# THE NEPTUS C4ISR FRAMEWORK: MODELS, TOOLS AND EXPERIMENTATION

**Gil M. Gonçalves and João Borges Sousa**
**{gil, jtasso}@fe.up.pt**

*Faculdade de Engenharia da Universidade do Porto*
*Rua Dr. Roberto Frias s/n*
*4200-465 Porto, Portugal*

Abstract: The *Neptus* distributed command and control framework for operations with vehicles, sensors, and human operators in inter-operated networks is presented. This is done in the context of applications, technologies, and field tests.

Keywords: Command and Control Systems, Systems Engineering, Communication Networks, Autonomous vehicles.

## INTRODUCTION – WHAT IS NEPTUS?

This paper presents a C4ISR (Command, Control, Communications, Computing, Intelligence, Surveillance and Reconnaissance) framework for the development of systems for the coordinated operation of networked vehicles (teams of multiple autonomous and semi-autonomous vehicles), sensors, and human operators. The interactions with human operators are classified according to the phases of a mission life cycle: world representation; planning; simulation; execution and post-mission analysis. There are applications for world representation and modeling, planning, simulation, execution control, and post-mission analysis.

This mix-initiative environment supports the activities of the Underwater Systems and Technology Laboratory (USTL/LSTS – http://whale.fe.up.pt/ ). This includes the support to joint operation of multiple underwater vehicles. In the context of this work, operation means the wide variety of possible interactions between the pilot (human or automated) and the vehicles including: pre-mission setup and preparation of a vehicle (or multiple vehicles) mission; real-time data acquisition and visualization; pilot intervention during mission execution (mixed initiative operation); coordinated control of multiple vehicles (fleet control); and post-mission review and data analysis.

## MOTIVATION – WHY DO WE NEED NEPTUS

Researchers and technology developers are devoting significant efforts to the development of concepts of operation for networked vehicle systems. In these systems vehicles come and go and interact through inter-operated networks with other vehicles and human operators. Surprisingly, or not, the role of human operators is receiving significant attention in new concepts of operation for future systems. In fact, this is the reason why researchers and technology developers have introduced the concept of mixed initiative interactions where planning procedures and execution control must allow intervention by experienced human operators.  In part this is because essential experience and operational insight of these operators cannot be reflected in mathematical models, so the operators must approve or modify the plan and the execution.  The design and deployment of mixed initiative frameworks in a systematic manner and within an appropriate scientific framework requires a significant expansion of the basic tool sets from different areas (computation, control, communication, and human factors) and the introduction of fundamentally new techniques that extend and complement the existing state of the art.  The major challenges come from the distributed nature of these frameworks and from the human factors.  This is why we need to couple the development of scientific frameworks with field tests with human operators.

At the Underwater Systems and Technology Laboratory (USTL) from Porto University we have been designing and building ocean and air going autonomous and remotely operated vehicles with the goal of deploying networked vehicle systems for oceanographic and environmental applications (Sousa *et al.*, 2003).  We have developed a framework for the mixed initiative coordination and control of networked vehicle systems and a tool set for deploying applications.  The tool chain comprises the *Neptus* command and control framework (Dias *et al.,* 2005) and the *Seaware* middleware publish/subscribe framework for distributed real-time systems (Marques *et al.,* 2006).

## NEPTUS C4ISR FRAMEWORK – HOW DOES IT WORK AND HOW DO WE USE IT?

Neptus uses the *Seaware* middleware framework for network communication (Marques *et al.,* 2006). *Seaware* is a publish/subscribe framework for dynamic and heterogeneous network environments oriented to data-centric network computation. Publishers and subscribers communicate transparently to any node that is registered in the network. Nodes can either be vehicles that publish sensor data and receive operator commands or consoles that subscribe to the data provided by vehicles and sensors and publish operator commands. Seaware uses the RTPS (Real Time Publish Subscribe) protocol and other forms of network transport.

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. A 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 the vehicle’s command interface in XML format.

There is a set of modular software components – Map Editor, Mission Planner, Mission Processor, Console Builder, Variable Tree, Renderer2D, Renderer3D – which can be used by developers to build *Neptus* applications. This is especially useful when it comes to integrate new vehicles in the framework. The *Neptus* software components and interactions are briefly described next.

The Mission Map Editor (MME) component is a GIS-like application that allows the creation and manipulation of three dimensional world maps. Maps are stored as XML files.

The Mission Planner (MP) component is a top-level application for single and multi-vehicle mission planning.  Mission planning is vehicle specific. There is a library of vehicle models and interfaces. Mission plans are stored as XML files. A mission plan is composed of world maps (links to other XML files), vehicle mission plans (a graph with nodes representing maneuvers and transition conditions among them) and additional data like local information, checklists for operations, and specifications for tests.

The Mission Processor (MProc) component translates *Neptus* mission files (XML) to the native formats used by different vehicles. We use this module to generate vehicle-specific mission files. These are then uploaded to a vehicle for execution.

There are vehicle-specific and mission-specific operational consoles. We use the first to supervise single vehicle operations and the latter to supervise multi-vehicle operations. We use the Console Builder (CB) component to build operational consoles and to tailor these consoles to each vehicle and to each operator. Initially the CB application presents an empty window which serves as a canvas for adding various visual components. The visual components are then connected to variables that might be available on the network. These include, for example, the state of the vehicle, or the motor RPMs. The configurations for each console are saved as XML files for reuse.

The two dimensional (R2D) and three dimensional Renderer (R3D) components are used to visualize the motions of the vehicles and the state of the world. These can be used simultaneously. The Renderer components are connected to VT module in each console to subscribe to the data for visualization. The R3D version proved extremely useful to support the human operator in remotely operated vehicle (ROV) operations. This is because video from the vehicle does not provide enough visual clues for tele-operation in low-visibility areas. The R2D module is quite useful to supervise operations that take place over a large area. Additionally, R2D is also used for map edition, allowing the user to interact with the existing objects (images, paths, marks, etc.).

