Hierarchical Agent-based Command and Control System for Autonomous Underwater Vehicles

Tan Yew Teck
ARL, Tropical Marine Science Institute
National University of Singapore
Singapore
william@arl.nus.edu.sg

Mandar Chitre
ARL, Tropical Marine Science Institute
National University of Singapore
Singapore
mandar@arl.nus.edu.sg

Prahlad Vadakkepat
Electrical and Computer Engineering
National University of Singapore
Singapore
prahlad@nus.edu.sg

***Abstract*—Over the past decades, the design and development of mission based Autonomous Underwater Vehicle (AUV) continues to challenge researchers. Although AUV technology has matured and commercial systems have appeared in the market, a generic yet robust AUV command and control (C2) system still remains a key research area. This paper presents a command and control system architecture for modular AUVs. We particularly focus on the design and development of a generic control and software architecture for a single modular AUV while allowing natural extensions to multi-vehicle scenarios. This proposed C2 system has a hybrid modular-hierarchical control architecture. It adopts top-down approach in mission level decision making and task planning while utilizing bottom-up approach for navigational control, obstacle avoidance and vehicle fault detection. Each level consists of one or more autonomous agent components handling different C2 tasks. This structure provides the vehicle developers with an explicit view of the clearly defined control responsibilities at different level of control hierarchy. The resultant C2 system is currently operational on the STARFISH AUV built at the ARL of the National University of Singapore. It has successfully executed some autonomous missions during sea trials carried out around the Singapore coastal area.**

***Index Terms*—Autonomous Underwater Vehicle (AUV), Command and Control System, Software Architecture, Hybrid Architecture, Modularity.**

I. Introduction

The Command and Control (C2) system onboard an Autonomous Underwater Vehicle (AUV) is responsible for the high level mission and task planning, as well as the low level vehicle sensors and actuators control to achieve mission objective without human interaction. Despite substantial progress in AUV technologies over the last few years, it still remains a challenge for researchers to develop a C2 system that is robust, adaptive, and able to cope with the changes in dynamic and uncertain environments.

During the early stages, C2 system fell into one of the categories: reactive or deliberative, centralized or distributed, top-down or bottom-up ([4], [5]). However, as AUV technology advances and the need for better functionality and capability arises in the AUV's working environment, a C2 system adopting only one of the architecture mentioned above is no longer able to handle complicated tasks in partially unknown environments. In order to solve this problem, majority of the C2 systems nowadays use a hybrid architecture.

Instead of developing complex, expensive monolithic AUVs for underwater missions, researchers nowadays are moving their attention towards building simpler, low-cost modular AUVs [1]. Modularity in AUV development at software and hardware level provides benefits to the developers and users. Different sections of an AUV can be built separately by different group of developers in parallel and they can also be exchanged depends on the functionalities needed for a particular mission task.

Every changeable section has its own software modules that implement different algorithms depending on the section's responsibilities in the overall AUV setup, and when put together, they form a complete working AUV. However, this plug-and-play capability can only be achieved if the underlying C2 system is capable of adapting to the various AUV configurations for different missions.

In this work, we have developed a generic C2 system for a single modular AUV and it can be naturally extended to be used for a team of collaborative AUVs. The C2 system is flexible enough for changes to be made and extra functionalities to be added depending on the mission tasks assigned and the configuration of each individual AUV in the team. The C2 system has been implemented and tested on the STARFISH AUV (Figure 1). The STARFISH project ([2], [3]) is an initiative at the ARL of the National University of Singapore (NUS) to study collaborative missions carried out by a team of low-cost, modular AUVs. Field trials have been carried out at lake as well as around the Singapore to test the performance and validate the functionalities of the developed C2 system.

Fig. 1. STARFISH AUV in action.

The remainder of this paper is organized as follows. Section II presents a brief discussion of related works. Section III illustrates the architectural overview of the component based C2 system. Section IV shows the results from the lake and

978-1-4244-7107-2/10/$26.00 ©2010 IEEE

sea trials. Finally, section V concludes the paper and discusses future work.

