

Autonomous vehicles in the response to maritime incidents

J. Borges de Sousa, G. Andrade Gonçalves
F. Lobo Pereira

*Electrical and Computer Engineering Department, Faculty of Engineering, Porto University,
R. Dr. Roberto Frias, 4200 Porto, Portugal (Tel: 351-22-5081690; e-mail: jtasso@fe.up.pt).*

Abstract: The future role of autonomous vehicles in the emergency response to maritime incidents is discussed and a framework for their integration into existing response plans is proposed. This is done in the context of the developments on autonomous vehicle systems from the Underwater Systems and Technologies Laboratory from Porto University.

Keywords: Autonomous vehicles; mixed-initiative planning and control; control and coordination frameworks; maritime incidents.

1. INTRODUCTION

Ocean and air going autonomous vehicles are capable of performing “dirty, dull and dangerous” (3D) missions. 3D missions arise not only in military operations, but also in civilian operations, such as those concerned with emergency response to maritime incidents. However, the current best practice of response plans to maritime incidents does not include an explicit role for autonomous vehicles.

This paper discusses roles for autonomous vehicles in the emergency response to maritime incidents and proposes a framework for their integration into existing response plans. This is done in the context of the Maritime Incident Research and Innovation Network (MARINE¹) project funded by the Interreg III B Programme «Atlantic Area».

The paper is organized as follows. In section 2 we present an overview of autonomous vehicles technologies that can be used in maritime incidents. In section 3 we briefly discuss action plans for maritime accidents and identify the role and challenges associated to the future utilization of these technologies in maritime incidents. In section 4 we propose a framework for this integration and discuss technical approaches to several problems. In section 5 we discuss the conclusions.

2. OVERVIEW OF AUTONOMOUS VEHICLES

The last decade has witnessed unprecedented interactions between technological developments in computing, communications and control, and the design and implementation of networked vehicles and systems. These developments enable engineers to design new systems, and in turn, the implementation of these systems leads to a better understanding of the underlying technological issues, and to the development of new concepts of operation for new, or existing applications. Emergency response and preparation to maritime incidents is one of those applications, and the one of

interest to the MARINE project. This requires an in-depth understanding of the current capabilities and limitations of ocean and air going vehicles. We discuss this in the context of the developments from the Underwater Systems and Technology Laboratory from Porto University (USTL)². At the USTL we have an inter-disciplinary approach for the design, implementation and deployment of networked vehicle systems for oceanographic and environmental field studies. A brief description of our systems follows.

2.1 Vehicle systems

Isurus is a modified version of a Remus class (Remote Environment Measuring UnitS) class autonomous underwater vehicle (AUV), built by the Woods Hole Oceanographic Institution, MA, USA, for low cost and lightweight operations in coastal waters. *Isurus* has a torpedo shaped hull about 1.6 meters long, with a diameter of 20 cm and weighting about 35 kg in air. The maximum forward speed is 4 knots, being the best energy efficiency achieved at about 2 knots. The maximum operating depth is 200m. For navigation *Isurus* uses a PNI TCM2 digital compass and Long Baseline acoustic beacons (20-30 Khz). In the standard configuration, *Isurus* is equipped with a Ocean Sensors 200 conductivity, temperature and depth sensor, a Wet Labs optical backscatter sensor, a Marine Sonics side scan sonar and an Imagenex altimeter. The communications suite includes a Benthos acoustic modem and WiFi.

Our most recent vehicle, the Light Autonomous Underwater Vehicle (LAUV) is a prototype of a low-cost submarine for oceanographic and environmental surveys. It is a torpedo shaped vehicle made of composite materials (110x16 cm) with one propeller and 3 (or 4) control fins. The *LAUV* has an advanced miniaturized computer system running modular controllers on a real-time Linux kernel. It is configurable for multiple operation profiles and sensor configurations. In the standard configuration it comes with a low-cost inertial

¹ <http://www.project-marine.eu/>

² <http://paginas.fe.up.pt/lsts/>

motion unit, a depth sensor, a LBL system for navigation, GPS, GSM and WiFi.



