In Proceedings of the Eleventh International Symposium on Unmanned Untethered Submersible Technology, August 23–25, 1999, Durham NH. Published by Autonomous Undersea Systems Institute, 86 Old Concord Turnpike, Lee NH, 03824.

Cooperative AUV Development Concept (CADCON) An Environment for High-Level Multiple AUV Simulation*

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`

Liang Peng
Yingchun Lu
Harbin Engineering University
2 Yiman Street
Harbin
Heilongjiang 150001, China
`lpeng@ausi.org`

Abstract

As an aid to our investigations in the area of multiple cooperating vehicles, AUSI has developed the CADCON facility. This facility employs a distributed multi-agent simulation, visualization system, and control harness designed to simulate a fairly accurate underwater environment, which can be shared by simulated or real vehicles connected via the Internet. To date, this facility has been used to support two AUSI projects. The facility is available via the Internet for others to use and it has been employed by independent workers in industry and academia to support their own research projects.

Introduction

A strong component of current underwater robotics work is 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 lack of tools geared towards the needs of researchers working with multiple cooperating VIPs[1]. 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.

The main idea behind CADCON and its simulator is that it provide an open and flexible environment for use by as many researchers as possible. 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 the CADCON Simulator to 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. This broad participant heterogeneity pushes us to refer to them as *agents*. To enable this, we have endeavored to streamline the complexity and the costs associated with an agent's connection to the CADCON Simulator. In particular, we have implemented the system's various components on ubiquitous hardware (Pentium class computers). These components embody the client/server model and communicate via a simple connection

---

*This work was supported by AUSI and the U. S. Office of Naval Research (ONR) under grant number N0001-14-96-1-5009.

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

protocol with rides on the lingua franca of the Internet: TCP/IP. This non-reliance on exotic hardware and/or proprietary communications protocols allows other researchers to leverage the simulator's utility for their own work.

Background

For the most part, "traditional" 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, 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 become secondary to the *behaviors* exhibited by the participating agents. Thus, it is more concerned with how agents interact with each other and the big features of their environment, than how 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 simulator's agent-to-agent communication mechanism, which provides a communication network at an idealized high level.

The ideas and features supported in the CADCON Simulation harness have their roots in our old (1980's) Environ Simulator [Momenee, 1987]. With new technology available, we have completely reworked the old Environ into a new system which:

- is more modular,
- is distributable across a network (communicating via sockets),
- simulates a variety of participants instead of just our open frame EAVEs,
- 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,
- provides a cyberspace for real vehicles that are undergoing development and testing,
- 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 AOSN [Curtin *et al.*, 1993] mission rehearsal, test, and display tool.

In this paper, we describe the CADCON facility components, the environmental features provided by the Simulator, the agent models employed to represent participants, the logistics of connecting to the Simulator, and mechanisms of agent interaction. In particular, we report our use of the CADCON Simulator in the support of two AUSI projects; building generic cooperative vehicle behaviors, and Solar AUV development [Ageev *et al.*, 1999].

Top Level Features

As currently implemented, there are four major components in the CADCON facility; the *Environment Server* and three different clients: an *AUV Simulator Client*, a *Visualizer Client*, and a *Status Client*. Figure 1 diagrams the distributed nature of these components. The first two clients are available for downloading from the Simulation Home Page at the AUSI web site[2]. The Status Client has been implemented as a Java applet that will run in a Java-enabled web browser.

---

[2] See `http://www.ausi.org/`.

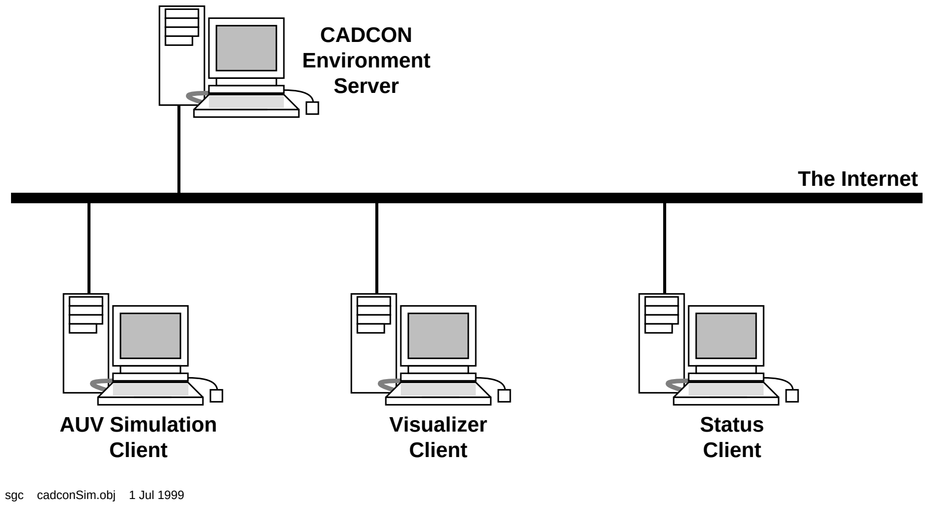


Figure 1: CADCON Simulator Facility.

