2007-08

Autonomous Unmanned Vehicle (AUV) Workbench Rehearsal and Replay: Mapping Diverse Vehicle Telemetry Outputs to Common XML Data Archives

AUTONOMOUS UNMANNED VEHICLE WORKBENCH (AUVW) REHEARSAL AND REPLAY: MAPPING DIVERSE VEHICLE TELEMETRY OUTPUTS TO COMMON XML DATA ARCHIVES

Don Brutzman
Code USW/Br, Naval Postgraduate School, Monterey California USA 93933-5000
+1.831.656.2149 brutzman@nps.edu

Abstract. The Autonomous Unmanned Vehicle Workbench (AUVW) supports physics-based mission rehearsal, real-time task-level control of robot missions, and replay of recorded results in support of unmanned underwater, surface and air vehicles. Geographic information system (GIS) layers, a 2D geographic plot and a 3D visualization capability provide operators with multiple views and an improved understanding of robot operations. Extensible 3D (X3D) graphics and other open data standards are used throughout, implemented using open-source software for maximum repeatability and usefulness. This paper first provides a synopsis of AUV Workbench functionality and then examines how cross-platform compatibility of mission-output telemetry from various robots is achieved using the Autonomous Vehicle Command Language (AVCL). Experimental results, online assets and future work are also presented.

AUV Workbench background. The NPS AUV Workbench supports physics-based modeling and visualization of autonomous vehicle behavior and sensors. Rehearsal animations are configured to model arbitrary vehicles, based on high-fidelity vehicle-specific hydrodynamics and aerodynamics equations of motion. Extensible 3D (X3D) graphics models are provided for numerous kinds of robots to permit behavior visualization in combination with sensor propagation and bathymetry backgrounds. Chat-based networking allows multiple operators to share visualizations via the Distributed Interactive Simulation (DIS) protocol. A geographic information system (GIS) view is synchronized with a 2D tactical view, the X3D-based virtual-environment view, and detailed time-parameter plots. These combined capabilities support control-algorithm development, tuning of hydrodynamics response constants, mission generation and rehearsal, and replay of completed missions in a benign laboratory environment.

The AUV Workbench software architecture is designed to permit identical rehearsal, real-time control and replay of robot missions. Data flow among the various system modules is shown in Figure 1.

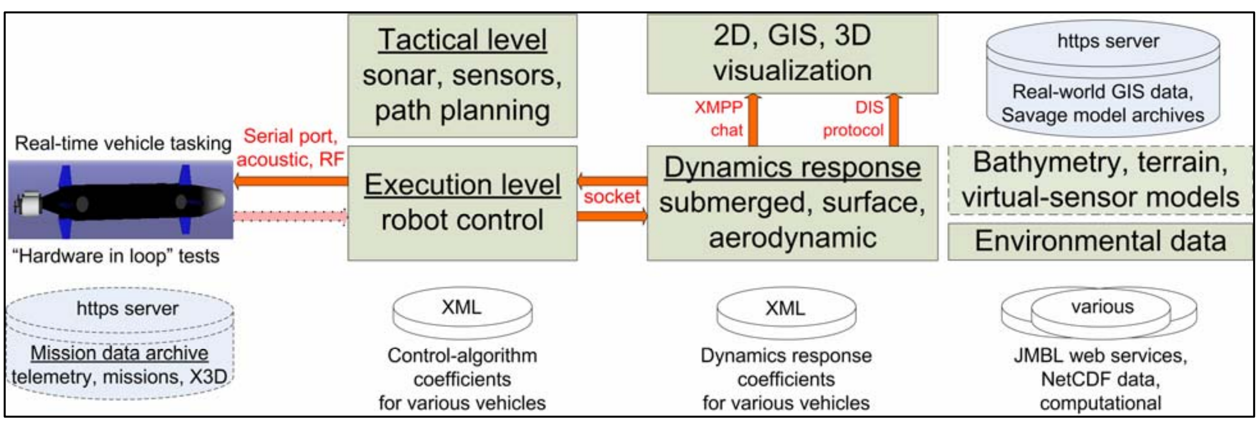

Figure 1. AUV Workbench data flow architecture enables consistent physics-based rehearsal, real-time control, or variable-rate replay of both real and simulated robot missions.

Software version control, autoinstallers, online software update capability, bug tracking and an archived mailing list enable group usage of shared assets. Cross-platform portability with a consistent user interface and internationalization (I18N) of menus is provided by an implementation that follows relevant coding conventions for the Java programming language.

The AUV Workbench has numerous panels and displays. Figure 2 shows graphical user interface (GUI) naming, primary capabilities, and typical layout for the various panes.

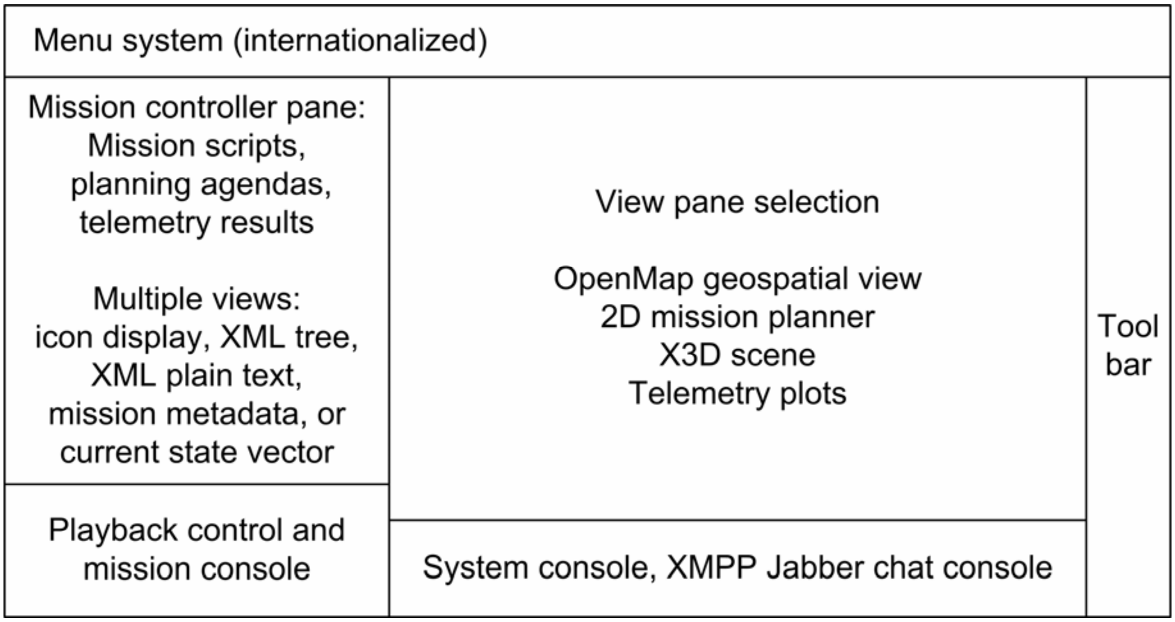

Figure 2. AUV Workbench display layout. Additional popup panels are also invocable.

