

Creating artificial vessel trajectories with Presto

Jeroen H.M. Janssens ^{a,†} Hans Hiemstra ^b Eric O. Postma ^a

^a *Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands*

^b *Thales Nederland B.V., P.O. Box 42, 7550 GD Hengelo, The Netherlands*

Abstract

The automatic detection of anomalies in the maritime domain requires representative anomalous instances. We developed Presto, an application that enables maritime-domain experts to create artificial anomalous vessel trajectories that may be characteristic of traffic violations, illegal fishing activities, drug smuggling, or piracy. When merged with existing real-world data, the artificial trajectories make possible the usage and evaluation of machine learning algorithms for the automatic detection of anomalies.

1 The need for maritime anomalies

A Maritime Safety and Security (MSS) system aims at guiding a surveillance operator to find maritime anomalies, such as vessel traffic violations, illegal fishing activities, and drug smuggling [5]. In real-world data, serious maritime anomalies rarely occur. Because the MSS system might be deployed in any maritime circumstance, we need to evaluate if our machine learning algorithms [6] can properly detect such anomalies. To this end, we have developed Presto, an application which enables maritime domain-experts to easily create artificial scenarios. In contrast to existing simulation applications, such as VR-Forces [7], which impose restrictive behavior models, our application gives the expert full control over the vessel trajectories.

2 Overview of Presto

The main concept in Presto is a scenario, which can contain one or more trajectories. Each trajectory is defined by several waypoints. A waypoint is a location on the world map and contains the additional parameters velocity, time, and curvature. Figure 1 shows the user interface of Presto, which consists of three main elements: (a) the world map, (b) the timeline, and (c) the property editor. Using the world map, the user can edit the position of the trajectories and their waypoints. The timeline gives an overview of the trajectories within the scenario and is used to navigate through time. The property editor with its three tabs “scenario”, “track”, and “waypoint”, is used to edit element specific parameters (e.g., for a scenario: name and description; for a trajectory: name and flag; and for a waypoint: location and velocity).

Presto has the ability to load existing, real vessel-trajectory data as so-called background data. This background data contains the positions and bearings of all vessels for a certain period (e.g., a day). It serves as a reference when creating new artificial trajectories, so that, for instance, collisions can be avoided or enforced. The generated data is used to create real-world vessel-trajectory data by filtering the appropriate variables such as location, velocity, and bearing, such that data points mimic messages according the Automatic Identification System (AIS) protocol. The exported data is fused with real-world data, such that the artificial data cannot be distinguished up front from the original data.

Presto is entirely programmed in Java, and makes extensive use of the Eclipse Rich Client Platform [3]. The world map is based on NASA World Wind, an open-source alternative to Google Earth [1]. The use of these software solutions makes Presto suitable for recent versions of Windows, Mac OS X, and Linux based operating systems. The world map requires a hardware-accelerated 3D graphics card and an internet connection for downloading the high-resolution terrain images, although these can also be cached locally.¹

[†]This work has been carried out as part of the Poseidon project under the responsibility of the Embedded Systems Institute (ESI), The Netherlands. This project is partially supported by the Dutch Ministry of Economic Affairs under the BSIK03021 program.

¹The first author is the main developer and will demonstrate Presto and all its features at the BNAIC in approximately 20 minutes.