II. Related Works

Mission controllers based on deliberative reasoning [9] are both hierarchical and top-down in its control structure. However, it relies heavily on the information of the world model generated from sensors' input and suffers from computational latency during the sense-model-plan-act process. On the other hand, mission controllers based on reactive reasoning [5] consists of a set of elemental behaviors that defines the AUV's capabilities. Global behavior emerges from the combination of several elemental behaviors activated in parallel when interacting with the world without involving any high level reasoning or re-planning process. This approach may lead the AUV into dead ends while navigating because only the immediate sensing is utilized to react with the environment.

Due to the requirement of self-supervisory, goal-oriented and complex nature of autonomous mission, most of the autonomous robot's mission controllers adopted hybrid approach that integrates different controller architectures to utilize the advantages of some architecture while minimizing the limitations of others [5]. For example, Ridao [6] reported the implementation of three layers of hierarchical mission controller which combined deliberative and reactive control architecture in their semi-AUV, SAUVIM to allow both predictability and reactivity. Elsewhere, Bhattacharyya [7] implemented a hybrid mission controller for AUV simulation for rapid development, while Yavnai [8] developed a reconfigurable mission controller called ARICS that combines the characteristic of both reasoning-based and reactive-reflexive behavior to provide goal-directed planning and good responsiveness. In AUV research recently, developers have started to adopt modular based software developments for the control system for the benefits of component extendability and exchangeability. Examples are MOOS [10] and Neptus [11]. MOOS runs a suite of distributed software modules each communicating through a central database process in order to carry out mission operations while the Neptus is a service oriented distributed architecture which consists of several software components that work together for missions. Although these approaches have produced some cutting-edge control systems for AUV operation, little attention has been put to explore the benefit of integrating them.

Inspired by the command structure in real submarine, we have developed a C2 system that clearly allocates navigational, mission and vehicle tasks into individual self-contained agent components, each with its own responsibilities. The vehicle's C2 tasks are achieved via the interaction among the agent components. These agents are distributed over three different levels in a hybrid control hierarchy where they behave deliberatively at the mission supervisory and command level and reactively at the vehicle and navigational level. Besides that, the C2 system is built on top of DSAAV [12] which allows changeable agent components to be built for specified mission tasks and various AUV setup without affecting the overall C2 system structure.

III. Architectural Overview

A C2 system performs tasks ranging from planning, coordinating, directing and controlling various activities in an AUV. It receives the processed data from the sensors as inputs and then outputs the control commands to the AUV's actuators or control systems to generate desired maneuvering behavior to achieve the mission objective while keeping the AUV safe throughout the mission execution.

In the STARFISH project, we have developed a C2 system based on a hybrid hierarchical model as shown in Figure 2. It adopts a deliberative-reactive architecture and consists of a set of interacting agent components arranged in hierarchical order to depict different level of command responsibilities. Our architecture consist of three levels: Supervisory level, Mission level and Vehicle level. The Supervisory level is in charge of monitoring the high level mission and vehicle status as well as corresponding and sending the information to the operator/mothership. The Mission level is responsible for mission/tasks planning and finally, the Vehicle level carry out the mission tasks and perform obstacle avoidance by utilizing different Sentuators (sensors and actuators) to generate the desired maneuvering behaviors. A communication component (Signaling_Officer) also is designed to provide a communication link with the mothership/operator or with another AUV. Chart Room is the database where a map of the mission areas are stored while Mission Script consists of different mission files identified by their mission numbers.

Each agent component implements its own algorithms and named depending on the assigned tasks to mimic the real control structure in a submarine. All the components are self-contained and have a uniform software interface to facilitate inter-component communication by using a message passing mechanism. The vehicle's C2 tasks are achieved via the interaction and cooperation among the involved agent components.

An agent component's internal activity is governed by a finite state machine which processes its tasks continuously depending on the current state of the component. The transitions between states are triggered by commands from components higher up in the control hierarchy and/or the component's internal events.

The following paragraphs give a detailed description of the responsibilities and tasks of different agent components:

*A. **Supervisory Level: Captain, Chief_Scientist and Safety_Officer***

There are three components under the Supervisory level: the Captain, the Chief_Scientist and the Safety_Officer. Components in this level carry out the main decision making with respect to the mission and vehicle safety. Any one of these components has the right to modify or abort a mission if found necessary.

