Nested Autonomy with MOOS-IvP for Interactive Ocean Observatories

Henrik Schmidt and Arjuna Balasuriya
Department of Mechanical Engineering
Laboratory for Autonomous Marine Sensing Systems
Massachusetts Institute of Technology
Cambridge, MA 02139
henrik@mit.edu

Michael R. Benjamin
Naval Undersea Warfare Center
Newport RI, 02841
Department of Mechanical Engineering
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
michael.r.benjamin@navy.mil

EXTENDED ABSTRACT

The growing desire for autonomy in unmanned marine systems is driven by several trends, including increased complexity in mission objectives and duration, increased capability in on-board sensor processing and computing power, and an increase in the number of users and owners of unmanned vehicles. The MOOS-IvP project is an Open Source project designed and developed in this context. It is an implementation of an autonomous helm and substantial support applications that aims to provide a capable autonomy system out of the box. It also has an architecture, software policy, documentation, and support network that allows this newer generation of scientists, with newer vehicles and mission ambitions, to be nimble to build innovative autonomy algorithms to augment an existing set of capabilities. This paper describes the MOOS-IvP autonomy architecture and software structure, and describes how groups of vehicles, each with different sensors, processing power and communications capabilities, may be combined together to form a nested autonomy architecture with identical core autonomy software running on each platform.

MOOS-IvP is comprised of two distinct Open Source software projects. The Mission Oriented Operating Suite (MOOS) is a product of the Mobile Robotics Group at the University of Oxford, and provides core middleware capabilities in a publish-subscribe architecture, as well as several applications ubiquitous in unmanned marine robotic and land robotic applications using MOOS. Additional MOOS applications, including the IvP Helm, are available in the MOOS-IvP project. IvP stands for Interval Programming and refers to the multi-objective optimization method used by the IvP Helm for arbitrating between competing behaviors in its behavior-based architecture.

The MOOS-IvP software is available on the web via anonymous read-only access, [3]. It consists of more than 120,000 lines of C++, comprising about 30 distinct applications and over a dozen vehicle behaviors. It represents about 20 work years of effort or more from individual contributors. Autonomy configurations and missions in this environment have been tested in several thousands of hours of simulation and several hundred hours of in-water experiments, on platforms including the Bluefin 21-inch UUV, the Hydroid REMUS-100 and REMUS-600 UUVs, the Ocean Server Iver2 UUV, the Ocean Explorer 21-inch UUV, autonomous kayaks from Robotic Marine Systems and SARA Inc, and two larger USVs from the NATO Underwater Research Center in La Spezia Italy.

1.1 Trends in Unmanned Marine Vehicles Relating to Autonomy

The algorithms and software described in this paper have their genesis in unmanned underwater vehicles. Unlike unmanned sea-surface, ground and aerial vehicles, underwater vehicles cannot be remotely controlled; they must make decisions autonomously due to the low bandwidth in acoustic communications. Remote control, or teleoperation, in land, air, or surface vehicles may be viewed as a means to allow conservative, risk-averse operation with respect to the degree of autonomy afforded to the vehicle. In underwater vehicles, similar conservative tendencies are realized by scripting the vehicle missions to be as predictable as possible. Missions typical of early model UUVs were comprised of a pre-planned set of waypoints accompanied with depth and perhaps speed parameters. The on-board sensors merely collected data which was then analyzed after the vehicle was recovered from the water.

Advances in sensor technologies include greater capabilities, at lower cost, lower size and lower power consumption. The same is true for the on-board computing components needed to process sensor data. Increasingly underwater vehicles are able to see, hear and localize objects and other vehicles in their environment and quickly analyze an array of qualities in water samples taken while underway. Likewise, the available mission duration at-depth has grown longer due to improvements in inertial navigation systems, which have become cheaper, smaller and more accurate, and due to improvements in platform battery life. Each of these trends has contributed to making a UUV owner less satisfied with simply collecting the data and analyzing the results in a post-mission analysis phase. The information and analysis are available in-stride, in situ, why not act on that information in-stride to the advantage of the mission objectives? Enter adaptive autonomy.