Fig. 1. Sea and air going autonomous vehicles from Porto University

IES is a modified Phantom 500 remotely operated vehicle (ROV) model from Deep Ocean Engineering. The innovations include on-board power and computer systems (to minimize the number of wires in the tether cable), tele-operation and tele-programming modes and an integrated navigation system which fuses data from an external acoustic system and internal navigation sensors. The inspection package includes a video camera (Inspector, zoom 12:1) mounted on a pan and tilt unit (Imenco) with 600W of light (DSP&L). The navigation package includes a Doppler Velocity Log (Argonaut/Sontek), an Inertial Unit (HG1700 /Honeywell), a Digital Compass (TCM2/PNI) and acoustic beacons (20-30KHz).

KOS is a modular ROV for underwater inspection and intervention in three basic configurations. It is made of composite materials to reduce weight and for added performance. It has advanced thrust and power control for operations in difficult environments. Dimensions: 120 x 70 x 90 cm; weight: 90 kg; 5 Seaeeye SI-MCT01 Thrusters; max operating depth: 200m; Power: 3Kw. It has the same inspection and navigation packages installed on the IES ROV plus a 2-degree of freedom robotic arm for interventions.

Swordfish is a 4.5m long autonomous surface vehicle (ASV) based on an ocean-going catamaran (200kg) equipped with two Seaeeye SI-MCT01 thrusters and a docking station for AUVs. Power is provided by batteries. It has a GPS unit and a miniature IMU for navigation. *Swordfish* is a powerful communications node with WiFi and broadband radios, GSM and a Benthos acoustic modem for underwater communications. The standard payload includes a wireless video camera and a distributed meteorological station based on a Mote sensor network. It is used both as a gateway buoy for underwater communications and as docking station for autonomous underwater vehicles.

Lusitânia is an unmanned air vehicle (UAV) based on a remotely controlled model airframe equipped with one OS 91-FX, 15cc, 2.9HP, 2 stroke engine. *Lusitânia* is equipped with the Piccolo autopilot (Vaglianti *et al.* 2004), with a small video camera and with Telos motes (with

meteorological sensors optimized for use on a UAV platform). The camera can be remotely controlled, and provides the operator with a video feed in real-time. This is done through a 2.4GHz wireless transmission system with a range of 8Km.

Antex-M is a family of UAV platforms developed by the Portuguese Air Force Academy (AFA). ANTEX-M is a 6 meter wingspan platform with a 220cc, 22HP, 2 stroke 3W engine for a payload weight exceeding 30kg. ANTEX-M X02 is a 1:2 scale model of ANTEX-M with a 15cc, 2Hp, 4 stroke Saito100 engine, for a maximum payload takeoff weight of 7Kg. The ANTEX-M UAV family has a standard computational and sensor configuration. It is configured to fly with two different autopilots (Piccolo and Micro-Pilot).

In addition to autonomous vehicles, we have been developing drifters to monitor ocean currents. In their simplest version, our drifters consist of a simple computer system and a GPS/GSM board installed on a waterproof ocean-resistant container. The position of the drifter is monitored in real-time with the help of GSM/GPS communications.

2.2 Mixed initiative interactions

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 (Girard *et al.* 2004). Surprisingly, or not, the role of human operators is receiving significant attention in the development of concepts of operation for future robotic 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 (Sousa *et al.* 2007). Also, it is impossible to design (say) vehicle and team controllers that can respond satisfactorily to every possible contingency. In unforeseen situations, these controllers ask the human operators for direction.

The USTL approaches the problem of mixed initiative interactions in the context of the USTL command and control framework. The operational deployment of this framework is done with the help of the USTL toolset. This is briefly described next.

2.3 Command and control framework

We use the concept of manoeuvre – a prototype of an action/motion description for a vehicle – as the atomic component of all execution concepts. We abstract each vehicle as a provider of manoeuvres and services. A simple protocol based on an abstract vehicle interface governs the interactions between the vehicle and an external controller: the external controller sends a manoeuvre command to the

vehicle; the vehicle either accepts the command and executes the manoeuvre, or does not accept the command and sends an error message to the controller; the vehicle sends a done message or an error message to the controller depending on whether the manoeuvre terminates successfully or fails. This protocol facilitates inter-operability with other platforms. Actually, the same protocol is used on-board each vehicle for autonomous execution control (Almeida *et al.*, 2006).

