Multiple Vehicle Sampling and Survey for MCM

Sam Smith, Stanley Dunn, Edgar An
Department of Ocean Engineering
Florida Atlantic University
Sea Tech
Dania Beach, Florida 33004
(*report prepared by Edgar An*)
Phone: (954) 924 7231   Fax: (954) 924 7233   E-mail: ean@oe.fau.edu

LONG-TERM GOAL

The general objective is to investigate basic and applied problems associated with the efficacious reconnaissance of littoral waters in support of mine warfare and oceanographic tasks.

OBJECTIVES

This proposal addresses several control system software development tasks necessary to the achievement of more robust and optimized single and multiple vehicle mine reconnaissance capabilities. In particular, this proposal will focus on multiple vehicle co-ordination and control including the infra-structure for more sophisticated mission plans. Part of this infra-structure is the new hierarchical state machine based on high level control software. Another key part of this infra-structure is multiple vehicle hardware in the loop simulation. The idea is to extend the current hardware in the loop simulation to include better multiple vehicle capability so that we might simulate the logic of multiple vehicle missions including communications to reduce at-sea testing costs.

APPROACH

High Level Control

The term Convenient Hierarchical Autonomous State Machine is used to describe a single hierarchical state machine (in which states are hierarchically organized) with additional features. Each CHASM represents a particular kind of high-level controller that is assigned a given task (for example the control of the speed, or the depth of the AUV). CHASM's are built dynamically from some text streams. Each CHASM is under the responsibility of a manager, which runs it on a cyclic basis. Each manager runs a separate process and Inter-Process Communication (IPC) is done through a dynamic shared memory (SM) (see figure). In particular, the Navigator manager has to be entirely rewritten as a CHASM created from a mission plan whose syntax remains accessible to the non-expert users.

Each state of a CHASM represents a mode of the system and is associated with some actions (or behaviors) that are components that have to be fired. We transit from state to state based on the progression of the AUV. The CHASM framework is therefore made of generic components, i.e. components that are both hardware and application independent. Behaviors are application dependent but can be configured for different types of hardware. The CHASM framework supports the plug in of

Report Documentation Page		*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302 Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number

1. REPORT DATE **30 SEP 2001**	2. REPORT TYPE	3. DATES COVERED **00-00-2001 to 00-00-2001**
4. TITLE AND SUBTITLE **Multiple Vehicle Sampling and Survey for MCM**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Department of Ocean Engineering,,Florida Atlantic University,Sea Tech,,Dania Beach,,FL, 33004**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES

14. ABSTRACT **The general objective is to investigate basic and applied problems associated with the efficacious reconnaissance of littoral waters in support of mine warfare and oceanographic tasks.**

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a REPORT **unclassified**	b ABSTRACT **unclassified**	c THIS PAGE **unclassified**	**Same as Report (SAR)**	**9**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

reusable behavior objects that could implement different portions of the command and control logic (i.e. the plug-in of new components that can be triggered by the CHASM).

Instead of developing a solution with built-in behaviors, some work has to be done to provide a solution that offers a modular design to add, remove or modify control features. A modular solution can be assimilated to what is commonly called 'plug-in'. That is, modules that can be easily inserted into an application to offer new features. The design of the architecture was developed so that new functionalities can be added and included into the software without modifying the code or recompiling it. There is no need to stop the mission and the software to update or add a new behavior. There isn't additional installation in order to use the plug-ins that can be downloaded through RF and initialized automatically by the software if they are used in the mission plan for a CHASM manager (i.e. only when needed). Developers only need to know how to design a new behavior - a new plug-in - to extend the application without knowing any deep details about the CHASM architecture.
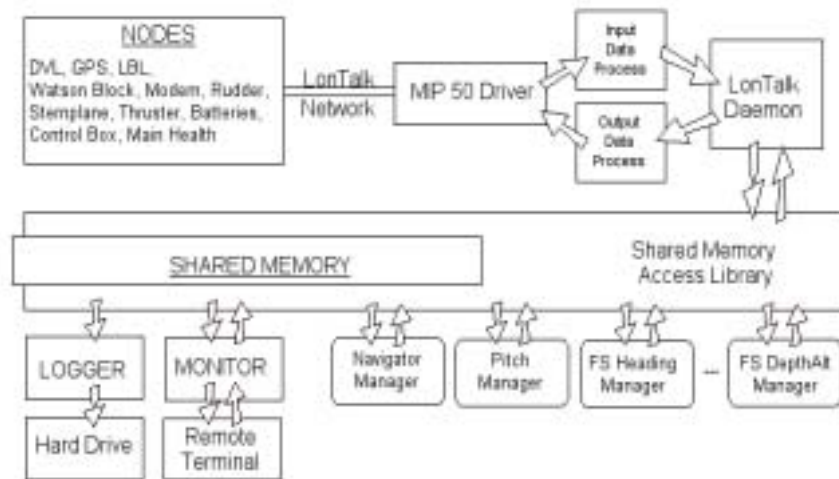
Figure 1: Communication Model of the Morpheus AUV:
Each manager is implemented as a Convenient Hierarchical
Autonomous State Machine (CHASM)

