An Environment for High-Level Multiple AUV Simulation and Communication[1]

Steven G. Chappell
Rick J. Komerska
Autonomous Undersea Systems Institute
86 Old Concord Turnpike
Lee, NH 03824
(603) 868-3221

http://www.ausi.org
chappell@ausi.org
komerska@ausi.org

Introduction

A strong component of current underwater robotics work lies in the realm of multiple vehicles and instrument platforms (VIPs). Successful fielding of functional multiple VIP systems will require that we first gain experience with the issues pertaining to the interactions between the participants, who must cooperate in order to accomplish complex and dynamic long term missions. Many current simulation tools are directed towards investigations into the motion characteristics of vehicle body styles and their propulsion systems. While these tools are essential in the design of vehicle bodies, they do not directly address the issues of cooperative behavior among a *team* of VIPs. There is a scarcity of tools geared towards the needs of researchers working with multiple cooperating VIPs[2]. To answer this need, we have developed the Cooperative AUV Development Concept (CADCON).

At the heart of the CADCON idea is a simulation harness expressly designed to provide a shared environment for multiple distributed interacting VIPs. Fundamental to the CADCON idea is the notion that this environment be open and flexible, allowing it's use by a wide range of research initiatives. Understanding that no single simulation harness could capture the full fidelity of real open water environments, nor the complexity of every sort of VIP that might participate, we have focused our efforts on designing CADCON to primarily address one important aspect of multiple VIP systems. That is, dealing with those issues associated with the higher level interactions among multiple heterogeneous participants; be those participants real VIPs, simulations of VIPs, or even human users. To enable this, we have

---

[1] Adapted from Chappell *et al.*, [1999].

[2] For an exception, see Turner & Turner, [1998], which describes a high level multi-agent simulator called CoDA.

endeavored to streamline the complexity and the costs associated with a remote participant's use of CADCON. In particular, we have implemented the system's various components on ubiquitous hardware (Pentium class computers). The architecture is based upon a client/server model and communication is done using TCP/IP networking over the Internet. Groups can use the existing client programs (a VIP simulator and a 3-d visualization application) which are Win32 applications, or make use of the open application level protocols established for CADCON to write their own clients. This non-reliance on exotic hardware and/or proprietary communications protocols allows other groups to leverage the simulator's utility in their own research.

Background

For the most part, "traditional" underwater vehicle simulators deal with the complex hydrodynamics of specific vehicles operating in simple idealized environments. In these simulators, emphasis is placed on the fidelity of the vehicle model to the real thing and less so on the "situation" model. The scope of those *hydrodynamic* simulators is at the small scale physics level of the dynamic interactions among the parameters of vehicle shape, vehicle mass, thrust vectors, inertia, friction, medium characteristics, and the like.

The CADCON simulator is *not* of this variety; instead, it stresses the complexity of a *situation model* over that of vehicle dynamics. Its scope is at the level where the dynamics of agent movement are of secondary importance to the behaviors exhibited by the participating agents[3]. Here we are more concerned with how agents interact with each other and some features of their environment than how individual agent bodies interact with the medium in which they move. This means that the system simulates only the larger scale physics of those agents/craft. The chief feature for enabling agent interaction is the simulation's agent-to-agent communication mechanism, which provides a communication network at an idealized high level.

The ideas and features supported in CADCON have their roots in the Environ Simulator developed by AUSI in the 1980's [Momenee, 1987]. CADCON differs from this previous work in that it:

- is more modular,
- is distributable across networks (communicating via sockets),
- simulates a variety of participants instead of a single AUV type (AUSI's open frame EAVE),
- simulates tens of participants rather than only three or four,
- separates data generation from data display,
- is more interactive,
- concentrates on simulating complex inter-participant activities,

---

[3] The wide range of participant heterogeneity pushes us to use the broader term *agents* in describing CADCON participants.

- provides a cyberspace for real vehicles that are undergoing development and testing, thus enabling hardware-in-the-loop experiments,
- simulates non-numerical aspects of situations (such as participant organizations),
- will leverage higher levels of simulated "realism" via runtime links to Internet available oceanic data bases and models,
- will serve as an Autonomous Oceanographic Sampling Network (AOSN) [Curtin *et al.*, 1993] mission rehearsal, test, and display tool.

