Australian Government
Department of Defence
Defence Science and
Technology Organisation

Simulation Trial Results for the Cooperative Autonomous Underwater Vehicle Demonstration (SA15)

Anthony Travers

Maritime Platforms Division

Defence Science and Technology Organisation

DSTO–TN–1171

ABSTRACT

A simulation trial was conducted to test the connectivity between the two heterogeneous software systems that were part of a cooperative Autonomous Underwater Vehicle Demonstration being undertaken by the Australian and Singaporean defence science agencies. Since the final, demonstration trial was to be undertaken in Singapore, with no possibility of undertaking repeat trials, a risk management strategy was developed. This included, where possible, using the internet to simulate connectivity and communciation between the two nations' vehicles. The trial, which involved connecting the mission management systems of the two vehicles, and testing the High Level Protocol's functionality for the prescribed Mission, culminated in a successful demonstration of all relevant systems functioning together.

APPROVED FOR PUBLIC RELEASE

Published by

DSTO Defence Science and Technology Organisation
506 Lorimer St,
Fishermans Bend, Victoria 3207, Australia

Telephone: (03) 9626 7000
Facsimile: (03) 9626 7999

AR No. 015-9585
May, 2013

APPROVED FOR PUBLIC RELEASE

Simulation Trial Results for the Cooperative Autonomous Underwater Vehicle Demonstration (SA15)

Executive Summary

Australian/Singapore Subsidiary Agreement 15 "COOPERATIVE AUV DEMONSTRATION" involved the development of Mission Management software to autonomously manage the vehicles for a trial. This software was used to maintain an overall Mission-level representation and to trigger vehicle-level behaviours in the Autonomous Underwater Vehicles participating in the mission.

Rapid development software techniques provided the ability to address changing requirements for the Mission Manager. This proved very effective in facilitating the preparations for the final demonstration.

From July to August 2008, a number of internet-based simulation trials were conducted to test the ability for the Singaporean Meredith Autonomous Underwater Vehicle and DSTO's Mullaya Autonomous Underwater Vehicle to communicate via an agreed protocol.

The simulation trial provided an important opportunity to test software integration and thus providing a more mature robust end product. This delivered significant benefits to the International Collaboration in terms of minimising risk, facilitating development of solutions to identified problems, and reducing the cost of the cooperative program.

For future similar trials, it is recommended that an equivalent simulation trial be conducted to de-risk trials by testing as much of the interconnectivity of the software systems as practicable. It is also recommended that more detailed simulations of the communication mediums also be trialled to help develop and test the robustness of the proposed solutions (especially in challenging domains like underwater).

THIS PAGE IS INTENTIONALLY BLANK

Author

Anthony Travers
MPD

Tony Travers is a Senior Research Scientist at Maritime Platforms Division of DSTO. He is responsible for the Autonomous Mission Management and Collision Avoidance Systems for the Automation and Unmanned Maritime Systems Group. Tony is also part of a team that develops scientific visualisations for use in concept evaluation and forensic investigation within Defence. He joined DSTO in 1999.

Tony has a PhD in Artificial Intelligence from Curtin University of Technology in Perth. He has expertise in areas such as computer programming, 3D graphics, artificial intelligence and simulation.

THIS PAGE IS INTENTIONALLY BLANK

Contents

THIS PAGE IS INTENTIONALLY BLANK

1 Introduction

Under Subsidiary Arrangement Number 15 to the Agreement between the Government of the Republic of Singapore and the Government of Australia for Cooperation in Defence Science and Technology (Subsidiary Arrangement Number 15, Cooperative AUV Demonstration 2005), a Cooperative Autonomous Underwater Vehicle (AUV) Demonstration was organised between the Singaporean Defence Science and Technology Agency (DSTA) and the Australian Defence Science and Technology Organisation (DSTO). The Singaporean Defence Science Organisation (DSO) provided the technical capability on behalf of the DSTA.

Under Subsidiary Arrangement Number 15 (SA15), it was proposed that the DSTA/DSO vehicle "Meredith" and the DSTO vehicle "Mullaya" conduct a cooperative demonstration in Singaporean waters. The key planned events for the trial were: the deployment of the vehicles, the vehicles transiting to a given location, the survey of an area and the reacquisition of a target that had been detected by one of the vehicles. Both vehicles used a common type of acoustic modem to provide the sole mechanism for communication between the vehicles. The cooperative trial, including details of the demonstrated mission, are described in detail in the final report of the Trial (Neill 2012).

