

Simulating Multistatic ASW Operations with the Multistatic Tactical Planning Aid (MSTPA)

Armand Wathelet, Christopher Strode, Arjan Vermeij
NURC – A NATO Research Centre
Viale San Bartolomeo 400
19126 La Spezia
ITALY

wathelet@nato.nurc.int strode@nato.nurc.int vermeij@nato.nurc.int

SIMULATING MULTISTATIC ASW OPERATIONS WITH THE MULTISTATIC TACTICAL PLANNING AID (MSTPA)

The concept in countering a submarine threat using a multistatic system is cooperation of the platforms and the sharing of sonar information amongst the ASW assets to improve detection, tracking and classification of the threat submarine. The manoeuvres of this networked system of platforms are complex and must be coordinated to achieve the mission objectives, for example protecting the afloat NATO Response Force during an amphibious operation. A key question is: what is the best use of these assets to successfully counter a submarine threat in a range of NATO scenarios? A multistatic tactical planning aid (MSTPA) is in development at NURC to address this by simulating multistatic ASW operations to derive successful tactics against a credible submarine threat. This paper outlines the approach, the models and the Monte-Carlo framework developed for MSTPA to represent multistatic detections, tracking and fusion of contacts, the communication network and classification of tracks. The core modules of MSTPA simulate detections variability and uncertainty and realistic false contacts. This paper describes the efforts made to address reusability and interoperability of the core modules. In particular, the experience in developing MSTPA's user interface with existing tools and data sets, to provide a rich visual environment for tactics development. The paper also reports the development of MSTPA using standard data formats and open source packages to facilitate interoperability with other systems and to facilitate the development, deployment and reuse of the modules. The result of these efforts allows compatibility with a vast array of existing data such as maps, meteorological, bathymetry and oceanographic information providing a rich context for objective analysis of the simulated operations. This paper reports the efforts made to verify the simulation by comparing simulated multistatic scenario with at-sea experiments. MSTPA is initially aimed at supporting the design of concept demonstration experiments, that is, to assist in making the best use of the multistatically enabled assets during the experiments to highlight the benefits of networking sonar information.

INTRODUCTION

NATO maritime concepts of operation include expeditionary operations in the littoral, typically shallow, waters. These expeditionary operations are supported by the tasks of clearing the area of threat submarines and keeping the area safe for the duration of the operations. The advent of small quiet submarines operating in these shallow waters makes these tasks difficult using current sonar systems. One concept to increase anti-submarine warfare effectiveness is to use multistatic sonar systems.

Wathelet, A.; Strode, C.; Vermeij, A. (2007) Simulating Multistatic ASW Operations with the Multistatic Tactical Planning Aid (MSTPA). In *Improving M&S Interoperability, Reuse and Efficiency in Support of Current and Future Forces* (pp. 6-1 – 6-10). Meeting Proceedings RTO-MP-MSG-056, Paper 6. Neuilly-sur-Seine, France: RTO. Available from: <http://www.rto.nato.int>.

