

# Authority Sharing in Mixed Initiative Control of Multiple Uninhabited Aerial Vehicles

Rui Gonçalves<sup>1</sup>, Sérgio Ferreira<sup>1</sup>, José Pinto<sup>1</sup>, João Sousa<sup>2</sup>, and Gil Gonçalves<sup>1</sup>

<sup>1</sup> Department of Informatics Engineering

<sup>2</sup> Department of Electrical and Computer Engineering

School of Engineering, Porto University (FEUP)

Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal

{rjapg, ei05092, zepinto, jtasso, gil}@fe.up.pt

**Abstract.** In this paper we discuss a conceptual framework that supports operational scenarios with multiple UAVs and operators. These UAVs possess different levels of autonomy while the operators have variable skill sets. The scenarios themselves encompass different missions, with different phases (requiring different levels of attention from the operator) and with the occurrence of various exogenous events. This framework was employed in the development of a Command and Control (C2) application which is capable of operator advisement, self adaptation, and automatic task distribution among operators and UAVs, depending on mission objectives, phase and occurrences. This C2 application enables a clear overview of the remote environment by placing the operator closer to the control loop, whether it is at an abstract or low level of control. Consequently there is an improvement of task redistribution and situation awareness, as well as reduction of workload.

**Keywords:** Operator, UAV, Interoperability, Autonomy Levels, Command and Control, UAS, Situation Awareness, Workload.

## 1 Introduction

The last decades have witnessed unprecedented technological developments in computing, communications, navigation, control, composite materials and power systems, which have led to the design and deployment of the first generations of unmanned aerial vehicles (UAV) and unmanned aerial systems (UAS). These vehicles have already seen action in many scenarios and proved their value.

As the operational capacity of UAS continues to grow, these systems can include multiple UAVs operating as a team, furthermore solidifying their employment in military and civilian scenarios. With the aid of these systems it is possible to remove the human element from “dirty, dull, and dangerous” situations and relocate it to a less operational and more supervisory role. However, with the rise of their operational capacity so rose the complexity of tasks they could perform.

At the Underwater Systems and Technology Laboratory (LSTS) [1] we have been designing, building and operating a significant number of heterogeneous unmanned

vehicles. These include Remotely Operated Vehicles (ROV) [2], Autonomous Underwater Vehicles (AUV) [3, 5, 6], and Autonomous Surface Vehicles (ASV) [4].

We have been also developing UAVs [7] as a result of our collaboration with the Portuguese Air Force Academy.

Throughout this paper we describe a conceptual Framework for optimal inclusion of the operator in the control loop and the application of its concepts in a C2 software interface. The objective is to distribute and reduce the workload of a decentralized team of operators controlling multiple UAVs. To achieve this goal we intend to advise operator's actions and reconfigure C2's layout using an automated methodology. The combination of the different mission intervenient entities (UAVs, Plan State, Operators, Consoles Profiles, and Mission Workload) can be used to inform the operator about the ideal operation console layout to be used and the ideal workload for each operator. The properties of these mission entities can change, during the mission execution, making this a dynamic process. The operator can have different levels of situation awareness, at different stages of the mission. The system will help operators to dynamically configure an optimal view of the mission state from a set of predefined console layout profiles.

In section 2 we introduce the C2 framework inner workings and its operation method. In section 3 we give a complete overview of the LOA framework and it's execution principles. In section 4 we present an example of the LOA framework at work and in section 5 we introduce two examples of operation consoles, to be used in conjunction with de C2 framework.

## 2 Networked Vehicle Systems and Supervisory Control

Unmanned vehicle systems are currently being employed in the field for very distinct purposes. For instance, considering just individual UAVs, these can be used for precision sensing, aerial imagery, surveillance, etc. The full potential of these systems, however, requires the management of multiple networked vehicles operating as a whole, sharing their workload and knowledge about the environment.

The concepts of operation for multi-UAV teams differ from single UAVs in the sense that in the former there exist common objectives like maintaining a common knowledge database [8] and redundant execution of crucial actions [9]. Moreover, operators are required to quickly perceive the entire system state, so that they can re-organize themselves in the face of unpredicted situations. All this while taking into account the different levels of attention all the vehicles demand. In order to decrease the number of operators' necessary on a multi-UAV deployment, we use mixed-initiative interaction for controlling the network at a system-level.