As part of the preparation for the Demonstration, a simulation of the mission was conducted with the aim of identifying and mitigating risks and testing the integration of the two vehicles' software systems. The internet was used to mimic the connectivity provided by the acoustic modems, thus enabling remote testing of the two research teams' software systems. This Technical Note details the design, implementation and the results of the Simulation Trial (conducted during July and August 2008).

2 Communication System Design

Both Meredith and Mullaya are custom built vehicles with their own individual software systems for control and mission management. In order to facilitate the interaction between the vehicles, an abstract High Level Communications protocol (hereafter called the "Protocol") was developed specifically for this Demonstration. The Protocol was designed under the assumption that there was no need to interact at vehicles' control system level. Hence it was necessary and sufficient that the Protocol provide limited command and confirmation messages that would be sent between the two vehicles in order to coordinate each vehicle's mission state. The Protocol also contained data messages needed to determine Mullaya's current position.

The development of the Protocol was driven by two key aspects of the Demonstration. Firstly, underwater acoustic communications is highly error prone and has an extremely low bandwidth (the amount of data that can be sent over a given time). Secondly, the two heterogeneous systems participating in the trial need to be able to communicate using the same "language" in order for them to communicate effectively. The Protocol provided a simple set of commands with an element of redundancy built in to the basic command structure (simple letter repetition inside the commands).

Because of the complexity and variability of the physical environment, through-water digital acoustic communications is very prone to transmission errors (see, for example Chitre, Shaabudeen, Freitag & Stojanovic (2008)). Consequently, wherever long messages or commands must be sent, sophisticated error correction and/or redundancy schemes need to be adopted. In the present case, however, given the simplicity of the commands and the fact that new commands would be transmitted relatively infrequently, it was decided that simple repetition of entire commands would be used to counter the expected errors in the data (resending entire commands if an acknowledgement was not received in due time).

A simulation trial was designed to test the integration of the two heterogeneous software systems. The testing mimicked the full mission being conducted and used the communications Protocol to send messages between the two vehicles via an internet connection.

This approach enabled the two teams to significantly de-risk the whole programme in an environment where each team had access to a full suite of diagnostic hardware. If the approach proved to be effective it was anticipated that the use of simulation trials, utilising the internet as the underlying communications framework, could become an often-used feature of international collaborative trials.

2.1 Connection

The DSTO and DSO AUVs were developed independently, using different operating systems, different design approaches and were implemented in different programming languages. The nature of the systems was also significantly different. The only effective means of communication between these two vehicles was via low data rate acoustic modem system (the only practical form of underwater communication currently available to underwater vehicles). Data transmission via underwater acoustic communications is subject to high error rates due to the nature of the transmission medium.

A common abstract communication Protocol was developed to provide connectivity between the heterogeneous systems. Each team would independently develop its own implementation of the functionality detailed in the Protocol. The Protocol consists of a number of predetermined text messages that would be sent between the vehicles. The text messages detailed prescribed command and information messages.

For the simulation trial, an internet connection was used to replace the functionality of the acoustic modem. No attempt was made to simulate the speed or lossy nature of the environment, as the focus of this simulation trial was upon the basic functionality of the mission and the Protocol. Simulation of the underwater environmental factors would have been a valid test to perform but given the restrictions in resources it was decided that a simple functionality test was of higher priority. Given the planned vehicle separations for the mission and the anticipated time between transmission of mission-critical commands, it was estimated that – in cases where transmission errors occurred – commands could be re-transmitted several times without compromising the mission.

2.2 The Protocol

This section details the elements of the protocol that were implemented by the individual teams. The specification for the protocol defined both message formats and an agreed time subdivision multiplexing scheme. Only a subset of the commands have been included in this report - sufficient to enable the reader to understand the following discussion. A full set of command definitions are included in Neill (2012). Protocols are reported as defined in the present tense.

2.2.1 Buoy Position (Buoy ↔ Mullaya)

The Mullaya vehicle utilised a self-calibrating, long-baseline navigation system called NavSong. The operational principles of NavSong are described in Neill, Knox, Ward & Lawrence (2006) and the specific implementation of NavSong is described in a companion report (Neill 2012).

The positions of each of the two buoys are used by the NavSong System to triangulate the vehicle's position. The following format was used by Mullaya's control systems to request the buoy's position and receive the consequential position update. This format was specified by the buoy's developer Nautronix (now called L-3 Nautronix).

Description	Code	Notes
Buoy Position Request	$L*Buoy*Z	where *Buoy* is "A" or "B".
MGRS[1]Most Significant	$L*Buoy pos*	where *Buoy* is "A" or "B" and *pos* is the most significant MGRS data.
MGRS Least Significant	$L*Buoy pos*	where *Buoy* is "A" or "B" and *pos* is the least significant MGRS data.