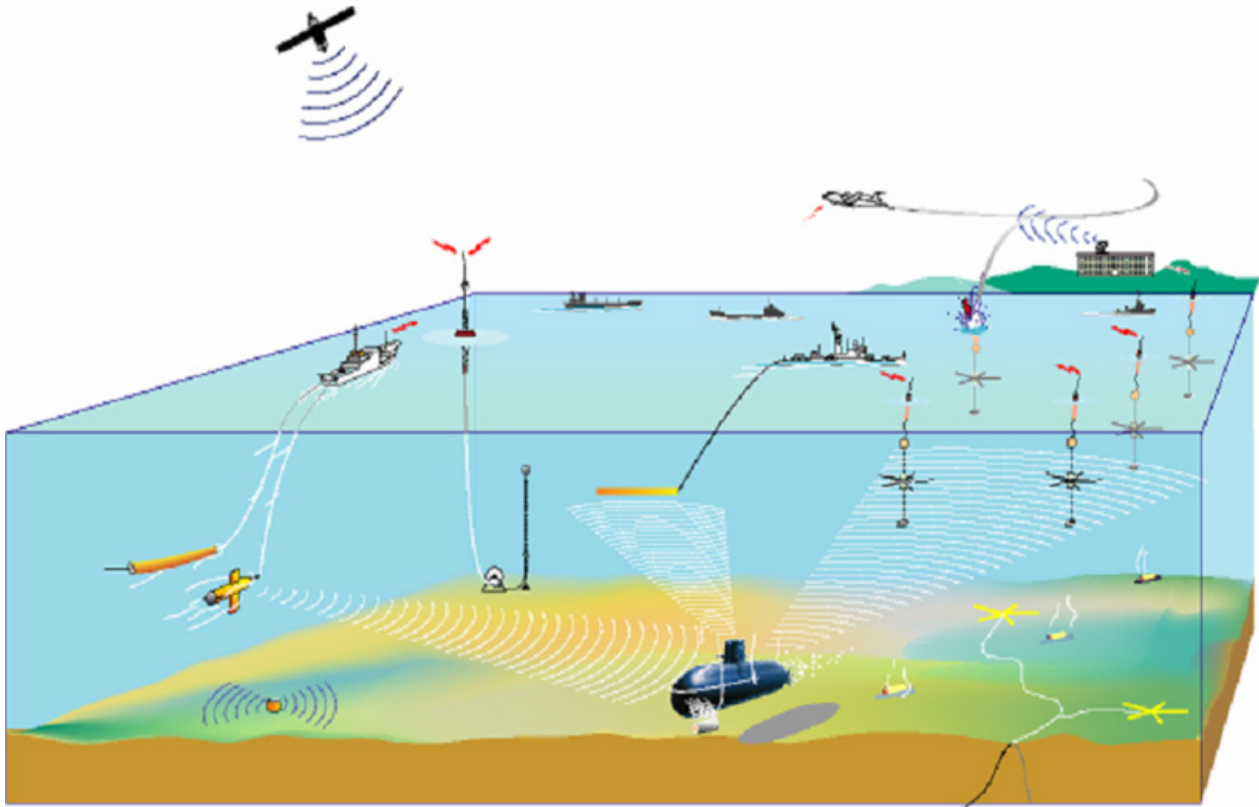


Figure 1: Multistatic overview

The concept in multistatic ASW operations is to share sonar information amongst a number of dispersed ASW units using a communications network to increase the ASW effectiveness of the assets, Figure 1. The information is used both for augmenting a shared view of the ASW operation by contributing to the common operating picture and for improving sonar performance. In essence multistatic ASW is an example of a networked enabled capability since sharing sensor information to increase the effectiveness of the whole system is central in a multistatic sonar system.

Multistatic systems have been proposed specifically for improving sonar effectiveness in the detection, tracking and classification of target submarines. However, in order to fully utilise the acoustic benefits, the asset geometries and transmission policy must be carefully considered. A key issue therefore is how to best employ the multistatic assets to successfully counter a credible submarine threat. Employing multistatic systems in a complicated and dynamic scenario requires automated support. Project MSTPA was initiated in NURC to assist analysts, both military and civilian, in gaining insights into the effects of different tactical decisions. The goal of MSTPA is to provide a tool to analyse and quantify multistatic employment in a range of relevant NATO scenarios to support maritime commanders for operational planning and decision makers for the acquisition of such systems. The initial focus in developing MSTPA is for supporting the design of experiments to evaluate the benefits of multistatic ASW concept.

This paper outlines the work conducted to date in MSTPA, with emphasis on the functions and the development using existing packages and the efforts made to allow this tool to be reused by others.

APPROACH

Evaluating tactics when using multistatic systems to successfully counter a threat submarine is a difficult problem as it involves a range of interrelated complex elements such as uncertainties in sensor performance in a difficult environment, fusion of information, multi-step and multi-objective decisions over time, the simultaneous coordination of a number of assets, and threat tactics. Deriving analytic formulae that relate measures of effectiveness (MOE) whilst taking into account operational constraints is difficult if not impossible. Hence the technique of simulation is used to evaluate the MOEs that relate employment of multistatic systems and the ASW objectives. The approach consists of modelling the sequence of events resulting from an acoustic ping. This includes acoustic detection, tracking, classification and the communication of detections for multistatic fusion. A Monte-Carlo simulation framework is then used to obtain statistical results. In addition an optimiser is used to calculate optimum asset employment in the selected scenario. One important element that is also under development is modelling the behaviour of the threat submarine.