### 2.1 Simultaneous Control of Multi-UAV Teams

In our C2 framework, UAVs can be tasked either individually by an operator or they can be tasked by a software agent that acts as an operator (Team Supervisor). The team supervisor divides work among the vehicles according to a multi-UAV mission

specification and simple task-allocation algorithms. If the control over the UAV is not overridden, they carry out planned behavior until they are faced with failures, or any other unpredicted situations in which they contact the ground station and require human intervention. When operators have sufficient authority, they can cancel the current vehicle's planned behavior and replace it with other tasks or tele-operate the vehicle (control override). This may result in the cancellation of tasks that were generated by the team supervisor and thus they will be postponed for execution by another free UAV. UAVs may also actively contact the base station asking for human intervention when there is an onboard malfunction or a potential risk is detected. In this case, the C2 framework will try to allocate the vehicle supervision to a free operator or will suggest switching of coupling between vehicles and operators.

## 2.2 Team Supervisor

To provide system-level control of multiple vehicles, we use a software agent that holds a multi-UAV mission specification. This mission specification is currently a list of individual plans that need to be executed by UAVs. The importance of this software module is that it allows the interaction with UAV network simply by adding plans that need to be carried out. The team supervisor then captures the capabilities among existing UAVs, their current tasks and also the availability of operators. Tasks are divided among UAVs in a way that workload is shared among capable vehicles. Some tasks however also require the intervention of human operators for correct execution, so the availability of operators is taken into account by the team supervisor while tasking the network.

## 3 Concepts for the Framework

This section presents how we interpreted and adapted the original LOA matrix (Table 1) into our Framework for optimal inclusion of the operator in the control loop. We will describe the LOA matrix and how we intend to categorize the operator skills. Then we describe the methodology used to advise one Console Profile (CP) to the operator for a given LOA on a mission stage, combined with the Operator Skills data. The Operator Workload management is made by the Mission Team Supervisor as described in the previous section.

### 3.1 Levels Definition

The LOA Table [10] is based on Sheridan's 10-level of autonomy scale [11] and simplified to present only eight levels of autonomy. The lower the task is on the scale, the more authority the human operator has over the automate. The two dimensions of the matrix (Table [10]) are the eight levels (matrix rows) crossed with four functional categories (matrix columns). The second dimension presented in this matrix is the division of each task into four functional steps. These tasks present human decision-making processes as a set of OODA cycles (Observe, Orient, Decide, and Act).

**Table 1.** Partial LOA matrix as originally published in [10]

Level	Observe	Orient	Decide	Action
8	The computer gathers, filters, and prioritizes data without displaying any information to the human.	The computer predicts, interprets, and integrates data into a result which is not displayed to the human.	The computer performs ranking tasks. The computer performs final ranking, but does not display results to the human.	Computer executes automatically and does not allow any human interaction.
...	...	...	...	...
1	Human is the only source for gathering and monitoring (defined as filtering and prioritizing) all data.	Human is responsible for analyzing all data, making predictions and interpretation of the data.	The automate does not assist in or perform ranking tasks. Human must do it all.	Human alone can execute decision.

Table 2 is used to categorize the operator skills using the LOAs he is certified to respond, the CP (CP-Console Profile) the operator is familiarized and the number of vehicles he can handle safely at a certain LOA. With this data we can infer about the training and education level. Notice that the LOA entry table, of operator skills, has correlation whit the number of vehicles the operator can handle.

**Table 2.** Fields used to infer about the operatos skills in the framework

Certified Type of LOA	Certified Consoles Profiles	Number of Vehicles
Type of maneuver the operator is certified.	Set operation Consoles the operator is familiarized. By preference order. (for one LOA)	Operator fan-out of vehicles (for one LOA)

### 3.2 LOA Combination and Consoles Profiles

The matrix represented in Table 1 can be related with the creation of different types of console profiles. Different console profiles can be associated to different combinations of the four functional categories (OODA) - operational modes. For the presented framework we have a direct relation of LOA and CP.

The formal representation for CP-LOA tuple is:

$$\text{CP-LOA} = (\{\text{Obs}_1 \dots \text{Obs}_n\}, \{\text{Ori}_1 \dots \text{Ori}_n\}, \{\text{Dec}_1 \dots \text{Dec}_n\}, \{\text{Act}_1 \dots \text{Act}_n\})$$

For example a CP specialized for UAV “fly-by-wire” (direct control) operational mode would be based on the following tuple:  $\text{CP-LOA} = (\{1\}, \{1\}, \{1\}, \{1\})$  for OODA combination. The elements on the tuple are represented as sets so we can group the OODA functional categories. This way it is possible to have one CP capable of handling different Operational Modes. Grouping some intervals of OODA levels in the LOA matrix has proved to be useful in practical application. Another example of grouping the LOA of the proposal matrix in [10] can be consulted in [12].

