a compound task, an *or* relationship to specify options for the completion of the goal, or some other logical combination. Whereas *and* relationships are be used almost exclusively for disseminating larger tasks to more simple ones, the *or* relationship is used to reduce the need for re-planning by encoding the potential actions capable of completing a task into the plan itself. This helps minimise the need to alter the plan during execution.

Dependencies and Constraints

The dependencies and constraints of a goal determine its availability based on the states of the other goals in the plan and the state of the world respectively. A goal is considered to be *ready* when its dependencies and constraints are met, and *locked* when they are not. Dependencies are essentially links to the status of other goals in the plan; this may be a link to the status of single goal, or a logical set of the statuses of many. Constraints function under a similar mechanism, although instead of the states of other goals in the plan, they refer to the state of the world (this information is received in the form of messages from other modules in the system). A similar technique can be found in (Goldman et al. 2002).

Dependencies and constraints can be demonstrated in a scenario where multiple robots are required to cross a mine field. Here *navigate across field* would depend upon *clear field of mines*. Additionally, a goal might have a constraint requiring that the goal only be attempted if there is a certain amount of battery power remaining.

Execution and Completion Locks

The BIIMAPS system can further constrain mission execution through the use of execution and completion locks. An execution lock can be applied to a super-goal indicating that if one of its sub-goals is being executed no other agent acting in the system can attempt another sub-goal. Completion locks function in a similar manner and require that if one sub-goal is executed, the remaining sub-goals must be completed before any other available goals. A good example of the application of this functionality would be two dependent goals such as *goto mine* and *destroy mine*. In this case the vehicle that goes to the mine should also be the one that destroys it.

Priorities

A priority is a way of indicating the importance of goals in relation to each other. This helps the goal selection phase during the execution of a BIIMAPS plan because it allows more important goals to be given more weight. Priorities range from 1-10 (1 being the lowest, 10 the highest) and in this way goals with high priorities are executed first. This is particularly useful in the situation where two sub-goals are related by an *or* logical. Priorities can also be used to suggest the order in which goals should be attempted; if a high priority goal is not available the system will move to one with the next highest priority. This utilisation of priorities prevents the goal sequence from being fixed as it would be if a dependency was used to impose the ordering. Goal priority is currently fixed for simplicity however these priorities could be modified at runtime should the need arise in the future.

Floating Goals

Floating goals are goals that do not contribute to the logical set of their parents condition. They are heavily constrained but given a high priority, ensuring they will run if (but only if) they are required. They do not have a condition as such, but run until their constraint becomes false. Subsequently they are not required for the completion of the plan but are used as event handlers. Floating goals are also intended to further reduce the need for mission re-planning by anticipating events which could occur during the run time of a plan and specifying the behaviour which should then occur.

A good example of a use for a floating goal is a leak manager for an AUV. In this case the floating goals constraint is *if there is a leak* and its operation would be to return to the surface and abort the mission.

Blackboard Functionality

The BIIMAPS system has been designed so that it contains much of the functionality of a blackboard; the plan taking the role of the blackboard with the agents working on the plan taking the roll of the knowledge sources. As information is received from other AUVs their current goal and list of completed goals and newly discovered targets are parsed into the receiving mission plan. This is done by iterating through the list of completed goals and based on their ID updating the corresponding goal status in the local plan to *complete*. New target goal IDs are allocated in such a way so as to avoid any overlap. These targets are then added to the root goal of the plan and given a priority of 10. Finally the goal corresponding to the sending AUV's current goal is set to *current* to prevent another agent from attempting it.

```
<goal name="Waypoint 1">

   <dependency ref="Clear Area of Mines"/>

   <condition>
      <completed ref="this"/>
   </condition>

   <operation>
      <waypoint enable="111000" absolute="true" local="true" mode="10">
         <request>
            <coordinate x="0" y="20" z="0"/>
            <heading ref="stdSpeed"/>
         </request>
         <tolerence>
            <coordinate ref="stdPTol"/>
            <heading ref="stdHTol"/>
         </tolerence>
      </waypoint>
   </operation>

</goal>
```

Figure 2: Excerpt from the XML version of a BIIMAPS plan illustrating a waypoint goal with dependencies.