The chart in Figure 1 below conveys a rough time-line and relationship between the evolution of UUV autonomy capabilities and the evolution of other critical UUV technologies. The notion of adaptive in adaptive autonomy is a sliding scale, and refers to the ability to allow increasing degrees of sensor information to affect in-stride autonomy decisions. On one end of the scale, even a vehicle that deterministically follows a set of waypoints may be adapting its heading decisions based on an INS or GPS sensor. However, sensors that are capable of perceiving qualities about the vehicle's environment, including water quality, bottom type, artifacts, and other moving vehicles, are able to alter the flow of autonomy decisions in a much more profound manner.

The notion of collaboration in collaborative autonomy may be viewed as a sliding scale as well. At one end of the spectrum are vehicles deployed alongside each other, executing a mission independently but each contributing to a joint mission. In this case, the collaboration occurs in the pre-deployment mission planning process. When at least periodic communication between deployed vehicles is feasible, a whole different kind of collaboration is possible, especially when each vehicle is able to adapt components of its mission to both its sensed environment and incoming communications from other vehicles. Advances in underwater acoustic communications (ACOMMS) in terms of reliability, range, flexibility in defining message sets, and bandwidth, have enabled the development of adaptive, collaborative autonomy [14, 15]. This trend also occurs in the context of declining cost and size of commercially available UUVs, making it possible for even medium-sized organizations to own and operate several vehicles.

The MOOS-IvP autonomy architecture has been developed and refined in this context of migration to adaptive, collaborative autonomy. Mission structure is less defined in terms of a sequence of tasks, but rather as a set of autonomy modes with conditions, events and field commands defining the transitions between modes. The modes correlate to a set of one or more active behaviors, where each behavior may be its own substantial autonomy sub-component. An autonomy system that includes the ability to adapt its mission to the environment, other collaborating vehicles, and periodic messages from within a field-control hierarchy will inevitably need to balance competing objectives in a way that reflects a singular mission focus. This paper also discusses how multi-objective optimization is used at the behavior coordination level in the helm to achieve this design objective.

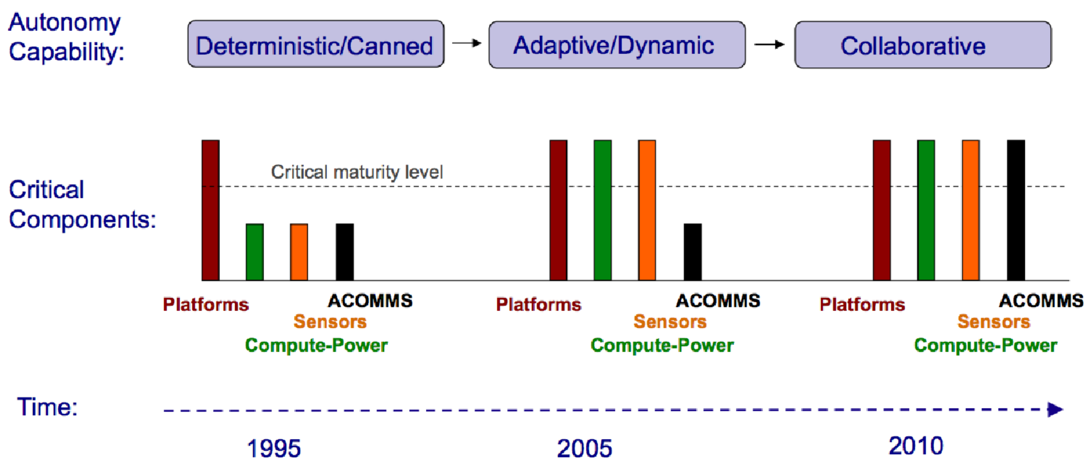

Figure 1. **UUV Technologies and Autonomy:** A rough time-line and relationship between UUV autonomy and other critical UUV technologies. Critical components include (a) the platform itself in terms of reliability, cost, and endurance, (b) on-board computing power and sensor processing, (c) on-board sensors in terms of resolution, size, and cost, and (d) acoustic communications (ACOMMS). Each of these maturing technology trends affects what is possible and desired from the on-board autonomy system. The corresponding trend in autonomy is from deterministic vehicles acting independently, toward adaptive vehicles acting in collaboration.