The control architecture consists of two main layers: multi-vehicle control and vehicle control. Each layer, in turn, is further decomposed into other layers. The vehicle control architecture is standard for all the vehicles (see Figure 2). The multi-vehicle control structure is mission dependent. We use our vehicle abstractions in multi-vehicle controllers that may reside in some remote locations or in some other vehicles. This leads to different control configurations and strategies. We do this in the framework of high level hybrid automata in which the state variables include sets of vehicles and links to these vehicles.

The vehicle control architecture consists of four layers: low-level control, manoeuvre control, vehicle supervision and plan supervision. The concept of manoeuvre plays a central role in this architecture: it facilitates the task of mission specification, since it is easily understood by a mission specialist; it is easily mapped onto self-contained controllers, since it encodes the control logic; and is a key element in modular design, since it defines clear interfaces to other control elements. Each manoeuvre controller is encoded as a hybrid automaton. Each transition is labelled with a guard, the condition under which the transition can take place, and an event, the message sent out when the transition is taken. We allow the operator to interact with the execution of some manoeuvres. This is encoded in the manoeuvre automaton.

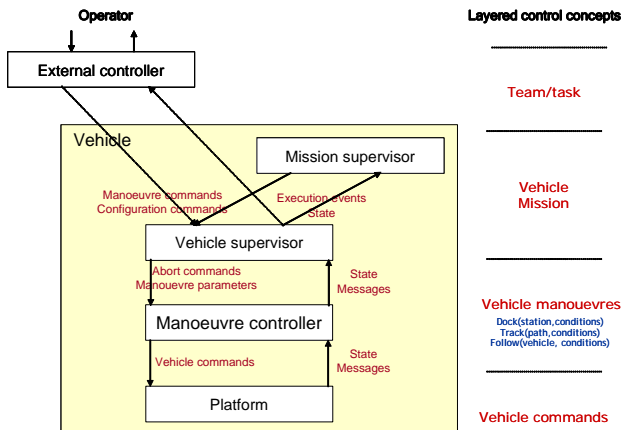


Fig. 2. USTL generic vehicle control architecture

There is a library of manoeuvres/manoeuvre controllers. Example manoeuvres include: *Hover*, *FollowTrajectory*, *Surface*, *Goto*, *Rows* and *Tele-operation*. The addition and deletion of manoeuvre to the library does not require changes to the control architecture (Sousa *et al.* 2004).

The vehicle supervisor controls all of the onboard activities and mediates the interactions between an external multi-vehicle controller or the internal mission supervisor and the manoeuvre controllers. The vehicle supervisor is encoded as

a hybrid automaton. It has an internal state representing the state of all physical components in the vehicle and of all software modules. Mixed initiative control is allowed by the enabling and disabling of transitions in the automaton. This can be done by an operator if communication with the vehicle is available. The supervisor accepts manoeuvre commands (or commands to abort the current manoeuvre) and passes the manoeuvre parameters to the corresponding manoeuvre controller for execution, and signals back the completion or failure of the manoeuvre. The basic structure of the automaton encoding the vehicle supervisor is very simple. It has 4 states: *Init*, *Exec*, *Error* and *Idle*. The vehicle supervisor is initially in the state *Idle*. Upon the reception of a manoeuvre specification it creates a manoeuvre controller if the enabling condition is true. When the manoeuvre is completed it goes to the *Idle* state again, otherwise, the transition to the *Error* state is taken, and it sends an *error(id)* event to the plan supervisor, and the plan fails. The vehicle supervisor maintains the state of all the components of the vehicle and also encodes error handling logic.