Environment Server

This is the heart of the CADCON Simulator. 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 set of magnetic flux values emanating from a dipole source [Lu *et al.*, 1999]

These features are controlled via a set of configuration files. The Simulator 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 cyberspace. When the last participant leaves, a timer is set, and when that expires, the simulation takes itself down to its dormant state, where only the connection daemon remains to wait for a new connection request. This scheme allows the simulator 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 cyberspaces.

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 the CADCON Simulator 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 a communication string to any and all other participants. This is the Generic Behavior (GB) [Komerska *et al.*, 1999a] transport mechanism.

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 log administrative type information (user account names, remote hosts names, times, etc) as well as agent specific traces of that agent’s movement through the cyberspace.

The Environment Server runs on a Linux Pentium host.

AUV Simulation Client

This is the Simulator component that allows users distributed around the Internet to participant in joint mission scenarios. Using this client, a user can configure a local simulation of a particular kind of submersible vehicle on his own workstation, login to the remote Environment Server, start at any desired position, and move the simulated vehicle through the cyberspace using the GB command *Maneuver*[3]. Through the benefit of the Environment Server's agent-to-agent communication harness, the user can also send GB commands from his AUV client to other AOSN participants, causing them to move around as well. There are also provisions for the user to set up local log files, which will collect a data trace of the AUVsim's participation in a scenario.

This client is single vehicle centric. That is, it can present only its single view of 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 AUVsims 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 runs in a Win32 environment (Windows 95/98/NT).

Visualizer Client

This Simulator component was implemented in order to provide users with a real time, three dimensional, animated image of the ongoing simulation scenario. In contrast to the AUVsim, the Visualizer 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. The Visualizer provides its user with a set of controls, which allow him to manipulate the position and direction of the viewpoint, scene lighting, wire frame or surface drawing, agent marking, and display of various environmental features.

The Visualizer client runs in a Win32 environment.

Status Client

The Status client was developed to provide a simple and easily accessible means of interrogating the Environment Server for a report on its current state. If a simulation is running, this client receives and displays a few lines of information giving the Simulator's local time, simulated environment's name, startup time, elapsed time, and a count of the number of clients being served. If the simulation is dormant or completely unavailable, the user is notified of this as well.

The Status client runs in any Java 1.0 enabled web browser.

Using the CADCON Simulator

In typical use, most apparent activity is on the client side. This is because the Environment Server has been designed to run completely autonomously. Since late 1998, we have made the Environment Server available on as much of a 24/7 basis as our phone modem connection allowed. Our recent cable modem installation has vastly improved the connection reliability, stability, and bandwidth.

To use the Simulator, the prospective user follows links in the AUSI web site to the Simulation Home Page. From there, the user can download the current versions of the AUVsim and Visualizer clients. After fetching those clients and unbundling them, the user interrogates the Simulator by clicking the "Check Status" button found at the bottom of the web page. (This executes the status client in the browser.) If the Simulator is completely unavailable (say, due to maintenance or a broken Internet connection) the user will see a terse message of a "connection refused" nature. Otherwise, the user will be informed that the simulator is either "available, but dormant" or "running". In either case, activating either the AUVsim or the Visualizer client will allow the user to join the scenario or initiate one.

---

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

Once the AUVsim client is up and running, the user configures the agent, and then logs in to the Environment Server giving the host name (or sometimes IP address) and port number found at the top of the status report output. After an AUVsim login, the user is free to direct the agent about the environment using control mechanisms in the AUVsim client. If the user is only interested in his own agent's experience in the scenario, there is no need to activate a Visualizer client, since the AUVsim can report all of its sensor readings in tabular form and send copies to local log files. To get an image of what is happening, the user will have to bring up a Visualizer. When finished with his session, he simply commands his client(s) to disconnect from the Environment Server.

More complete usage instructions are available from our Simulation Home Page in the `ReadMe` files.