The Captain component starts, coordinates, oversees and controls the execution of all other components while keeping track of mission progress. Every component in C2 keeps a record of their internal state. When a start command is received from the operator, the Captain checks the internal

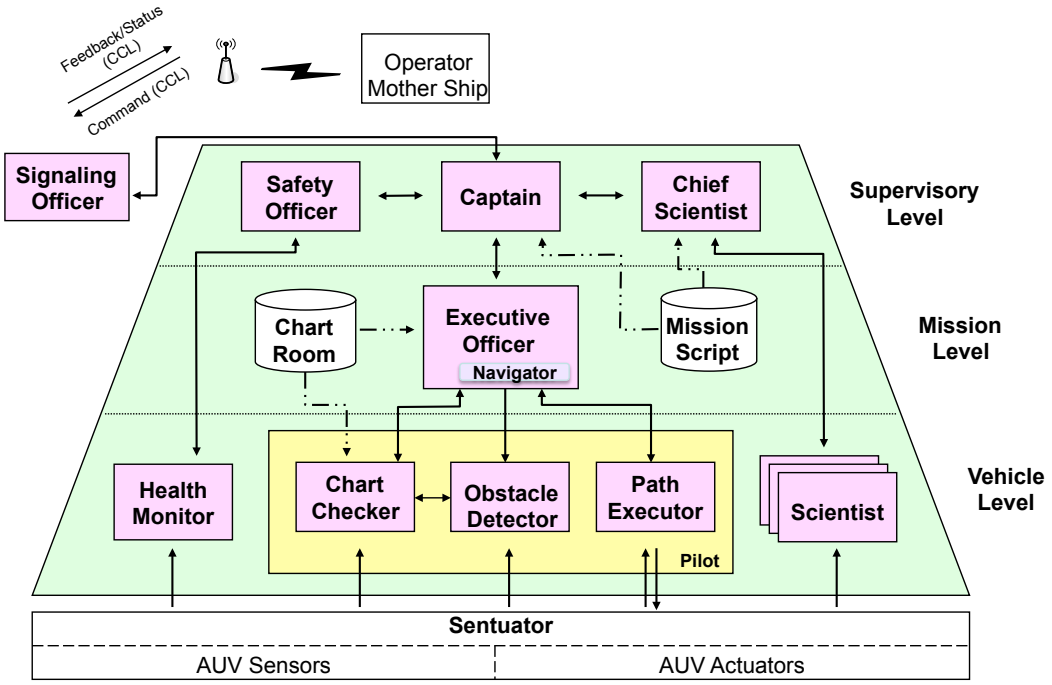

Fig. 2. Hybrid Control Architecture for the AUV.

states to make sure all the components are in "Standby" state before passing the command down to the Executive_Officer. In situations where the AUV encounters problems caused by software errors or hardware failures, the Captain determines the source of the problem and attempts to solve it by either resetting the component's internal states or restarting the component if the first attempt fails. However, if the problem continues to exist, the current mission is aborted and the operator is notified. The decisions are made based on inputs from components within the C2 system and a simple rule-based system with knowledge represented as IF-THEN rules. For missions that involve multiple-AUVs, the Captain is also involved in cooperation and coordination among the AUVs.

The STARFISH AUV can have different payload sections added or exchanged to meet the requirements of mission or to perform underwater scientific experiments. The Chief_Scientist is responsible for command and control of payload sections. It detects payload sections attached to the AUV and based on the mission tasks specified in the mission file, controls and coordinates the Scientist components at the vehicle level. When the AUV is in the mission area, the Chief_Scientist enables the corresponding Scientist components and starts analyzing the obtained information. When necessary, the Chief_Scientist informs the Captain to modify its navigational plan, or to abort the current mission if it fails to perform the assigned tasks.

It is important to ensure an AUV's safety throughout mission execution. For an autonomous mission, the AUV must be able to detect any abnormality that might arise and take necessary steps to make sure that it is not lost during the mission. The Safety_Officer polls data regarding the health conditions of all the devices (sensors and actuators) from Health_Monitor components. It then analyzes the health condition of each device and informs the Captain if one or more malfunction is detected. Besides that, Safety_Officer also looks at sensor data to determine unsafe conditions and perform emergency abort if necessary. This provides a safeguard against agent component malfunction. Furthermore, whenever critical events such as a leak develop, the Safety_Officer will cut off the entire AUV's power without consulting the Captain to prevent the hardware from being damaged.