Motivating problems and prior work defining AVCL. The most widespread problems facing operators of multiple robots are compatibility and interoperability. While robot communication schemes and data formats vary widely, the types of commands that are sent (and the types of data that are recorded) are often quite similar. This logical consistency provides a significant opportunity for achieving common operating capabilities despite apparent robot dissimilarities.

Previous work has shown that interoperability among diverse robots types can be provided by the Autonomous Vehicle Command Language (AVCL), which maps frequently used data structures into a common representation (Davis 2005a, b, 2006). AVCL itself is expressed using the Extensible Markup Language (XML), which in turn provides consistent naming and annotation of data elements, validation of data values, and data compression superior to conventional techniques.

AVCL can express three distinct types of robot information, summarized as follows.

- *agenda-based mission descriptions* for defining area search and threat avoidance, enabling consistent application of planners that implement navigation and search algorithms
- *imperative task-oriented mission scripts* for explicitly directing robot actions using open-loop or closed-loop control, and
- *telemetry-data recording* of state variables and sensor information recorded during the conduct of a mission.

Domain differences between underwater, surface and airborne robot operations are carefully distinguished in order to strictly maintain unambiguous semantics. Import and export routines for both mission scripts and telemetry data have demonstrated that this common-denominator approach can work compatibly for multiple robot types.

The primary motivating problem described in this paper is the consistent conversion and usage of dissimilar robot mission data as AVCL telemetry files.

Telemetry recording. Current work explored in this paper focuses on the recording and conversion of telemetry data into AVCL, XML-based data compression, data communication using file transfer or Web services, automatic preparation of post-mission analytic products, and then automatic uploading of online mission archives for visualization study and long-term reference.

Unprocessed telemetry data can be produced either by robots or high-fidelity simulations. Various telemetry formats can be imported into AVCL syntax by the AUVW tool, either as a mission proceeds or upon reporting completion of mission results. Consistent naming of individual data elements enables resulting AVCL telemetry files to be consistently readable over the long term, even if recorded-data formats change on a day-to-day basis during real-world operations.

Figure 3 shows an example excerpt of the AVCL telemetry format, showing a single set of state-vector values at a given timestamp. Units for each element and attribute are implicit and do not need to be reported since the AVCL schema strictly constrains each data type.

```
<SampledResults>
  <UAVTelemetry timeStamp="6235.231">
    <GeographicPosition description="30.01901583333471,-85.54705610955124">
      <XYPosition x="2116.87" y="1246.85" />
    </GeographicPosition>
    <VerticalPosition altitudeAGL="722.5" altitudeMSL="697.73" />
    <Orientation phi="359.0259717482776" psi="15.412564689019145"
       theta="329.9197157556318" />
    <BodyCoordinateVelocity p="-2.521014298575622" q="4.870141258611998"
       r="-9.854874076250159" />
  </UAVTelemetry>
</SampledResults>
```

Figure 3. AVCL telemetry excerpt expresses arrays of floating-point data in a self-describing XML format.

Figure 4 shows the complete AUV Workbench interface, highlighting the Geographic Information System panel. Displayed data is from NPS Unmanned Air Vehicle (UAV) mission testing off Panama City Florida. This mission was conducted from Tyndall Air Force Base (AFB) during the June 2007 AUV Fest, sponsored by the Office of Naval Research (ONR) and hosted by Naval Surface Warfare Center (NSWC) Panama City.