The key to developing MSTPA is to reuse existing models, algorithms and packages. The computer language Java was selected as the main language for MSTPA development. This brings the benefits of running MSTPA on a wide range of computer systems without expensive and time consuming code rewrite or re-compilations. It also allows access to a wide range of good scientific open source libraries and packages and the possibility of using the tool through the internet or corporate intranet.

TOOL REQUIREMENTS

MSTPA has three main user requirements as shown in Figure 2. Firstly, the user needs to be able to create a specific scenario and explore the results of that scenario through the interactive use of MSTPA. Secondly, the user requires assistance in optimising the employment of the multistatic assets. And thirdly, the user requires assistance in deriving successful tactics against a competent threat submarine. The main requisite is for the user to interact with the model and visualize the impact of the interactions as well as the results calculated by the model. The broad system requirements for MSTPA are shown through the use case diagram of Figure 2.

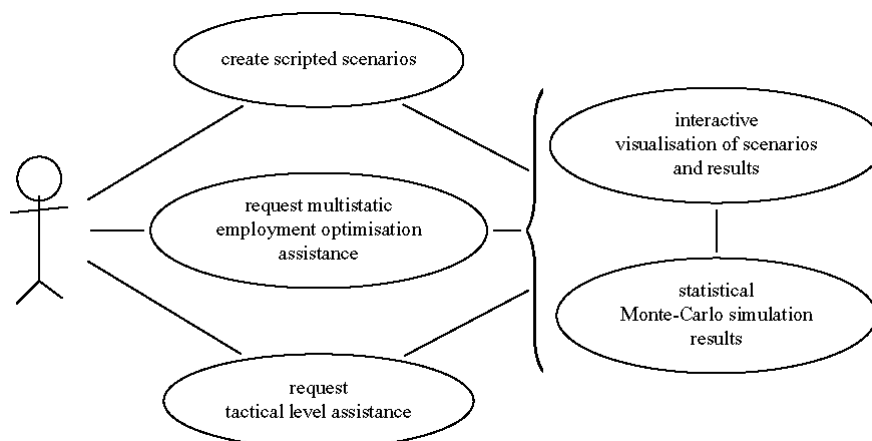


Figure 2: MSTPA use case diagram

CORE MODULES

The core modules in MSTPA focus on the acoustic detections and tracking of target submarines as well as the Monte-Carlo structure. Figure 3 shows the broad architecture and major modules relating to the acoustic sensor part of MSTPA. The greyed boxes are modules that are not yet implemented. As can be seen most decision modules are yet to be developed.