B. Mission Level: Executive_Officer

The Mission level is responsible for the translation of given mission points into individual tasks and the dissemination of those tasks to the components at the vehicle level for mission execution. The Executive_Officer converts mission points to tasks, plans the task sequence and outputs the task commands as well as mission path for the mission execution. Whenever a mission number and START command are received from the Captain, the Executive_Officer reads the mission file to retrieve the task sequence, mission parameters as well as mission points. The retrieved mission points are then fed to the Navigator for mission path planning. If a feasible path is found between the start and target mission point, the resultant tasks are passed to vehicle level for navigation while the mission points and task sequence are reported back to Captain for mission monitoring.

*C. **Vehicle Level: Path_Executor, Obstacle_Detector, Chart_Checker, Scientist, Health_Monitor***

The Vehicle level consist of five components: Path_Executor, Obstacle_Detector, Chart_Checker, Scientist and Health_ Monitor. They are the reactive components that interact with the vehicle's sensory and actuator level - the Sentuator level. The control and processing at the vehicle level is distributed among its components. Every component has its own set of responsibilities and operates asynchronously based on the commands from the higher level of control hierarchy. Among the components at this level, Path_Executor, Obstacle_Detector and Chart_Checker together play the role of a pilot. They handle tasks ranging from translation of mission tasks into control signal, depth and position keeping, obstacle avoidance and mission chart updating. The Scientist component is responsible to interact with devices in payload section while the Health_Monitor component keeps track of the overall AUV's health condition.

1) Obstacle_Detector and Chart_Checker
   During mission execution, floating obstacles and sea floor are threat to the AUV's safety. Collision with any threats may jeopardize the mission as well as the AUV. The Obstacle_Detector reads the data from Forward Looking Sonar (FLS) to determine the location of objects that exist along the AUV's mission path. Early detection of unknown obstacles lying along the AUV's path is crucial to make sure it has enough time and space to perform the avoidance maneuver.
   The obstacle data are sent to the Chart_Checker which in turns checks for existence of the obstacles in the Chart Room. ChartRoom is the central storage place for the maps of the mission area. Obstacles that are known in the Chart Room will be taken into account when planning mission path. Obstacles that do not exist in the Chart Room are marked and the corresponding location and depth are updated in the maps. Collision checking is then be performed by the Chart_Checker along the mission path. If any of the newly found obstacles lie in the mission path, the Chart_Checker modifies the section of the mission path with simple local reactive avoidance behavior. However, if it failed, the Executive_Officer is notified for mission re-planning. Mission is aborted if the Executive_Officer can not find a new collision free path.
2) Path_Executor
   The Path_Executor is responsible for translating the high level mission tasks into vehicle's low level maneuver control. This component implements a library of basic functions that the vehicle can use to generate the desired maneuvers. One or more basic functions can be invoked concurrently to achieve a high level mission task. This is bottom-up approach where distributed simple vehicle behaviors can be merged to form complex maneuvers. Currently, there are two basic functions implemented to fulfill the mission's requirements: *SteerToXY* and *GoToDepth*. The selection of basic functions to be performed is depends on the current mission task as well as the state/activity of the component's state machine.
   - *SteerToXY:* steers the vehicle from current position to a given mission point in two dimensional plane, (*x,y*). This function takes in the current AUV's bearing and output the bearing offset to adjust the AUV's heading to the target point. Bearing is defined as the direction in which the AUV is pointing while heading is the direction of actual AUV motion over time. When the distance between the AUV's current position and the target point is less than the *WaypointRadius* specified in the mission script, the target is considered reached and new target point from the path will be loaded to continue the navigation. This process is repeated until the AUV reaches the final mission point. Whenever sea current exists during AUV's operation, *SteerToXY* will be operating in current compensation mode. The implementation of navigation with sea current compensation at the reactive level minimize the control signal delay and thus, resulting in better path following behavior.
   - *GoToDepth:* attempts to bring and maintain the AUV to the mission specified depth (z value of the mission) by outputting the depth setpoint for AUV's depth controller. Since STARFISH AUV is not designed for hovering operation or steep angle diving, this behavior has lower priority compared to *SteerToXY* behavior. This means a target point is considered reached when the sensory data satisfy condition set for *SteerToXY* behavior even though it is not at the desired depth.
3) Scientist
   Scientist component is responsible for processing and analyzing the data obtained from payload sensors. To allow exchangeable payload sections, one Scientist component is built per payload section and loaded by the Chief_Scientist depending on the final setup of the AUV. More than one Scientist components can exist in a single AUV if there are several payload sections attached. They are coordinated and controlled by Chief_Scientist component throughout mission. Two payload sections are currently built for STARFISH project. They are the Advanced Navigational payload and Side Scan Sonar payload. The AUV configuration shown in Figure 1 consists of both Doppler Velocity Log (DVL) payload as well as Side-Scan Sonar payload.
4) Health_Monitor
   Health_Monitor component keeps track of the health conditions for all the devices in the AUV including the optional payload section. This is particularly important to make sure the AUV is in its optimum working condition. Every hardware interfacing component (Sentuator component) in the AUV implements an internal health monitoring method which checks the health status of the hardware it attached to. This information is collected pe-