The system is constantly refreshing itself and as agents post goal completions new goals are made available based on their dependencies. If at any point a goal x, which was previously believed to be completed, is found to in fact be incomplete, the system will refresh and all goals which depend upon goal x will be rolled back. This functionality is extremely important for multi-AUV mission execution because it allows for goals to be accomplished concurrently and more importantly, for recovery should any conflicts arise between vehicle plans after a period of downed communication.

Representation

While in use by a mission executive, a BIIMAPS plan is stored in memory as a structure of interlinked objects which represent the goal tree and the linkages between its elements (with each object implementing the functionality required of it). When stored and not in use, BIIMAPS plans are represented in the form of XML files, which describe the object structure. XML was selected due to its suitability for describing object structures and it's easily machine parsable and human readable nature. A parser is used to convert between these two representations and has been implemented as part of the current system. An excerpt of a BIIMAPS plan represented in XML can be seen in Figure 2.

Applications

The BIIMAPS system was motivated and developed to serve as the mission model in the DELPHÍS multi-AUV coordination architecture (Sotzing, Evans, and Lane 2007; Sotzing and Lane 2008).

The DELPHÍS system is an intelligent mission executive designed to fit in the executive layer of a hybrid tri-level architecture (Gat 1997). In architectures such as these missions are planned in the deliberative layer using an adaptive mission planning module. These plans (Such as BIIMAPS) are then passed to the executive layer where they are completed by the mission executive (Such as DELPHÍS). When faced with problems or new information, the mission executive traditionally requires the mission planner to perform a mission re-plan. This process is time consuming and consequently not practical in most real world situations as it can result in a lack of synchronisation between agents. The DELPHÍS system aims to coordinate vehicles in communication poor environments while keeping the need for a mission re-plan to a minimum.

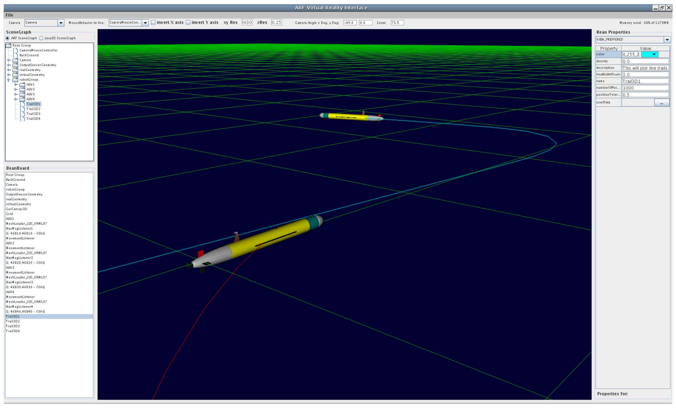
The incorporation of the BIIMAPS mission model in the DELPHÍS system allows for much of the system's coordination functionality. Goal selection works in conjunction with the Mission Model to select the best possible goal for the AUV to achieve. The BIIMAPS system is able to return a list of only the goals in the mission that are available for execution. This list is then passed to the goal selection algorithm where it is pruned down to a single goal representing the best available task for the given AUV. Once the goal is executed, its status is updated and the loop repeats until the mission is complete.

AUVs are coordinated via acoustic broadcasts containing information about each vehicle. This information includes AUV data as well as mission information, consisting of current goal, a list of previously achieved goals and newly discovered targets. Because all vehicles in the system utilise copies of the same BIIMAPS plan, goals can be referenced solely by their ID number, thereby keeping communication bandwidth low.