One high level control CP should be able to handle high LOA values for OODA. For example one representation of LOA for a console of this type can be: CP-LOA= ( $\{5-6\}$ ,  $\{6\}$ ,  $\{5-6\}$ ,  $\{6-7\}$ ). In conceptual terms we can have one very generic CP that responds to all possible combinations of LOA (CP-LOA= ( $\{1-8\}$ ,  $\{1-8\}$ ,  $\{1-8\}$ ,  $\{1-8\}$ )). Since the system is composed by several CP's, one LOA required by the UAV can have different CP's to handle interaction. This means the system will have different CP's to choose, to advise the operator to use, in a given Operational Mode of the plan state. This choice can be automated by looking to the preference order of CP's that the operator is certified.

In Fig. 2 is illustrated the process of advising on CP to the operator. First the UAV starts a manoeuvre which requires a LOA. The system searches for the catalogued CPs in the system and operators capable of handling that LOA (manoeuvre). The listed CPs are filtered by the ones that the operator is certified. Finally the system advises the best CP to the operator. The operator is selected by the Mission Team Supervisor based on the workload of the mission operators.

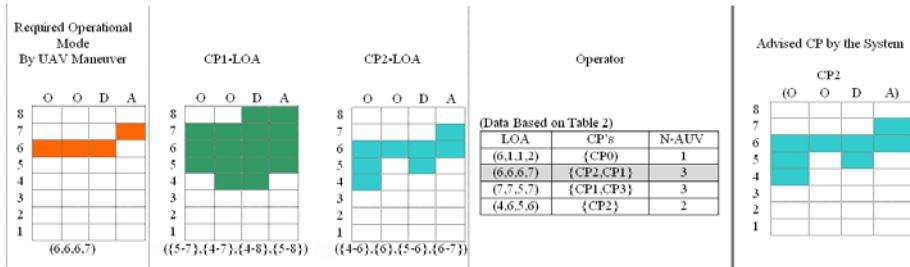


Fig. 1. Decision process of Console Profile to advise the operator

## 4 Scenario Definition

To exemplify the framework's execution we will use one mission scenario where the operators have to find a target and follow it. There will be two operators and five UAVs in this example.

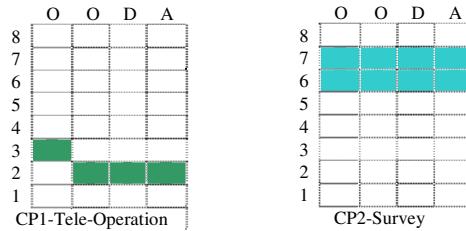
Current UAVs offer little adaptability in terms of automation: operators can leave the UAV do the flight by itself, following a pre-defined flight plan, or they can control it manually. For this example we will use 2 LOAs for the operators and another one of full autonomy, for handover and emergency manoeuvres. The operators LOAs are further sub-divided into a high level control LOA and low level control LOA. All three LOAs are described as follows:

- **Operational Mode 1 – Tele-Operation or Direct Control :** LOA= (3, 2, 2, 2)
- **Operational Mode 2 – Survey :** LOA= (6, 6, 7, 6)
- **Operational Mode 3 – Full Autonomy :** LOA= (8, 8, 8, 8)

Once the target is identified by the operator Operational Mode 1 will be used to follow it. When the vehicle is in “search mode” the operator sees the payload data

(video) and tries to identify the targets, this is Operational Mode 2. In Operational Mode 2 the operator can also define survey areas for each UAV. Finally Operational Mode 3 is a full autonomy mode used for the handover of UAV control logic. Its premise is that the operator must free the UAV so other operators can own it (request control of it). In this mode the system only knows about the UAV existence.