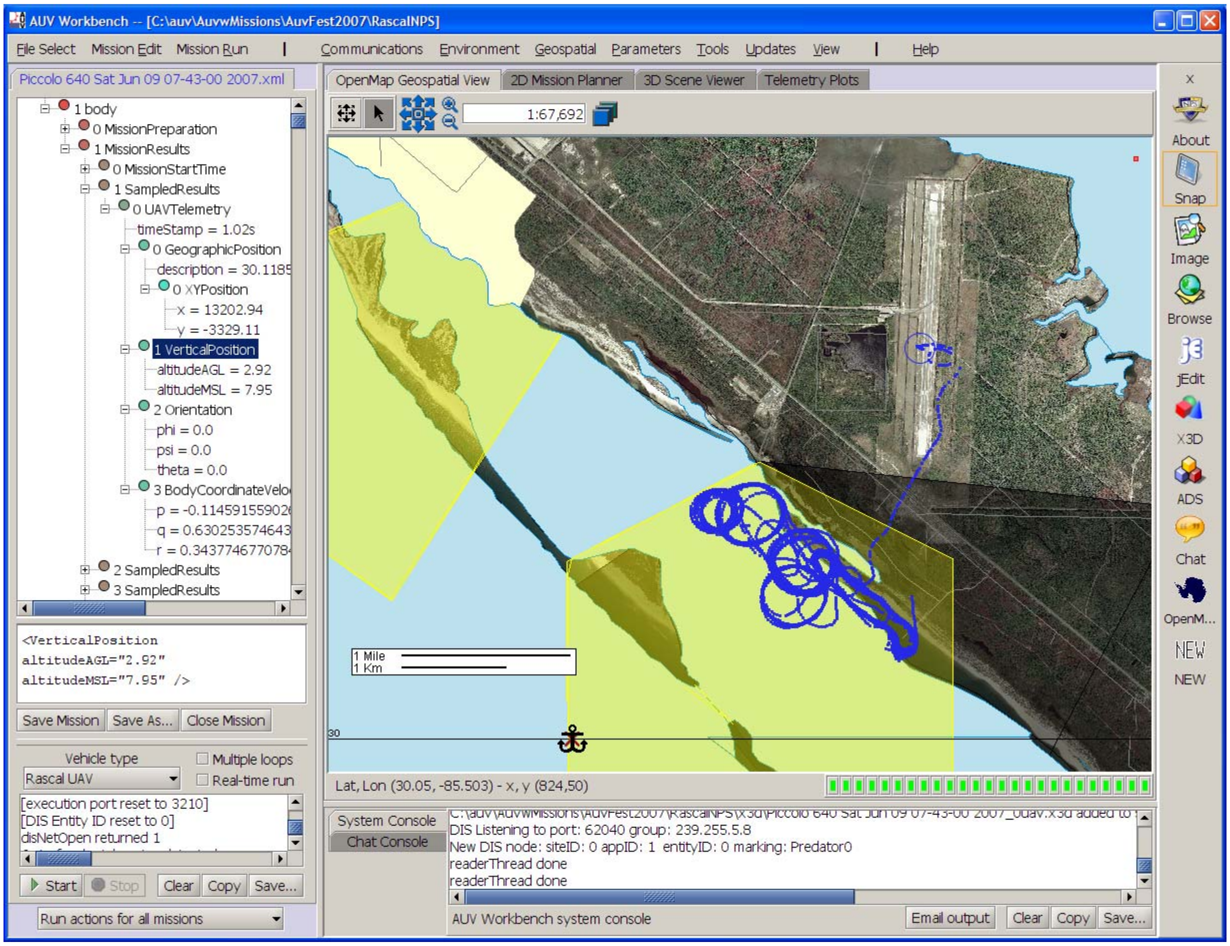

Figure 4. Geographic Information System (GIS) view shows NPS Rascal UAV at AuvFest 2007.

The same data records are shown rendered in the 2D Mission Planner display in Figure 5.

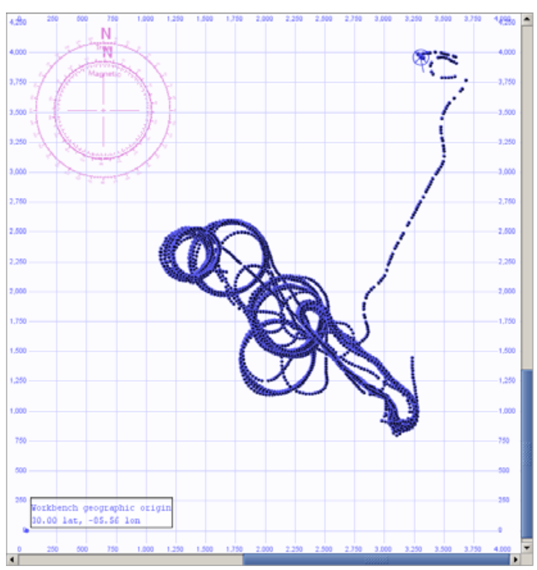

Figure 5. Tactical 2D mission display of recorded telemetry data is shown in meters for precise measurement.

Telemetry-derived data products. Typically most teams developing new robots are repeatedly adjusting and modifying plot-producing scripts to accommodate frequently changing telemetry data formats. Improved availability of telemetry data means that improved post-mission data products are possible. Originally the Workbench used either the C++-based *gnuplot* package or *Matlab*-compatible plain-text outputs to produce plots. State-variable plotting routines are now provided as an embedded capability, in the telemetry plot display pane, using the Java-based *JFreeChart* packages. Example plots are shown in Figure 6 and Figure 7.

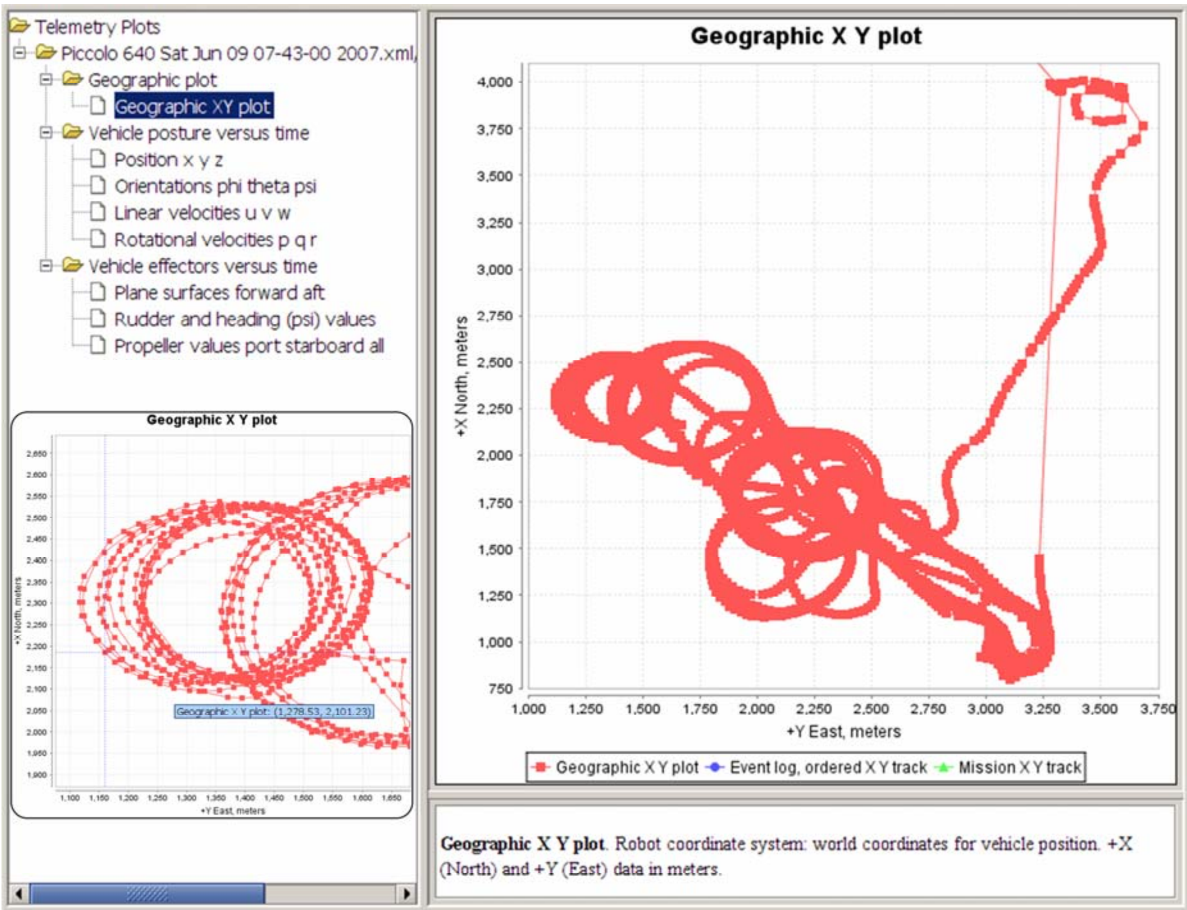

Figure 6. Geographic plot of X-Y telemetry data, with lower-left inset showing closeup detail of data points.