Simulation

One of the modeling and simulation design principles is to de-couple the vehicle dynamics model from the sensor and environment models. By passing information over a global shared memory, these models do not directly interact with each other. As a result, it is inherently flexible to exchange different vehicle geometries, sensor suites and waves characteristics, and this provides a basis for simulating the low-level motion responses for a range of existing vehicles. In our hardware in the loop (HIL) simulation, actual vehicle software runs on a PC104 486 platform with a QNX 4.23 operating system, whereas the simulator runs on another PC platform with a RedHat 6.1 Linux operating system. The simulator generates the relevant sensor data, and sends them to the vehicle, and it replies to the simulator with a set of actuator commands (fins and prop).

To establish an HIL simulation, a bi-directional connection must be established between two computers. We have chosen a network connection based on TCP/IP protocol because it is widely used in industry, can sustain high-speed data transfer, and very portable. Although this protocol is not deterministic and network traffic could introduce unexpected delays, results collected thus far indicate that with our cycle rate and packet size, there has not been any drop-out even the platforms are connected to a LAN. To carry this further, our multi-Hardware-In-the-Loop simulation is to establish a modeling and simulation framework for multiple AUV cooperative missions. Multi-HIL simulation involves several PCs running simulator and several PC104s running their own vehicle software. An important difference between a single and multiple HIL simulation is that in the multiple HIL simulation, all the vehicles need to have their environmental information synchronized. In addition to that, the clocks of each AUV should also be synchronized. This requirement is necessary since the validity and post-mission verification are all based on the timestamps from the individual AUV clocks.

Environment information consists of waves, currents, LBL reply times, and bathymetry although additional information can be included in the future. A data structure that encapsulates all these information is defined which is then translated into a message and broadcasted to all other simulators over the network. In order for multiple vehicles to cooperate with each other, a vehicle needs to know the position, attitude and speed of other vehicles during a mission. Inter-vehicle communication is thus an important attribute for multiple HIL simulation. The physical and data layer for modem communication is currently not available.

WORK COMPLETED

1. High Level Control

Controller Design

We have designed a generic architecture for the different managers. Each manager tailors this generic architecture so that it fulfills its specific needs. Most of the objects can be easily identified in the CHASM formalism.

A CHASM consists of a tree structure of states, with one or several root states at the top. Actions are associated to these states. An entity has to manage this set of states and to trigger the transition and the different actions associated to the states in the current active path, on a regular basis. The CHASM manager is this entity and is responsible of initializing the CHASM, managing the CHASM tree, and keeping track of the active state path. Indeed, a CHASM manager has to run the CHASM structure at a regular frequency to ensure the good execution of the mission. Consequently, the different objects of the system are the CHASM manager, the parser (or Loader) that loads the CHASM, the actions, the states, the transitions, etc.

State Data Structure

It is much more convenient to generate all the states at once, and then to transit from one to another. There are several reasons to this:

• To deal with a Hierarchical Finite State Machine, not just one state is active at any one time but a path of states. A check of the transitions of the state with a higher level of priority is done first.

• There are special relations between the states: parent/child, default next, default child, etc.

• Do not want to access a file or a database every time we make a transition, it's time consuming. All the objects are loaded at init-time. States are generated by the Loader. The role of the Loader is to parse the mission plan and to generate all the objects of the CHASM architecture.

Analysis Class Diagram

The CHASM manager manages the CHASM data structure (a forest of state objects) and hence acts as a container for the different states of the system. A state may have several sub-states to allow for the hierarchy property, or can be a leaf state. Only certain states are active in the CHASM at one given moment. They belong to the current active path, which starts from one root node and ends with a leaf state. Some actions are associated to the states, and they are run as long as the system is in this state. There are different types of actions with a different time model, so all the actions associated to the current active states are not fired every time the CHASM manager is fired. *Producer* represents any entity we can have data from, mostly sensors or payloads (INS). The role of the INS is to merge many sensor data and to synthesize different parameters to provide the position and the speed of the vehicle. This is done at the low-level in our system, and data are emitted by one of the nodes in the network. *Consumer* represents an entity we can send data to, mostly actuators, but also low-level controllers that are connected to the network. The Actions get the data from the sensors, from the INS system, or from the results of other actions and do some processing, before publishing the results, either to some consumers or to other actions. The class diagram allows us to make appear the 'one writer-many readers rule'.