One of the main tools the DELPHÍS system uses to coordinate multiple vehicles is prediction. Using the recursive modelling method (RMM) (Durfee 1995), vehicle action can be predicted in the likely event of limited or no communication. The BIIMAPS system helps facilitate this by including a *predicted* flag in each goal node which is set to true when a status update is predicted. When communication returns, the model can refresh itself with the new received data and using its blackboard functionality roll back any incorrect predictions that may have been made. The BIIMAPS system's ability to reconcile an incorrectly predicted plan is paramount to coordinating multiple AUVs. Data supporting this will be shown in the next section.

Results

In order to demonstrate the benefits of the BIIMAPS mission representation system over current script-based systems, the BIIMAPS enhanced DELPHÍS system was compared to a script-based stoplight system (the current state of the art in multi-AUV coordination) and evaluated in terms of efficiency. In stoplight systems current single-AUV script execution systems are modified to include stoplight points where vehicles can synchronize with each other. The benefit of such systems is that current AUV control architectures can be utilised with few modifications. However, autonomy is sacrificed because the user has to plan the static sequence of the mission in advance including all synchronization points.



Property	Value
color	0,255,2
density	0.0
description	This will plot line trails
lineWidthPixels	3.0
name	Trail3D1
numberOfPoi...	1000
positionToler...	0.5
userData	...

Figure 3: An augmented reality visualization of a simulated multi-AUV MCM mission.

Both systems were given an identical mine countermeasures (MCM) mission and then graded for efficiency. In this scenario, efficiency of multiple AUV operations is defined as a combination of the following:

- t - Mission speed (expected mission time / actual mission time)
- r - Redundancy (% of goals that were achieved only once)
- m - Missed goals (% of goals that were accomplished)
- x - Mine discovery (% of mines that were discovered)

Efficiency (e) was calculated using the following formula:

$$e = 100(t * r * m * x)$$

Experiments were conducted in simulation to determine the efficiency of the system in deteriorating communication environments. A high-fidelity AUV simulator in combination with an augmented reality framework (ARF) (Davis et al. 2007) provides a realistic environment for the software-in-the-loop evaluation of vehicle execution code. An image of the system coordinating a simulated MCM mission is shown in Figure 3.

To evaluate efficiency this research compared three different multi-AUV control architectures: a stoplight system and two versions of the DELPHÍS system (optimised and with 2X prediction). The 2X prediction system was the DELPHÍS system with the prediction module predicting actions happening two times faster than in reality. This system was used to test the BIIMAPS system's ability to roll back the plan in the face of incorrect predictions. The MCM mission was executed by each system for 2-4 vehicles in varying degrees of communication success ranging from 100-10%. Results can be seen in Figures 4, 5and 6.

The efficiency of the three coordination systems shows clear trends as the communication rate drops. The opti-

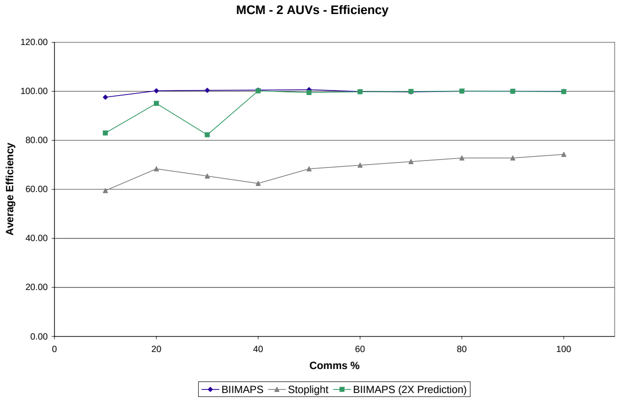


Figure 4: Coordination efficiency data for 2 AUVs. The BIIMAPS enhanced DELPHÍS system is shown in blue, stoplight system in gray and BIIMAPS enhanced DELPHÍS system with 2X prediction is in green.