1.2 The Backseat Driver Design Philosophy

The main idea in the backseat driver paradigm is the separation between vehicle control and vehicle autonomy. The vehicle control system runs on a platform's main vehicle computer and the autonomy system runs on a separate payload computer. This separation is also referred to as the mission controller - vehicle controller interface. A primary benefit is the decoupling of the platform autonomy system from the actual vehicle hardware. The vehicle manufacturer provides a navigation and control system capable of streaming vehicle position and trajectory information to the payload computer, and accepting a stream of autonomy decisions such as heading, speed and depth in return. Exactly how the vehicle navigates and implements control is largely unspecified to the autonomy system running in the payload. The relationship is depicted in Figure 2.

The autonomy system on the payload computer consists of a set of distinct processes communicating through a publish-subscribe database called the MOOSDB (Mission Oriented Operating Suite - Database). One such process is an interface to the main vehicle computer, and another key process is the IvP Helm implementing the behavior-based autonomy system. The MOOS community is referred to as the "larger autonomy" system, or the "autonomy system as a whole" since MOOS itself is middleware, and actual autonomous decision making, sensor processing, contact management etc., are implemented as individual MOOS processes.

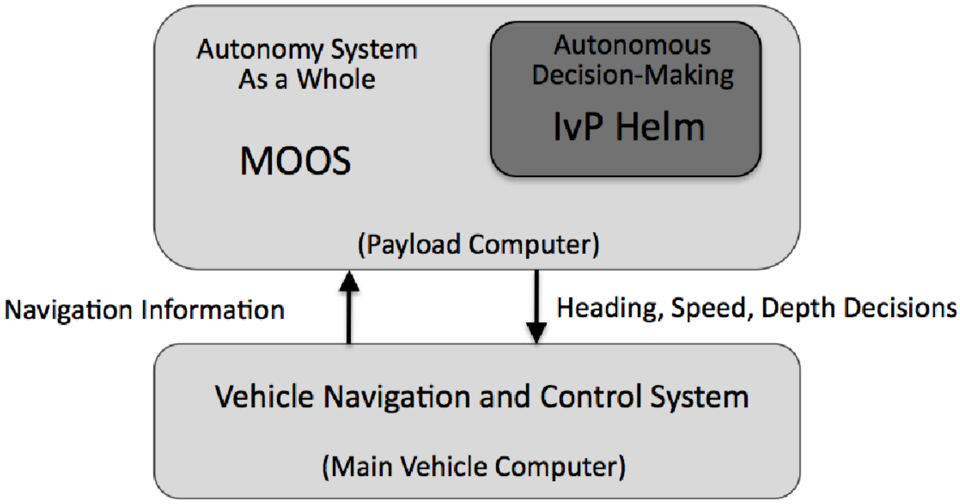

Figure 2. **The backseat driver paradigm:** The key idea is the separation of vehicle autonomy from vehicle control. The autonomy system provides heading, speed and depth commands to the vehicle control system. The vehicle control system executes the control and passes navigation information, e.g., position, heading and speed, to the autonomy system. The backseat paradigm is agnostic regarding how the autonomy system implemented, but in this figure the MOOS-IvP autonomy architecture is depicted.

1.3 The Publish-Subscribe Middleware Design Philosophy and MOOS

MOOS provides a middleware capability based on the publish-subscribe architecture and protocol. Each process communicates with each other through a single database process in a star topology (Figure 3). The interface of a particular process is described by what messages it produces (publications) and what messages it consumes (subscriptions). Each message is a simple variable-value pair where the values are limited to either string or numerical values such as (STATE, ‘‘DEPLOY’’), or (NAV_SPEED, 2.2). Limiting the message type reduces the compile dependencies between modules, and facilitates debugging since all messages are human readable.