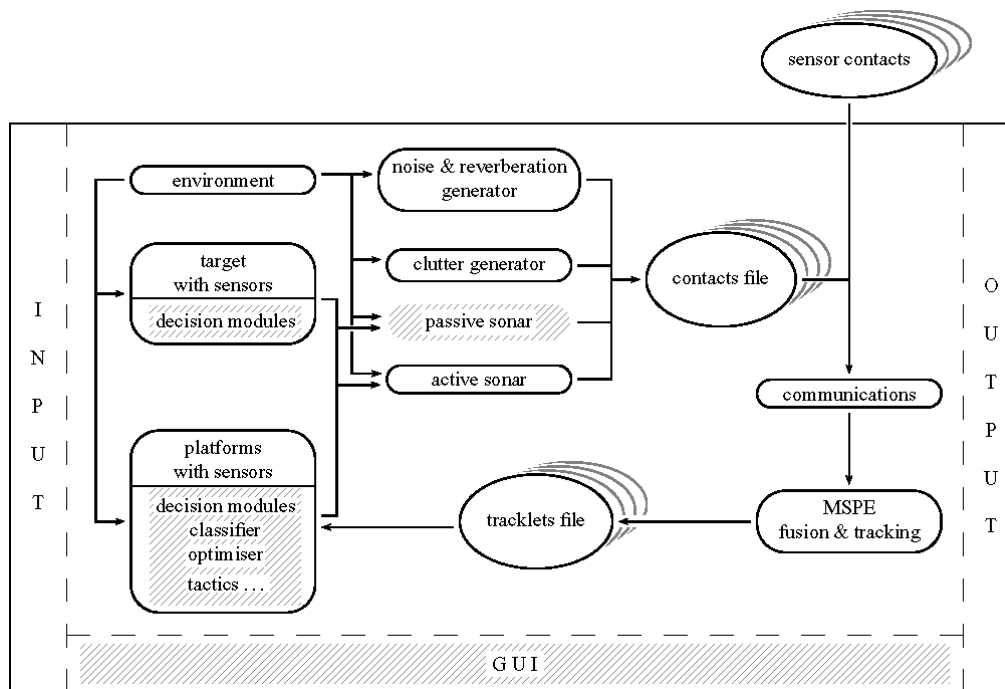


Figure 3: MSTPA Core modules overview

As shown in Figure 3, MSTPA has the capacity to read data generated by real sonars delivering contacts in the format developed by the NATO Multistatic Sonar Working Group [1]. These contacts can be processed in combination with simulated input if desired allowing us to output simulated scenarios in the same format. This provides valuable data to researchers for their research on tracking and fusion.

At present MSTPA simulates generic platforms that can support a number of active monostatic and multistatic sonars. Platforms such as surface combatants, submarines, autonomous underwater vehicles and sonobuoys can also be modelled. Other platforms such as helicopters and maritime patrol aircraft will be incorporated in future iterations of MSTPA. The target submarine is simulated with basic kinematics and a fully bistatic target strength model.

Central to MSTPA's acoustic sensor is a generic multistatic sonar performance model. This model, with appropriate changes is also used to represent a generic monostatic sonar. These sensor models, combined with false contact simulators for clutter and noise, and a tracker/fusion module, allow for realistic generation of both target and false tracks. The output of the MSTPA tracker is a series of simulated tracks that have resulted from the generation of contacts originating from a target, clutter object or from random fluctuations in the background noise level. These contacts are then sent through a simulated communication link, such as a satellite or VHF tactical link to a multistatics processing/fusion engine. This engine is composed of a Kalman filter tracker and fusion module to produce tracks.

The process to arrive at the simulated tracks is outlined in the following sequence of steps.

- Determine signal excess for target and clutter objects,
- Determine whether to generate a contact based on the assumed distribution of signal excess together with an inter ping correlation factor,
- Generate a contact taking into account bistatic localisation errors,
- Generate false contacts due to random fluctuations in background noise level,
- Transmit all contacts to the processing engine through a communication link,
- Generate tracks from the tracker.

The process described above essentially simulates the chain of events from the generation of an active acoustic transmission to the formation of tracks. This allows evaluation of the impact of a range of factors on target detection and tracking. For example changes in the communication link availability, the multistatic geometry and clutter density.

Some of the core modules introduced above were derived from existing algorithms that were initially developed by scientists as part of their research. Most of the acoustic modules, such as the multistatic reverberation module, were developed in MATLAB. Although a key element in developing MSTPA was to reuse existing code, in this particular instance it was more productive to convert it. In doing this, the code was restructured to define a clean interface, such that the acoustic modules can be replaced with a higher fidelity model if need be. This modular feature runs throughout MSTPA's architecture. Converting the existing MATLAB code to Java was done using an in-house developed automated procedure, which converted the code and made extensive verification of the results. The conversion removed the undesirable and expensive dependencies on MATLAB.

Other core modules were developed using a number of open source libraries [6][7][8][9][10][11][12]. Using these proved to be extremely valuable and saved a lot of time and effort. Java was chosen for its intrinsic benefits including cross platform compatibility and also because there is a large pool of good scientific and mathematical libraries to construct the algorithms needed in MSTPA.