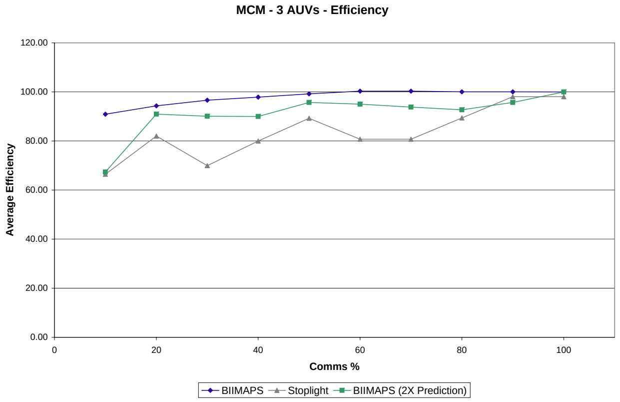


Figure 5: Coordination efficiency data for 3 AUVs. The BIIMAPS enhanced DELPHÍS system is shown in blue, stoplight system in gray and BIIMAPS enhanced DELPHÍS system with 2X prediction is in green.

mised DELPHÍS system is able to maintain a high level of efficiency throughout, with only a minor drop towards the 10-20% communication success rate. The 2X prediction DELPHÍS system was only marginally less efficient reporting data that was just under that of the optimised system. This is logical as incorrect predictions would result in some decrease in efficiency. The blackboard functionality of the BIIMAPS system however was able to reconcile most of these incorrect predictions resulting in efficiency only just below that of the optimised system with correct prediction.

The stoplight system showed lower efficiency than the DELPHÍS systems, however this was mostly due not to coordination errors (since stoplight systems are by definition scripted a priori) but to the time it took for the mission to be accomplished. Because the mission couldn't be optimised during execution, the stoplight system was often slower than the other three systems and this had a major effect on efficiency. A good example of this can be seen in the 2 vehicle

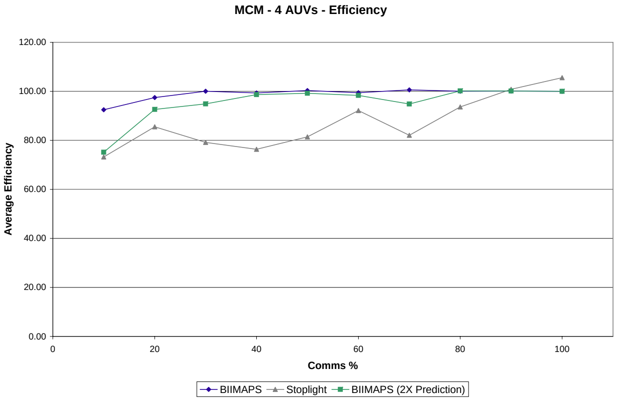


Figure 6: Coordination efficiency data for 4 AUVs. The BIIMAPS enhanced DELPHÍS system is shown in blue, stoplight system in gray and BIIMAPS enhanced DELPHÍS system with 2X prediction is in green.

data where the stoplight system recorded significantly lower efficiency than the other systems. This was because in the 2 vehicle scenario, the MCM mission took significantly longer when controlled by the stoplight system (one vehicle performing the lawnmower search pattern, one vehicle investigating targets) than it did when controlled by the DELPHÍS system.

Stoplight efficiency data wasn't only affected by time however. There were also coordination errors regarding the newly discovered mines which were not initially scripted in the mission but added at runtime. These targets were chosen as they came available and without the mission optimisation techniques of the BIIMAPS enhanced DELPHÍS system goal conflicts couldn't be reconciled. This was particularly common in the experiments with 3 and 4 vehicles where due to the larger group size coordination was more complex .

In some cases efficiency of the runs was above 100%. In this research mission time was compared to the *expected* data which is the time data returned by the DELPHÍS system at 100% communications. In the cases where efficiency was above 100% this simply meant that the mission was accomplished faster than the DELPHÍS system. This is evident in the 4 vehicle data (Figure 6) where the stoplight system starts off more efficient than the others.