In this paper, we describe the various components of CADCON, it's environmental features, the agent models employed to represent participants, the logistics of connecting clients to the CADCON server, and mechanisms of agent interaction. In particular, we report our use of CADCON to support several AUSI projects, including the development a common control language for VIPs, support for our Solar-powered AUV (SAUV) development, and as a debugging tool for a unique magnetic navigation algorithm.

Top Level Features

As implemented, there are currently three major components in the CADCON facility; the *Environment Server* and two clients: the *AUV Simulator Client and* and the *Visualizer Client*. Figure 1 diagrams the distributed nature of these components. The two clients actually represent particular implementations based upon an established CADCON application communication interface; the user is free to implement their own instances as long as they adhere to this interface protocol. This strategy allows users to develop clients tailored to their own research, as well as potentially leveraging their own existing tools in CADCON .

Also shown in Figure 1 are a pair of clients we have in development; an *Autonomous Systems Monitoring and Control* (ASMAC) client and an *External Model Interface* (EMI) client. Since these clients are not yet fully developed, they are briefly discussed in the Future Work section.

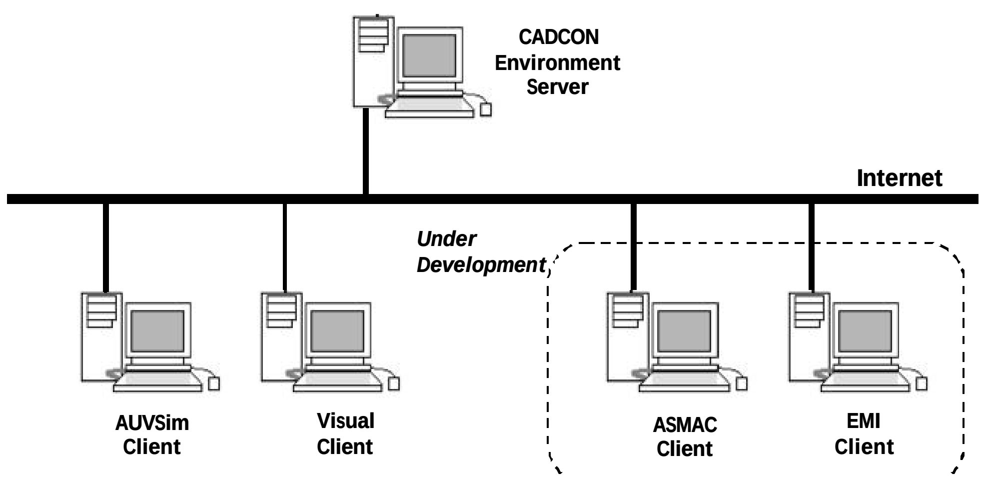

Figure 1. CADCON Components

Environment Server

The Environment Server forms the core of the CADCON facility. Its purpose is to generate the cyberspace in which CADCON (e.g. AOSN) participants can operate and interact. The cyberspace generated by this component is a volume of water bounded by or containing one or more of the following features:

- a topographically interesting bottom,
- a topographically interesting ice cover,
- a simple thermocline model,
- a simple salinity model,
- a simple water current model,
- a variable number of agents,
- a variable number of inanimate objects,
- a magnetic dipole source [Lu *et al.*, 1999].

These features are controlled via a set of configuration files. The Environment Server operates only when it is needed. It is triggered by a *daemon*; that is, a process that runs on its host machine "in the background" and simply waits for connection requests. When the first such request is detected, the daemon spawns a server process, which accesses the scenario configuration files, builds the specified environment, and then places the new participant in that environment. Subsequent connection requests simply place those new participants in the already existing scenario. When the last participant leaves, a timer is set, and when that expires, the server takes itself down to its dormant state, where only the connection daemon remains to wait for a new connection request. This scheme allows the server to have minimal impact on the resources of its host machine while it is not being used. By running more than one instance of the connection daemon (monitoring different sockets) and having multiple sets of configuration files, the host machine can simultaneously provide many different and separate scenarios.