riodically by the Health_Monitor component for vehicle health status updating and forwarded to Safety_Officer for further action.

D. *External Communication: Signalling_Officer*

Signaling_Officer acts as the AUV's external communication node, and is represented outside the overall control hierarchy. This component encodes and decodes the messages among AUVs or between AUV and operator. Besides that, Signaling_Officer is also responsible for updating the operator with the current mission and AUV status periodically. For the STARFISH AUV, the Common Control Language (CCL) is adopted as the message format for acoustic communication [15]. CCL is developed to establish a standard for communication between different agents during operation and provides support for interaction between machines and operators. During mission execution, different CCL messages are sent periodically to the operator for display on the GUI interface. Figure 3 shows a sample CCL message string with their corresponding representations and the screen capture of Command and Control GUI module during operation.

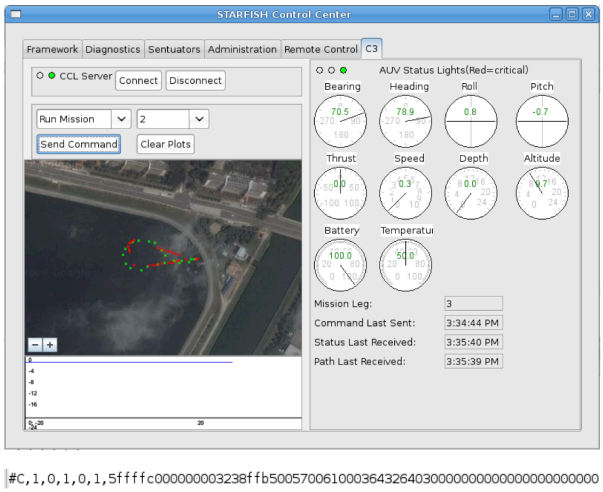

Fig. 3. Screen capture of the C2 GUI interface and a sample CCL message.

E. *Benefits from the architectural design*

The resulting C2 system's architectural design offers many benefits. The hybrid modular-hierarchical control architecture adopts both top-down and bottom-up approach in its control structure. This allows deliberative high level mission control while decouples the low level reactive vehicle control. Moreover, the breaking down of C2 tasks into individual agent components presents an explicit view of the clearly defined control responsibilities at different level of control hierarchy and make it easier for C2 system designers to fine tune the performance of individual agent component.

The implementation of state machine in the agent components facilitate controllability and observability [16] in the control architecture. This allows the agent components at the Supervisory level to monitor and command the behavior of the Mission and Vehicle level components.

The modular software based design of the C2 system on top of the DSAAV [12] makes exchangeable components possible and provides flexibility in terms of software implementation and alternation. Instead of modifying the existing software components, new components with identical interfaces but different algorithms can be built and loaded when necessary. Besides that, the Scientist component can be configured to adapt to the AUV's final payload setup without affecting the overall control structure. This can be done easily by changing the entries in the configuration file.

Since the components are self-contained and the inter-component communication is carried out through message passing, the internal operation of the components do not interfere with each other. This provides fault tolerance if errors occur in one component, as they do not cause the whole C2 system to malfunction.

IV. Field Trials

Several field trials have been carried out at the Pandan Reservoir and Selat Pauh Channel, Singapore. We present data from the autonomous missions carried out for verification of the overall C2 system in a single as well as multi-AUVs scenarios. The trials were conducted using our STARFISH AUV.

