

Mission Review and Analysis

Paulo Sousa Dias
Rui Gonçalves

José Pinto
João Borges Sousa

Gil M. Gonçalves
Fernando Lobo Pereira

LSTS – Underwater Systems and Technology Laboratory
Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias s/n
4200-465 Porto, Portugal
{pdias,zepinto,gil,rjpp,jtasso,flp}@fe.up.pt

Abstract - *This paper presents the Mission Review and Analysis module of a C3I (Command, Control, Communication and Information) infrastructure - the Neptus framework. This is a mix-initiative environment that it's being developed in the Underwater Systems and Technology Laboratory (USTL/LSTS) with the goal to support the coordinated operation of heterogeneous teams of vehicles. This includes autonomous and remotely operated underwater, surface, land, and air vehicles and people. People perform a fundamental role, not only in the case of remotely operated vehicles, but also with autonomous vehicles where mix-initiative operation is a requirement. The Neptus is very modular and will be heavily based on services with a distributed architecture. This paper focus mainly on the mission review and analysis where the data collected in a mission is prepared for analysis.*

Keywords: Control Networks, Middleware, Systems Engineering, Systems networks, Underwater Vehicles.

1 Introduction

This paper presents the Mission Review and Analysis module of a C3I (Command, Control, Communication and Information) infrastructure for the coordination and control of teams of multiple autonomous and semi-autonomous vehicles. The mission review and analysis module displays the data collected during the missions.

The Neptus framework is a mix-initiative environment that supports the activities of the Underwater Systems and Technology Laboratory (USTL/LSTS). 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.

The USTL is currently operating two underwater vehicles: one underwater ROV (Remotely Operated Vehicle) and one AUV (Autonomous Underwater Vehicle). Shortly the number of the laboratory vehicles

will grow to six with the addition of an UAV (Unmanned Air Vehicle) from the Academy of Portuguese Air Force, one more ROV [1] being developed under project KOS (Kits of underwater operations) and two more AUVs being developed under PISCIS (Prototype of an Integrated System for Coastal waters Intensive Sampling) project. These joint operations, with heterogeneous multiple vehicles, and the need to control the complete fleet in a coordinate manner set the motivation and the requirements for the development of a framework to support the coordinated operation of multiple vehicle.

This operating scenario is very rich and the amount of data collected during a typical mission is huge. The Mission Review and Analysis module helps the users to prepare the data and make post mission analysis right after its conclusion,

This paper is organized as follows. Section II presents the requirements that guided the development of the Neptus infrastructure, the definition of subsystems, components and architecture. Section III describes the architecture and the common operational scenarios of the Missions review and analysis module. Section IV presents a short guided tour of the Mission Review and Analysis. Finally, in section V, conclusions and an outline of future work are presented.

2 The Neptus framework

2.1 Motivation

The Neptus framework is a mixed-initiative environment for the coordination and control of teams of multiple autonomous and semi-autonomous vehicles. This infrastructure (C3I – Command, Control, Communication and Information) is used in the context of the activities of the Underwater Systems and Technology Laboratory (LSTS/USTL). This framework is intended to support the joint operation of multiple autonomous and semi-autonomous vehicles.

As presented in [2] and [3], exists several applications to support the operations but each one giving attention to different aspects of a mission. But for the USTL operations it was needed a more completed and integrated tool. The most promising tool was the AUV Workbench [4] from Naval Postgraduate School (NPS) which allows

the visualization of vehicle behavior by means of simulation of vehicle's physical equations. It also helps the user to see collected data during the mission. Vehicle behavior visualization is ensured by 2D or 3D (VRML) screen and can be monitored in a standard web browser. Although AUV workbench was designed for use with multiple vehicles it does not allow cooperative missions using hybrid systems concepts.

On-going projects at USTL require vehicle interactions in the context of hybrid systems ([3] and [6]). The main target here is to design a framework that allows cooperative mission planning and visualization. Some effort is also being done to integrate sensor networks in future in such a way that missions can be planned or on-line re-planned with the information gathered by these networks [6].