Participants are modeled as simple geometric shapes which are moved around using a Newtonian motion model. These shapes can collide with each other, the bottom, the ice cover, and the inanimate objects.

One of the chief functions of CADCON is to provide a means for experimenting with high level AOSN participant interaction. This is accomplished with an agent-to-agent communication scheme. This machinery allows any participant to send communication strings to any and all other participants. This could be common VIP commands [Komerska *et al.*, 1999a] or any other message which agents or their developers may find useful to communicate among themselves.

Given the autonomous nature of the Environment Server, it has been designed to leave a fair amount of evidence of its functioning in a set of log files. These files capture administrative type information (user account names, remote hosts names, times, etc) as well as agent specific traces of that agent's movement through the scenario.

The Environment Server is currently implemented in C, and runs at AUSI on a 300 MHz Pentium II computer hosting Linux 2.x.

“AUVSim” VIP Simulation Client

The AUVSim program was written by AUSI to simulate a VIP within the CADCON environment. It forms the basic platform for which we represent vehicles and test high-level multiple agent communication and control in CADCON. AUVSim builds upon the VIP client interface protocol, which allow users to represent their VIPs to other CADCON clients distributed across the Internet. These VIP representations, be they simulations or proxies for actual situated hardware platforms, can then communicate and participate in joint mission scenarios within the same simulated underwater context.

With AUVSim, a user can configure a local simulation of a particular kind of submersible vehicle on their own workstation. Available body/control types include an open space-frame, torpedo, wing/cylinder, bottom crawler and mooring. In addition to body type, the user also chooses other subsystem options, including available sensors. This sensor selection is registered with the Environment Server, which then will tune its data flow to that client accordingly. Thus, a user is able to register for none, some, or all of the possible data for his agent. Sensor data available for selection include horizontal position, depth, attitude, altitude, linear 6-dof velocity, temperature, salinity, magnetic flux and ocean current. After configuring a particular instance of a VIP, the user logs into the remote Environment Server, and chooses a scenario to join. Upon acceptance into the scenario, the user places the VIP at a desired position. Movement as well as other VIP activities are instigated using a VIP common control language[4] incorporated in the program. Through the benefit of the Environment Server's agent-to-agent communication harness, the user can also send these commands to other CADCON VIP participants, causing them to react as well. There are also provisions for users to set up their own local log files, which will collect a data trace of the AUVSim's participation in a scenario.

AUVSim, and the VIP interface to the server, is single vehicle-centric. That is, it can present only its single view situated somewhere in the simulated environment to the user. It was designed to operate this way in order to keep its “experiences” as realistic (when compared with real vehicles) as possible. In other words, individual AUVSim clients do not benefit from receiving global knowledge about the scenario. It does not take much “flying” time with this client for the user to get a feel for just how impoverished a submersible vehicle's sensor suite can be.

The AUVSim client was written in C++ using Microsoft Foundation Classes (MFC) and it runs in a Win32 environment (Windows 95/98/2000/NT 4.0). It functions well on Pentium 133 MHz class and better computers.

[4] Explained in detail in [Komerska *et al.*, 1999b].

"Visual" Scenario Visualization Client

The Visual program was written by AUSI to provide CADCON users with a real time, 3-dimensional image of an ongoing simulation scenario. Visual builds upon the visualization client interface protocol, which allow users to view global information regarding other CADCON VIP clients distributed across the Internet. In contrast to the AUVSim program, Visual presents a global, multiple VIP view of the simulated environment. Thus it is an observation tool, used only as an aid in understanding the scenario ground truth. As such, it has no means for controlling scenario participants.

Visual provides the user with a set of controls, which allow for the manipulation of the position and direction of the viewpoint, scene lighting, wire frame or surface drawing, agent marking, and display of various environmental features.

The Visual client was written in C++ using MFC and OpenGL and runs in a Win32 environment (Windows 95/98/2000/NT 4.0). It functions well on Pentium 200 MHz class and better computers.