The main input and output files of MSTPA are in XML format. The use of XML allows for input file generation using existing XML editing packages. This is particularly useful in batch mode where incremental changes may be required to parts of the scenario, for example changes to sonar parameters or the addition of a platform. Indeed the input files may be completely generated by other applications. Similarly, using XML format for output files enables other applications to process MSTPA's results easily. All this is facilitated by the existence of off-the-shelf tools such as XSLT processors and the work done as part of another NURC project.

VISUALISATION

Viewing the progress and interactions amongst the ASW assets during an ASW operation is a feature that is required in MSTPA. To provide this we opted to interface the core modules to a GIS package developed at NASA called "World Wind" [2]. Figure 4 shows a typical view of a MSTPA scenario.

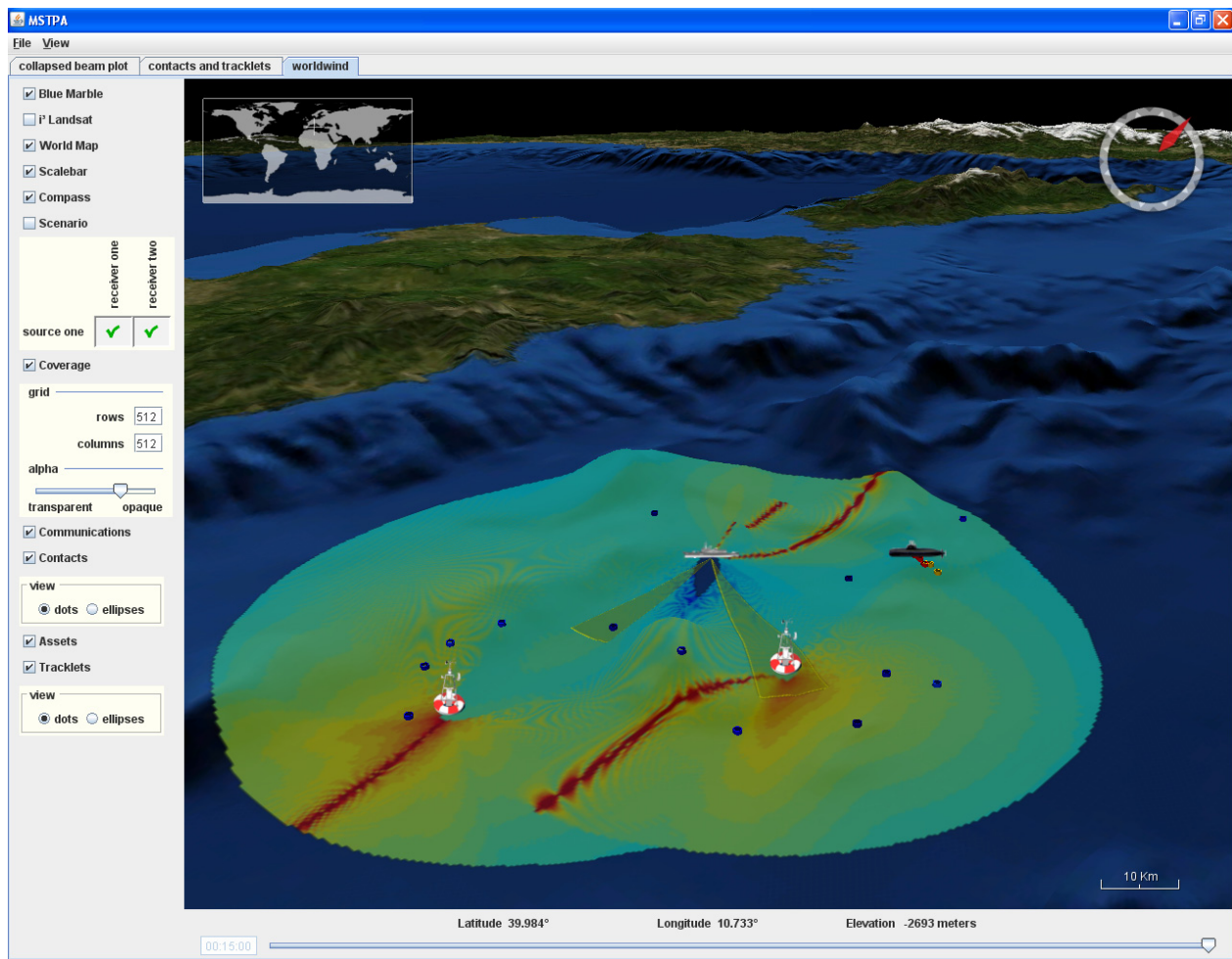


Figure 4: MSTPA WorldWind display

NASA's World Wind Java SDK is a three dimensional Geographic Information System (GIS) software development kit. It is built on top of Sun Microsystem's Java OpenGL (JOGL), which is a wrapper library giving access to the Open Graphics Library (OpenGL) for producing 2D and 3D computer graphics. OpenGL is a software interface to the graphics hardware making the rendering process very fast. The software is open source and therefore does not have expensive licensing and restrictive distribution requirements.

World Wind Java provides generic layers for the rendering of satellite imagery, bathymetry data, political boundaries, and the likes. Of particular interest to MSTPA is the bathymetry data. For our purposes additional layers were implemented to display application specific information such as asset placement, sonar coverage, communication links, contacts and tracks. In addition to the standard navigation controls provided by World Wind Java we added further controls through which the user can turn on or off layers, navigate through time and include or exclude certain assets from the display.

Interfacing MSTPA to World Wind was done in two ways. One way was to use GeoRSS interface to directly show the input scenario and the other was to write additional layers to display our specific results. Since World Wind uses accepted standards, a wide range of data can be displayed. In particular meteorological, oceanographic and other ASW relevant information such as ship wreck locations.

Future extensions to the architecture include using World Wind to generate the input scenario files. That is, constructing the scenario by adding the desired assets at the chosen location using the mouse and setting the parameters required for a simulation run from popup menus.