The data in the Most Significant and Least Significant fields is quite distinct (the Most Significant contains 7 Alphanumeric characters whereas the Least Significant is 8 numeric characters only) and thus can be used to differentiate between the different contexts.

2.2.2 Vehicle Position (Mullaya ↔ Meredith)

Each vehicle will report its current position when it is able (i.e. not transmitting Mission Messages), by default sending the Least Significant MGRS field only.

[1]As a descriptive example, the Military Grid Reference System (MGRS) message 48NUG9076044909 can be interpreted as: 48N is a Gridzone designator, where the numeral 48 corresponds to the 6 degree wide UTM zone (of which the Earth is broken down into zones 1-60) and N is a latitude descriptor, defining an 8 degree wide strip in latitude (zone N covers latitude 0-8 degrees North) - UG is a 100 km x 100 km square area located within the 6 degree by 8 degree zone indicated by the 48N prefix - 90760 is a 5 digit east-west coordinate, defining the longitude to an ultimate precision of one metre - 44090 is a 5 digit north-south coordinate, defining the latitude to an ultimate precision of one metre. (Taken from Neill (2012))

Description	Code	Notes
Vehicle Position	PPP *id pos*	where *id* is the vehicle id, *pos* is the position of the vehicle (either Most or Least significant depending on the context).
Most Significant MGRS request	QQQ *id*	where *id* is the vehicle id that the request is sent to, to request the retransmit of the Most Significant MGRS component.

Upon receiving the Most Significant MGRS request, the recipient will transmit the Most Significant MGRS value at the first opportunity. It will then revert back to sending Least Significant MGRS data.

2.2.3 Message Check (Mullaya → Meredith)

During deployment, the Meredith vehicle will be sent a Message Check[2] message from Mullaya. Meredith's acknowledgement of this check demonstrates the two way communications between the vehicles is functional.

Description	Code	Notes
Message Check	CCC	
Acknowledge	AAA CCC	Acknowledge vehicle check

2.2.4 Reacquire (Meredith ↔ Mullaya)

The Meredith vehicle transmits the location of the target to Mullaya. Mullaya acknowledges the location and then replies when it has found the target.

Description	Code	Notes
Hunt Target	HHH *pos*	where *pos* is the MGRS location of the target to hunt
Acknowledge	AAA HHH *pos*	where *pos* is the MGRS location of the target to hunt
Found Target	VVV	

2.2.5 Meredith MMI (Singapore Control Modem → Meredith)

Time was reserved for the Meredith vehicle to communicate back to the Singaporean Mission Management Interface (MMI).

The Meredith vehicle can send a simple status message to provide an update of the vehicle's current status.

[2] The Singaporean team, on occasion, also identified this as the Deploy Command meaning "I have been deployed and am ready to commence my mission".

Description	Code	Notes
Meredith Status	LLL *state*	where *state* is short code indicating the vehicle status.

In the case of a situation where the Meredith vehicle must abort the mission, an emergency message is sent to the vehicle to tell it to abort. The vehicle will acknowledge the receipt of the message and immediately surface and shutdown.

Description	Code	Notes
Stop/Abort	SSS	
Acknowledge	AAA SSS	

2.2.6 Time Division Multiplexing Scheme

While operating within one particular frequency band, an acoustic modem can either send or receive. When the modem is sending then it cannot receive and vice-versa. Thus if there are multiple, frequency matched send/receive acoustic modems using the same trial area, they will need to coordinate their send/receive times so that they are not sending at the same time. For this trial, a Time Division Multiplexing Scheme was chosen, utilising a thirty second cycle, where different acoustic modems were assigned different times to perform their acoustic sends, as detailed in the Table 1. When a modem was not in send mode it was able to receive messages.

Table 1: *Multiplexing windows for the different systems.*

Time (Window)	User
1-4	Meredith's position/message
7-8	Emergency Message from MMI
11-14	Mullaya's position/message
17-21	Navigation Buoy1
24-28	Navigation Buoy2

Each system would have its local clock synchronised to GPS time at the beginning of the trial. Gaps were allowed between the send times to allow for messages to complete before the new sender began its turn.