Because of this and more requirements presented in the following section, we decided to build from the bottom a new application expanding preliminary work [7].

2.2 Requirements

The system requirements for the Neptus systems are presented from the user's point of view. The exposition is partially based on IEEE Std. 1362-1998 [8], which is a standard for system characteristics description based on the definition of Concept of Operations (ConOps).

Let us take as example an USTL mission near Porto in the Douro River, Portugal, with two vehicles, the Isurus (Figure 1) and the ROV-IES (Figure 2).

Before the actual mission execution, a site assessment is needed along with the gathering of the mission objectives. In this stage the map and the setup for the mission are prepared. This falls in the *operational setup* that is the first stage of a typical mission life cycle. Then we start with the *mission programming* where the path (or area of operation) of the vehicle(s) is defined and a mission is specified by selecting a pre-defined set of maneuvers and tasks.



Figure 1 : AUV Isurus vehicle mission setup (vehicle from Hydroid Inc. [9] but with some hardware and software redone [10])

There might be a simulation of the actual mission to prevent possible mistakes and accidents. The next stage is the *mission execution* that is the actual execution of the mission on site. Then it's time to analyze the collected

data. This is the *mission analysis* stage. In a typical AUV mission the results are analyzed upon return home but in a ROV mission this analysis can happen on site. Upon this stage follows the *dissemination* stage where the completed collected data and mission report are published and stored.



Figure 2 : ROV-IES starting an inspection mission (vehicle from Deep Ocean Engineering [11] but with the hardware and software completely redone [12])

The Neptus framework [13] should be designed to fulfill the operational mission requirements described above (see Figure 3). Thus, the top level requirements are listed below:

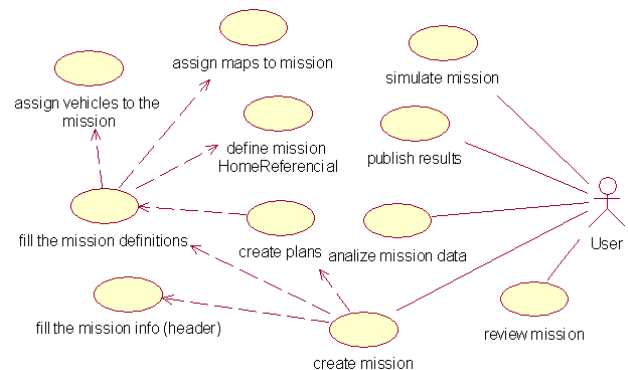


Figure 3 : Requirements use case diagram

- An application to define the environment of the operational mission. Including navigational references, bottom profile, obstacles, and mark points;
- An application for the mission programming;
- A console that establishes the link between the support and the on-board computers. In the ROV class, the console should enable users to send joystick commands and visualize sensors' data in real-time;
- A simulation platform to allow the user to verify the conformity of the mission program. This tool will give the user the chance to debug its own mission plan;
- A tool for mission review and analysis of the sensors' data;
- A repository for the gathered data with associated query services.

2.3 Architecture

The Neptus architecture is a modular one. It is organized in several modules/applications that together create the Neptus environment. In Figure 4 is depicted the main applications: Mission Planner (MP), Mission Console (MC), Mission Reviewer & Analysis (MRA), Mission Data Broker (MDB) and the Multiple Vehicle Simulator (MVS).