MSTPA's output can be delivered in a range of formats. This allows the possibility for MSTPA to interface to other simulations or visualisation applications quite easily. For example if required, MSTPA could output the results in shape files format for display using ESRI ArcView. The output format includes MATLAB, making its plotting functionality available to display selected results. This feature was used extensively during the development of MSTPA to explore options for displaying results. Selected plots are now being implemented in MSTPA using another open source package called JFreeChart. This open source package once again was very useful and provides a free and stable plotting functionality to MSTPA.

OPTIMISATION AND TARGET BEHAVIOUR

ASW assets placement

MSTPA calculates a number of MOEs, such as the detection area coverage and the probability that a submarine will be detected before crossing a barrier. A number of algorithms are used in MSTPA to optimise the placement of the ASW assets by maximising these MOEs. These algorithms are packaged in an open source library of evolutionary optimisation algorithms [3]. The principal technique used in MSTPA is a genetic algorithm to optimise the placement of ASW assets, such as ASW frigates and multistatic capable sonobuoys.

This open source package called JavaEvA can be described as [3] "... a general modular framework with an inherent client server structure to solve practical optimization problems. This package was especially designed to test and develop new approaches for Evolutionary Algorithms and to utilize them in real-world applications. JavaEvA already provides implementations of the most common Evolutionary Algorithms, like Genetic Algorithms, CHC Adaptive Search, Population Based Incremental Learning, Evolution Strategies, Model-Assisted Evolution Strategies, Genetic Programming and Grammatical Evolution. In addition the modular framework of JavaEvA allows everyone to add their own optimization modules to meet their specific requirements. The JavaEvA package uses a generic GUI framework that allows GUI access to any member of a class if get and set methods are provided and an editor is defined for the given data type. This approach allows very fast development cycles, since hardly any additional effort is necessary for implementing GUI elements, while still at the same time user specific GUI elements can be developed and integrated to increase usability".

The use of this type of open source package has again proved to be very valuable and very effective in developing MSTPA.