Using CADCON

To participate in CADCON, the prospective participant uses their web browser to visit the CADCON home page at the AUSI web site[5]. From there, the user can download the current versions of the AUVSim and Visual clients. After fetching those clients and unbundling them on the local workstation, the user can interrogate the Environment Server by clicking the "Environment Server Status" link on the CADCON home page. The user's browser will then present a page containing the information relating to available simulations, some details on their states (dormant or usable), the number of participants (if being used), and the host name and port number needed for connection.

The AUVSim folder contains the following files: AUVSim.exe, EAVE.auv, and several support files. Double-clicking on AUVSim.exe executes the program. The user can create a new AUV configuration by utilizing the "AUV Configuration" dialog from within the "Edit" menu, or use the sample configuration "EAVE" supplied in the folder. To use this file, the user simply uses the "Open" command from within the "File" menu. Once the vehicle is configured, the next step is to connect to the Environment Server. First, the user must ensure that a working network connection to the Internet is in place on the host computer. Next, the user should inspect the "Connection Preferences" dialog within the "Connect" menu. The default server host and port information should be valid; if this has changed since the software was released, up-to-date information can be found on the CADCON Environment Server Status page. User name and password can both be set to "guest". Finally, the connection time can be specified; the server will disconnect the AUVSim application after this period of time elapses. Having verified this connection information, the user can now connect to the server by selecting the "Connect to Server" item within the "Connect" menu. This will initiate contact with the server and will provide the user with a list of currently

[5] http://www.ausi.org/cadcon/simulation.html

available scenarios in which to join. The user selects one of the these scenarios, and presses “Enter”. After a moment, the user is prompted with a dialog requesting the initial starting location of the AUV. This can be anywhere within the scenario boundary limits. After entering this data, the user presses “Enter Sim” to enter the virtual environment.

From this point onward, the simulated AUV is now a part of the scenario. The operator can view the position, energy status, communication messages sent and received, and mission sensor status of the AUV through the appropriate dialogs within the “Display” menu. The “Generic Behaviors” dialog within the “Command” menu provides a means to control this AUV as well as remote AUV simulations, based upon current AUSI research into a common control language for VIPs. Finally, the “Monitor” dialog within the “Environment” menu can be used to view client connection status. When the user is finished with the session, the “Disconnect” item can be selected from the “Connect” menu to disconnect the client from the Environment Server.

The Visual client folder contains the following files: Visualizer.exe and several support files. Double-clicking on Visualizer.exe executes the program. Connection to the Environment Server is handled similarly to AUVSim; after verifying connection information in “Connection Preferences”, the “Connect to Server” menu item is used to initiate the link. The user is then requested to select a scenario to join, and the connection process is completed.

Visual provides a complete view of all VIP participants in the environment. These participants can be viewed graphically to scale through the 3-d window. Numerical data for the environment features as well as each VIP client is also available through the “Environ Info” and “Vehicle Info” tabs, respectively, positioned to the right of the 3-d view. The user can manipulate their view using the 2-button mouse or by using the “Scene Maneuver Tool” within the “View” menu. The various scenario features can be located within the environment by checking their names within the “View” menu. In addition to the default 3-d view, the user can also plot 2-d x-y position history and energy vs. time for any VIP in the environment. As with the AUVSim client, the “Disconnect” item can be selected from the “Connect” menu to disconnect the client from the Environment Server.

More complete usage instructions for each client are available from the client help files located off of the CADCON home page.

Environment Server Mechanics

The CADCON architecture is based upon the *client-server* model [Comer & Stevens, 1993]. The *concurrent server* sits on an AUSI host waiting for users to initiate or join a simulation session via their *client* software. The services provided by the Environment Server are classified as being *nonstandard application services*; that is, they use a custom applications level protocol and nonstandard protocol ports. The clients are fully parameterized to provide access to different scenarios on different server ports. The client interaction with the server is of the *connection-oriented style* (TCP/IP based) for increased reliability and simpler applications software design.