In some of the data there are bumps and dips in otherwise clear trends. These anomalies are due to the fact that each experiment was only run ten times. The trial size was deliberately limited to keep the study manageable since multitudes of experiments needed to be run. However, with a trial size of only ten, random events can have large effects on the data. A good example can be seen it the two vehicle data where the 2X prediction system recorded a drop in efficiency at the 30% communication level. In three of the ten trials one redundant goal was recorded. In addition, in one of these ten trials four goals were missed. The redundant goal trials resulted in longer mission times, and the combination of these factors led to the dip in efficiency. Large trial

Figure 7: Ocean Systems Laboratory REMUS Autonomous Underwater Vehicle.

Figure 8: Ocean Systems Laboratory Nessie III Autonomous Underwater Vehicle.

sizes are expected to average out this anomalous data.

Real World Validation

To validate the simulated results trials were conducted with the REMUS and Nessie III AUVs in Loch Earn, Scotland. Using the BIIMAPS enhanced DELPHÍS system these two AUVs coordinated a mine countermeasures mission similar to the one tested in the simulated experiments. This section will first introduce the vehicles used in the trials followed by a presentation of the results.

REMUS AUV

The Remote Environmental Monitoring UnitS (REMUS) AUV (Figure 7) is the industry standard AUV. This is a transit AUV which means it flies through the water much like a plane through the air. Though it lacks the hovering capabilities of an intervention AUV, it is significantly faster and can cover a lot of sea in a relatively small amount of time. Sensors include sidescan sonar, Doppler Velocity Log (DVL) (both downward and upward facing) and acoustic modem in addition to a host of environmental sensors (water temperature, salinity, etc.).

Figure 9: REMUS and Nessie III AUVs coordinating in Loch Earn, Scotland.

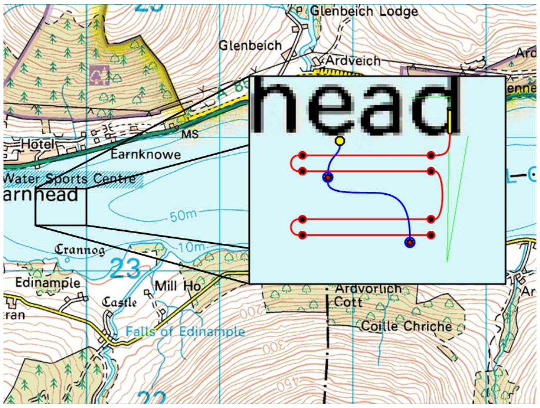


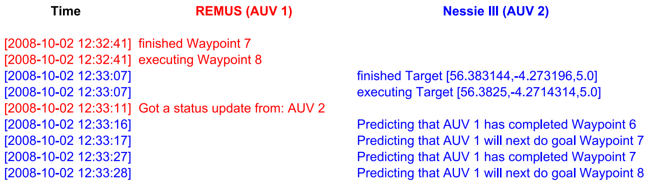
Figure 10: Multi-AUV mission Loch Earn, Scotland.

The REMUS AUV is the choice of most navies throughout the world including the US, UK and New Zealand services among others. This widespread usage and its proven service as an area search vehicle (von Alt et al. 2001) make it an excellent choice for this research, particularly when combined with an intervention AUV such as the Nessie III vehicle.

Nessie III AUV

Nessie III (Figure 8) is the third generation of an intervention AUV created by the Ocean Systems Laboratory to compete in the Student Autonomous Underwater Challenge - Europe (SAUC-E) (DSTL 2008) competition, of which it was the 2008 champion. Like all intervention AUVs, Nessie III can move in 4 degrees of freedom and has the ability to maintain position with a high amount of accuracy. Sensors on board include binocular forward and down facing cameras, DVL and acoustic modem, with additional sensors easily accommodated.