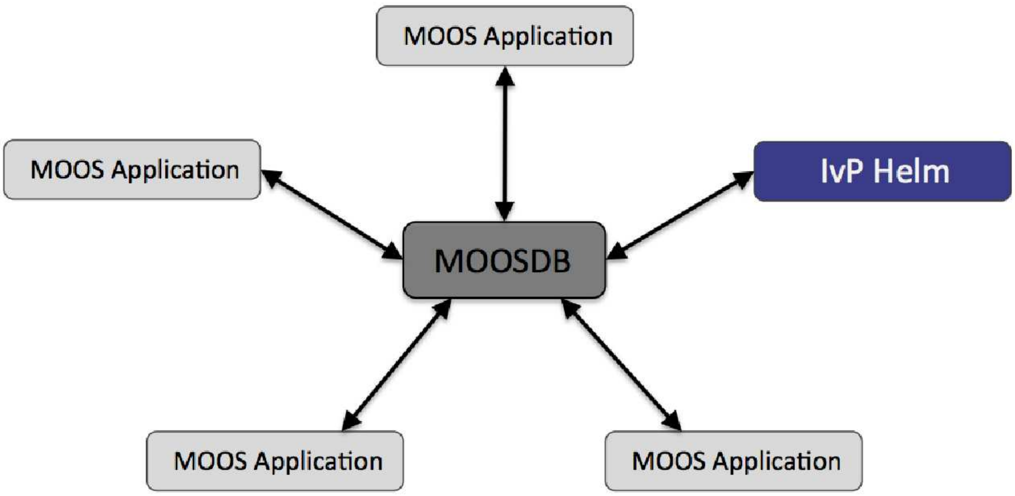

Figure 3: **A MOOS community:** is a collection of MOOS applications typically running on a single machine each with a separate process ID. Each process communicates through a single MOOS database process (the MOOSDB) in a publish-subscribe manner. Each process may be executing its inner-loop at a frequency independent from one another and set by the user. Processes may be all run on the same computer or distributed across a network.

The key idea with respect to facilitating code re-use is that applications are largely independent, defined only by their interface, and any application is easily replaceable with an improved version with a matching interface. Since MOOS Core and many common applications are publicly available along with source code under an Open Source GPL license, a user may develop an improved module by altering existing source code and introduce a new version under a different name. The term MOOS Core refers to (a) the MOOSDB application, and (b) the MOOS Application superclass that each individual MOOS application inherits from to allow connectivity to a running MOOSDB. Holding the MOOS Core part of the codebase constant between MOOS developers enables the plug-and-play nature of applications.

1.4 The Behavior-Based Control Design Philosophy and IvP Helm

The IvP Helm runs as a single MOOS application and uses a behavior-based architecture for implementing autonomy. Behaviors are distinct software modules that can be described as selfcontained mini expert systems dedicated to a particular aspect of overall vehicle autonomy. The helm implementation and each behavior implementation exposes an interface for configuration by the user for a particular set of missions. This configuration often contains particulars such as a certain set of waypoints, search area, vehicle speed, and so on. It also contains a specification of mission modes that determine which behaviors are active under what situations, and how states are transitioned. When multiple behaviors are active and competing for influence of the vehicle, the IvP solver is used to reconcile the behaviors (Figure 4).