The plan Supervisor commands and controls the execution of the mission plan. It commands the vehicle supervisor to trigger the execution of a manoeuvre specification and waits for the acknowledgment of its completion, or for an error. When it receives the acknowledgement, the plan supervisor selects the next manoeuvre to be executed. The process is repeated until the plan is successfully terminated, or it fails. The plan also has provisions for mixed initiative control by allowing the operator to enable and disable some of the transitions.

2.4 Software tool set

We are using the *Neptus/Seaware/DFO/Dune* tool set, developed at USTL, to support the implementation of this command and control framework. These tools and the technologies they use are described next.

Neptus is a distributed command, control, communications and intelligence framework for operations with networked vehicles, systems, and human operators (Dias *et al.*, 2005) (Dias *et al.*, 2006). 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. *Neptus* 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. *Neptus* has a Console Builder (CB) application. This facilitates the addition of new vehicles with new sensor suites to *Neptus*. *Neptus* supports the control of several UAVs, AUVs and ASV concurrently. There is a *Seaware* node per vehicle and per operator console (one per vehicle). Each vehicle node is characterized by a topic domain identifying the vehicle to allow for a set of messages to be exchanged with the corresponding operator console. *Neptus* implements a subset of the NATO standard STANAG 4586 (NSA 2006) for communications with unmanned air vehicles.

Seaware is a middleware framework that addresses the problem of communications in heterogeneous environments with diverse requirements (Marques et al., 2006). *Seaware* adopts publish/subscribe based messaging, defined by anonymous message exchange between data subscriptions and publications. Each application dynamically registers itself, specifying the topics it wishes to publish and subscribe, without the need to know in advance who its peers are or where they are located.

At the core of *Dune* is a platform abstraction layer, written in C++, enhancing portability among different CPU architectures (Intel x86 or compatible, Sun SPARC, Intel XScale/StrongARM and IBM PowerPC) and operating systems (Linux, Sun Solaris 10, Apple Mac OS X, FreeBSD, NetBSD, Microsoft Windows 2000 or above and QNX 6.3). *Dune* can be extended in the native compiled programming language C++ or using an interpreted programming language such as Python or Lua.

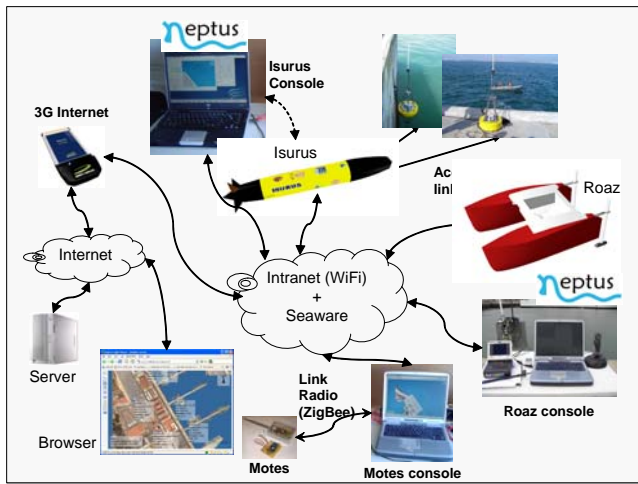


Fig. 3. Example of USTL operations and tools integration.

We are currently developing a programming language called DFO (“Data Flow Objects”) for embedded control software specification (Marques 2007). DFO allows the specification of objects with “data flow”. The aims of DFO are basically two-fold: firstly, to allow the definition of “data flow objects” with sound and clear semantics; secondly to provide good performance and abstract details of native support in a particular platform, operating system or use of programming language for specification of “user code”. Initially, DFO is being developed to support a set of core language constructs for: input-output data flow; mode switching (in the sense of a finite state machine); and object composition in sequential, concurrent or hierarchical fashion. The core properties we wish to attain from derived programs are determinism in execution and high performance and low memory footprint.

2.5 Challenges to deployments

The deployment of ocean and air going autonomous vehicles is not a trivial matter. There are several reasons for this: 1) lack of legal frameworks for the operation of these vehicles (e.g. UAVs); 2) lack of standards for inter-operability; 3) lack of concepts of operation that include a role for autonomous

vehicles; 4) lack of guidelines for best practice; 5) relative infancy of the technologies and of the commercially available solutions; 6) lack of trained operators; and 7) high cost of commercially available solutions. This state of affairs should not prevent us from deploying autonomous vehicle systems. On the contrary, we are learning important lessons from our deployments (Marques et al., 2007).