Figure 2 shows a typical "snapshot" in time of active processes during a scenario to illustrate the major software components of the CADCON architecture. Shown are five participating agents, with their corresponding Environment Server processes represented by the column of ellipses just to the right of the shared memory. They each access their particular piece of the shared memory to obtain and deposit pertinent information about the agent they are serving. In this diagram, ellipses denote running processes while rectangles denote files, memory segments, and machine boundaries. Communication paths, are depicted with arrows of various thicknesses to qualitatively indicate directionality and bandwidth of that communication.

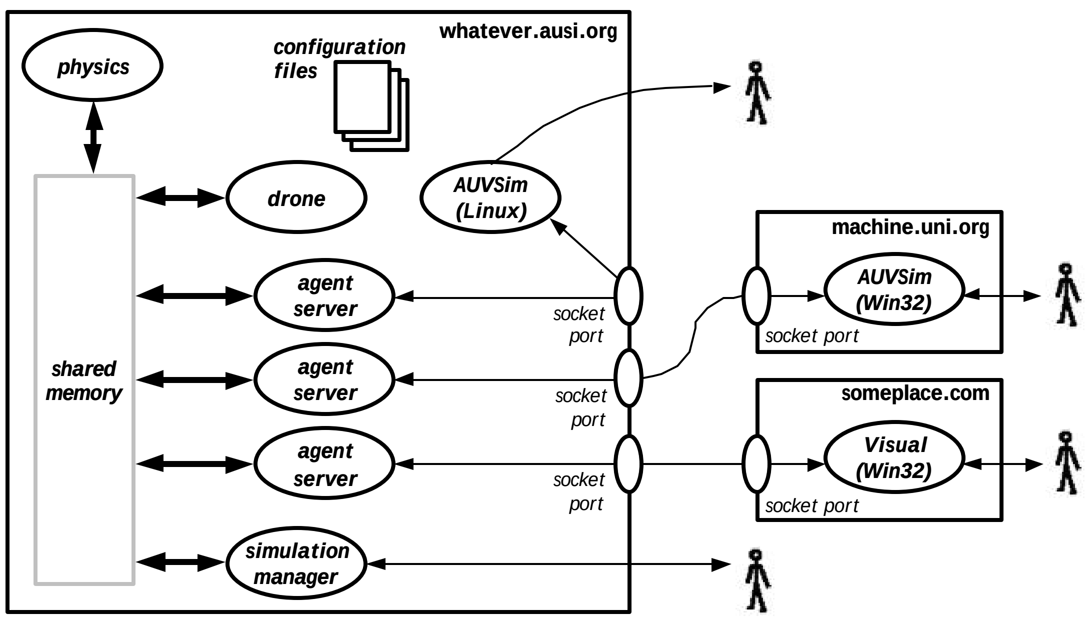

Figure 2. CADCON Architecture Detail

The physics daemon process is where the large grain physics of the system are implemented; it is the source of individual agent motion. It is also responsible for checking to see that agents' bodies do not overlap with each other or environment features. When this occurs, collisions are signaled and the involved agents' motion is modified appropriately. This process also manages the data structures which are used to represent the features of the simulated aquatic environment. During operations, it reads the shared memory structure to find out the state of each participant, computes a new state for each one, and writes the results back to the memory structure. The cycle rate of this process is configurable. We have been running it at 10Hz.

The agent server process is what actually performs the traditional serving function to a remote client. It deposits data from the client in the shared memory data structures and retrieves required data from same for transmission to the client. Updates to the client occur at a slower rate, typically 1 Hz to the Visual client and 2 Hz to the AUVSim client. AUVSim and Visual client updates to their respective agent servers occur at 1 Hz.

The simulator manager process provides the server administrator (local to the server machine) with a window into ongoing simulations. Via this process, the administrator can view the absolute ground truth of a scenario as it unfolds. This process can be activated when needed during execution of a scenario. In fact, the physics daemon and the agent server(s) are fully capable of running complete end to end scenarios without any administrator interaction. The simulation manager program was originally intended as an observation tool only; recently however, it has been outfitted with some control capabilities. We expect this trend to continue.

At the top is shown a completely autonomous drone process. Being autonomous, the drone process does all the computations necessary for participating in the scenario; that is, it has no corresponding client. Drones are intended to add “background color” to scenarios by populating them with simplistic agents. Drones are specified in the scenario configuration files, and are considered part of the virtual environment feature set.