Telemetry data import and conversion. Although few robot telemetry formats are identical, most are similar. Typically line-oriented, a single entry starts with a timestamp value and is followed by a sequence of position, orientation, velocity, acceleration and sensor data. Some telemetry file formats start with a comment line indicating the intended order of values. The vast majority of values tend to be floating point.

Telemetry file formats often change over time, sometimes even during the course of a single day's tests, since the data-recording requirements for different experiments may vary. Such variations, while simple to program, can easily become the cause of robot-software error since recompilation at sea tends to be error prone. Of course such changes similarly make it quite difficult for playback and data-plotting software to remain stable. Thus the generation of data plots (such as those shown in Figure 7) requires corresponding software changes.

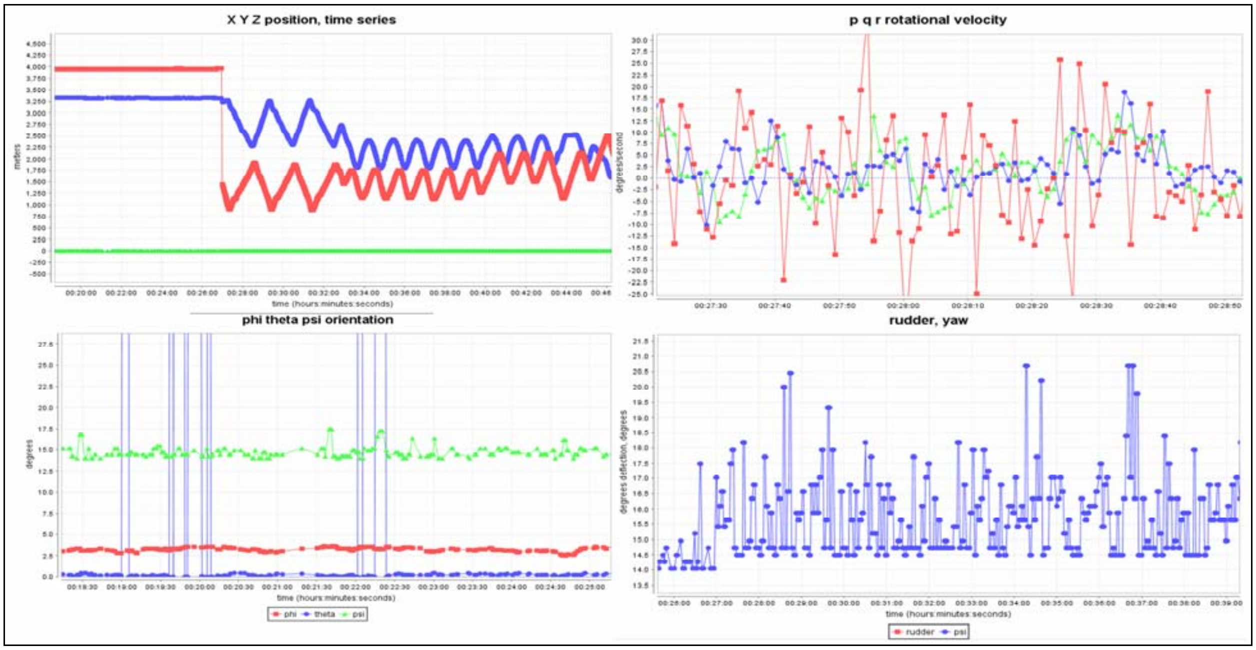

Figure 7. Selected telemetry plots created using the JFreeChart package, with user customizations.

Some files are recorded as plain text (which are easy for humans to read but often quite massive), while other telemetry files are recorded as a binary serialization of the data being recorded. The Import Telemetry panel includes both types of importers, shown in Figure 8. In order to account for improvements and changes over time, both the original data files and the derived AVCL telemetry files are typically retained in mission archives. Creating a new importer for a different format is not difficult. Less than a day is usually required to adapt a prior importer and then perform testing to match a new vehicle's telemetry. UAV, Unmanned Underwater Vehicle (UUV), and Unmanned Surface Vehicle (USV) formats are supported.

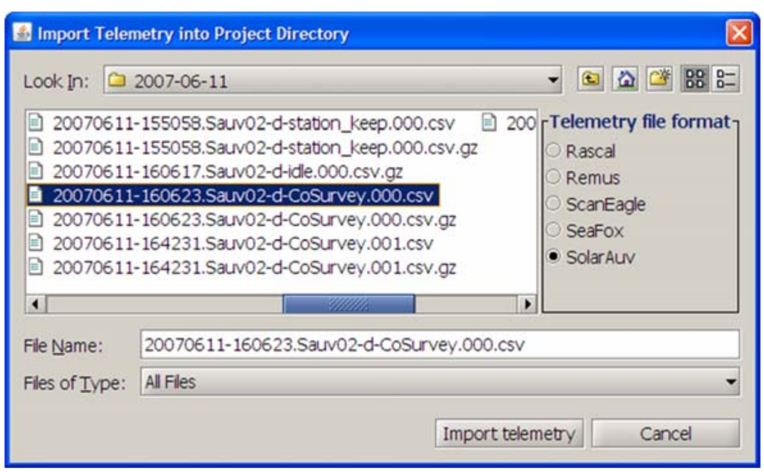

Figure 8. AVCL Import Telemetry panel currently supports five robot-telemetry formats: Rascal and ScanEagle UAVs, Remus and Solar UUVs, and SeaFox USV.

Post-import inspection of the comma-separated-values (.csv) data format reveals that the telemetry data has indeed been converted into AVCL XML form. Figure 9 is an excerpt that shows an alternate tree-based view of this archivable AVCL telemetry mission.