Mechanics

The CADCON Simulator is constructed on 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 Simulation server are classified as being *nonstandard application services*; that is, it uses a custom applications level protocol and nonstandard protocol ports. The Simulation 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.

The major software components of the CADCON Simulator are shown in generalized terms in Figure 2. In this diagram (and those following) 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.

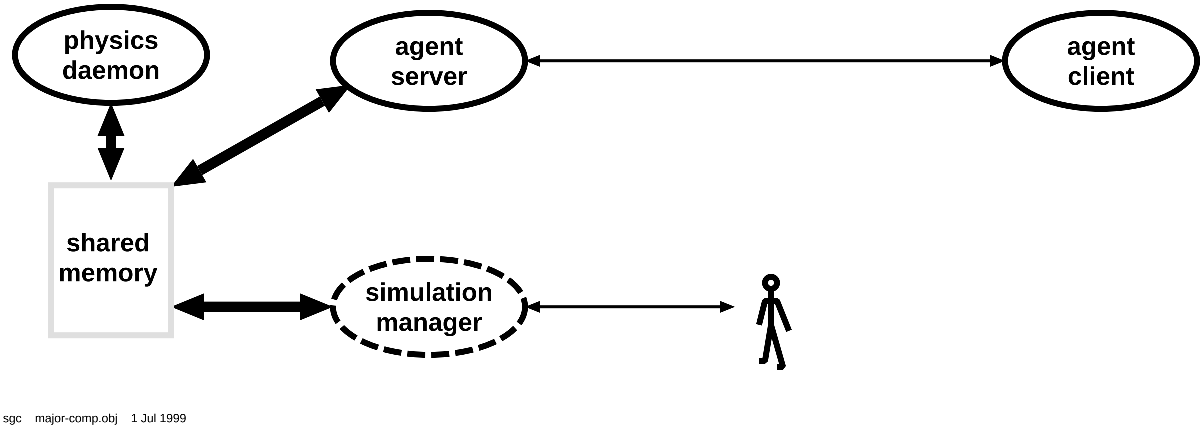


Figure 2: CADCON Simulator Major Software Components.

The components on the left of Figure 2 make up the Simulator's Environment Server while those on the right make up the client(s). The Environment Server is really a family of processes, the three most active being shown in the Figure. All three of these processes run asynchronously to each other, inhabit the same host machine, and communicate with each other through the indicated shared memory.

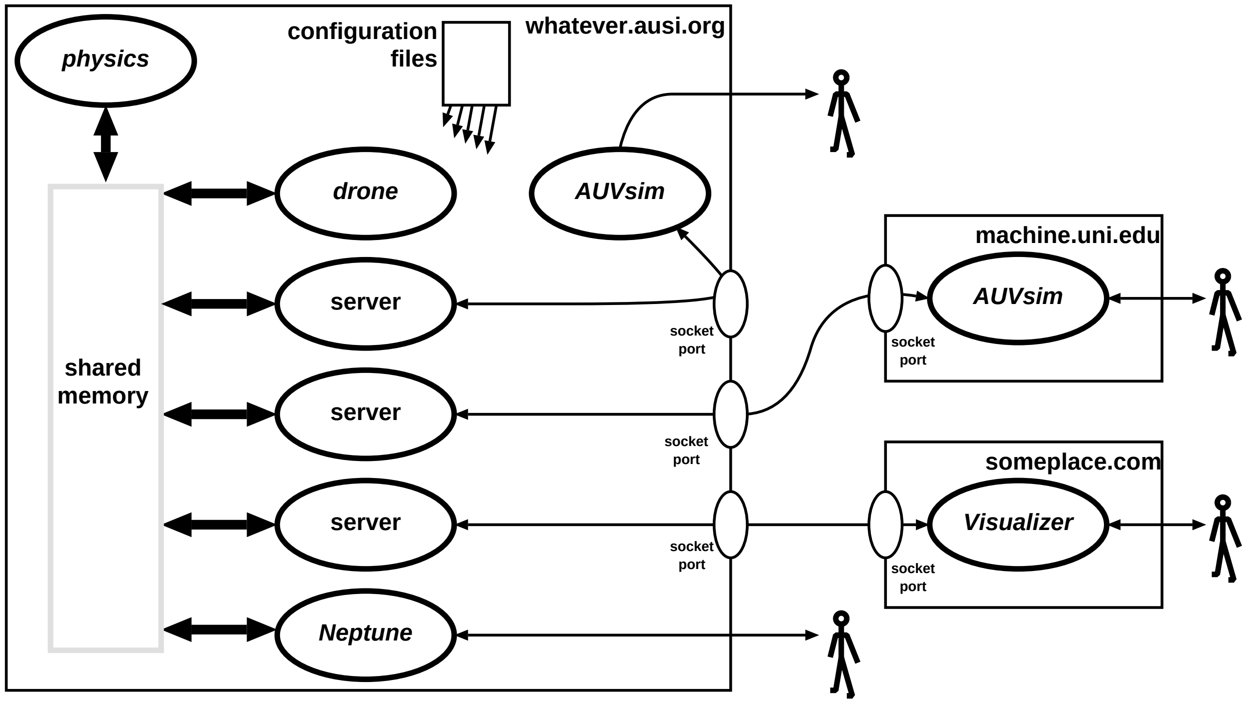
The **physics daemon** process is where the large grain physics of the system are implemented; it is the source of individual agent motion. It also is responsible for checking to see that agents' bodies do not collide with each other or environment features. When this occurs, the involved agents' motion is modified appropriately. This process also manages the data structures which are used to represent the features of the simulated aquatic environment.