3. MARITIME INCIDENTS

There are a significant number of types of maritime incidents in which the operation of autonomous vehicles may have a tremendous impact. Here, we will focus on incidents for which emerges the need to manage a maritime pollutant spill. Typically, this requires the integration of data with incident-specific requirements for the affected region (CEDRE, 2004). One such data set concerns the a priori characterization of the region which might encompass:

- Environment characterization. This may include the natural geophysical, meteorological or biological dynamics (currents, waves, winds, life web, sediments, sea states, etc.), renewable and non-renewable resources (either pristine or subject exploitation), and existing anthropogenic impacts, among others.
- Risk assessment of pollutant spills. This may include the characterization of maritime traffic (volume, substances, types of vessels, social responsibility, etc.), port activities, pipe-lines safety, land originated effluents (agriculture, cattle, industry, storages, etc.), as well as the mapping of incident risks, and of the environment sensitivity to incident damages.
- Characterization of stakeholders. For each stakeholder, this encompasses role, environmental footprint, sensitivity to incident impact, stakes, cost-benefit analysis, insurance policies, cultural and political issues, etc.

Once a baseline characterization is available, the management of maritime pollutant spills requires appropriate monitoring policies. The two general types of monitoring activities that are typically considered are discussed next.

3.1 Operational monitoring

The operational monitoring concerns essentially the gathering of episodic data targeted to define and guide the response to maritime incidents. The main operational monitoring features are: a) Determination of a small set of indicators key to the specification of the intervention objectives: pollutant type, pollutant quantity, geographic evolution, distribution of response resources; b) Well established methods and straightforward implementation; c) Limited scope (area, time and scale); and d) Real-time constraints.

Typically, the response to a maritime incident requires timely and pertinent feedback data – rapid data gathering and easy processing for a meaningful intervention – with a sampling rate and accuracy lower than those required for scientific characterization. Typical information relevant for an operational monitoring activity involves estimates of the size and of the distribution of the spill, type of substance,

environmental features, resources at risk, specific spill response needs, underlying legal and insurance issues, as well as sensitive political and cultural issues, etc. The specification of operational monitoring activities involves the consideration of six phases (AMSA, 2004):

1. Pre-Spill. It consists in the determination of the true baseline for operational monitoring ("control" sites, time and space scales).
2. Post-Spill. It identifies the pre-impact conditions. This is a naturally reactive activity and designed and implemented at short notice and within a very short time span in order to collect a "snapshot" of pre-impact conditions.
3. Post-Impact. It determines the pre-cleanup conditions by monitoring the impacted shorelines, waters, and resources. Obviously, the execution of this phase is required to take place in a very short time span.
4. Cleanup. This phase occurs concomitantly with the cleanup activity and has to be performed in real-time. The response strategies should be designed having in mind the: a) greatest potential of recovery; b) most sensitive (socially, environmentally, economically) areas; and c) the minimal global (pollutant spill plus cleanup effort) negative impact.
5. Post Cleanup. This phase determines the pre-response termination. It consists in the monitoring of resources, water and shorelines after the completion of the cleanup activities but before the response has been terminated. Usually, these programmes are short-term and short time span.
6. Post-Response. This phase consists of all monitoring after the formal termination of the response.

3.2 Non-operational monitoring

The relevance of non-operational monitoring for maritime incidents concerns the environment characterization to assess, as rigorously as possible, the impact of the incident and of the response, as well as the recovery progress. The results for this kind of monitoring are pertinent for both the scientific and the legal contexts. Clearly, non-operational monitoring general involves: a) More complex objectives since it aims at the quantification of impact, establishment of cause-effects relations of spill and cleanup, and it may encompass short and long term; b) Wider scope since more detailed models, taking into account natural variations and "higher-order" effects are required; and c) Stronger involvement of highly trained experts since sophisticated data analysis and in-depth studies are required to reach solid conclusions.