Transition Design.

Evaluating a transition in the CHASM formalism is done in two states:

Step 1 of evaluating whether a transition should be made consists in defining some test Actions and running them. They define some criteria and test some variables in shared memory to see if they satisfy these criteria. The Test Action objects contain the extrinsic data for the test, i.e. the criteria value, and the tolerance value, and the Test Action Handler is the object that performs the actual test and knows how to access shared memory to do it. There are a lot of different criteria that can be tested: pitch, altitude, etc. There are therefore as many Test Action Handler subclasses as there are criteria that can be tested. When a test Action is triggered, it fires the associated test Action Handler passing it the parameters for the test and this one performs the test. The result is Boolean which is stored in the Action Handler object.

Step 2 of evaluating whether a transition should be made is to evaluate the transition condition. The transition condition is a Boolean expression that can use combinations of several criteria. The Transition objects of our system must have a mean to obtain the result of the requirements that appear in the Transition Condition. They must have a reference to the appropriate Test Action Handlers. This design is very flexible and scalable because it allows de-coupling functions of the CHASM. Transition conditions can be composed of as many criteria to test as wanted.

2. Simulation

A thread called environment server is established when the Multiple HIL simulation starts. Other simulator that joins the simulated mission later will create a thread called environment client. This client connects to the server and starts receiving data from it. Each simulator has its own configuration file that specifies a set of shared memory environment variables needed, and the server will respond accordingly to fit different clients' needs.

Robustness and transparency are useful design attributes for the Multiple HIL implementation. These attributes refer to the stability of a simulated mission when one or several simulators fail unexpectedly, and the flexibility to initiate multiple server/client sessions. As a result, the following criteria were established:

- An HIL simulation joins the Multiple HIL simulation as a client first, if it is the first HIL simulation, it will re-configure itself as a server.

- If the environment server terminates unexpectedly, another client will automatically change itself to a server. If more than two clients remain, the client with the highest default priority will become the server. This priority information is defined in the configuration file.

During a simulation, each HIL will have its local clock or time. The local time with which the environment server is associated is also called the global time. During the environment synchronization, the global time is sent to all the clients so that their local time based on the received global time can be updated. This is how the clocks in multiple HIL simulation are synchronized. Currently, the environmental information synchronization and clock synchronization was implemented, and several HIL simulations can run different missions in the same environment simultaneously.

RESULTS

High Level Control

The CHASM architecture has not been implemented on our vehicles.

Simulation

The multiple HIL simulation has been implemented, and has been tested with multiple clients running on separate PCs. The robustness of the client-server architecture was demonstrated.

IMPACT/APPLICATION

This project will foster synergistic development of advanced mine reconnaissance operations with multiple vehicles and further the development of high level tactical command and control.

TRANSITIONS

None

RELATED PROJECTS

- Very Shallow Water Mine Reconnaissance with Multiple AUVs
- Node-based Adaptive Sampling and Advanced AUV Capabilities
- Sampling and Survey with AUVs in Adverse Weather Conditions

REFERENCES

[1] A. R. Girard, "A Convenient State Machine Formalism for High Level Control of Autonomous Underwater Vehicle", Thesis, Florida Atlantic University, Boca Raton, May 1998.

[2] B. F. Dupire, "Pattern-Oriented Design of a Dynamically Reconfigurable Software Architecture for AUV High-Level Control", Thesis, Florida Atlantic University, Boca Raton, August 2001.

[3] H.Lin and S.Hsu, "Development and Simulation of Vertical Profiling Capability for FAU Autonomous Underwater Vehicle", Thesus, Florida Atlantic University, Boca Raton, Mars 1998.

[4] K. Ganesan, S. M. Smith, K. White, T. Flanigan, "A pragmatic Software Architecture for UUVs", IEEE symposium on Autonomous Underwater Technology, 1996.

[5] Girard, A.R., Smith, S.M., Ganesan K., "A Convenient Format for Discrete Event Control of Autonomous Underwater Vehicles", IARP Workshop on AUVs for shallow water and Coastal Environments, February 17-19, 1998, Lafayette, LA.

[6] F. Bushmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, "Pattern Oriented Sotware Architecture: A system of patterns", 1996, Wiley & Sons, ISBN 0-471-95869-7

[7] D. Schmidt, M. Stal, H. Rohnert, F. Bushmann, "Pattern-Orientted Software Architecture, volume 2: Patterns for concurrent and networked Objects", 2000, Wiley & Sons, ISBN 0-471-60695-2.

[8] E. Gamma, R. Helm, R. Johnson, J.Vlissides, "Design Patterns", 1995, Addison- Wesley, ISBN 0-201-63361-2.