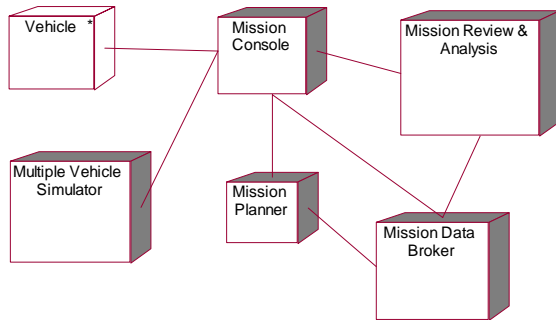


Figure 4 : Deployment diagram

Each of these modules gives support to one or more stages of the mission life cycle. So in the *operational setup* we use MP and MDB in order to layout the mission objectives. In the *mission programming*, in addition to the applications used in the previous phase in order to prepare the mission plans and the site's map, the MVS is used to simulate the mission. Next in the *mission execution* phase we add the use of the MC to setup the mission plans into the vehicles, see the created mission data flow, and in the case of a ROV type of vehicle to pilot it throughout the mission. Then in the *mission analysis* it will be used the MRA to further analyze the mission captured data and to produce the mission reports. In the *dissemination* of the mission results the MDB is used to publish the results.

In Figure 5 can be seen a screenshot of Neptus desktop used in one of our latest missions. In the following section the Mission Review & Analysis is presented in more detail.

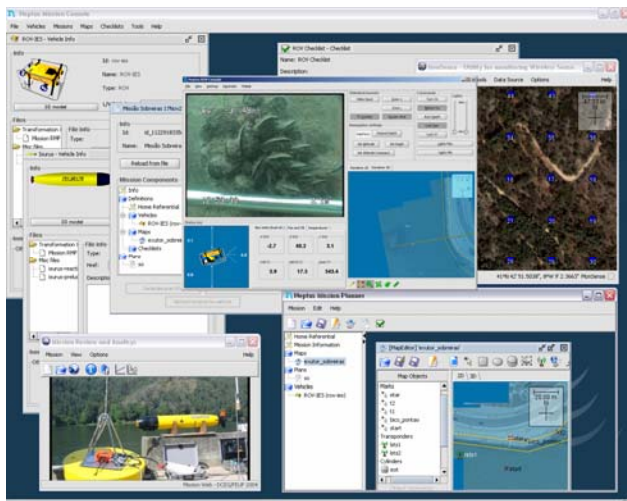


Figure 5 : Neptus desktop environment

3 MRA

Whenever an autonomous vehicle is sent on mission, it collects large amounts of data. It is of great importance to visualize this information in an easy and fast way. This will enable the interaction with the vehicle, in order to guide it accordingly to the mission's objectives.

In the case of Autonomous Underwater Vehicles (AUVs), these data can't be visualized in real time due to the low bandwidth of existing acoustic modems. Practically speaking, AUVs can only communicate with a base station when they are off water or at the water's surface, where common wireless network technologies, like *Wi-Fi*, or Iridium can be used.

The application Mission Review and Analysis (MRA) was developed to help an AUV pilot / commander to view the data that was gathered in a previously executed mission, allowing him to take actions accordingly. The pilot can be near the vehicle or at any other place in the world, using the Internet as the communication mean. Additionally, the developed system can also be used as a catalog of previously executed missions, allowing anyone with an Internet connection to view public data.

3.1 System Requirements

The application was developed with various initial requirements in mind. The application should be accessed by anyone with an internet connection, regardless of its software or hardware platform. This implied that the gathered data should be transferred over the Internet. For this to be possible, an efficient communication protocol to transfer the information should be encountered or created.

The visualization of the data should be precisely stored and represented, allowing a scientific usage of the software.

Despite the large amounts of collected data, the information has to be provided in a way that allows a pilot to control the vehicle operation safely.

The development of the software should, whenever possible, reuse existing open source components that provide desired features.

3.2 System Architecture

The system is composed by various modules that communicate using Web Services.

The client module is a Java™ Swing application that is packaged and deployed using Java Web Start (JWS). This technology allows the launch of the application with a single mouse click over a hyperlink. After the application is started, it downloads any missing or newest packages to the local machine and then runs the main application. The usage of the Java™ language and JWS allowed software and hardware independence along with seamless integration of new improvements to the software.