4. INTEGRATION OF AUTONOMOUS VEHICLES IN EMERGENCY RESPONSE AND PREPARATION

The role of autonomous vehicles in emergency response to maritime incidents is far from being fully understood. Technological advancements allow us to envision the design of systems which could have not been imagined before. But design is a process that greatly benefits from previous experience, in this case that of deploying autonomous

vehicles as part of the response plans to maritime incidents. This is why we propose to use our expertise at the USTL to develop scenarios against which practitioners and technology developers can refine concepts of operation, and identify the main difficulties facing the operational deployment of autonomous vehicles in maritime incidents.

4.1 Storyline

Consider the case of a maritime incident spanning a wide geographical area. With the current technologies, tools and models, it is simply not possible to bring together, in a systematic manner, vehicles, sensors and communication networks from all over Europe to address this problem. This is simply because these vehicles, sensors and communication networks are not design to work together. In what follows, we discuss what could be possibly done in a maritime incident with the tools and technologies from the USTL. We consider, for the purpose of illustrating the main ideas, the case of an oil spill resulting from the collision of two ships in the Gulf of Biscay. The year is 2015.

The response plan from the maritime authority identifies the need for the use of several types of autonomous systems: 1) UAV to track the oil spill from the air and to inspect one of the ships; 2) UAV for local communication relays; 3) drifters to track the currents; 4) AUVs for detailed inspections of a protected marine area; and 5) ASV for a close inspection of one ship.

The maritime authority accesses the USTL database to confirm that the required autonomous vehicles are available with a 12h advance notice. Upon confirmation of availability, the deployment protocol is invoked. A crew from USTL is dispatched together with the vehicles to the site of operations. Meanwhile, the logistical support required for the operation of the vehicles is prepared prior to their arrival. The situation maps are shared with the USTL to develop the deployment plan in coordination with the maritime authority. This takes place in transit to the site of operations. The maritime authority and the USTL crew exchange standardized plans and data for this purpose. The vehicles are unloaded and prepared for the deployment at a staging area close to the site of operations. This is done by elements from the maritime authority following the deployment protocol, with the help of the USTL crew. Operations are run by a joint team, with full integration of communications and data exchanges. The operational deployment is supervised remotely by a team from USTL in Porto, Portugal. The team not only provides technical advice, but also develops and uploads new code to the vehicles.

4.2 Tool set

The USTL toolset is easily extended to accommodate the requirements for this deployment. In what concerns the publish/subscribe framework *Seaware* the required extensions are: 1) new transports to accommodate other means of communication; 2) security levels and encryption; and 3) delay/disruptive tolerant network capabilities. In

concerns the command and control framework *Neptus* the required extensions are: 1) tactical channels for coordination; 2) compliance with inter-operability and data exchange standards; 3) graphical user interfaces for logistical support and operational procedures.

4.3 Control

The control requirements for this deployment are easily described in terms of a set of manoeuvres which, if not available, can be coded in our control framework. This is because of the modular design. In this case, the required manoeuvres include: 1) UAV providing area coverage for communication relays; 2) ASV finding and tracking a ship with the help of vision-guided algorithms; 3) AUVs executing space-filling curves for an exhaustive area search; and 4) UAV tracking an oil spill using vision-based algorithms.

4.4 Concepts of operation

We are still far from being able to design and deploy networked vehicles and systems in a systematic manner and within an appropriate scientific framework. This is why we propose to develop concepts of operation based on mixed initiative operations. This allows intervention by experienced operators, thus increasing the probabilities of success, and contributes to build a body of knowledge which will be fundamental to the development of the next generation of control frameworks with increased levels of autonomy.

5. CONCLUSIONS

The problem of integrating autonomous vehicles in the response plans to maritime incidents is discussed along with scenarios for their future utilization. This is done with reference to the vehicles and technologies from the USTL. The contribution to this discussion is descriptive, not prescriptive, in nature. This paper could potentially be used in a number of ways: to evaluate technologies; to define windows of feasibility for the introduction of new capabilities in the near term on existing systems or for starting new programs and; to identifying key enabling technology development efforts to support now for use in the far term.