We will use two CPs (CP1 and CP2) to handle this mission example as follows:



**Fig. 2.** Two Console Profiles used in mission

For this mission example we will have two operators with the following Skill Tables:

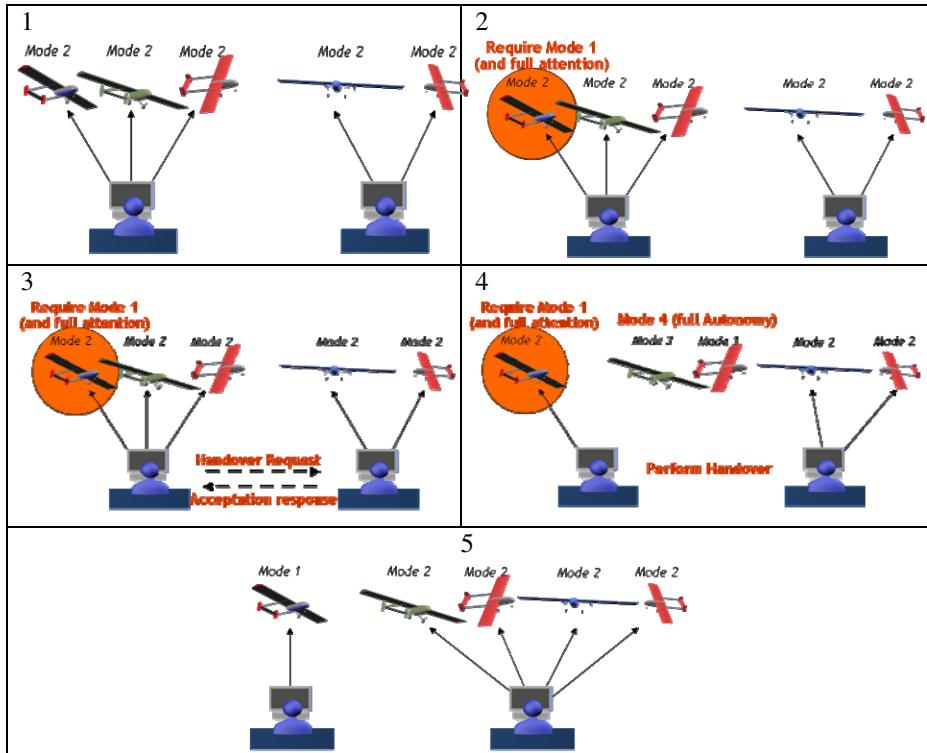
**Table 3.** Skills Table for Operator 1, based on table 2 – This operator can handle 3 UAVs in high level control and 1 UAV in low level control

Certified Type of LOA	Certified Consoles Profiles	Number of Vehicles
(3,2,2,2)	{CP1}	1
(6,6,7,6)	{CP2}	3

**Table 4.** Skills Table for Operator 2, based on table 2 – This operator can handle 4 UAVs in high level control

Certified Type of LOA	Certified Consoles Profiles	Number of Vehicles
(6,6,7,6)	{CP2}	4

Fig. 3 illustrates the 5 most important steps taken when one of the operators find the target. Initially all the UAVs are in survey mode – mode 2 of our LOA definition – and both of the operators are using CP2 to operate the UAVs: defining survey areas and looking at the payload data (video). In step 2 Operator 1 finds the target, which must requires on UAV to enter mode 1. This leads to a workload overload for Operator 1 that must be solved by the mission Team Supervisor. The only operator capable of handling UAVs in mode 1 is Operator 1, as defined in table 3. Since Operator 1 is only capable of handling one UAV in mode 1 the mission supervisor will advise Operator 1 to hand-over the other 2 UAVs to Operator 2. Here starts step 3 with the handover process: Operator 1 frees the UAVs putting them in mode 3. Finally, in step 5, Operator 2, takes over these UAVs in mode 2 and Operator 1 can now follow the target. In this step the Mission Supervisor advises Operator 1 to use CP1-Tele-operation to respond mode 1 LOA, according to his skills table 3.



**Fig. 3.** Logic of operation example for mission workload distribution

## 5 Framework Components

As stated before, this framework was employed in an existing C2 application, Neptune, which has an underlining architecture and provides de means of creating the various consoles used in the different CP's. This section introduces Neptune and an example of such consoles.

### 5.1 Neptune

Neptune is a distributed C2 framework for operations with networked vehicles, systems, and human operators. Neptune supports all the phases of a mission life cycle: planning, simulation, execution, and post-mission analysis. Moreover, it allows operators to plan and supervise missions concurrently [13].