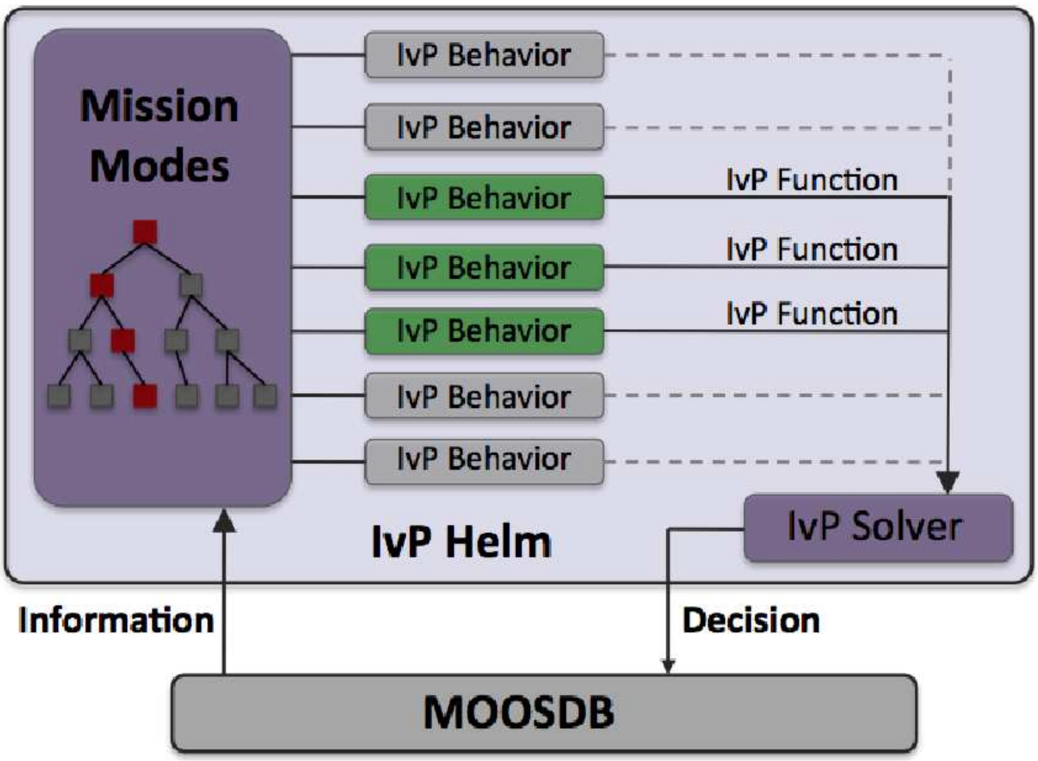

Figure 4: **The IvP Helm:** The helm is a single MOOS application running as the process pHelmIvP. It is a behaviorbased architecture where the primary output of a behavior on each iteration is an IvP objective function. The IvP solver performs multi-objective optimization on the set of functions to find the single best vehicle action, which is then published to the MOOSDB. The functions are built and the set is solved on each iteration of the helm – typically one to four times per second. Only a subset of behaviors are active at any given time depending on the vehicle situation, and the state space configuration provided by the user.

The solver performs this coordination by soliciting an objective function, i.e., utility function, from each behavior defined over the vehicle decision space, e.g., possible settings for heading, speed and depth. In the IvP Helm, the objective functions are of a certain type - piecewise linearly defined - and are called IvP Functions. The solver algorithms exploit this construct to find a rapid solution to the optimization problem comprised of the weighted sum of contributing functions.

The concept of a behavior-based architecture is often attributed to [5]. Since then various solutions to the issue of action selection, i.e., the issue of coordinating competing behaviors, have been put forth and implemented in physical systems. The simplest approach is to prioritize behaviors in a way that the highest priority behavior locks out all others as in the Subsumption Architecture in [5]. Another approach is referred to as the potential fields, or vector summation approach (See [1], [7]) which considers the average action between multiple behaviors to be a reasonable compromise. These action-selection approaches have been used with reasonable effectiveness on a variety of platforms, including indoor robots, e.g., [1], [2], [9], [11], land vehicles, e.g., [12], and marine vehicles, e.g., [4], [6], [8], [13], [16]. However, action-selection via the identification of a single highest priority behavior and via vector summation have well known shortcomings later described in [9], [11] and [12] in which the authors advocated for the use of multi-objective optimization as a more suitable, although more computationally expensive, method for action-selection. The IvP model is a method for implementing multi-objective function based action-selection that is computationally viable in the IvP Helm implementation.