In the summer of 2008, the USTL will demonstrate the operation of autonomous vehicles in the response to maritime incidents in a demonstration of the MARINE project. In the envisaged scenario, one source of pollution, a small container, is dropped from one ship in the harbour. The *Swordfish* ASV will run a sonar survey of bottom of the harbour to search for the container. The *LAUV* will be used to map the plume of pollutants. This data will be assimilated with the estimated currents to forecast the evolution of the plume. Drifters will be used to estimate the currents. Finally, the *KOS* ROV will be used to inspect the container. The control and supervision of the operations will be done by a team of experts in a mixed initiative fashion. The team includes harbour specialists and researchers from the USTL.

ACKNOWLEDGEMENTS

This research was partly supported by the Maritime Incident Research and Innovation Network (Marine) project funded by the Interreg III B Programme «Atlantic Area».

REFERENCES

- Almeida, P., G. M. Gonçalves, J. B. Sousa (2006). Multi-UAV Platform for Integration in Mixed-Initiative Coordinated Missions. In: *Procs. First IFAC Workshop on Multi-vehicle Systems (MVS'06)*, Salvador, Brazil.
- AMSA (2003). *Oil Spill Monitoring Handbook*, Wardrop Consulting (Australia) in association with the Cawthron Institute (New Zealand).
- CEDRE (2004). *Oil Spill Waste Management - Operational Guide*, CEDRE, 2004.
- Dias, P., R. Gomes, J. Pinto, G. M. Gonçalves, J. B. Sousa and F. L. Pereira (2006). Mission planning and specification in the *Neptus* framework. In: *Procs. IEEE International Conference on Robotics & Automation*. pp. 3220-3225, IEEE, USA.
- Dias, P., S. Fraga, R. Gomes, G. M. Gonçalves, F. L. Pereira, J. Pinto and J. B. Sousa (2005). *Neptus* - a framework to support multiple vehicles operation. In: *Procs. Oceans '05 Europe*, pp. 963 - 968 Vol. 2, IEEE, France.
- Girard, A., J. B. Sousa and J. K. Hedrick (2005). A Selection of Recent Advances in Networked Multi-Vehicle Systems. In: *Procs of the 1 MECH E Part I Journal of Systems & Control Engineering*, Number **II**, February 2005, pp. 1-14.
- Marques, E., J. Pinto, S. Kragelund, P. Dias, L. Madureira, A. Sousa, M. Correia, H. Ferreira, R. Gonçalves, D. Horner, A. Healey, G. M. Gonçalves and J. B. Sousa (2007). AUV control and communication using underwater acoustic networks. In: *Proceedings of the Oceans 07 Conference*, Scotland.
- Marques, E. (2007). DFO – Data Flow Objects for embedded software control specification, *Internal Report*, FEUP, Portugal.
- Marques, E., G. M. Gonçalves and J. B. Sousa (2006). Seaware: A Publish/Subscribe Middleware for Networked Vehicle Systems. In: *7th Conference on Manoeuvring and Control of Marine Craft (MCMC'2006)*, Lisboa, Portugal.
- NSA Standardization Agreement (STANAG) (2006). *Standard Interfaces of UAV Control Systems (UCS) for NATO UAV Interoperability*, NATO.
- Sousa, J. B., T. Simsek and P. Varaiya (2004). Task planning and execution for UAV teams. In: *Proceedings of the Decision and Control Conference*, Bahamas.
- Sousa, J. B., K. H. Johansson, J. Silva and A. Speranzon (2007). A verified hierarchical control architecture for coordinated multi-vehicle operations. In: *International Journal of Adaptive Control and Signal Processing*, Volume **21**, Issue 2-3, pp. 159 – 188 (Special Issue: Autonomous and adaptive control of vehicles in formation . Issue Edited by Sandor M. Veres).
- Vaglianti, B., R. Hoag and M. Niculescu (2004). *Piccolo system user guide*. www.cloudcaptech.com.