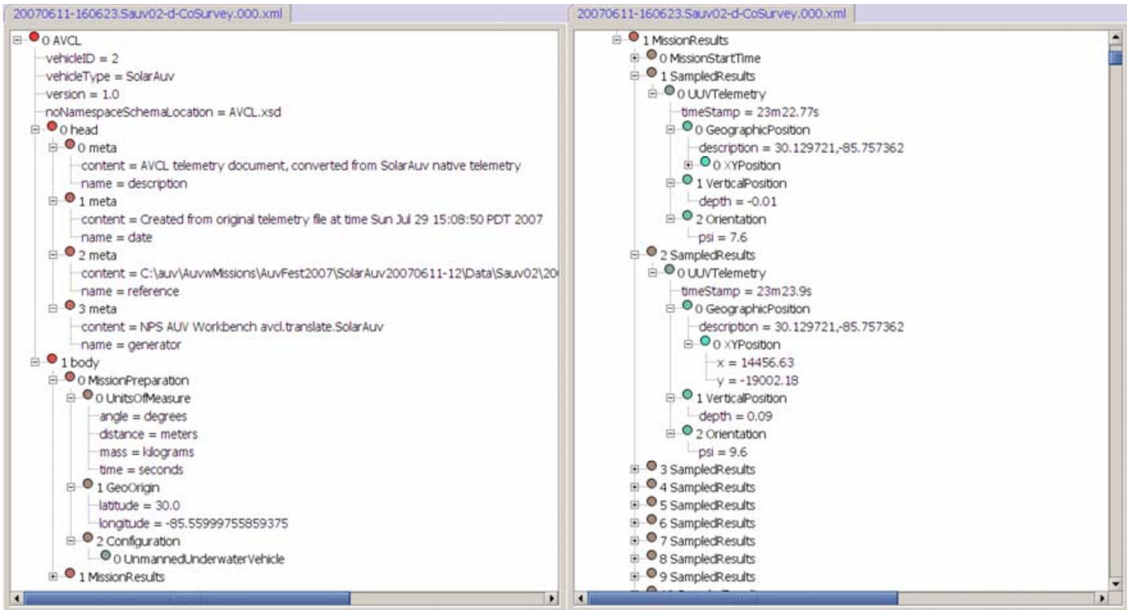

Figure 9. Excerpted tree view of SolarAUV telemetry data converted from comma-separated-value (.csv) format into plain-text AVCL telemetry XML.

Environmental information. Unmanned vehicles are typically limited in power capacity, which in turn restricts both endurance and ability to maneuver in the presence of wind and waves. Realistic values for key environmental parameters enable better mission rehearsal, and can also help match modeled values to recorded telemetry values during mission reconstruction. Figure 10 shows the environmental data-source selection panel, along with the configuration panel for setting constant-value settings for wind, sea state, ocean current, etc. Future work is expected to include the ability to directly utilize robot-collected environmental data, perhaps encoded using Sensor Markup Language (SensorML), an XML-based language for describing the geometric, dynamic, and radiometric properties of dynamic in-situ and remote sensors.

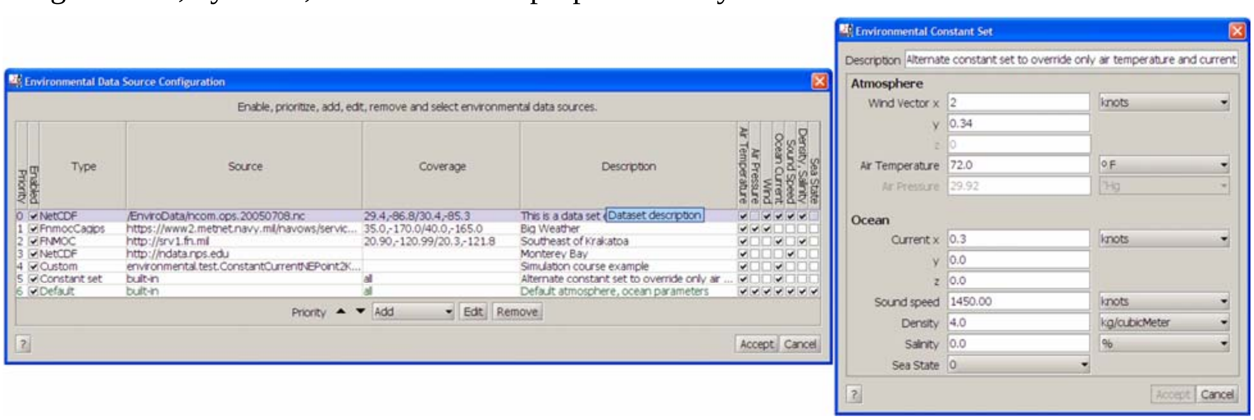

Figure 10. A variety of environmental data sources are selectable and configurable, enabling high-fidelity mission rehearsals to better match expected and actual results.

Detailed analysis and evaluation of at-sea missions becomes possible by superposition of commanded mission orders, high-fidelity simulation of dynamics response to parameterized control systems, environmental effects, and actual recorded telemetry tracks. Retrieval of relevant environmental datasets (again via Workbench-supported file transfer or web service) allows high-fidelity simulation across a range of operating conditions.

Correlation of such track plots into a larger real-world context is provided by construction of large-scale 3D visualizations corresponding to each robot mission. Data-driven XML-stylesheet conversions of AVCL telemetry produce X3D visualizations of completed mission tracks. Users can adjust playback speed and navigate freely to study mission details. Because X3D is not a platform-specific application but rather a presentation language for the Web, composition of X3D models is easily accomplished without requiring reprogramming. Available assets to populate such replays include numerous X3D models from the SAVAGE model archive and bathymetry/terrain from the X3D Earth project. This combination of numerous interactive visualization capabilities can significantly improve overall operator understanding, providing broader situational awareness of the many relevant actors in each robot's environment.

Telemetry-based mission replay. Two-dimensional (2D) plots with GIS and mission-planning layouts are helpful for viewing planned missions and cumulative results. Nevertheless it is important to remember that most robot interactions are inherently three-dimensional (3D) in nature. Even the simplest real-time control of robot motion tends to result in 3D changes to robot position and orientation. Sensor coverage is typically described by a 3D volume. Furthermore, propagation of sonar is spatial and nonlinear due to refractions and reflections underwater. Thus the AUV Workbench includes an X3D view to provide more intuitive displays of robot motion and sensor interaction. Relevant X3D features are listed in Figure 11.

As shown in Figure 1, computational physics results are communicated to the X3D display via either the IEEE Distributed Interactive Simulation (DIS) protocol, or an experimental DIS-XML encoding passed over Extensible Messaging and Presence Protocol (XMPP) chat channels. This means that shared X3D views for multiple robots running on multiple AUV Workbench instances are possible. Mission-rehearsal and telemetry-playback animation can proceed either in real time or using a fixed-interval speedup rate.

The Scenario Authoring and Visualization for Advanced Graphical Environments (SAVAGE) project is an archive of military and robotics X3D models. SavageDefense is a similar archive providing access-controlled X3D models that are designated For Official Use Only (FOUO) by U.S. government projects. Together these provide a steadily growing resource for X3D models under an unrestricted-use open-source license.

Current bathymetry locations of interest include Monterey Bay California, Panama City Florida and a short list of other locations. Preparation of matching X3D geometry and GIS image-layer assets can be time consuming and difficult. The X3D Earth project (http://www.web3d.org/x3d-earth) is a working group effort with the vision and mission to make it easier to create and use 3D spatial data, by promoting spatial data use within X3D via open architectures. The design and production of full-globe X3D data archives for terrain, bathymetry, cartography and other data overlays is already in progress. X3D Earth assets are expected to eventually make AUV Workbench usage at arbitrary locations a fairly easy process.