The process labeled **agent server** 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.

The **simulator manager** process provides an operator (local to the server machine) with a window into an ongoing simulation. Via this process, the operator can view the absolute ground truth of the scenario as it unfolds. In the Figure, its outline is dashed to indicate that it need not be present all the time in order for a simulation to run. In fact, the physics daemon and the agent server(s) are fully capable of running complete end to end simulations without any operator interaction. To date most of our simulation time has been accomplished without the simulator manager running at all. This program was originally intended to be more of an simulator operator's observation tool. Recently, it has been outfitted with a few control capabilities. We expect this trend to continue.

The process labeled **agent client** represents the software which is making requests of the Simulator and providing necessary data about one particular participant in the simulation. It may or may not be running on the same host as the server processes with which it is communicating.

Some More Detail



sgc simtop2.obj 1 July 1999

Figure 3: CADCON Simulator Structure.

The deeper structure of the CADCON facility is shown in Figure 3. At this level of examination, machine boundaries, some actual process names (in italics), and the different client types may be seen.

The large rectangle on the left in the Figure represents the host machine (named **whatever.ausi.org** here), which is providing the Simulator as a service for other machines. Three example clients are shown running on their hosts in the center and the right of the Figure: one AUVsim on the same host[4] and two (another AUVsim and a Visualizer) on two other hosts. Client/server communication between remote hosts is effected over the Network via sockets, which are diagrammed as small ellipses in the various host boundaries.

The *physics* daemon is found at the top of the Figure. 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.

---

[4] The current Linux version of the AUVsim client is limited in functionality.

Figure 3 shows a “snapshot”, where five agents are participating in a simulation: their individual servers being represented by the column of ellipses just to the right of the shared memory. They all access their particular piece of the shared memory to obtain and deposit pertinent information about the agent they are serving. At the top is a completely autonomous *drone*. Being autonomous, the **drone** process does all the computations necessary for participating in the scenario; that is, it has no corresponding client. Drones are supposed to add “background color” to scenarios by populating them with simplistic agents. Drones are specified in the scenario configuration files. Below the **drone** are two **server** processes handling user controlled AUVsims, which are arranged to show the Simulator’s multi-machine adaptability: one is being controlled locally on **whatever.ausi.org** and the other is being controlled remotely on **machine.uni.edu**. The third **server** process is for a Visualizer running on host **someplace.com**. That user is viewing a 3D representation of the entire scenario. Not shown here is a Status client because its effect on a scenario is minimal and only transitory.

Given the administrative privileges provided the operator when accessing the Simulation via the management process, we have chosen to call it *Neptune*. This process (shown at the bottom of the figure) presents the ground truth of the Simulator’s cyberspace to the operator.

As currently implemented, both **drone** and **Neptune** don’t strictly adhere to the client/server model. They both must run on the same machine as the rest of the Simulator. This weakness will be addressed in the future.

Both the Visualizer Client and *Neptune* use OpenGL graphics libraries[5] for rendering 3D views of the Simulator’s cyberspace. In keeping with the desire to implement this on commonly available hardware, the current *Neptune* uses the Linux port of OpenGL. The Visualizer Client uses the Microsoft Foundation Class compatible version of the OpenGL Library.