As currently implemented, both the drone and simulation manager processes don't strictly adhere to the client/server model. They both must run on the same machine as the remainder of the Environment Server. This weakness will be addressed in the future. Both the Visual client and simulation manager use OpenGL graphics libraries[6] for rendering 3-d views of the underwater cyberspace. In keeping with the desire to implement this on commonly available hardware, the current simulation manager uses the free Linux port of the OpenGL library (called Mesa) while the Visual client uses the MFC compatible version the library.

Below the drone process are shown two agent server processes handling AUVSim clients, which are arranged to show the distributed capability: one is being controlled locally on the server host while the other is being controlled remotely from *machine.uni.edu*. A third agent server process is for a Visual client running on host *someplace.com*. Client/server communication between remote hosts is effected over the network via sockets, which are diagramed as small ellipses in the various host boundaries.

CADCON Simulator Experience and Conclusions

The primary utility of CADCON is to serve as a testbed infrastructure to support research activities at AUSI. To date, it has been employed in the following endeavors:

VIP Common Control Language Development

CADCON has allowed us to test and evaluate portions of a VIP Common Control Language (CCL), currently under research at AUSI [Komerska *et al.*, 1999b]. Such a language is critical in enabling multiple heterogenous VIP command and control such as in an AOSN application. The current AUVSim clients implement language and behavior aspects which include basic maneuvering, waypoint transiting, and

---

[6] http://www.opengl.org

communication of capabilities and status. Using CADCON, we have connected numerous agents in joint situations and exercised these language aspects. This has allowed us to test the agents' maneuvering and communication capabilities, including situations where agents have command over the movements of teams of other agents. Doing this provides a powerful way to bring a static language specification "to life" and to gain meaningful experience with it.

The CCL specification has been incorporated in the AUVSim client in a modular fashion, allowing us (and other researchers) to incorporate newer language versions into existing AUVSim clients relatively easily.

Solar AUV Hardware-in-the-Loop Experiments

By design, the CADCON infrastructure abstracts the notion of a VIP to include both simulated and actual platform representations. Having demonstrated the purely simulated capabilities of CADCON (through the AUVSim client), we then carried out an effort to implement simple hardware-in-the-loop monitoring and control of a prototype SAUV energy subsystem [Ageev *et al.*, 1999]. This energy subsystem, known as the Energy System Testbed (EST), was developed at AUSI to investigate SAUV energy management strategies [AUSI, 1999]. It consists of a photovoltaic panel array, batteries and micro-controller integrated into a watertight SAUV-shaped platform. We utilized a modified AUVSim client to both interface to the EST hardware as well as the Environment Server, thereby representing the EST "by proxy" within the virtual CADCON environment. We could then choose to use CADCON as a pure distributed EST monitoring tool (via the Visual client) or have the AUVSim client represent an SAUV, with a real energy subsystem integrated with other simulated subsystems. This latter approach allowed us to test the previously developed energy management strategies in (simulated) motion oriented experiments, as well as exercise inter-VIP communication-related CCL elements, by allowing other VIPs to query the Solar AUVSim client for its energy state. The energy state responses corresponded to the actual measured subsystem values calculated on the EST hardware.

In November 1999, the EST participated in the Office of Naval Research (ONR) sponsored "AUV Demo," held in Gulfport, Mississippi. Our objective during this event was to exercise the EST situated in a remote water bound context, collect solar energy data, communicate with the platform over several unique communication links, and distribute that data in near real time to Visual clients over the Internet via the CADCON communication harness. Principal components linked together during the demonstration included the EST platform, the research vessel, the shore station and CADCON elements. These components used links based on both Ethernet and serial protocols, RF and cellular telephone modems, and the Internet. In addition to this, several PC-based applications were utilized, including an early version of the ASMAC client for vehicle control and monitoring, and the CADCON AUVSim and Visual clients for remote representation and monitoring of the vehicle, respectively. By using this approach, we gained valuable experience in networking representative AOSN components, along with an understanding of issues related to public versus private vehicle control, communications time delays between nodes, impact of node failures upon both command and monitoring, and understanding general aspects of interfacing mature nodes with experimental nodes.