1.5 The Nested Autonomy Paradigm

For large scale ocean monitoring and observation systems, no single unmanned platform has the ability in terms of sensing, endurance and communications to achieve large scale, long endurance system objectives. Even if multiple platforms are applied to the problem, effectiveness may be substantially diminished if limited to a single platform *type*. The *nested autonomy* paradigm, depicted in Figure 5, is an approach to implementing a system of unmanned platforms for large scale autonomous sensing applications. It is based in part on the objective of making seamless use of heterogeneous platform types using a uniform platform-independent autonomy architecture. It also assumes the platforms will have varying communications bandwidth, connectivity and latency.

The *vertical* connectivity allows information to pass from sensors to the on-board sensor processing and autonomy modules, or from each node to other nodes in the cluster, or up to the field operator, and thus forms the basis for the autonomous *adaptive control* which is a key to the capability in compensating for the smaller sensor apertures of the distributed nodes. Similarly, the *horizontal* connectivity forms the basis for *collaboration* between sensors on a node (sensor fusion) or between nodes (collaborative processing and control).

The three layers of horizontal communication have vastly different bandwidths, ranging from 100 byte/min for the inter-node acoustic modem communications (ACOMMS) to 100 Mbyte/sec for the on-board systems. Equally important, the layers of the vertical connectivity differ significantly in latency and intermittency, ranging from virtually instantaneous connectivity of the on-board sensors and control processes to latencies of 10-30 minutes for information flowing to and from the field control operators. This, in turn, has critical implication to the time scales of the adaptivity and collaborative sensing and control. Thus, adaptive control of the network assets with the operator in-the-loop is at best possible on hourly to daily basis, allowing the field operator to make tactical deployment decisions for the network assets based on e.g. environmental forecasts and reports of interfering shipping distributions, etc. Shorter time scale adaptivity, such as autonomously reacting to episodic environmental events or a node tracking a marine mammal acoustically must clearly be performed without operator intervention. On the other hand, the operator can still play a role in cuing forward assets in the path of the dynamic phenomenon, using the limited communication capacity, taking advantage of his own operational experience and intuition. Therefore, as much as a centralized control paradigm is infeasible for such systems, it is also unlikely that a concept of operations based entirely on nodal autonomy is optimal. Instead, some combination will likely be optimal, but in view of the severe latency of the vertical communication channels, the *nested autonomy* concept of operations described is heavily tilted towards autonomy.

The MOOS-IvP autonomy implementation discussed in this paper is situated primary at the node level in the nested autonomy structure depicted in Figure 5. However, aspects of the MOOS-IvP architecture are relevant to the larger picture as well. A key enabling factor to the nested autonomy paradigm is the platform independence of the node level autonomy system. The backseat driver design allows the decoupling of the vehicle platform from the autonomy system to achieve platform independence. The MOOS middleware architecture and the IvP Helm behavior-based architecture also contribute to platform independence by allowing an autonomy system to be comprised of modules that are swappable across platform types. Furthermore, collaborative and nested autonomy between nodes is facilitated by the simple modal interface to the on-board autonomy missions to control behavior activations.