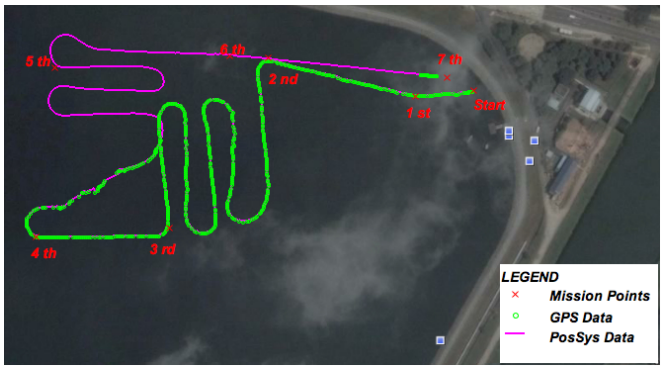

Fig. 4. Plot of AUV's mission path in the lake.

In the reservoir trial, the AUV carried a DVL payload for position estimation and was given a mission to perform lawn mowing navigational pattern simulating surveying mission around an area marked by mission point No. 1 - 7 as shown in Figure 4. The AUV positioning was based on raw GPS data (green dot) or approximated by the AUV's positioning system (PosSys, pink line) when AUV is submerged. The mission was executed autonomously by the onboard C2 system and the DVL scientist agent. The C2 system managed to navigate the AUV through all the mission points while performing surveying pattern at the designated area and surface after the mission was completed. The trial successfully demonstrated the functionality of different control levels in the C2 system in executing an autonomous mission.

In the Selat Pauh trial, the C2 system was extended to perform cooperative mission mentioned in [17]. There were two AUVs involved in the cooperative mission; the Positioning

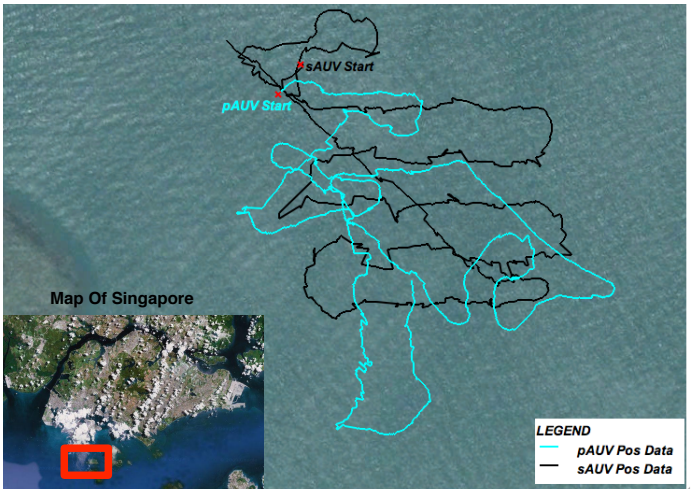

Fig. 5. Plot of Cooperative mission paths around Singapore coastal area.

AUV (pAUV) was equipped with DVL payload while the Surveying AUV (sAUV) was equipped with side-scan sonar payload. During mission, the pAUV (with better positioning capability) is to move in such a way that it can help improve position estimates of the sAUV (only navigate with dead reckoning). In this case, the C2 system of the pAUV has been loaded with cooperative positioning algorithm so that it can calculate the optimum moves.

Figure 5 shows the resulted trajectory of both the AUVs during the cooperative mission. The sAUV planned and executed a lawn mowing pattern around the surveying area while the pAUV calculated, at every time step, the best move to provide position estimates for the sAUV. During the mission, the AUVs had to navigate through strong sea current and surface wind without colliding with each other. The mission was successfully carried out with only minimum modifications to the Captain components of both the AUVs while leaving the rest of the C2 system untouched. This trial clearly shown the flexibility of the C2 system being able to be adapted and extended for different kinds of autonomous missions and its capability in operating under the harsh sea condition.

V. Conclusion

In this paper, a C2 system has been developed for modular AUVs. The focus has been to develop a generic control and software architecture for a single AUV and easily expendable to multi-AUV operations. The proposed control architecture has a hybrid structure. It contains a set of interacting agent components and are grouped into three hierarchical control layers to show an explicit view of the clearly defined control responsibilities at each layer. The control architecture also allows multiple Scientist components to be added to adapt to the AUV's final setup without affecting the overall control structure. The adoption of modular based software design principle for the agents provides flexibility in terms of software implementation or alteration. Since the components are self-contained and communicate by message passing, system malfunction due to error in one of the components can be contained. The resultant C2 system is currently operational and has successfully executed autonomous missions around the Singapore water. Future work includes introducing more computational intelligent techniques in implementing the algorithms in the agent components as well as expanding the C2 system's capability in handling heterogeneous autonomous underwater vehicles.

