

A SHIPPING SIMULATION THROUGH PATHFINDING: *SEL* WITHIN THE *MSP CHALLENGE* SIMULATION PLATFORM

Phil de Groot, Wilco Boode, Carlos Pereira Santos, Harald Warmelink, Igor Mayer
Academy for Digital Entertainment
Breda University of Applied Sciences
Monseigneur Hopmansstraat 2, 4817 JT BREDA, The Netherlands
E-mail: groot.p@buas.nl

KEYWORDS

Shipping Simulation, Maritime Spatial Planning, Simulation Gaming, Serious Gaming

ABSTRACT

The authors present the design of the shipping simulation *SEL* and its integration in the *MSP Challenge Simulation Platform*. This platform is designed to give policymakers and planners insight into the complexity of Maritime Spatial Planning (MSP) and can be used for interactive planning support. It uses advanced game technology to link real geo- and marine data with simulations for ecology, energy and shipping. The shipping sector is an important economic sector with influential stakeholders. *SEL* calculates the (future) impact of MSP decisions on shipping routes. This is dynamically shown in key performance indicators (e.g. route efficiencies) and visualised in heat maps of ship traffic. *SEL* uses a heuristic-based graph-searching algorithm to find paths from one port to another during each simulated month. The performance of *SEL* was tested for three sea basins: the firth of Clyde, Scotland (smallest), North Sea (with limited data) and Baltic Sea regions (largest, with most complete data). The behaviour of the model is stable and valid. *SEL* takes between 4 and 17 seconds to generate the desired monthly output. Experiences in 20 sessions with 302 planners, stakeholders and students indicate that *SEL* is a valuable addition to *MSP Challenge*, and thereby to *MSP*.

INTRODUCTION

The *MSP Challenge Simulation Platform* (henceforth *MSP Challenge*) is designed to give policymakers and planners insight into the complexity of Maritime Spatial Planning (MSP). It can be used for interactive planning support and general learning purposes. MSP is a process by which a country ‘analyse[s] and organise[s] human activities in marine areas to achieve ecological, economic and social objectives’ (European Union 2014), ending in a spatial plan. This spatial plan is essentially a highly annotated map of the sea area with spatial designations for specific human activities and marine protection measures for the medium-term future, often a period of 5 to 10 years. The *MSP Challenge* was first conceived and developed as a computer simulation game in 2011, and has been applied in sessions with MSP authorities, stakeholders and students many times since (Mayer et al. 2014, 2013; Stolte et al. 2013). Since early 2016, it has been further developed at Breda University of Applied Sciences within the context of the EU projects

and consortia *NorthSEE*, *Baltic LINES* and *SIMCelt*. It has now become a platform allowing for all sorts of simulation game sessions: in different sea basins, with different data sources, and with different simulation models running in the background.

Shipping is an important sector to take into consideration for three reasons. First, it is one of the oldest and thus best-established sectors to use the seas and oceans. Second, the sector has been one of the key drivers of global economic prosperity by transporting goods and people all over the world (Ferreira et al. 2018). Third, it is legally a strong sector as well; freedom of navigation has for centuries been an important principle in international maritime law (Wolfrum 2008). For this reason, shipping has always been an important theme and consideration in *MSP Challenge* sessions, especially since recent technical and social developments are creating new offshore human activities (e.g. wind farms, aquaculture) or new environmental protection measures (e.g. Marine Protected Areas, MPA) which are directly impacting the shipping sector.

To better involve shipping in *MSP*, different governments and private companies developed spatial maps of specific sea regions (e.g. the North Sea region) showing ship movements over a certain period of time (Nilsson et al. 2018). These static ‘heat maps’ are developed using ship movement data captured through the ships’ Automatic Identification System (AIS). The compendium of all ship movements generates heat maps which are very useful since they indicate the intensity of ships in specific areas over a specific time period. The maps can be used to recognise patterns, identify congestion areas, and evaluate risks, thus allowing planners to take important shipping information into account when they plan the use of sea space.