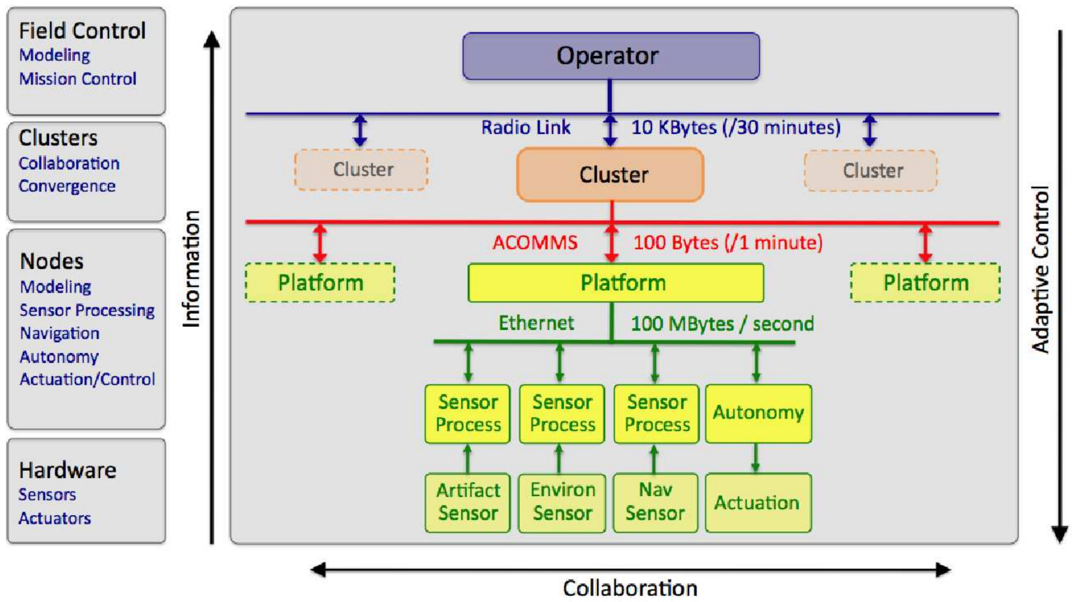

Figure 5: **The Nested Autonomy Paradigm:** Field control operators receive intermittent information from field nodes as connectivity and bandwidth allow. Elements of clusters may serve a heterogeneous role as a gateway communications agent. Likewise, nodes receive intermittent commands and cues from field operators. Node autonomy compensates for and complements the sporadic connectivity to field control and other nodes in a cluster or network of clusters.

References

[1] Ronald C. Arkin. Motor Schema Based Navigation for a Mobile Robot: An Approach to Programming by Behavior. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 264–271, Raleigh, NC, 1987.

[2] Ronald C. Arkin, William M. Carter, and Douglas C. Mackenzie. Active Avoidance: Escape and Dodging Behaviors for Reactive Control. *International Journal of Pattern*

Recognition and Artificial Intelligence, 5(1):175– 192, 1993.

[3] Mike Benjamin, Henrik Schmidt, and John J. Leonard. http://www.moos-ivp.org.

[4] Andrew A. Bennet and John J. Leonard. A Behavior-Based Approach to Adaptive Feature Detection and Following with Autonomous Underwater Vehicles. *IEEE Journal of Oceanic Engineering*, 25(2):213–226, April 2000.

[5] Rodney A. Brooks. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, April 1986.

[6] Marc Carreras, J. Batlle, and Pere Ridao. Reactive Control of an AUV Using Motor Schemas. In *International Conference on Quality Control, Automation and Robotics*, Cluj Napoca, Rumania, May 2000.

[7] Oussama Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 500–505, St. Louis, MO, 1985.

[8] Ratnesh Kumar and James A. Stover. A Behavior-Based Intelligent Control Architecture with Application to Coordination of Multiple Underwater Vehicles. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Cybernetics*, 30(6):767–784, November 2001.

[9] Paolo Pirjanian. *Multiple Objective Action Selection and Behavior Fusion*. PhD thesis, Aalborg University, 1998.

[10] Kanna Rajan, Frederic Py, Conor McGann, John Ryan, Tom O'Reilly, Thom Maughan, and Brent Roman. Onboard Adaptive Control of AUVs using Automated Planning and Execution. In *International Symposium on Unmanned Untethered Submersible Technology* (*UUST*), Durham, NH, August 2009.

[11] Jukka Riekki. *Reactive Task Execution of a Mobile Robot*. PhD thesis, Oulu University, 1999.

[12] Julio K. Rosenblatt. *DAMN: A Distributed Architecture for Mobile Navigation*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1997.