In February, 2000, the EST platform was transported to New Castle, NH for another set of remotely situated demonstrations. The focus of these experiments was to complete a series of validation tests begun at the AUV Demo of a prototype satellite communication system called the Ocean Data Link (ODL), under development for ONR by ViaSat Corporation. ODL employs existing C-band geosynchronous satellites to provide two-way communications with full ocean coverage utilizing low power transceivers. An ODL transceiver prototype was mounted on the EST and was placed 5-10 m out in a mild ocean surf off of New Castle. The demonstration consisted of regular polling of the EST status and GPS information over the satellite datalink using a modified AUVSim client. The EST/ODL equipment responded with appropriate energy status or GPS information. This information was then made available on the Internet through the CADCON Visual client.

Movement Simulator for Magnetic Navigation Experiments

As part of an effort to support Foster-Miller, Inc. (FMI), researchers from AUSI and FMI were able to utilize CADCON to test and evaluate magnetic navigation algorithms under development at FMI and MIS Ltd. of Nova Scotia. This was accomplished by modifying the AUVSim client to incorporate the magnetic navigation algorithms in conjunction with adding a magnetic dipole model to the Environment Server. Dipole flux information is delivered to the AUVSim client as “sensed data”, where it is then used as input into the FMI positioning algorithms. The “true” vehicle position can then be compared to the calculated “magnetic” position, and the error assessed over a range of vehicle trajectories. The magnetic positioning algorithms are provided to a user (with FMI/MIS permission) in binary format, in the form of a Windows dynamic link library (DLL). The AUVSim client searches for this library during AUV configuration. It provides the extra functionality if the DLL is present or the default functionality (no magnetic positioning) if it is not found. This paradigm offers other researchers the ability to explore the use of these magnetic positioning algorithms within their own VIP simulations (within the CADCON simulation context) while protecting the developers' intellectual property rights in this area.

An interesting outcome of this work is we now have a method for researchers to add their own custom functionality to the AUVSim client. Because of its modular object-oriented design, the AUVSim client serves as a good foundation application whose functionality can be extended by other research groups for use in the CADCON facility in order to address their own research issues.

Future Work

We have identified a number of areas for further CADCON development. At the front of the list is the design and implementation of a new kind of client called the *Autonomous Systems Monitoring and Control (ASMAC)* client. A user actually controlling and monitoring a fleet of heterogenous VIPs will need more than a single agent perspective into the system. Such a user faces issues beyond those associated with remotely piloting single vehicles. The user will want to control and monitor the fleet in terms of teams of VIPs cooperating to complete some mission, not in terms that function at the level of individual fleet participants. The development and use of this client will form part of this research into multi-agent monitoring and control at AUSI. A simplified version of ASMAC is currently under development to support a joint AUSI/IMTP/University of Hawaii extended duration ocean sampling demonstration of the SAUV during Spring of 2001.

We have also started work on an *External Model Interface (EMI)* client. The purpose of this new client will be to access higher fidelity ocean databases and models (for example, those generated by Rutgers University models during LEO-15 testing[7]) and provide that information to the Environment Server. Part of this work will involve adapting our software to utilize standard data access formats, such as NetCDF[8]. This improvement will allow CADCON users to participate in more realistic and relevant simulations.

Plans are also underway to implement a family of motion models, which will also be selectable by the participant at login time. These will range from a fairly realistic force based model to the abstract "achieve next position" mechanism used in our original Environ simulator. Additionally, each participant will be able to register for a specially defined "self" motion model, where the participant supplies its position/attitude data to the Environment Server. This opens the door to utilizing far more accurate client side motion and control models [Peng *et al,*. 1999] in scenarios. It will also provide the means for real VIPs operating in situ to inject data into the scenario (assuming a method exists for getting that agent's communication attempts onto the Internet).