There was some fluidity to the Mullaya and Buoy times. Rather than the buoys sending their messages at predetermined times they required a prompt from the Mullaya Vehicle (that is, they operated in responder mode). Mullaya sent a message to the Buoys to prompt them to send their position message. Buoy 2 sent its position at a fixed period after Buoy 1. There was, in principle, a potential for this to overrun the designated send period (and end up in Meredith's transmit time) but this should only happen under extreme circumstances (when Buoys are very far from the vehicle or Buoy 1 message is never received).

The experimental design for the cooperative trial virtually ensured such overruns could not occur: 1) the vehicles and the buoys were confined to an area which required two-way acoustic transmission times of less than one second; 2) by design the low power of the acoustic transmissions set an upper operational range which could be accommodated within the defined time slots.

3 Mullaya Mission Management

The Mullaya Mission Manager provided functionality to allow the mission to run through each of its phases automatically. It also provided a User Interface that allowed Supervisors to monitor and, if necessary, intervene in the mission.

3.1 Mission Management System Support Infrastructure

Flex and LiveCycle Data Services were chosen to provide the functionality of the Mullaya Mission Manager. Flex is a modern development environment (based on Actionscript 3) that runs in the Flash Player. It allows rapid construction of Graphical User Interfaces (GUIs) for use in situations such as applied here, where experimental results may determine that ongoing refinement of both the underlying Mission Manager and the GUI are necessary or desirable. Flex is computationally resource efficient and uses a Messaging system that allows near real-time communications between its components. Being web based, it also allows the Flex programs to be run anywhere on the Internet. The Singapore Trial had the potential to be monitored (via Flex components) from Australia in real time, assuming there was appropriate internet connectivity available to the trial site.

The Flex system requires a Webserver to provide some of this functionality and a Tomcat 6 Server was chosen. It also uses a server side system called LiveCycle Data Services (LCDS) to provide the messaging system. The design of the Flex system supports individual programs, all performing small sets of tasks whilst communicating to each other. This allowed for a flexible design that facilitates rapid change.

3.2 Integrating the Mission Manager with the Mullaya Infrastructure

The backbone of Mullaya's internal communications system was based on a system called Centrale (Valentinis, Wharington & Dunn 2005). Connecting the Mission Manager to Centrale proved to be a non-trivial problem. Theoretically, the Monitor could be connected as a Java Client (as Tomcat is a Java based system). Due to resourcing issues this option was not pursued, instead a simple Remote Method Invocation (RMI) Client Server was created to connect a simple Centrale Java client to the Tomcat Server.

3.3 Mission Manager Design Concept

The design of the Mission Manager focused on the mission being conducted as a series of events and states that progressed the mission from the start to the finish. Inputs from the vehicle would be received via Centrale and used as trigger states in the Mission Manager

Variable ID	Variable Name	Publish	value	Type
14	CollAvoidStatus	✓	Off	string
11	ComsStatus	✓	Off	string
17	MRecACOMS	✓	Placeholder	string
18	MSendACOMS	✓	Placeholder	string
8	MeredithStatus	✓	Off	string
6	MeridithReceive	✓	Placeholder	string
5	MeridithSend	✓	Placeholder	string
4	MissionStatus	✓	Placeholder	string
7	MullayaStatus	✓	Off	string
10	NavStatus	✓	Off	string
9	PayloadStatus	✓	Off	string
12	PlatformStatus	✓	Off	string
16	SoSinWater	✓	1491.415	double
15	WSMSeparation	✓	-1	double
13	WaterMStatus	✓	Off	string
2	lat			other
3	long			other
-1	test name		Test content	string

Get Data

Figure 1: *The Viewer provided a link between the Centrale Infrastructure (which contained the Mission Management system) and the Web-based Visualisation software.*

to change states and trigger new events. Figure 1 shows the Mission Manager subscribing to the Centrale Variables via the RMI connection.

The system was designed to inform the human supervisor of the current state of the mission and new information was sent to the Centrale system to update the waypoints (via the system's mission visualisation tool Third Eye (Knox 2012)). The Mission Manager would process information about the vehicle's positions and perform Waterspace Management (essentially informing the supervisor of the distance between the vehicles and flashing an alert if the distance went below a set minimum).

In the Mission Manager (see Figure 2), the communications system and the navigation system where used as triggers to change between mission states.

There were two key states in the mission: Deploy and Reacquire.