Furthermore, Neptune encompasses a console building application which facilitates the rapid creation of new operation consoles for new vehicles with new sensor suites, as well as the remodelling of old consoles for current vehicles. There are two important aspects to console configuration: visual components and event communications. The internal Neptune event communication system is based on a tree structure (following the blackboard design pattern [14]), where nodes indicate the subject of data values in leafs.

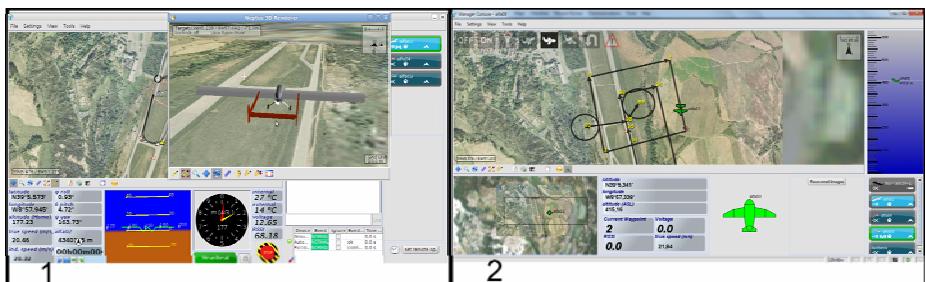
Neptus visual components can become listeners of a single variable (tree leaf) or of a defined variable domain (tree branch). Whenever a message arrives, using the IMC[15] communication protocol, that data is stored in a specific branch of the tree and listeners are then informed of the incoming network data. In a similar way, output data is sent by Neptus console components through the variable tree. The variable tree system is also used for event communication between Neptus console components.

There are two states in the Neptus generic console builder application: Editing and Operational. In Editing mode, users can then add and place components freely inside the console's main panel. Component properties can be edited to connect the panels to different systems and variables. When all components are ready, correctly placed and connected to the system variable tree, the user can switch the state of the application to the Operational mode.

## 5.2 Operation Consoles

Besides having the capability of dynamically creating new consoles during a specific mission, Neptus also has predefined consoles already available for the LOA switches the presented framework requires. These consoles go from standard tele-operation consoles, as seen as example 1 of Fig. 4, to supervision consoles, as seen as example 2 of Fig. 4. These consoles have different layouts depending on the central function they have. For instance a tele-operation console will typically have more detailed data about the UAV under its control, whereas a supervision console will only have a simplified view of the current UAV to allow a broader view of the whole team.

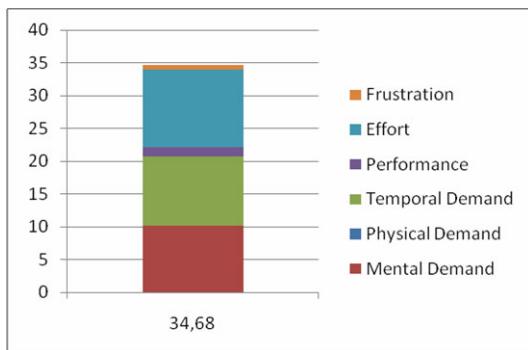
As an example of said consoles we introduce de details behind the current flight manager console used for UAV mission supervision at the LSTS.



**Fig. 4.** Tele-Operation and a Supervisory control consoles

This console, as seen in Fig 4, was developed based on an RTS paradigm with the intent of applying the concepts, learned by these types of games, on how to efficiently control and supervise groups of units of various dimensions and with different capabilities. This approach, while not being new, has allowed the implementation of a console which supports high LOA levels CP-LOA=  $\{\{6-7\}, \{6-7\}, \{6-7\}, \{6-7\}\}$  while, at the same time, enables the supervision of UAV teams with a low workload rating value for the operators, as can be seen in Fig. 5.

On par with workload evaluation there has been, as well, situation awareness evaluations of these consoles in order to guarantee flight manager focus and to maximize UAV team fan-out.



**Fig. 5.** Flight manager console's total workload rating, using NASA-TLX [16], in a 3 UAV scenario

## 6 Conclusions

Throughout this paper we referenced the growing importance of multi-UAV systems, paying special attention to the needs of the successful use of these systems.

We presented the concepts behind a framework for managing UAV task and workload allocation between various operators in a mission scenario. This framework was applied in the development of a command and control (C2) application which is capable of self adaptation, operator advisement and automatic task distribution amongst operators and UAVs depending on mission objectives, phase and occurrences. An example scenario of this framework, as well as an example of the details around one of the consoles used by the operators, was presented and discussed.