[9] P. Dhaussy, J. Champeau, R. Moitie and A. Pringent, "Object-oriented and Formal Methods for AUV development", proceedings of the MTS/IEEE Oceans 2000 conference, volume 1, pp73-78, September 11-14, 2000, Providence, Rhode Island.

PUBLICATIONS

P.E. An, M. Dhanak, N. shay, J. Van Leer, Sam Smith, Coastal Oceanography Using a Small AUV, submitted to Journal of Atmospheric and Oceanic Technology, Sept 1998

A. Healey, P.E. An, S.M. Smith, "Multi-Sensor Asynchronous Extended Kalman Filtering for AUV Navigation, Submitted to the IEEE Transactions on Oceanic Engineering.

GS Rae, D. Kronen, S. Smith, FAU Technical Report, Docking Navigation Experiment. September 1999

S. Smith, L. Marquis, S Snowden, FAU Technical Report, MCM Experiment Dec 1998.

S. Smith, L. Marquis, S. Snowden. FAU Technical Report, AUV Fest 1998,

S. Smith, S. Snowden, FAU Technical Report, ACOMS Hawaii Experiment

J. Jalbert, FAU Technical Report, Co-Bop Experiment

J. Jalbert, FAU Technical Report, 4D Current Experiment

J. Jalbert, FAU Technical Report, Adverse Weather Experiment.

Feijun Song, Samuel Smith, Charbel Rizk, "A General Cell State Space Based TS Type Fuzzy Logic Controller Automatic Rule Extractions and Parameter Optimization Algorithm", IEEE Industrial Electronics Conference, IECON 99, San Jose CA November 29 - December 3rd 1999.

Feijun Song, Samuel Smith, Charbel Rizk, "A Fuzzy Logic Controller Design Methodology for 4D Systems with Optimal Global Performance Using Enhanced Cell State Space Based Best Estimate Directed Search Method", 1999 IEEE International Conference on Systems Man and Cybernetics , Tokyo Japan, October 12-15, 1999

Feijun Song, Samuel Smith, Charbel Rizk, "Reducing Memory Requirement of Cell State Space Based Fuzzy Logic Controller Design Approaches Using K-d Trees" 1999 IEEE International Conference on Systems Man and Cybernetics, Tokyo Japan, October 12-15, 1999

Xiaohong Yuan, K. ganesan, Matthew Evett, Samuel M. Smith, "Providing Real-time Data Trajectory Access in Autonomous Underwater Vehicles", Proceeding IEEE Oceans 99 Conference, September 13-16, 1999, Seattle WA

Xiaohong Yuan, K. Ganesan, Scott Snowden, Samuel M. Smith, Matthew Evett, "Mission Command Macros For Autonomous Underwater Vehicles, Proceeding IEEE Oceans 99 Conference, September 13-16, 1999, Seattle WA

Alexandre Delarue, Samuel Smith, Edgar An, "AUV Data Processing and Visualization Using GIS and Internet Techniques", Proceeding IEEE Oceans 99 Conference, September 13-16, 1999, Seattle WA

M. Dhanak, E. An, K. Holappa, S. Smith, "Using Small AUVs for Oceanographic Measurements" Proceeding IEEE Oceans 99 Conference, September 13-16, 1999, Seattle WA

G. Grenon, E. An, S. Smith, "Enhancement of the Inertial Navigation System" Proceeding IEEE Oceans 99 Conference, September 13-16, 1999, Seattle WA

D. Mallinson, D. Naar, A. Hine, S. Smith, S. Schock, d. Wilson and G. Gelfenbaum, "Sea.oor Mapping and Target Identi.cation Using AUVs: Applied AUV Experiments, UUST 99 Durham NH August 23-25, 1999.

Feijun Song, Samuel Smith, Charbel Rizk, "Optimized Fuzzy Logic Controller Design for 4D Systems Using Cell State Space Technique with Reduced Mapping Error", IEEE Conference On Fuzzy Systems, FUZZ-IEEE 99 Seoul Korea, August 23- 25 1999

S.M. Smith, D. Kronen, R. Dunn, J. Whitney, J. Frankenfeld, E. An, T. Pantelakis, A. Burns, E. Heatzig, "An Ultra Modular Plastic Mini AUV Platform for VSW Mine Reconnaissance". SPIE Aerosense, 1999, Orlando Florida April 5-9, 1999.

S.M. Smith, P.E. An, R. Christiansen, J. Kloske, S. Snowden, D. Kronen, L. Marquis. "Results of an Experiment Using AUVs for Shallow Water Mine Reconnaissance, SPIE Aerosense, 1999, Orlando Florida April 5-9, 1999.