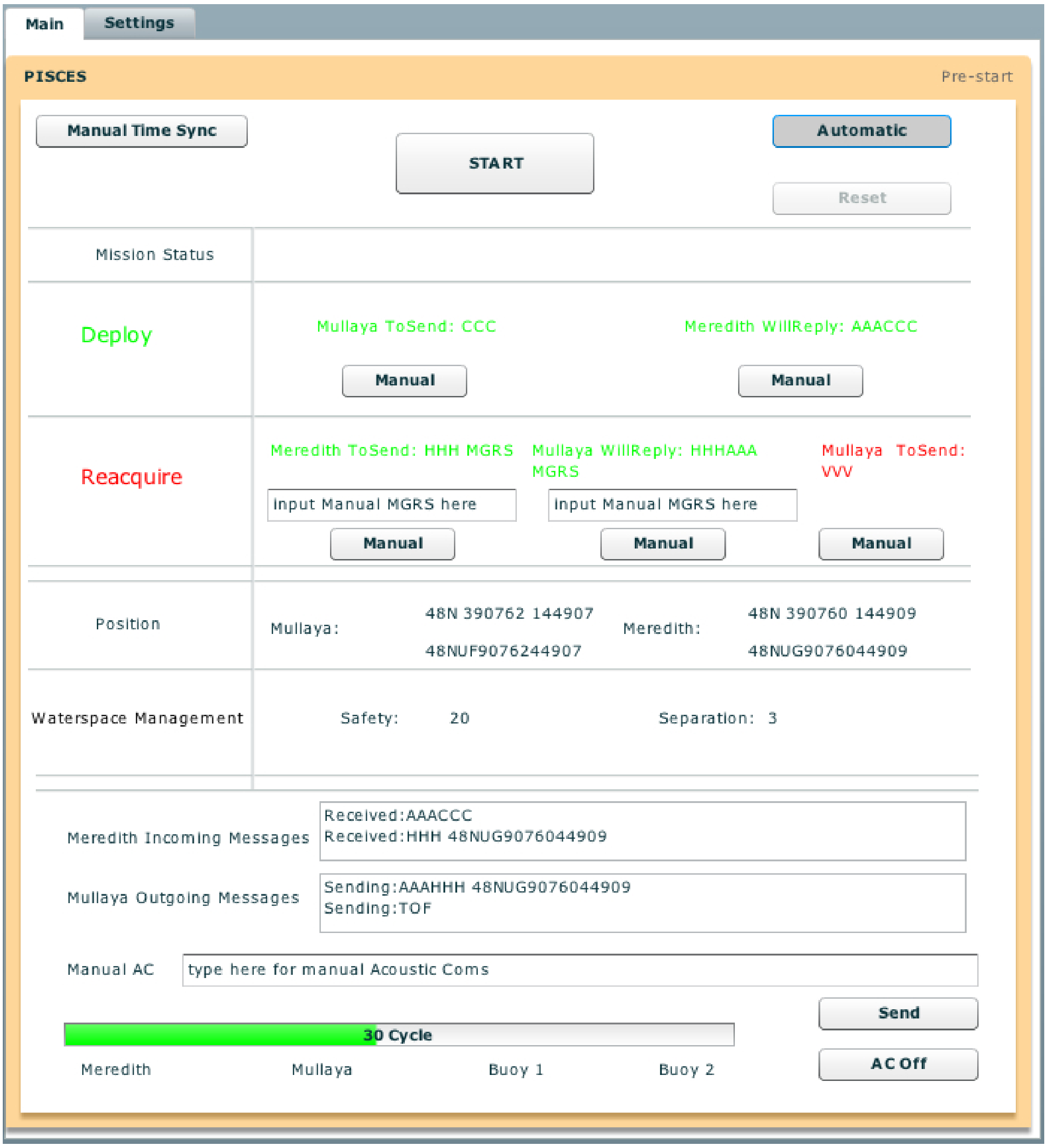

Figure 2: *The Mission Manager Monitor Screen. As the mission proceeds, the system "steps through" the various stages of the mission. The Supervisor had the ability to manually progress the mission if required.*

When the mission started, the Mullaya vehicle would automatically send a Check Status message ("CCC") to Meredith via the acoustic comms system. Meredith would reply with the acknowledgement ("AAA CCC") if it received that message. This demonstrated successful two way communication between the vehicles had been established and ended the Deploy phase.

In an operational context this could correspond to a situation where a second UUV is being introduced to an operational area to undertake a specialist task (such as identification of a mine-like contact or uploading of survey data to a so-called "data truck" - see, for example (Kragelund 2004) for discussion of the latter concept). The vehicle "declares" its presence so that the mission managers of both vehicles can implement appropriate waterspace management procedures.