The Mission Review and Analysis (MRA) component provides support for the analysis of mission data. This includes provisions for replaying missions in a virtual world and also to graph mission variables.

Together, these modules enable the specification of abstract missions with coordinated vehicle plans and world maps (Dias *et al.,* 2006a, b, c).

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. Fig. 1 depicts multi-vehicle interactions under *Neptus* and *Seaware*. There is one operational console for an autonomous surface vehicle (ASV) and another one for a remotely operated vehicle (ROV).

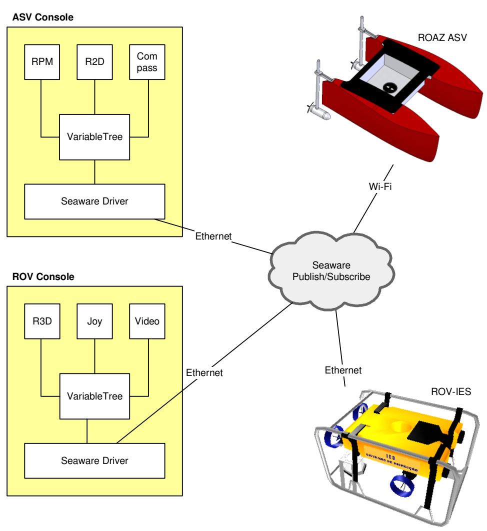


Fig. 1. Interactions under *Neptus*

## OPERATIONS

Field tests are essential to validate developments and to evaluate mixed initiative interactions. We have field-tested *Neptus* in our operational deployments.

We started field tests with single vehicle operations and recently moved into multi-vehicle operations with wireless sensor networks (WSN). In the first field test we used Neptus for mission planning and control of the IES ROV in the inspection of an underwater pipeline. The use of the same map for mission planning and execution greatly reduced the number of human errors. We were able to visualize the mission in simulation and to use the experience acquired in simulation to operate the vehicle in real time. The 3D visualization of the real motions in a virtual world proved quite useful for operations in waters with poor visibility. We tested the mission planning GUI and the generation of mission files through XSLT in operations with the Isurus AUV (Fig. 2) which took place in the Montemor-O-Velho nautical center. This represented a great advance since we used to edit Isurus native mission files by hand. The number of planning errors was greatly reduced with the help of the 2D/3D maps and of the visual aids of our planning GUI. We have also built a new console to track the motions of Isurus with the help of data provided by the acoustic localization system. This console enabled us to evaluate mission performance in real-time. We had to provide consistency checks for displaying data coming from different sources. These developments have been extensively tested with heterogeneous vehicles in missions on the Port of Leixões.

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


Fig. 2. Vehicles support by the Neptus Framework

## CONCLUSIONS AND FUTURE WORK

The *Neptus* framework has already proven invaluable in operational deployments with ROVs, AUVs, ASVs, UAVs 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. The ability to define an abstract mission and to translate the resulting XML by using XSLT is also a much appreciated feature because it allows the integration of new vehicles without changes to the *Neptus* code. In the same manner, the ability to build operating consoles with a GUI is quite important for anyone trying to use *Neptus* to interact with a new vehicle in a new operational scenario.

*Neptus* is a work in progress. New releases incorporate lessons learned from operational deployments. The available functionality is being extended and improved. This includes: a simulation service to support operator training and validation of mission specifications for generic vehicles (currently this is restricted to one ROV); GUI for mission specification in the framework of hybrid automata (currently mission plans have a linear structure); data logging onto a central database which will be accessed by the MRA application for mission revision or through a web page. This will allow to display data gathered anywhere in the world by any vehicle connected to *Neptus*.

## REFERENCES

Dias, P. S., R. Gomes, J. Pinto, S. L. Fraga, G. M. Gonçalves, J. B. Sousa and F. L. Pereira (2005), Neptus – A framework to support multiple vehicle operation. In: *Today's technology for a sustainable future, OCEANS Europe 2005*, Brest, France, June 20-23.

Dias, P. S., R. Gomes, J. Pinto, G. M. Gonçalves, J. B. Sousa and F. L. Pereira (2006a), Mission Planning and Specification in the Neptus Framework. In: *Humanitarian Robotics, ICRA 2006 IEEE International Conference on Robotics and Automation*, Orlando, Florida, USA, May 15-19.

Dias, P. S., J. Pinto, G. M. Gonçalves, R. Gonçalves, J. B. Sousa and F. L. Pereira (2006b), Mission Review and Analysis. In: *Fusion 2006 The 9th International Conference on Information Fusion*, Florence, Italy, July 10-13.

Pinto, J., P. S. Dias, R. Gonçalves, E. Marques, G. M. Gonçalves, J. B. Sousa, and F. L. Pereira (2006c), Neptus – A Framework to Support THE Mission Life Cycle. *7th Conference on Manoeuvring and Control of Marine Craft (MCMC'2006)*, Lisbon, Portugal, from September 20-22.

Marques, E.R.B., G.M. Gonçalves and J.B. Sousa (2006). Seaware: a publish/subscribe middleware for networked vehicle systems. *7th Conference on Manoeuvring and Control of Marine Craft (MCMC'2006)*, Lisbon, Portugal, from September 20-22.

Sousa, J. B., F. Lobo Pereira, P. F. Souto, L. Madureira and E. P. Silva (2003). Distributed sensor and vehicle networked systems for environmental applications. In *Biologi Italiani*, n. 8, pp 57-60.