X3D is an initiative to leverage 3D as digital media as easily as text and 2D graphics. It provides the technology to enable customers to view, modify, customize and reuse 3D visualizations in web applications, or on network devices from cell phones to supercomputers.

- **Industry Standard**. X3D is a royalty free, ISO-ratified format for quick and easy sharing of real-time 3D models, visual effects, behavioral modeling, and interaction. Using X3D, high-quality 2D, 3D and video information can be easily incorporated into technical publishing, maintenance manuals, websites, mashups, database applications, visual simulations, navigation systems and many other professional and consumer uses.
- **Collaboration**. With X3D, 3D data can be seamlessly integrated with the rest of a company's business processes. 3D information created by technical departments can be accessible to the extended enterprise. Numeric and geospatial data can be accessed using intuitive, interactive 3D visualizations.
- **Interactive**. X3D allows a user to associate behaviors and dynamic scripts with 3D objects so that others can interact with them, like playing a video game.
- **Easy to Adopt, Reliable for Archiving**. X3D provides an XML encoding so that interactive 3D and multimedia content can be read or written using standard XML tools and can be integrated seamlessly into any XML enabled application or web service. This enables any application to add components for 3D visualization, augmented reality and collaboration. Because X3D is an open XML standard, it is not dependent on the continued existence of any one organization. Assets exported to or created in X3D will be usable at any future date.
- **Extensible**. Developers can exploit X3D component architecture and XML schema to implement it in their own applications or extend/customize the schema to add vertical market-specific capabilities or custom data to the format.
- **Suitable for Networks**. X3D enables users to share live 3D data quickly and easily in a lightweight format. The binary encoding offers extreme compression while preserving geometry and is therefore well suited for networked consumption of 3D. File sizes are up to 90 percent smaller than those of existing formats so they can be loaded and transferred rapidly on the Web or cell phones for use in real-time collaborative systems.
- **Easy to Share**. X3D is an interchange format for integrated graphics and multimedia to enable content producers to share 3D, interactivity, and physics between applications.

Figure 11. What is X3D? Overview of X3D attributes and capabilities from http://www.web3D.org

Since AVCL missions and X3D scenes are both XML-based languages, converting data from one to another is straightforward using an Extensible Stylesheet Language for Transformations (XSLT) stylesheet. Figure 12 shows the typical dataflow for retrieving robot geometry and visualization widgets from the Savage X3D model archive, which are then superimposed over relevant underwater bathymetry and land-side terrain.

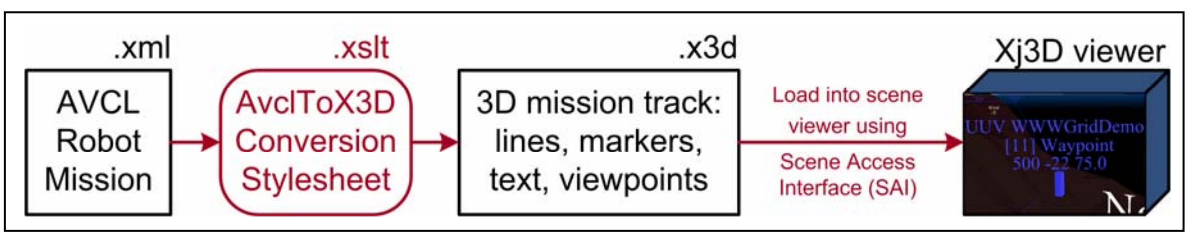

Figure 12. Conversion of XML-based robot AVCL data into labeled X3D scene geometry is performed by an Extensible Stylesheet Language for Transformations (XSLT) stylesheet.

Agenda missions. AVCL agenda missions are declarative, stating goals and constraints, rather than being imperative like mission scripts which explicitly direct a robot to follow a sequence of steps. The operator is able to define launch and recovery points, areas to search, search objectives, and avoidance areas. The subsequent goal following each search area can vary, depending on whether or not the search was successful. Thus, for example, if a mine is not found in the first area of interest then a secondary area can be tasked, while successful completion of any search might result in immediate return to the recovery point.

Interestingly, the search planner which evaluates each declarative agenda mission produces a mission script as an output. Thus planned missions can be run on any robot supporting the imperative AVCL mission-script commands, without requiring planner sophistication to be run on board. Similarly, the output of running mission scripts is telemetry output, regardless of whether the mission is conducted in the real world or in simulation. Thus AVCL telemetry outputs are a valuable product from each of the robot missions.

Figure 13 shows AVCL XML excerpts and a planner-output simulation track for an example mission that demonstrates use of a launch position, search-goal areas, avoidance areas and a recovery position.