The application can be used to view binary log files stored in the local or can be used to examine data transferred from the internet.

The Mission Data Broker (MDB) servers store data from previously executed missions and publish this information in the form of Web Services (WS).

There exist WS methods for authentication, listing of available information in a server and to get data from a selected mission.

The Base Station Module (BSM) is also a MDB server but connects directly to a vehicle and publishes the information right after it is retrieved. The gathered data is also stored in a local disk and it can later be transferred to any other MDB server for public visualization.

All the MDB servers are also registered in a central UDDI server, which is used for listing of the existing sources of information. When the client application is launched, it initially connects to one of these servers to show the user which servers are available (Figure 6).

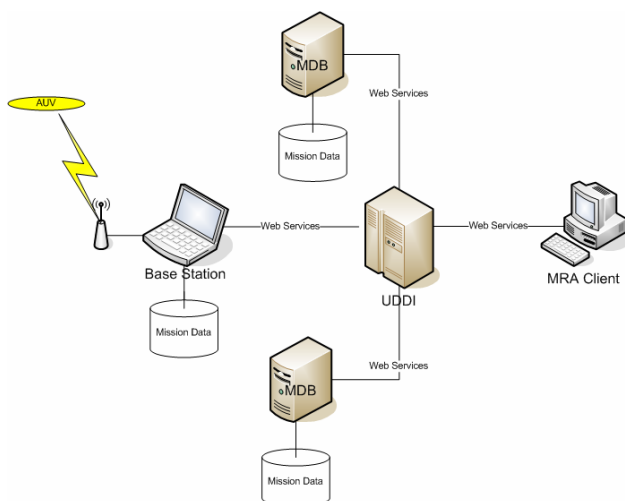


Figure 6: Example Architecture

3.3 Data Visualization

The collected data can be viewed in multiple ways, and given the modular approach to the development many others can be implemented in the future if found necessary.

The path of the vehicle is presented to the user in the form of a two dimensional line path, top-down and side view. Since the vehicle can undergo the same trajectory multiple times and at different depths, there is also the possibility to select the maximum and minimum depths to show in the top down view.

The data collected by the vehicle's sensors can be shown in the form of line plot versus time. These graphs can show data from a single sensor or from multiple sensors, adding the possibility to compare their values.

The collected sensorial data can also be represented in the form of a color map. This map shows the data in colors that are calculated from a selectable gradient. The area to be showed is also selectable, being possible to choose a "slice" of data or a volume to be viewed.

All the mission execution can be replayed in the form of an animation, being possible to see the vehicle's position over time and data that was being collected. There is the possibility to watch the mission in real-time (the replay velocity matches the vehicle velocity) or the time shortened by a factor (2x, 5x, 10x, 100x). A mission

timeline is also present, adding the possibility to advance or rewind the animation.

3.4 Data Sources

The data collected by the vehicle is stored in a binary log file with huge amounts of information, being some of it only for debug purposes like device tensions and inner temperatures.

Our system is able to parse this file and strip the irrelevant data, creating new log files. This is the intermediate format that is stored in the MDB servers and can also be loaded directly to the client application.

Even stripped, the log files can have sizes in the order of megabytes and so we had to take great caution in how to transfer this information safely over the internet.

We chose to transfer the information in the form of Web Services due to the seamless integration that is provided. Client applications can be created in all types of platforms and languages like .NET, J2ME or JavaScript.

We have used an incremental download algorithm in order to correctly download all the data but simultaneously present a rough overview of the data that is being transferred to the user. This feature was very important since it adds the possibility to interact with the vehicle right after the mission data starts to be downloaded (Figure 7).

The algorithm is very simple but efficient: After a mission is selected, the client application sets the number of vehicle states it wants to receive per message: N. Then, every time the MDB server is contacted to provide a "data chunk", it sends N states randomly chosen from the remaining (unsent) states.