The second phase, Reacquire, was initiated by the Meredith vehicle sending a Reacquire message with an MGRS coordinate to Mullaya ("HHH" *pos*). Mullaya would then respond with an acknowledgement ("AAA HHH" *pos*). Once Mullaya had reached that position (the distance between the current vehicle position and the coordinate in the Reacquire MGRS was within a selectable, preset tolerance) then it would send a Target Reached message ("VVV") to Meredith.

To support waterspace management, throughout the mission, the vehicles would send each other their current locations when possible. For Mullaya, this was managed automatically by the Mission Manager by taking the current Mullaya position, as provided via Centrale and sending it automatically to Meredith via the acoustic communications system. Meredith's current Position was also received via Mullaya's acoustic communications system and presented to the Mullaya operator (and an update send via Centrale).

Waterspace management was performed by comparing the positions of the two vehicles and notifying the operator of the current separation distance between the vehicles. The Mission Manager Monitor would flash if the separation was below a safety tolerance (10 metres in the case of the Singapore Trial).

Messages between the vehicles were sent strictly according to the 30 second cycle. In the case of the Mullaya Mission Manager, this was performed automatically by queuing messages to be sent and then sending them during the Mullaya's transmission window. If there were no mission messages to be sent then it would send the current position of Mullaya. It also automatically sent a message to the NavSong Buoys to initiate their broadcast.

4 Simulation Trial Design and Implementation

The overall design of the mission management system is illustrated in Figure 3. For the Simulation Trial, an alternative design was developed that provided the functionality to connect the Mission Management Systems over the internet (see Figure 4). This simplified design allowed the Mission Manager developers to focus upon how the mission managment systems interacted without having to consider the entire system. This design provided the capability to robustly test the mission management systems interacting, as it provided the flexiblity to test the system in ways that would have been time consuming and costly if the rest of vehicle's support system (Mullaya and Meredith) were also required when connecting the Mission Managers.

The connection of the two software teams' systems over the internet was implemented using a socket server. The server allowed clients to connect to it and then forward any communications between the clients.

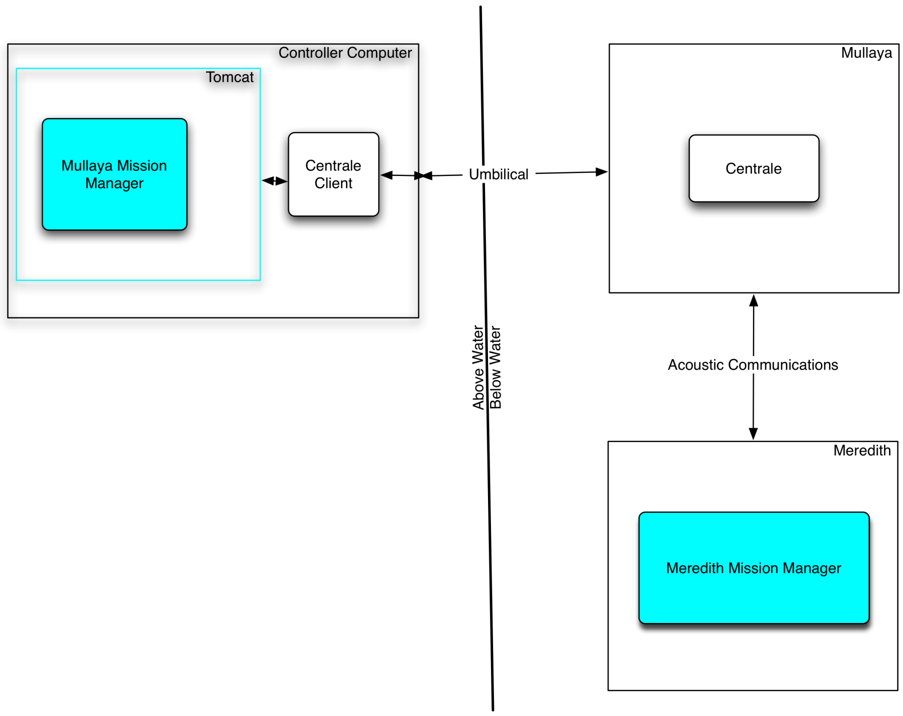

Figure 3: *During the full mission, the two Mission Managers communicate through their respective vehicle's systems via acoustic communications.*

Due to ease of use and security considerations Transmission Control Protocol (TCP) was used as the network protocol. User Datagram Protocol (UDP) is more representative of the acoustic communications (best effort rather than reliable, no check is made to see if all the packets were received) but it was more complicated to implement.

The socket server was set up by DSO in Singapore and DSTO wrote a client to connect to that server. The server provided a service at a specified address at a specified port. DSO provided the details of the server address and the 2200 port was chosen.