The Nessie III AUV is a relatively slow vehicle compared to other AUVs but due to its design is extremely manoeuvrable and can get in close to objects for investigation, an ability that most transit AUVs lack. When teamed with a



```
      Time                        REMUS (AUV 1)                              Nessie III (AUV 2)

[2008-10-02 12:32:41]  finished Waypoint 7
[2008-10-02 12:32:41]  executing Waypoint 8
[2008-10-02 12:33:07]                                        finished Target [56.383144,-4.273196,5.0]
[2008-10-02 12:33:07]                                        executing Target [56.3825,-4.2714314,5.0]
[2008-10-02 12:33:11]  Got a status update from: AUV 2
[2008-10-02 12:33:16]                                        Predicting that AUV 1 has completed Waypoint 6
[2008-10-02 12:33:17]                                        Predicting that AUV 1 will next do goal Waypoint 7
[2008-10-02 12:33:27]                                        Predicting that AUV 1 has completed Waypoint 7
[2008-10-02 12:33:28]                                        Predicting that AUV 1 will next do goal Waypoint 8
```

Figure 11: Multi-AUV mission logs illustrating prediction of intent.

transit AUV like REMUS this type of vehicle is an excellent choice for most multi-AUV missions.

Trial Results

To validate the BIIMAPS system it was tested as part of the DELPHÍS architecture to coordinate the REMUS and Nessie III AUVs in executing an MCM mission. Held at Loch Earn, Scotland the mission consisted of a 4 200m leg lawnmower search pattern and 2 simulated mines (Figure 10). Due to the large difference in speed between the REMUS and Nessie III AUVs (2.0 and 0.8 metres per second respectively) the lawnmower legs were contained within a “Search” super-goal that contained both execution and completion locks. This made it so that only one vehicle could attempt the search at any one time.

As mentioned earlier vehicles were coordinated via acoustic broadcasts. Due to the size limitations of these broadcasts (32 bytes) information had to be kept to a minimum. However, because each vehicle was pre-loaded with the same BIIMAPS mission plan vehicle intention could easily be transmitted in the form of goal IDs thereby significantly reducing the amount of transmitted information.

The mission started by first starting the REMUS AUV (represented in red in Figure 10) which began executing the lawnmower since at mission start no targets had been discovered. This was done by selecting the most suitable of the 4 legs based on the vehicles current state/position. The Nessie III AUV (represented in blue in Figure 10) was then started and the vehicles registered with each other via acoustic broadcast. Initially because the lawnmower was encompassed within a “Search” super-goal that was being achieved by the REMUS AUV there were no goals available to the Nessie III vehicle and it waited in a loiter position. The discovery of targets by REMUS resulted in their being added to it’s local BIIMAPS plan. This data was then trasmitted acoustically to Nessie III which took the information and passed it to it’s local BIIMAPS plan to be updated. The new target information was updated thereby synching the two vehicle plans and Nessie III began execution of the most suitable newly available target. A summary of an example trial run can be seen in the inset of Figure 10.

During execution the time between received broadcasts varied but ranged from about 20 to 120 seconds. In the longer communication blackout periods vehicles had to predict the progress of each other. This was particularly true of Nessie III having to predict the status of REMUS as it’s communications were more prone to dropout. This can be seen in an excerpt of the two vehicle logs shown in Figure 11. The BIIMAPS system was successfully able to update the mission with predicted data and then reconcile it once acoustic communication was returned and updated data received.

Conclusion

The use of the BIIMAPS mission representation system has shown to increase efficiency over the current state of the art in multi-AUV operations. Its hierarchical nature allows for missions to be executed dynamically in contrast to the strict order of script-based systems and its blackboard functionality facilitates vehicle prediction which is paramount in subsea operations. In addition real world, in water tests successfully validated the BIIMAPS system to work in the real world environment.