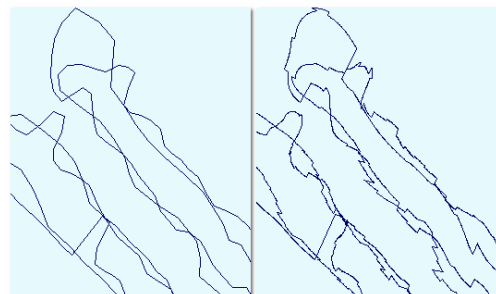


Figure 7: Progressive download of data

3.5 Neptus Integration

This system is also being integrated with the Neptus Framework, currently in development at our laboratory.

This framework provides visualization components like 3D and 2D renderers, vehicle communication protocols and implementations, utilities to create environmental maps, methods to locate any geographical coordinates in the world, among a long list of other functionalities.

After integration, the MRA will cease to be a parallel system, being transformed into a module (or modules) that compose the Neptus Framework [2].

4 Guided tour/case study

The system is already being used in real world missions for reviewing previously executed missions. Some of

these missions have been turned public, allowing almost anyone with an internet connection to access this data.

The client application is deployed using Java Web Start, allowing its installation using a single mouse click (see Figure 8).

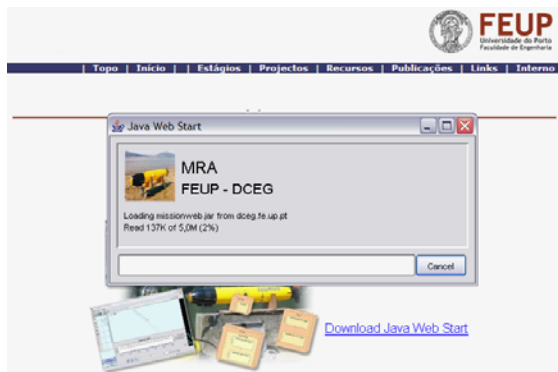


Figure 8: Launching the application using Java Web Start

After launching the client application there are two possibilities to load the mission data: by selecting a public Mission Data Broker and Mission and download the information over the Internet or load a file from the local disk (Figure 9).

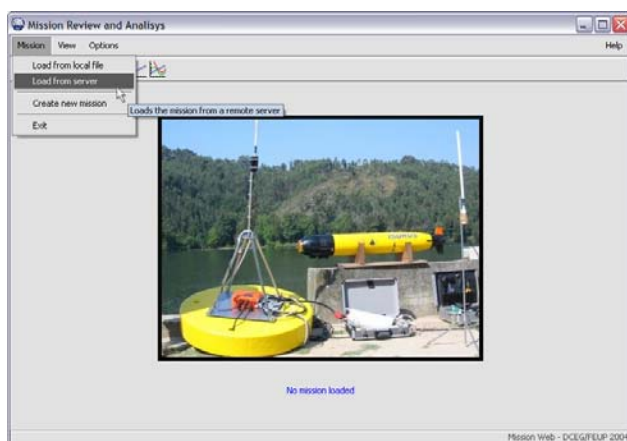


Figure 9 : Loading mission data



Figure 10 : Remote server and mission selection

When a user chooses to load a mission from a remote server he is then presented a dialog where all the currently active MDBs are listed (Figure 10). After selecting one of these servers, the user has access to all the missions that are published by the chosen MDB.

When a mission is selected, the data is then progressively downloaded into the client application, being possible to treat the partially deleted data exactly the same way as if all the data was downloaded.