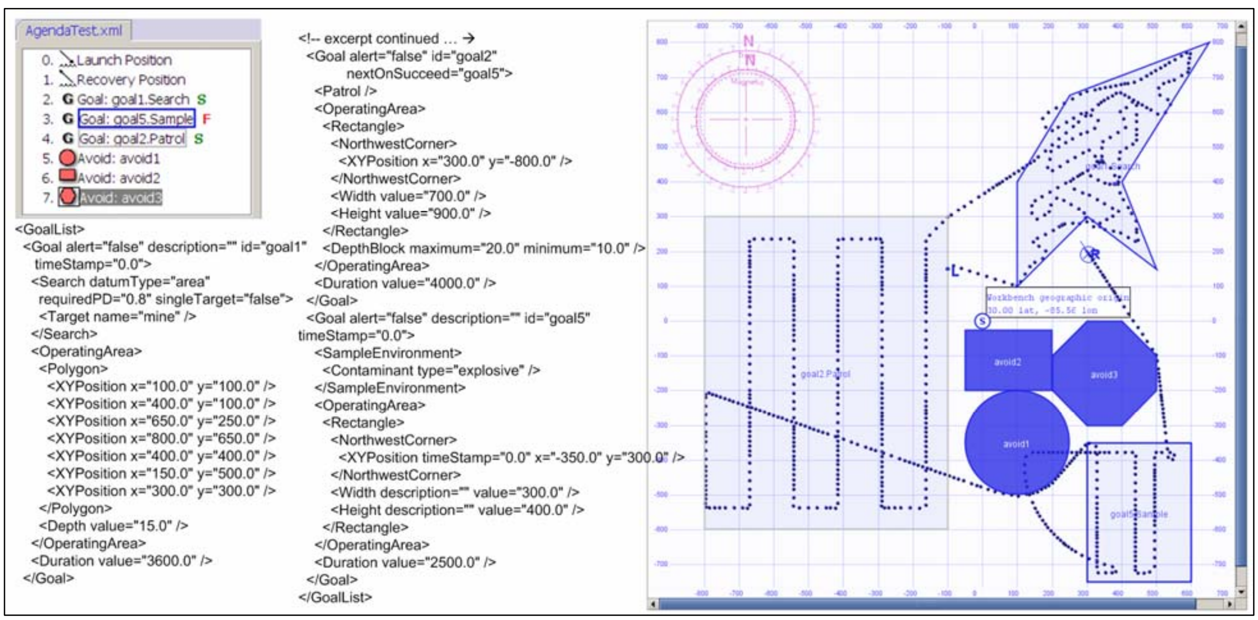

Figure 13. Declarative agenda test mission demonstrates planning results for a declarative mission that includes a launch point, multiple search areas, multiple avoidance areas and a recovery point.

Data compression. The voluminous nature of robot telemetry typically requires some form of compression to make archiving manageable. Transmission over low-bandwidth links is another possible application, but even compressed data can be quite large, and so low-bandwidth links are typically reserved for command and response message passing. Data compression concerns include size, which relates to storage requirements, and computational cost, which relates to processor capacity and power consumption. Thus multiple interrelated tradeoffs must be considered.

For underwater robots, which are severely limited by bandwidth and power-consumption constraints, the large sizes of both compressed and uncompressed telemetry data typically preclude transmission while operating. Three possible use cases are seen: radio transmission while at the surface, docking-station data synchronization while charging batteries, and post-mission download following robot recovery. Thus high bandwidth is a reasonable assumption for most download scenarios, and large-scale compression can be considered as primarily a prerequisite for initial collection and subsequent server archiving of mission data.

Although XML-ization of numeric data typically increases telemetry file size (often by a factor of two or three), the added structure and selective precision of AVCL enables significant telemetry improvements nevertheless. The XML Schema-based Binary Compression (XSBC) package compresses AVCL telemetry for data transfer, providing data-reduction ratios and decompression speed superior to zip and gzip algorithms. All such conversions are performed as post-production work to reduce impact on the robots themselves, minimizing the delicate balance of data storage, processing performance, power consumption, and communications throughput that already exists within each robot.

Figure 14 shows example compression results. Current efforts by the World Wide Web Consortium (W3C) Efficient XML Interchange (EXI) working group to produce a formal recommendation for XML compression are likely to provide even greater compression ratios.

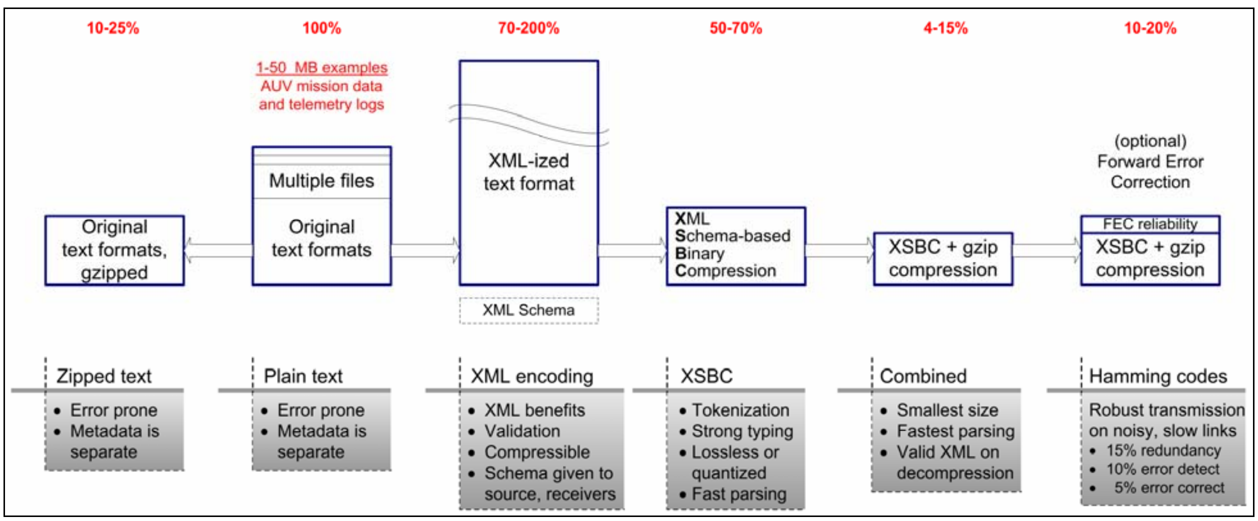

Figure 14. XML-aware compression is capable of smaller file sizes and faster decompression than .zip or .gzip algorithms. AUV telemetry example results shown here are demonstrated using XML Schema-based Binary Compression (XSBC).

Software and data-archive availability. A large set of resources are available for robot testing and software development using the AUV Workbench.

- Website online at https://savage.nps.edu/AuvWorkbench
- Concurrent Versioning System (CVS) source control viewable online at http://xmsf.cvs.sourceforge.net/xmsf/AuvWorkbench
- Mailing list (and message archive) for users and developers available at https://www.movesinstitute.org/mailman/listinfo/auvworkbench
- An issue tracker (also known as a bug tracker) is used to report problems and define new feature requirements, viewable at https://www.movesinstitute.org/bugzilla

- Autoinstaller for Windows and other Java-enabled platforms online at https://savage.nps.edu/AuvWorkbench/install.htm
- Chat channel available at xmpp://conference.savage.nps.edu/auvw
- Help system includes descriptive web pages and a large set of relevant papers, presentations, theses and dissertations.

Autoinstallers are rebuilt at approximately monthly intervals, but software updates are usually occurring at a much more rapid rate. Nightly developmental builds are available through the AUV Workbench interface, allowing end users to keep track with the most recent developments being discussed on the mailing list, if desired. Only the necessary changes are downloaded, making the Workbench automatic updater a relatively lightweight process. Additional autoupdate capabilities are provided for GIS data and X3D model archives. These should be invoked following initial installation.

A number of AUV Workbench mission library archives are currently available online, shown in Figure 15.