This C2 application enables a clear view and presence on the remote environment by putting the operator much closer to the control loop, whether it is high level or low level control, with the consequent improved redistribution of tasks and situational awareness. NASA Task Load Index (TLX) was used as a means to determine the adequacy of the C2 interface and functionalities. The preliminary results obtained with this framework are promising and we are confident its use will vastly improve the reliability of multi-UAV teams by augmenting their compatibility with more mission scenarios.

**Acknowledgments.** The authors acknowledge the contribution of AsasF of FEUP in the requirements identification and in the demonstration tests. Also acknowledge the Portuguese Air Force Academy (AFA) in the field tests and developments and design of ANTEXs X02 UAVs under the *PITVANT* project funded by the Portuguese Ministry of Defense. Neptus relies on other components of the LSTS tool chain for its correct execution. For this, we would like to thank Eduardo Marques, Paulo Dias,

and Ricardo Martins mainly for their efforts in developing the IMC protocol. Rui Gonçalves and José Pinto were partially supported by FCT grants.

## References

1. USTL. Underwater Systems and Technology Laboratory (February 2011), <http://whale.fe.up.pt>
2. Gomes, R., Sousa, A., Fraga, S.L., Martins, A., Borges Sousa, J., Lobo Pereira, F., et al.: A New ROV Design: Issues on Low Drag and Mechanical Symmetry. In: Oceans 2005, Europe, June 20-23 (2005)
3. Cruz, N., Matos, A., Borges de Sousa, J., Lobo Pereira, F., Estrela Silva, J., Coimbra, J., Brogueira Dias, E.: Operations with Multiple Autonomous Underwater Vehicles: The PISCIS Project. AINS
4. Ferreira, H., Martins, R., Marques, E., et al.: Swordfish: an Autonomous Surface Vehicle for Network Centric Operations. In: IEEE Oceans Europe (2007)
5. Madureira, L., Sousa, A., Sousa, J., et al.: Low Cost Autonomous Underwater Vehicles for New Concepts of Coastal Field Studies. In: CERF ICS (2009)
6. USTL. Seascout (January 2010), <http://whale.fe.up.pt/seascout>
7. Pereira, E., Bencatel, R., Correia, J., et al.: Unmanned Air Vehicles for Coastal and Environmental Research. In: CERF ICS (2009)
8. Jariyasunant, J., Pereira, E., Zennaro, M., Hedrick, K., Kirsch, C., Sengupta, R.: CSL: A Language to Specify and Re-Specify Mobile Sensor Network Behaviors. In: Proceedings of RTAS (2009)
9. Sousa, J.B., Simsek, T., Varaya, P.: Task planning and execution for UAV teams. In: Proceedings of CDC (2004)
10. Proud, R.W., Hart, J.J., Mrozinski, R.B.: Methods for determining the level of autonomy to design into a human spaceflight vehicle: A function specific approach. In: Performance Metrics for Intelligent Systems Workshop, Gaithersburg, MD (2003)
11. Sheridan, T.B.: Telerobotics, automation, and human supervisory control. MIT Press, Cambridge (1992)
12. Villaren, T., Madier, C., Legras, F., Leal, A., Kovacs, B., Coppin, G.: Towards a Method for Context-Dependent Allocation of Functions. In: HUMOUS 2010 conference Humans Operating Unmanned Systems ISAE - ONERA, Toulouse, France, April 26-27 (2010)
13. Dias, P.S., Pinto, J., Gonçalves, R., Sousa, J.B., Pereira, F.L., Gonçalves, G.: Neptune, command and control infrastructure for heterogeneous teams of autonomous vehicles. In: International Conference on Robotics and Automation ICRA 2007. IEEE, Los Alamitos (2007)
14. Deugo, D., Weiss, M., Kendall, E.: Reusable patterns for agent coordination. In: Omicini, A. (ed.) Coordination of Internet Agents, pp. 347–368. Springer, Heidelberg
15. Martins, R., Dias, P.S., Marques, E.R.B., Pinto, J., Sousa, J.B., Pereira, F.L.: Imc: A communication protocol for networked vehicles and sensors, Oceans (2009)
16. Hart, S.G., Staveland, L.E.: Development of a multi-dimensional workload rating scale: Results of empirical and theoretical research. In: Hancock, P.A., Meshkati, N. (eds.) Human Mental Workload, Amsterdam (1998)