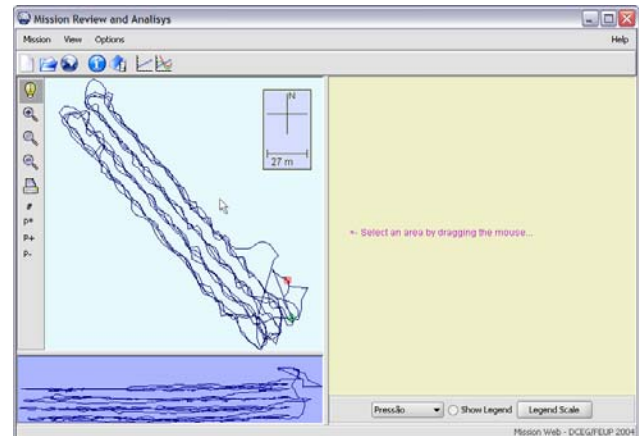


Figure 11 : Representation of the vehicle path

After any amount of data is received, it can be viewed simultaneously in various ways. The vehicle path is presented in the form of two views over the trajectory: A side view and a top-down view (Figure 11). These views can also be used to select a volume (parallelepiped) from the vehicle path. The selected volume can then be shown in the sensor data view (at right) in the form of coloured data grid.

When the volume selection is made, the sensor data is represented in the form of coloured grid. When more than one sensor has been logged, the user can select witch sensor to be rendered by means of a combo box. Another colour maps and grid sizes can also be selected to present the data in the most appropriate way (Figure 12).

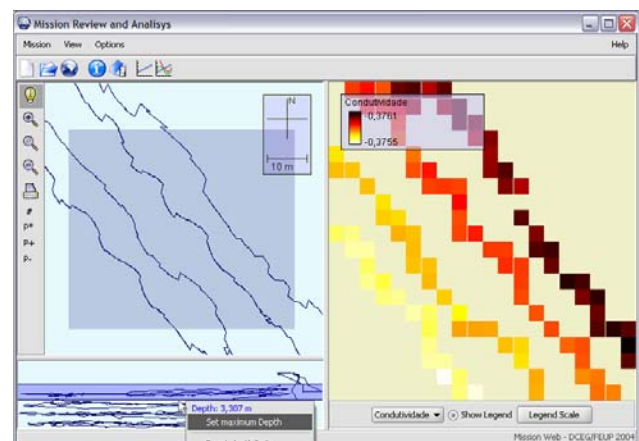


Figure 12 : Selection of the volume to be colored

Another way to view the gathered data is by plotting any of the sensors data versus the time it was collected or versus the location it was gathered (horizontal or vertical axis). Again these graphs can be customized by zooming in and out or mark the points where the data was actually collected (Figure 13).

Maybe the easiest way to view the mission data is by reviewing the vehicle actions and data gathered in real time. This feature is provided in the application by presenting the vehicle position (a dot) and the gathered data in real-time plot similar to an oscilloscope. The

animation can be rendered in real-time or in fast motion, being the time shortened by a chosen multiplier value like can be seen on Figure 14.

Another ways of presenting the data are being developed, most of them will reuse some of the components provided by the Neptus Framework.