Target model

One important aspect in ASW operation is the behaviour of the target submarine as a result of actions by the search platforms. The current iteration of MSTPA contains only scripted target behaviour that follows a predefined track. However, we are investigating a number of approaches such as intelligent agents and neural network to model an intelligent submarine behaviour. In addition path planning is being investigated to give the target submarine the ability to determine the optimum path through the sensor network. This could represent the worst-case scenario in which a target has complete knowledge of the network. Other work is also underway on a reactive target that is able to counter detect the source and alter its course appropriately, thus representing a more likely threat submarine behaviour. Such a target may or

may not be aware of the surrounding multistatic receivers. MSTPA will capitalise on the work being conducted under another project at NURC investigating intelligent agents. Much of this target behaviour will be able to be re-used by other simulations.

SIMULATION VERIFICATION

To show that MSTPA can produce valid results a study was conducted [4] that compared the results of an at-sea experiment with that of MSTPA. The study compared simulated contact signal to noise ratios and tracks to real data collected during an at-sea experiment. The ability of MSTPA to read in actual contacts files recorded at sea and to plot the results allowed for a direct visual comparison with the simulated case. The study showed good agreement between the simulated and actual data. This provides confidence in the tool when used to investigate various multistatic concepts of operation.

FUTURE WORK

Future development of MSTPA will focus on algorithms for ASW assets path optimisation and modelling target behaviour using intelligent agents. These efforts face a number of technical challenges in terms of speed of calculation for optimisation and validation of the model results when intelligent reactive targets are present and little at-sea experience is available for comparison. Subject to requirements, future efforts may also include interfacing MSTPA to existing systems to capitalise on the work done already.

MSTPA is already in use to support the design of a multistatic at-sea experiment [5]. The post-analysis of this experiment will feed back into the development of the tool and provide further validation of the model.

CONCLUSIONS

A multistatic tactical planning aid (MSTPA) is in development at NURC that will assist in the generation of successful tactics against a threat submarine. The tool simulates both the detection and tracking of submarines and can quantify ASW effectiveness for a range of multistatic assets. This tool will allow analysts to explore various tactical concepts based on sharing sonar information as part of the broader Nato Networked Enabled Capability.

The development of this tool using standard formats, such as XML for input and output, facilitates its reuse by other models. Developing MSTPA in Java and making extensive use of open source packages has proved to be very valuable and effective. This approach is recommended for similar projects.

Although interfacing MSTPA to other simulations has not yet been attempted, the efforts made at the beginning of the project to facilitate interoperability and reuse will without doubts help the next phase of the project.

REFERENCES

- [1] R Laterveer, Contact File Format, Technical Report RL-MAR05, NURC, 2005
- [2] World Wind Java, <http://worldwind.arc.nasa.gov/java/>
- [3] F Streichert and H Ulmer, JavaEvA – A Java Framework for evolutionary Algorithms, Technical Report, WSI-2005-06, Centre for Bioinformatics Tubingen, University of Tubingen, 2005
- [4] C Strode, R Wathelet, A Vermeij, Simulation of trail DEMUS04 using the multistatic tactical planning aid (MSTPA), NURC, TBP
- [5] C Strode, R Wathelet, A Vermeij, SEABAR07 Trial Planning using the Multistatic Tactical Planning Aid (MSTPA), NURC, TBP
- [6] Colt, a set of Open Source Libraries for High Performance Scientific and Technical Computing in Java, European Organization for Nuclear Research (CERN), <http://dsd.lbl.gov/~hoschek/colt/>
- [7] JTS Topology Suite, an API of 2D spatial predicates and functions, Vivid Solutions, Inc., <http://www.vividsolutions.com/jts/jtshome.htm>
- [8] Robert Harder, OpenTS, a Java tabu search framework, COmputational INfrastructure for Operations Research (COIN), <http://www.coin-or.org/Ots/index.html>
- [9] GeoAPI, a set of interfaces from OpenGIS® specifications, <http://docs.codehaus.org/display/GEO/Home>
- [10] David Gilbert, JFreeChart, a Java chart library, Object Refinery Limited, <http://www.jfree.org/jfreechart/>
- [11] Arjan Vermeij, A Generic MVC Model in Java, July 07, 2004, <http://www.onjava.com/pub/a/onjava/2004/07/07/genericmvc.html>
- [12] Erich Gamma and Kent Beck, JUnit, a regression testing framework, <http://www.junit.org/index.htm>