Although these maps offer a great utility for planners, they have no predictive power and do not allow *MSP* officials to forecast and develop different scenarios. While certain shipping patterns are generally constant (e.g. cargo or tanker vessels taking fixed routes and avoiding shallow waters), the influences of, for example, new wind farms or new traffic separation schemes introduced by the International Maritime Organization (IMO) are hard to grasp in a static map.

This is where the *MSP Challenge* could provide great value. *MSP Challenge* allows players to plan different scenarios for long periods of time (10 to 40 years), encouraging international discussion and cooperation to reach a coherent plan for an entire sea basin. However, the *MSP Challenge*

cannot rely on static shipping information, since any new plan made and implemented will invalidate the information. An example would be planning a wind farm over an existing shipping lane. A dynamic and responsive shipping model would be much more useful and insightful.

Several shipping simulations already exist and could theoretically be reused in the MSP Challenge. Integrating (parts of) any of the following existing simulators is technically possible:

1. MARIN's Vessel Traffic Service simulates individual ships over the course of a couple of hours. It is mainly used for training harbour personnel.
2. Sea Traffic Management, rolled out at several locations affiliated to the European Maritime Simulator Network, simulates individual ships in often particularly busy areas to test and teach novel traffic management technologies and techniques and thus optimise routes and reduce risks.
3. SEATRAS simulates sea traffic in particularly congested areas to e.g. enable calculations of collision risks and tests of collision avoidance technologies and techniques (Itoh et al. 2003).

However, when evaluating these existing solutions, we were concerned with the following:

1. None of these simulation goals are well-aligned with ours. The simulations are created for other purposes than those of MSP Challenge. They offer some functionalities that we could use, but need some functionalities that we would still need to develop.
2. Assuming we could adapt the existing simulations to fit our needs, MSP Challenge would simultaneously also handle dozens of large-scale data layers, as well as a simulation of offshore energy production and distribution (Hutchinson et al. 2018), and of ecosystem dynamics (Steenbeek et al. 2019). We thus require an efficient, well-targeted shipping simulation to keep system requirements at levels acceptable for our sessions and target audience.

We therefore decided to explore how we could create our own shipping simulation. In this paper, we offer an answer to the question of *how a convincing shipping simulation can be designed and implemented within the MSP Challenge, allowing for players to make MSPs that could include shipping measures and showing players within a reasonable timeframe the effects of their plans on ship traffic.*

We answer this question by explaining the design, implementation and results of the shipping simulation *SEL* (Shipping Emulation Layers) within the MSP Challenge. The bulk of this work took place over almost one year, involving co-design with shipping experts, programming and extensive testing and application in three sea basins through 20 MSP Challenge sessions since 2018. In the remainder of this paper we first formalise the requirements that SEL needed to fulfil, before we explain how SEL solves this pathfinding problem efficiently yet realistically.

FORMALISING REQUIREMENTS

The MSP Challenge platform architecture and desired shipping functionality led us to define a number of requirements for input, output and throughput of the shipping simulation. We explain the platform architecture and our chosen requirements in this section.

MSP Challenge Architecture

MSP Challenge is a data-driven client-server platform, enabling sessions with different scenarios, regions and time frames (Figure 1). The platform uses advanced game technology to link real geo- and marine data with simulators for specific maritime sectors, mainly ecology, energy and shipping. These simulators are satellite applications interconnected with the game server. They add dynamic data to the game on the levels of ecology, energy and indeed shipping. They have a discrete-event architecture, with each discrete event representing one simulated month. A single game client can act as a player or game master and connects to the server, which is responsible for maintaining the current game state and interfacing with the simulations. The actual time between each discrete event is defined by the game master and depends on how long they want the entire session to last. In this manner, the MSP Challenge simulates MSP in up to four rounds of planning and simulation, each round representing as many years as the game master defines.