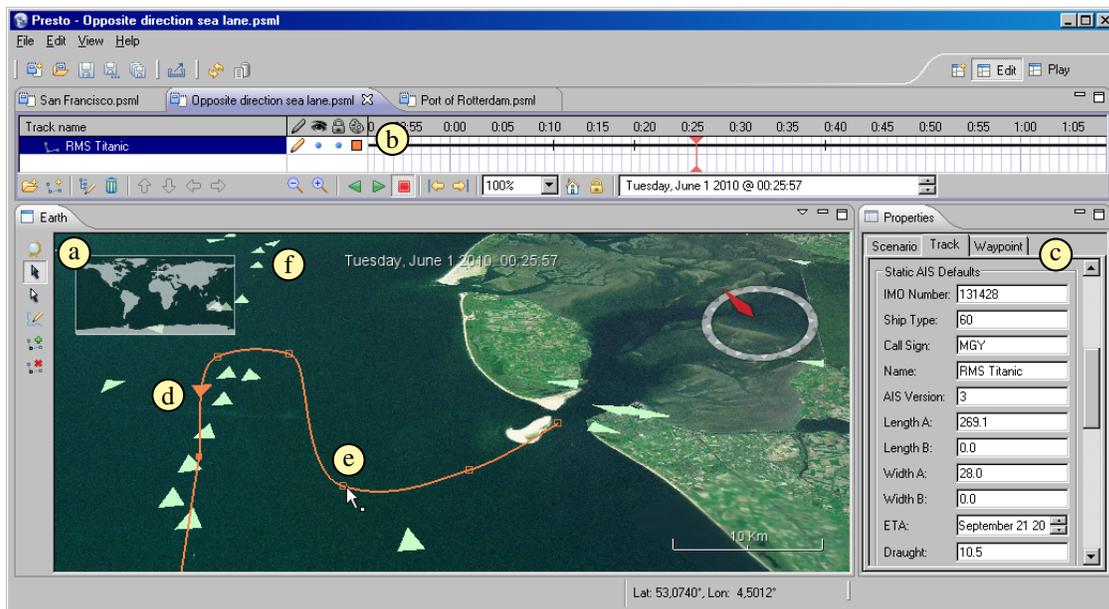


Figure 1: A screenshot illustrating the different user interface elements and concepts of Presto: (a) the world map, (b) the timeline showing the artificial trajectory in the scenario, (c) the property editor showing the trajectory properties, (d) the current location of the artificial vessel along the created trajectory, (e) one waypoint of the trajectory, and (f) the background data at the current moment.

3 Academic and industrial adoption of Presto

Currently, the beta version of Presto is successfully being used by researchers within the Poseidon project [5] and domain-experts at the Maritime Safety and Security division of Thales Nederland. It has proved to be a useful research tool for tasks including: anomaly detection [6], ontology engineering [2], runtime testing [4], and visual vessel analysis [8]. Additionally, Presto has enabled domain-experts to discuss and exchange potential maritime situations, and to challenge researchers with anomalies that are not present in real-world data. In future work we plan to implement additional import and export functionality such that Presto can be integrated more seamlessly with existing applications that are used within the MSS domain.

References

- [1] National Aeronautics and Space Administration. *NASA World Wind: Java SDK*, available from: <http://worldwind.arc.nasa.gov/java>. Accessed June 2010.
- [2] G.K.D. de Vries, V. Malaisé, M. van Someren, P.W. Adriaans, and G. Schreiber. Semi-automatic ontology extension in the maritime domain. In A. Nijholt, M. Pantic, M. Poel, and H. Hondorp, editors, *Proceedings of The 20th Belgian-Netherlands Conference on Artificial Intelligence*, pages 265–272, Enschede, The Netherlands, 2008.
- [3] Eclipse. *Rich Client Platform*, available from: <http://www.eclipse.org>. Accessed June 2010.
- [4] A. Gonzalez-Sanchez, É. Piel, and H.G. Groß. RiTMO: A method for runtime testability measurement and optimisation. In B. Choi, editor, *Proceedings of the 9th International Conference on Quality Software*, pages 377–382, Jeju, South Korea, 2009. IEEE Reliability Society.
- [5] Embedded Systems Institute. *The Poseidon project: System evolvability and reliability of systems of systems*, available from: <http://www.esi.nl/projects/poseidon>. Accessed June 2010.
- [6] J.H.M. Janssens, I. Flesch, and E.O. Postma. Outlier detection with one-class classifiers from ML and KDD. In M.A. Wani, M. Kantardzic, V. Palade, L. Kurgan, and Y. Qi, editors, *Proceedings of the 8th International Conference on Machine Learning and Applications*, pages 147–153, Miami, FL, 2009.
- [7] MÄK. *VR-Forces: The complete simulation toolkit*, available from: <http://www.mak.com/products/vrforces.php>. Accessed June 2010.
- [8] C.M.E. Willems, W.R. van Hage, G.K.D. de Vries, J.H.M. Janssens, and V. Malaisé. An integrated approach for visual analysis of a multi-source moving objects knowledge base. *International Journal of Geographical Information Science*, 2010.