We intend to implement simple models of various navigation sources, emitters, and networks so that participants may register for particular navigation aids and then receive their position and attitude updates in a more realistic fashion. The magnetic dipole is the first instance of such a navigation aid model. We also plan to improve the agent-to-agent communications harness to include representations of RF, cell phone, and satellite communications, as well as adding noise and attenuation effects to all simulated communications paths.

For collision computations, all agents are currently considered to be spheres of various sizes. Work is in progress to refine that representation using available "collision libraries". In particular, we are investigating the Robust and Accurate Polygon Interference Detection (RAPID) library[9].

We intend to participate in a joint effort to connect the University of Maine's high level multi-agent CoDA simulator[10] with CADCON in order to form an integrated AOSN testbed. In this setup, CoDA will be responsible for simulating high level AOSN cooperative control mechanisms as well as individual VIP

---

[7] http://marine.rutgers.edu/mrs

[8] http://www.unidata.ucar.edu/packages/netcdf

[9] http://www.cs.unc.edu/~geom/OBB/OBBT.html

[10] See Turner & Turner, [1998] and http://cdps.umcs.maine.edu/CoDA

intelligent controllers. CADCON will supply the individual agents' “presences” in the scenario, the communication channel, and the virtual underwater environment context.

Bibliography

[Ageev *et al.*, 1999] Ageev, M. D., Blidberg, D. R., Jalbert, J. C., Melchin, C. J., & Troop, D. J. Results of the evaluation and testing of the solar powered AUV and its subsystems. In [ISUUST, 1999].

[AUSI, 1999] AUSI Staff. Energy system testbed investigation. Technical Report 9909-01, Autonomous Undersea Systems Institute, 86 Old Concord Turnpike, Lee, NH 03824.

[Chappell *et al.*, 1999] Chappell, S. G., Komerska, R., Peng L., & Lu, Y. Cooperative AUV Development Concept (CADCON) An Environment for High-Level Multiple AUV Simulation. In [ISUUST, 1999].

[Comer & Stevens, 1993] Comer, D. E. & Stevens, D. L. Internetworking with TCP/IP Volume III: Client - Server Programming and Applications: BSD Socket Version. Prentice-Hall, Inc., Englewood Cliffs, NJ, 07632.

[Curtin *et al.*, 1993] Curtin, T. B., Bellingham, J. G., Catipovic, J., & Webb, D. Autonomous oceanographic sampling networks. Oceanography, 6(3):86-94.

[ISUUST, 1999] Eleventh International Symposium on Unmanned Untethered Submersible Technology, August 1999, Durham, NH. Autonomous Undersea Systems Institute, 86 Old Concord Turnpike, Lee, NH 03824.

[Komerska *et al.*, 1999a] Komerska, R. J., Blidberg, D. R., Chappell, S. G., & Peng, L. Progress in the development and evaluation of a standard AUV command and monitoring language. In [ISUUST, 1999].

[Komerska *et al.*, 1999b] Komerska, R. J., Chappell, S. G., Peng, L., & Blidberg, D. R. Generic behaviors as an interface for communication, command and monitoring between AUVs. Technical Report 9904-01, Autonomous Undersea Systems Institute, 86 Old Concord Turnpike, Lee, NH 03824.

[Lu *et al.*, 1999] Lu, Y., Blidberg, D. R., Aponick, A. A., & Jalbert, J. C. Simulation and error analysis of a VLF magneto-induction navigation system. In [ISUUST, 1999].

[Momenee, 1987] Momenee, M. L. An environment simulator for a KBS controlled underwater vehicle. In Fifth International Symposium on Unmanned Untethered Submersible Technology, pages 561-566, June 1987, Manchester, NH. Marine Systems Engineering Laboratory, Marine Program Building, University of New Hampshire, Durham NH, 03824.

[Peng *et al.*, 1999] Peng, L., Blidberg, D. R., Chappell, S. G., & Komerska, R. J. A new artificial neural network model of potential application in unmanned underwater technology. In [ISUUST, 1999].

[Turner & Turner, 1998] Turner, R. M. & Turner, E. H. Organization and reorganization of autonomous oceanographic sampling networks. In Proceedings of the 1998 IEEE International Conference on Robotics and Automation (ICRA'98), pages 2060-2067, Leuven, Belgium.