Building upon these results there are a number of aspects of the BIIMAPS system that can be studied further. Most planning systems with enough power and complexity to be useful give very little scope for the creation of plans by those with little experience in the area of planning (such as end users for many robotics systems). One of the motivations of the creation of BIIMAPS was the need for a representation that lent itself to graphical display, which could be used as part of a system to ease the creation, editing and understanding of plans created for complex robotic systems. It is hoped that further research will result in a successful system incorporating a powerful user interface tools in line with the principles suggested in (Johnson, Patrón, and Lane 2007) and following the methodology laid out by the Pilots Associate Program (Banks and Lizza 1991).

Acknowledgments

The authors would like to thank the members of the Ocean Systems Laboratory at Heriot-Watt University and SeeByte Ltd for their support.

References

Allen, B.; Stokey, R.; Austin, T.; Forrester, N.; Goldsborough, R.; Purcell, M.; and von Alt, C. 1997. Remus: A small, low cost auv; system description, field trials and performance results. In *Proceedings of OCEANS*, volume 1, 219–224.

Banks, S., and Lizza, C. 1991. Pilot’s associate: a cooperative, knowledge-based system application. *IEEE Expert* 6(3):18–29.

Belker, T.; Hammel, M.; and Hertzberg, J. 2003. Learning to optimize mobile robot navigation based on htn plans. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, 4136–4141.

Charles, F., and Cavazza, M. 2004. Exploring scalability of character-based storytelling. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems.*, 872–879. University of Teesside.

Corkill, D. D. 1991. Blackboard systems. *AI Expert* 6:40–47.

Davis, B. C.; Patrón, P.; Arredondo, M.; and Lane, D. M. 2007. Augmented reality and data fusion techniques for enhanced situational awareness of the underwater domain. In *Proceedings of IEEE Oceans*.

DSTL. 2008. Sauc-e http://www.dstl.gov.uk/news_events/competitions/sauce/index.php.

Durfee, E. H. 1995. Blissful ignorance: Knowing just enough to coordinate well. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, 406–413.

Erman, L.; Hayes-Roth, F.; Lesser, V. R.; and Reddy, R. 1980. The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys* 12(2):213–253.

Erol, K. 1995. *Hierarchical Task Network Planning: Formalization, Analysis, and Implementation*. Ph.D. Dissertation, Department of Computer Science, The University of Maryland.

Gat, E. 1997. On three-layer architectures. In *Artificial Intelligence and Mobile Robots*. MIT/AAAI Press.

Goldman, R. P.; Haigh, K. Z.; Musliner, D. J.; and Pelican, M. J. 2002. Macbeth: A multi-agent contraint-based planner. In *Proceedings of the 21st Digital Avionics Systems Conference*, volume 2, 7E3–1–7E3–8.

Johnson, N.; Patrón, P.; and Lane, D. M. 2007. The importance of trust between operator and auv: Crossing the human/computer language barrier. In *Proceedings of IEEE Oceans*.

Parker, L. E. 1998. Alliance: An architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation* 14:220–240.

Rowe, P., and Stentz, A. 1997. Parameterized scripts for motion planning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotic Systems*, volume 2, 1119–1124.

Sotzing, C. C., and Lane, D. M. 2008. Improving the coordination efficiency of multiple auv operations using prediction of intent. In *Proceedings of the 3rd SEAS DTC Technical Conference*.

Sotzing, C. C.; Evans, J.; and Lane, D. M. 2007. A multi-agent architecture to increase coordination efficiency in multi-auv operations. In *Proceedings of IEEE Oceans*.

Tate, A. 1977. Generating project networks. In *Proceedings of IJCAI-77*, 888–893.

von Alt, C.; Allen, B.; Austin, T.; Forrester, N.; Goldsborough, R.; Purcell, M.; and Stokey, R. 2001. Hunting for mines with remus: A high performance, affordable, free swimming underwater robot. In *OCEANS '01*, volume 1, 117–122.