The DSTO socket client used Flex to provide the socket functionality and there was some concern that the security implicit to Flash/Flex would become an issue. The security implicit in Flash Player requires that any external data have a crossdomain.xml file at the root of the webserver. DSO was able to provide this functionality but the risk was that, with all of the proxying via the firewalls at both ends, Flash would not permit the socket connection. Because this was an acknowledged risk, ample time was allowed to address connectivity issues if they arose.

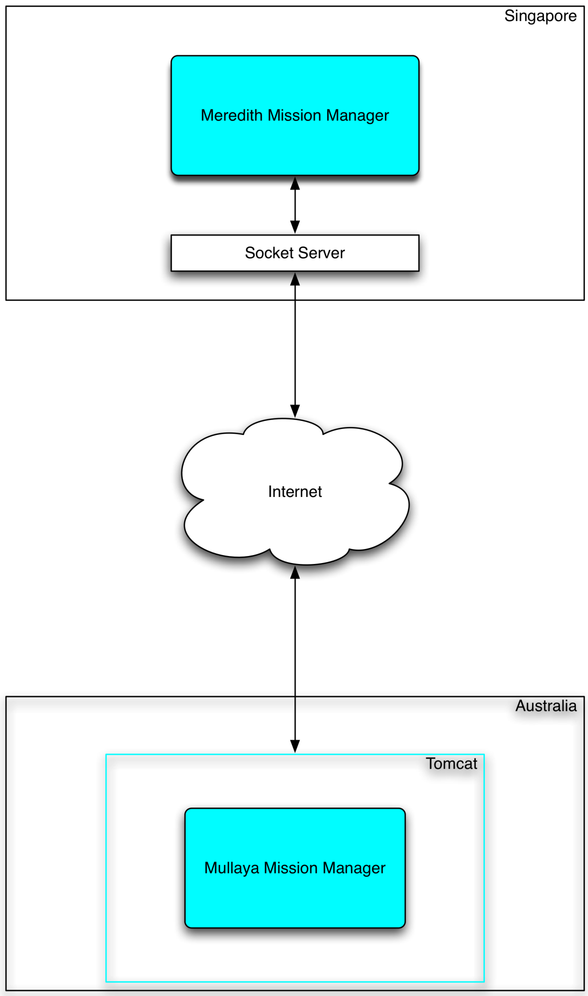

Figure 4: *Simplified Design allowing the Mission Managers to communicate over the Internet.*

5 Simulation Trial Results

The first goal of the simulation trial was to establish basic functionality between the DSTO socket client and the DSO Socket server. An initial connectivity test was scheduled for 14 July 2008. It proved impossible to establish any internet connection with Singapore due to a problem with internet connectivity. Connectivity was restored late on 15 July and the next trial was scheduled for 21 July and the days following.

On the 21 July, internet connectivity was available but the DSTO team was unable to establish a connection between the Socket Server and the Socket Client. It was suspected that this was due to the Flash security issues, but on this occasion a loss of internet connectivity (on the 22nd) due to routing issues prevented further investigation of the problem.

Nevertheless, it was decided that the Flash Security problems were a major risk, so the DSTO client was rewritten to use a JAVA socket client. Flex was then used to connect to the JAVA socket client, via Remote Procedure Calls, instead of behaving as a socket client itself. This worked around the Flash Security problems and provide exactly the same functionality as the original socket client. This was implemented for the next scheduled test on 28th July.

The 28 July test demonstrated, for the first time, a successful connection between the DSTO socket client and the DSO socket server. There were a number of basic data compatibility issues which necessitated modifications being made to the DSTO Java socket client to make the two connections compatible. There was an issue with the reading of the data at the DSO end that meant two copies of the message were received, but this didn't affect the overall functionality.

On the 29th of July a full test of the DSTO Mission Manager was conducted with the DSO Meredith system using the internet connection provided by the socket server. The full simulated mission proceeded as expected and the Protocol was tested. There were a few small problems with the Protocol at both ends but for the greater part the first test of the mission was a success. These issues were addressed in preparation for the next test on the 4th of August. An example of one of the problems encountered was that both teams had assumed they were vehicle 1 and thus both transmitted at the same time. This simple problem was easily sorted out at this stage but would have been very confusing if it had remained undiscovered until the final, at-sea trial.

On the 4th and 5th of August, several runs of the full simulated mission were conducted successfully. All of the functionality of the mission was performed by the actual software that was planned to be used for the final mission. There was still a small issue with DSO receiving duplicate messages but this did not affect the mission. The duplicates were determined to be caused by a problem at the DSO end and were addressed at a later date by DSO.