CADCON Simulator Experience and Conclusions

The CADCON Simulator as designed to provide the infrastructure for other work. It has been employed in the following endeavors:

Generic Behavior Development

This facility has allowed us to gain runtime experience with portions of the current specification of our Generic Behavior control language [Komerska *et al.*, 1999b]. The current AUVsims implement major aspects of the *Maneuver* GB, which is used to command agents’ movements. Using the Simulator, we have connected numerous agents in joint situations and exercised various *Maneuver* commands. This allows us to test the agents’ maneuvering 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 specification “to life” and, thus, allows us to gain meaningful experience with it.

Solar AUV Hardware-in-the-Loop Experiments

The AUVsim client also implements a portion of the *Monitor* GB. We have used this behavior to exercise the hardware-in-the-loop capability of CADCON by monitoring and controlling a prototype Solar AUV energy subsystem interfaced to an augmented AUVsim client. This energy subsystem, known as the Energy System Testbed (EST), was originally developed to investigate Solar AUV energy management strategies [AUSI, 1999]. It consists of a photovoltaic panel array, batteries and a microcontroller. By interfacing the EST to an AUVsim client, we are now able to better simulate a Solar AUV and more realistically exercise the EST. With this setup, we can test the previously developed energy management strategies and exercise the *Monitor* GB, by allowing other VIPs to query the Solar AUVsim client for its energy state. The energy state response will correspond to the actual measured subsystem values calculated on the EST hardware.

---

[5] See `http://www.opengl.org/`.

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 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 positioning algorithms. The true vehicle position can then be compared to the calculated position, and the error assessed over a range of vehicle trajectories. The magnetic positioning algorithms are provided to a user in binary format only, in the form of a Windows dynamic link library (`DLL`). The AUVsim client searches for this library on launch. It provides the extra functionality if the `DLL` is present or the default functionality if it is not found. This paradigm offers other researchers the ability to explore the use of these magnetic positioning algorithms in CADCON 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, this 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 CADCON Simulator enhancement.

At the front of the list is the design and implementation of a new kind of client we are calling the *External Model Interface Client*. The purpose of this new component will be to access high fidelity oceanic data bases and models (for example, those generated by Rutgers University's LEO-15 installation[6]) and send that information to the Environment Server. Part of this work will involve adapting our software to utilize standard data access formats, such as NetCDF[7]. This improvement will allow CADCON participants to be involved in more realistic simulations.

We anticipate the need of another client type. A user actually controlling and monitoring an AOSN 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 AOSN in terms of a desired mission, not in terms that function at the level of individual AOSN participants. In answer to this, an effort will be undertaken to investigate the characteristics of such a fleet level client for Autonomous Systems Monitoring and Control (ASMAC) and begin its development.

Along with improving the scenario presented by the Environment Server, we seek to improve the modeling of the individual participants. We are working on a scheme for flexible registration at participant login time. This will allow the participant to specify its own particular suite of subsystems, and the Environment Server will tune its communication to that client accordingly. Thus, a user will be able to direct his AUVsim client to register for none, some, or all of the possible data for his agent.

Beyond variable sensor registration, 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 an 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 actually 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 nets 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.

---

[6] See `http://marine.rutgers.edu/mrs`.

[7] See `http://www.unidata.ucar.edu/packages/netcdf`.

We will improve the agent-to-agent communications harness to include representations of RF, cell phone, and satellite communications, as well as adding noise sources 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 [8].

We intend to participate in a joint effort to connect the University of Maine's high level multi-agent CoDA simulator[9] with the CADCON Simulator 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 intelligent controllers. CADCON will supply the individual agents' "presences" in the scenario and the communication channels between them.

References

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

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

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

Curtin, T. B., Bellingham, J. G., Catipovic, J., & Webb, D. (1993). 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, R. J., Blidberg, D. R., Chappell, S. G., & Peng, L. (1999a). Progress in the development and evaluation of a standard AUV command and monitoring language. In [ISU, 1999].

Komerska, R. J., Chappell, S. G., Peng, L., & Blidberg, D. R. (1999b). 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, Y., Blidberg, D. R., Aponick, A. A., & Jalbert, J. C. (1999). Simulation and error analysis of a VLF magneto-induction navigation system. In [ISU, 1999].

Momenee, M. L. (1987). 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, L., Blidberg, D. R., Chappell, S. G., & Komerska, R. J. (1999). A new artificial neural network model of potential application in unmanned underwater technology. In [ISU, 1999].

Turner, R. M. & Turner, E. H. (1998). 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.

---

[8] See `http://www.cs.unc.edu/~geom/OBB/OBBT.html`.

[9] See `http://cdps.umcs.maine.edu/CoDA`.