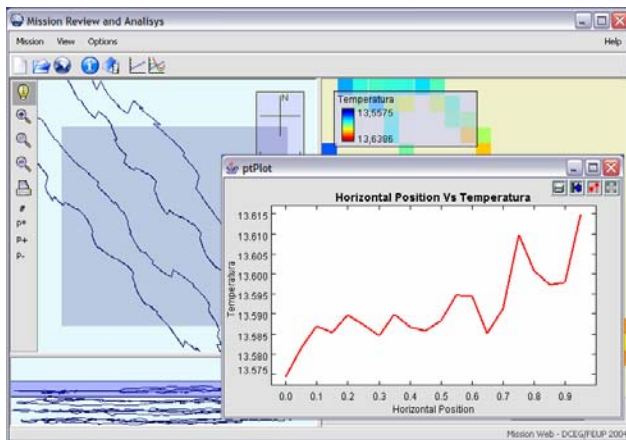


Figure 13 : Plotting temperature versus horizontal position

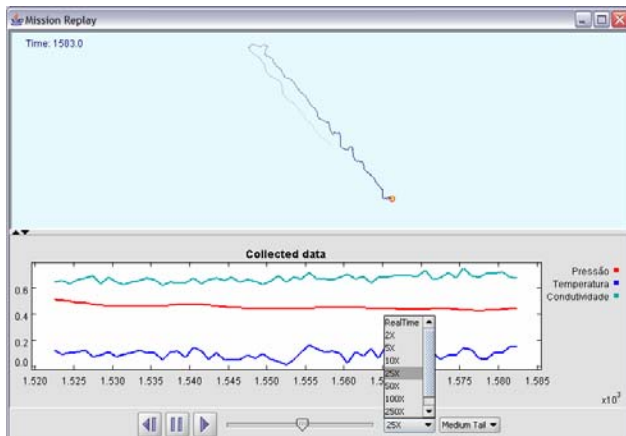


Figure 14 : Animation of the mission execution

5 Conclusions

This paper presented the current state of development of the Mission Review and Analysis Module. This application is part of the Neptus framework, a mixed-initiative environment to support the coordinated joint operation of multiple autonomous and semi-autonomous vehicles.

The developments in the Mission Review and Analysis are currently focusing on the integration and fusion of data from multiple sensors and vehicles in order to assist the user.

Acknowledgments

This research has been partly supported by AdI (Agência de Inovação) under projects PISCIS and KOS.

Paulo Sousa Dias would like to thank the financial support of FCT (Fundação para a Ciência e Tecnologia) in his work.

References

- [1] Rui Gomes, A. Sousa, S. L. Fraga, A. Martins, J. Borges Sousa and F. Lobo Pereira, "A new roV design: issues on low drag and mechanical symmetry", Today's technology for a sustainable future, OCEANS Europe 2005, Brest, France, June 20-23, 2005.
- [2] 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.
- [3] 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 (*in press*).
- [4] Naval Postgraduate School,
<<http://terra.cs.nps.navy.mil/AUV/workbench>> (Janeiro 2006)
- [5] Sérgio L. Fraga, João B. Sousa, A. Girard, A. Martins, "An automated maneuver control framework for a remotely operated vehicle", OCEANS, 2001. MTS/IEEE Conference and Exhibition, Vol.2, Iss., 2001 Pages: 1121-1128 vol.2.
- [6] José Pinto, Alexandre Sousa, Paulo Lebres, Gil Manuel Gonçalves, João Sousa, "MonSense – Application for Deployment, Monitoring and Control of Wireless Sensor Networks", RealWSN '06 (In conjunction with ACM MobiSys 2006), Uppsala, Sweden, June 19, 2006 (*in press*).
- [7] S. L. Fraga, J. B. Sousa and F. L. Pereira, "User-assisted trajectory generation for autonomous underwater vehicles", Marine Technology and Ocean Science Conf. – OCEANS 2003, San Diego, CA, USA, September 22-26, 2003, pp. 1234-1239.
- [8] IEEE guide for information technology - system definition - Concept of Operations (ConOps) document, IEEE Std 1362-1998, Vol., Iss., 19 Mar 1998.
- [9] Hydroid Inc.,
<<http://www.hydroidinc.com/>> (Janeiro 2006).
- [10] Aníbal Castilho Coimbra de Matos, Nuno Alexandre Cruz, A. Martins, Fernando Lobo Pereira, "Development and Implementation of a Low-Cost LBL Navigation System for an AUV", in Proceedings of the MTS/IEEE Oceans'99 Conference, pp.-, 1999
- [11] Deep Ocean Engineering,
<<http://www.deepocean.com>> (Janeiro 2006).
- [12] A. Matos, N. Cruz, J. Borges de Sousa and F. Lobo Pereira, "AUV and ROV Developments at Porto University", IFAC Workshop Modelling and Analysis of Logic Controlled Dynamic Systems, Irkutsk, Lake Baikal, Russia, July 30 - August 1, 2003 (CD-ROM)
- [13] Neptus,
<<http://whale.fe.up.pt/neptus>> (Janeiro 2006)