VI. Acknowledgment

The authors would like to express their gratitude to members of the STARFISH project team for their support and involvement in field trials and experiments. The authors would also like to thank the managements of the Pandan Reservoir, Singapore for allowing us to conduct our lake trials.

References

[1] Sangekar, M., Chitre, M., Koay, T.B., " Hardware architecture for a modular autonomous underwater vehicle STARFISH," *OCEANS 2008* , pp.1-8, 15-18 Sept. 2008

[2] P. D. Deshpande, M. N. Sangekar, B. Kalyan, M. A. Chitre, S. Shahabudeen, V. Pallayil, and T. B. Koay, " Design and Development of AUVs for cooperative missions," in Defence Technology Asia, Singapore, March 22 2007.

[3] STARFISH, "Small Team of Autonomous FISH Project," available at: http://arl.nus.edu.sg/twiki/bin/view/ARL/STARFISH/. Accessed 28 February 2010.

[4] H. Yavuz, A.B., " A New Conceptual Approach to the Design of Hybrid Control Architecture for Autonomous Mobile Robots." Journal of Intelligent and Robotic System, 2002. 34: p. 1-26.

[5] Arkin, R.C.," Behaviour-based Robotics", MIT Press, Cambridge, MA. 1998.

[6] Ridao, P., Yuh, J., Batlle, J., Sugihara, K. " On AUV control architecture," in *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on.*

[7] Bhattacharyya, S., Kumar, R., Tangirala, S., O'Connor, M., Holloway, L.E. " Animation/simulation of missions for autonomous underwater vehicles with hybrid-model based," in American Control Conference, 2006.

[8] Yavnai, A. " Architecture for an autonomous reconfigurable intelligent control system (ARICS)," in Autonomous Underwater Vehicle Technology, 1996. AUV '96., Proceedings of the 1996 Symposium on. 1996.

[9] A.M. Meystel, J.S. Albus, " Intelligent Systems: Architecture, Design, and Control," New York: John Wiley&Sons, 2002.

[10] P. M. Newman, " MOOS - A Mission Oriented Operating Suite," MIT Department of Ocean Engineering, Tech. Rep. OE2003-07, 2003.

[11] Dias, P.S., R.M.F. Gomes, and J. Pinto. " Mission planning and specification in the Neptus framework," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on.*

[12] Chitre, M., " DSAAV - A distributed software architecture for autonomous vehicles," *OCEANS 2008* , pp.1-10, 15-18 Sept. 2008

[13] M. Makatchev and S.K. Tso, " Human-robot interface using agents communicating in an xml-based markup language," in Proceedings of the 2000 International Workshop on Robot and Human Interactive Communication, Osaka, Japan, Sep 2000, pp. 270-275.

[14] Xiaoming Li, Yuzhen Jin, Xudong Hu, " An XML-Driven Component-Based Software Framework for Mobile Robotic Applications," Mechatronic and Embedded Systems and Applications, *Proceedings of the 2nd IEEE/ASME International Conference on* , pp.1-6, Aug. 2006

[15] Eugene Eberbach, Christine N. Duarte, Christine Buzzell, Gerald R. Martel, " A Portable Language for Control of Multiple Autonomous Vehicles and Distributed Problem Solving," Proceedings of the 2nd Intern. Conf. on Computational Intelligence, Robotics and Autonomous System, CIRA'03, Singapore, Dec. 15-18, 2003.

[16] Antonio C. Dominguez-Brito, Daniel h.S., Jose Isern-Gonzalez, Jorge Cabrera-Gamez., " CoolBOT: A Component Model and Software Infrastructure for Robotics. Software Engineering for Experimental Robotics," Vol. 30/2007: Springer Berlin.

[17] Gau Rui, Chitre, M. " Cooperative Positioning Using Range-Only measurements between two AUVs," *IEEE OCEANS 2010* - Sydney, 2010.