Video Summary – Neptus, Command and Control Infrastructure for Heterogeneous Teams of Autonomous Vehicles

Paulo Sousa Dias, José Pinto, Rui Gonçalves, Gil Gonçalves, João Borges Sousa, Fernando L. Pereira

Abstract – This video shows a brief overview over Neptus, a command and control infrastructure for heterogeneous teams of autonomous vehicles. Having different types of vehicles at our laboratory and from our partners, there was an increasing need to create a common infrastructure to all these systems. Additionally, a tool to support the entire mission life cycle (Planning, Execution, Review and Dissemination) was lacking. Neptus was created to provide vehicle independence and seamless inter-systems communications. Currently, Neptus has been already tested with Remotely Operated Vehicles, Autonomous Underwater Vehicles, Unmanned Air Vehicles, Autonomous Surface Vehicles and Wireless Sensor Networks. Some of these systems were operated simultaneously by various operating consoles that were sharing the same communication infrastructure. The received data was being relayed to a web server that allowed for the real-time mission following by using a common web browser.

AT the Underwater Systems and Technology Laboratory (USTL), we are currently working with different types of robotic vehicles. Our objective is to have heterogeneous teams of vehicles cooperating in order to achieve shared objectives. A common infrastructure was needed to decouple the details of vehicle control and provide inter-systems communication. *Neptus* is thus being developed to answer these needs.

Neptus is composed by a set of modular software components like the Map Editor, Mission Planner, Mission Processor, Console Builder, Variable Tree, Renderer2D, Renderer3D, and others. These modules can be used by developers to build *Neptus* applications. This is especially useful when it comes to integrate new vehicles in the framework because new specialized consoles can be built rapidly.

We have adopted XML for data representation in *Neptus*. This enables us to define a grammar for every data file and to specify the exact file format to be expected from potential users. XML can also be filtered and transformed into different formats like text, HTML or any kind of native mission file formats for existing vehicles. An eXtensible Stylesheet Language Transformations (XSLT) stylesheet gives the transformation rules from XML to the vehicle's mission language. This facilitates vehicle inter-operability and the integration of new vehicles. When we add a new vehicle to *Neptus* we must specify a new vehicle file in XML format.

The *Neptus* design supports concurrent operations. Vehicles, operators, and operator consoles come and go. Operators are able to plan and supervise missions concurrently. Additional consoles can be built and installed on the fly to display mission related data over a network.

The internal *Neptus* event communication system is based on a tree structure, where nodes indicate the subject of data values in leafs. *Neptus* visual components can become listeners of a single variable (tree leaf) or of a defined variable domain (tree branch). Whenever a message arrives, its data is stored in the tree at the right branch according to the XML definition of the message and the listeners are informed of the incoming network middleware data. In a similar way, output data is sent to middleware by *Neptus* console components through the same variable tree. The variable tree system is also used for event communication between *Neptus* local components.

We used *Neptus* to operate two Wireless Sensor Networks and two vehicles (AUV and ASV) in the NATO Swordfish exercise which took place in May 2006 at Tróia (Portugal). This was done in cooperation with the Portuguese Navy. There was one operator per vehicle and multiple consoles that subscribed to the data published by the vehicles and sensors. Data was published live to the Internet.

The *Neptus* framework has already proven invaluable in operational deployments with ROVs, AUVs, ASVs, and WSNs running different operating systems and using inter-operated communication networks (Wi-Fi, wired Ethernet, acoustic modems, ZigBee, etc.). This is in part because of its modular design and of the underlying communications infrastructure. The ability to create new specialized applications through the reutilization of existing components is very appreciated by developers. Heterogeneous vehicles and sensors are easily integrated into the *Neptus* framework and data is transparently shared across operational consoles.

REFERENCES

[1] Paulo Sousa Dias, R. Gomes, J. Pinto, S. L. Fraga, G. M. Gonçalves, J. B. Sousa and F. Lobo Pereira, "Neptus – A framework to support multiple vehicle operation", Today's technology for a sustainable future, OCEANS Europe 2005, Brest, France, June 20-23, 2005.

[2]   Paulo Sousa Dias, R. Gomes, J. Pinto, G. M. Gonçalves, J. B. Sousa and F. Lobo Pereira, "Mission Planning and Specification in the Neptus Framework", Humanitarian Robotics, ICRA 2006 IEEE International Conference on Robotics and Automation, Orlando, Florida, USA, May 15-19, 2006.

[3]   Paulo Sousa Dias, J. Pinto, R. Gonçalves, G. M. Gonçalves, J. B. Sousa and F. Lobo Pereira, "Mission Review and Analysis", Fusion 2006, Florence, Italy, 10-13 July 2006

[4]   J. Pinto, Paulo Sousa Dias, R. Gonçalves, G. M. Gonçalves, J. B. Sousa and F. Lobo Pereira, "Neptus – A Framework to Support the Mission Life Cycle", MCMC 2006, Lisbon, Portugal, 20-22 September 2006

[5]   USTL/LSTS, <http://whale.fe.up.pt/>, (January 2007)

[6]   Neptus, <http://whale.fe.up.pt/neptus>, (January 2007)