[13] Julio K. Rosenblatt, Stefan B. Williams, and Hugh Durrant-Whyte. Behavior-Based Control for Autonomous Underwater Exploration. *International Journal of Information Sciences*, 145(1-2):69–87, 2002.

[14] Toby Schneider and Henrik Schmidt. The Dynamic Compact Control Language: A Compact Marshalling Scheme for Acoustic Communications. In *Proceedings of the IEEE Oceans Conference 2010*, Sydney, Australia, May 2010.

[15] Toby Schneider and Henrik Schmidt. Unified Command and Control for Heterogeneous Marine Sensing Networks. *Journal of Field Robotics*, In Press, 2010.

[16] Stefan B. Williams, Paul Newman, Gamini Dissanayake, Julio K. Rosenblatt, and Hugh Durrant-Whyte. A decoupled, distributed AUV control architecture. In *Proceedings of 31st International Symposium on Robotics*, pages 246–251, Montreal, Canada, 2000.

「區間規劃模組-任務導向操作套件」
在互動式海洋觀測中之鑲嵌連結式自主結構

Henrik Schmidt and Arjuna Balasuriya
Department of Mechanical Engineering
Laboratory for Autonomous Marine Sensing Systems
Massachusetts Institute of Technology
Cambridge, MA 02139
henrik@mit.edu

Michael R. Benjamin
Naval Undersea Warfare Center
Newport RI, 02841
Department of Mechanical Engineering
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
michael.r.benjamin@navy.mil

摘 要

許多趨勢造成對無人水上載具的自主性產生越來越多的需求，這包括任務目標與執行期間的複雜性提高、艙上感測器處理能力與運算效能的增強，以及無人載具使用者與擁有者的增加。「區間規劃模組-任務導向操作套件」計畫是一項因應這個需求所設計與開發的「開放源碼」專案。這是自主式模組與許多支援應用程式共同執行，以提供即時可用的自主系統。它所具備的結構、軟體授權、文件與支援架構，也能讓新一代的科學家以更新型的載具與任務企圖心，靈巧地建構創新的自主運算系統，以便提升現有的系統的功能性。本研究將說明「區間規劃模組-任務導向操作套件」的自主結構與軟體架構，以及敘述如何將配備不同感測器處理能力以及通訊性能的各種載具群組結合，建構一個在各自平台上執行核心自主軟體的鑲嵌連結式自主結構。

「區間規劃模組-任務導向操作套件」是由兩個不同的「開放源碼」軟體專案所組成的。「任務導向操作套件」是由 Mobile Robotics 集團在牛津大學所開發的產品，能使用發佈-訂閱架構來提供核心中介軟體的功能，而一般無人水上自動機具與陸上自動機具中常見的一些應用程式也都使用「任務導向操作套件」。「任務導向操作套件」的其它應用程式還包括「區間規劃模組-任務導向操作套件」專案中涵蓋的「區間規劃模組」。IvP 指的是「區間規劃」，是一種「區間規劃模組」所使用的多重目標最佳化方法，用來在行為式架構的競比行為之間執行協調。

「區間規劃模組-任務導向操作套件」軟體在網際網路上能透過匿名唯讀存取來找到[3]。它包含 120,000 列以上的 C++原始碼，包括 30 種以上的不同應用程式以及十多種載具行為。它代表許多個別贊助者，總計超過 20 年以上工作時數的成果。在這個環境下的自主組態與任務，都已經通過數千小時的模擬測試與數百小時的水中試驗，操作平台包括 Bluefin 公司的 21 吋無人水下載具、Hydroid 公司的 REMUS-100 與 REMUS-600 無人水下載具、Ocean Server 公司的 Iver2 無人水下載具、Ocean Explorer 公司的 21 吋無人水下載具、Robotic Marine 系統公司與 SARA 公司的無人輕艇，以及義大利拉斯佩齊亞的「北約組織水下研究中心」的兩部大型水面無人載具。