AuvFest2007

- RascalNPS.132235ZJUN2007.zip (30.4 MB)
- RascalVirginiaTech.132203ZJUN2007.zip (30.4 MB)
- SeaFox20070609.132300ZJUN2007.zip (6.3 MB)
- SeaFox20070610.132316ZJUN2007.zip (6.1 MB)
- SolarAuv20070605.111921ZJUN2007.zip (1.9 MB)

MissionTranslationTests

- RascalTestMissions.020231ZJUN2007.zip (1 MB)
- RemusTestMissions.070653ZJUN2007.zip (856 KB)
- ScanEagleTestMissions.122015ZJUN2007.zip (437 kB)
- SeaFoxTestMissions.122016ZJUN2007.zip (319 KB)
- SolarAuv20070605.081923ZJUN2007.zip (1.9 MB)

Figure 15. Archived mission libraries at https://savage.nps.edu/RobotTelemetry

Recent work. The June 2007 master's thesis by Dennis Monroe, *Off-Board Robots using the Autonomous Unmanned Vehicle Workbench*, performed a comparative assessment of U.S. Navy master plans for UAV, UUV, and (in final-draft form) USV robotics. Monroe then compared aggregate master-plan requirements to AUV Workbench capabilities, in order to assess overall software relevance and also prioritize planned work for future development. Of additional note is that this work found excellent conceptual consistency between the UUV, UAV and (draft) USV master plans. The thesis abstract follows.

> The Autonomous Unmanned Vehicle Workbench (AUVW) is an ongoing project at NPS that allows rehearsal, real-time control, and replay of diverse autonomous unmanned vehicle (AUV) missions. The AUVW increases the

situational awareness of operators while allowing operators to learn valuable insights regarding a robot's performance before, during, and after a mission.

This thesis examines a variety of authoritative strategic plans for autonomous vehicles to determine functional mission requirements that autonomous vehicles are expected to be performing in the near future. Excellent agreement on tactical needs and requirements was found among these diverse documents. A series of exemplar missions corresponding to specific requirements are presented as a way to explore and evaluate different tactical capabilities. These missions are then compared to the current capabilities of the AUVW by planning, running, and evaluating them in the Workbench. Although the AUVW is a powerful tool, it still lacks some functionality to make it tactically usable. Nevertheless, perhaps two thirds of the necessary capabilities are already supported in the Workbench and further capabilities can be feasibly integrated. The result of this work is a roadmap for future work to add functionality so that the Workbench can thoroughly perform user tasks in all mission areas.

Ongoing work. A variety of capabilities continue to be designed and implemented to support field operations, graduate research, and educational course work. AUVW telemetry recording, transfer and playback supported multiple independent robots in the June 2007 AUV Fest exercises. Ongoing masters and doctoral research work continues to record mission preparations and results in AVCL for repeatable verification by others. Finally the AUVW is being used to support a new graduate-level course in Unmanned Systems, allowing novice operators to effectively monitor robot operations and record meaningful mission metadata. Chat capabilities using the Extensible Messaging and Presence Protocol (XMPP) for rapid exchange of text, message and file products are likely to eventually enable collaborative control among multiple unmanned systems, loosely supervised by multiple operators. Further work is also needed to show how AVCL capabilities might productively extend the Joint Architecture for Unmanned Systems (JAUS). Hopefully the ready availability of compatible comparable mission results from a variety of underwater, surface and airborne vehicles will support improved progress for the entire robotics-research community.

References.

AUV Workbench home page, https://savage.nps.edu/AuvWorkbench

Autonomous Vehicle Command Language (AVCL) references, online at https://savage.nps.edu/Savage/AuvWorkbench/AVCL/AVCL.html

Brutzman, Donald P., *A Virtual World for an Autonomous Underwater Vehicle*, Ph.D. Dissertation, Naval Postgraduate School, Monterey California, December 1995.

Brutzman, Don, Healey, Tony, Marco, Dave and McGhee, Bob, "The Phoenix Autonomous Underwater Vehicle," chapter 13, *AI-Based Mobile Robots*, editors David Kortenkamp, Pete Bonasso and Robin Murphy, MIT/AAAI Press, Cambridge Massachusetts, 1998.

Brutzman, Don, and Daly, Leonard, *X3D: Extensible 3D Graphics for Web Authors*, Morgan Kaufmann Publishers, 2007. http://x3dGraphics.com

Davis, Duane, "Automated Parsing and Conversion of Vehicle-Specific Data into Autonomous Vehicle Control Language (AVCL) using Context-Free Grammars and XML Data Binding," *14th International Symposium on Unmanned Untethered Submersible Technology (UUST)*, Autonomous Undersea Systems Institute (AUSI), Durham New Hampshire, 21-24 August 2005.

Davis, Duane, and Don Brutzman, "The Autonomous Unmanned Vehicle Workbench: Mission Planning, Mission Rehearsal, and Mission Replay Tool for Physics-based X3D Visualization," *14th International Symposium on Unmanned Untethered Submersible Technology (UUST)*, Autonomous Undersea Systems Institute (AUSI), Durham New Hampshire, 21-24 August 2005.

Davis, Duane T., *Design, Implementation and Testing of a Common Data Model Supporting Autonomous Vehicle Compatibility and Interoperability*, Ph.D. Dissertation, Naval Postgraduate School, Monterey California, September 2006.

JFreeChart plotting library, online at http://www.jfree.org/jfreechart

Joint Architecture for Unmanned Systems (JAUS), http://www.jauswg.org

Monroe, Dennis, Off-*Board Robots using the Autonomous Unmanned Vehicle Workbench*, Masters Thesis, Naval Postgraduate School, Monterey California, June 2007.

Norbraten, Terry D., *Utilization of Forward Error Correction (FEC) Techniques with Extensible Markup Language (XML) Schema-based Binary Compression (XSBC) Technology*, Masters Thesis, Naval Postgraduate School, Monterey California, December 2004.

SensorML home page, http://vast.uah.edu/SensorML

World Wide Web Consortium (W3C) Efficient XML Interchange (EXI) Working Group, http://www.w3.org/XML/EXI

X3D Help and X3D examples, online at http://www.web3d.org/x3d/content/examples/help.html

Technology keywords. AUV robotics, virtual environments, distributed control, X3D.

Acknowledgements. The work reported in this paper includes contributions by Terry Norbraten, Don McGregor, Mike Bailey, Jeff Weekley and ENS Dennis Monroe USN of the NPS Modeling, Virtual Environments and Simulation (MOVES) Institute. NPS students in the Unmanned Systems course provided extensive feedback while using the Workbench to document deployed robot testing. Collaboration efforts with Steve Chappell and Rick Komerska of the Autonomous Unmanned Systems Institute (AUSI) supported integration of Solar AUV (SAUV) telemetry. This work is supported in part by the NPS Undersea Warfare Research Center and OPNAV N87 Undersea Warfare.