6 Conclusion

The simulation trial was conducted between 21 July and 5 August 2008. It demonstrated that the two heterogeneous Autonomous Underwater Vehicles' Mission Management systems were able to communicate using the custom designed High Level Protocol. The test highlighted some issues that were easily corrected.

The two teams were able to demonstrate large parts of software functionality without requiring the actual hardware (and the teams needed to support them) to be co-located. Being able to test the software separately from the vehicle's hardware proved to be a valuable experience as it provided the flexibility needed to identify several problems that would have been very difficult to detect in a full system environment. It also provided an opportunity for DSTO and DSO to work together for the first technical part of the Demonstration.

The concept of using simulation based trials for both software and hardware systems is a powerful tool for de-risking international trials. It provides a cost effective means of ironing out many of the complications inherent in complex systems and international collaborations.

The rapid development that was achieved with the Flex based system allowed for flexible and productive software development. It also allowed for more thorough testing. Given the productivity of the Flex based system, it is recommended that the rapid development approach be adopted for creating custom-built software systems rather than trying to build a jack-of-all trades solution. In a research environment, with evolving scope and requirements, the rapid development approach is much more appropriate as it allows for great flexibility and agility. Some savings could be made in designing a framework that would allow reuse of components in subsequent projects.

Thus it is proposed that a "Mission" builder tool be developed to enable missions to be rapidly prepared. Initially this would involve a number of components to be manually constructed as required but would eventually result in a component based mix and match system where non-programmers could quickly construct missions from these components.

References

Chitre, M., Shaabudeen, S., Freitag, L. & Stojanovic, M. (2008), Recent advances in Underwater Acoustic Communications and Networking, *in* 'Proceedings of OCEANS 2008'.

Knox, B. (2012), The Wane Manual, General Document, DSTO-GD-0705.

Kragelund, S. (2004), NPS Center for Autonomous Underwater Vehicle (AUV) Research 2003 Annual Report, Technical Report NPS-MAE-04-002.

Neill, R. (2012), Final report of a cooperative research program undertaken through the Singaporean/Australian Subsidiary Agreement 15: "Cooperative AUV Demonstration", General Document, DSTO-GD-0678.

Neill, R., Knox, B., Ward, T. & Lawrence, C. (2006), The Navsong and Netsong Navigation, Command and Control Communications Demonstrator Project, *in* 'Pacific 2006 International Maritime Conference', pp. 416–425.

Subsidiary Arrangement Number 15, Cooperative AUV Demonstration (2005), To the Agreement between the Government of the Republic of Singapore and the Government of Australia for Cooperation in Defence Science and Technology.

Valentinis, F., Wharington, J. & Dunn, S. (2005), An Experimenation Framework to Support UAV Design and Development, *in* 'Proceedings of First Australian Air Vehicles Conference/Eleventh Australian International Aerospace Congress'.

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA	1. CAVEAT/PRIVACY MARKING

2. TITLE	3. SECURITY CLASSIFICATION
Simulation Trial Results for the Cooperative Autonomous Underwater Vehicle Demonstration (SA15)	Document (U) Title (U) Abstract (U)
4. AUTHOR	5. CORPORATE AUTHOR
Anthony Travers	Defence Science and Technology Organisation 506 Lorimer St, Fishermans Bend, Victoria 3207, Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO–TN–1171	015-9585	Technical Note	May, 2013

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. No. OF PAGES	12. No. OF REFS
2008/1108936/1	07/299	CDS	14	0

13. URL OF ELECTRONIC VERSION	14. RELEASE AUTHORITY
`http://www.dsto.defence.gov.au/publications/scientific.php`	Chief, Maritime Platforms Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for Public Release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SOUTH AUSTRALIA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS

No Limitations

18. DSTO RESEARCH LIBRARY THESAURUS

Unmanned Underwater Vehicle, Simulation, Trials

19. ABSTRACT

A simulation trial was conducted to test the connectivity between the two heterogeneous software systems that were part of a cooperative Autonomous Underwater Vehicle Demonstration being undertaken by the Australian and Singaporean defence science agencies. Since the final, demonstration trial was to be undertaken in Singapore, with no possibility of undertaking repeat trials, a risk management strategy was developed. This included, where possible, using the internet to simulate connectivity and communciation between the two nations' vehicles. The trial, which involved connecting the mission management systems of the two vehicles, and testing the High Level Protocol's functionality for the prescribed Mission, culminated in a successful demonstration of all relevant systems functioning together.