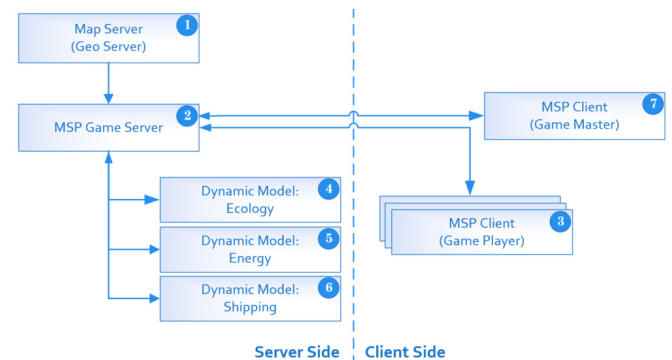


Figure 1: MSP Challenge high level architecture

A typical MSP Challenge session takes at least half a day, during which around 30 players are grouped into 5 to 9 country teams. They design and implement at least 20 independent maritime spatial plans that each alter any of the roughly 40 data layers, and analyse and evaluate resulting key performance indicators on the levels of ecology, energy, and indeed shipping. For players to analyse and evaluate results, the MSP Challenge needs to be able to obtain and pass through data of each month reasonably quickly.

The different background simulations must read the maritime spatial plans defined by all players, and calculate and feed back the combined results and consequences. Obviously, the quicker the simulation can do this, the better. Yet, if the shipping simulation would take between 5 to 10 seconds of computation per month, this would translate to around 15 minutes per 10 years. This is an acceptable

performance level, as it still fits the typical dynamic of an MSP Challenge session.

Shipping Emulation Layer Objectives

We defined SEL's objective as to generate reasonably realistic ship intensity information per discrete event (each simulated month) for a gameplay period of several decades. The information needed to be split over several different types of ships, each with different behaviour: cargo, tanker, maintenance, passenger and ferry ships.

The primary output of SEL needed to consist of rasterised heat maps showing the intensity of ship traffic in the simulated area. Given multiple ship types, multiple heat maps would need to be generated. The outputted heat maps would need to be shown in the MSP Challenge client, but would also need to be integrated into other simulations. Particularly, we would require the ability to define from ship intensity information particular ecological pressures that we could then feed into the ecosystem simulation MEL (Steenbeek et al. 2019).

Furthermore, SEL also needed to output certain key performance indicators (KPIs). KPIs give insight into certain aspects of the sea basin state, and are typically highly contextual quantifications. Interpretations of whether or not changes in KPI are improvements or setbacks is part of the game and thus up to the players. Some KPIs are directly influenced by players and others are more for informational purposes. Three main KPI types were defined for SEL to generate:

- The number of ships per port in the simulated month;
- The routing efficiency between any two ports (actual route distance compared to the rectilinear distance);
- The amount of ships travelled over a shipping lane.

As a third and final output, SEL also needed to create shipping routing issues. When SEL was unable to find a route between two points, it should report an issue to the MSP Challenge platform. The issue is then shown to the players in the game client indicating that possibly one of their plans has created a problem for shipping and needs to be investigated. For example, players might create plans which define restriction zones that prevent specific ships to reach destination areas or ports.

SEL Input Data

As input SEL would firstly need all the data in the MSP Challenge sea basin in question to obtain a representation of the simulated world. The server divides data into certain data layers, where data layers can contain any number of planned geometry instances. The data layers of the MSP platform can be classified as:

- *Constant data.* This is data that cannot be edited by players or other dynamic models while a session is running. This data is only requested and fed into the simulation upon startup. An example of static data would be of the landmass or bathymetry (sea depth).

- *Dynamic data.* This is data that may change throughout the session by players' actions, or as the result of another simulation. When a layer changes, it is flagged on the server and will be re-acquired and fed into SEL at the next discrete event (i.e. the next simulated month). Anything planned by players themselves, such as shipping routes, is dynamic data.

The following data is subsequently interpreted by SEL in particular ways:

- *Shipping lanes.* These are route segments in the sea basin. Ships might prefer to take such routes because it is, for example a mandatory route, company policy, or safer. Designated shipping lanes are mostly ship type specific and only present in busy and/or otherwise risky areas. Thus they never comprise complete routes from port to port, but are segments scattered over a sea basin.
- *Ports.* Ports are considered producers and consumers of ship intensity, and are defined as point geometry. Each port has relevant metadata, such as the available fuelling types, port facilities, and the expected number of vessels (per type) arriving or departing per simulated month.
- *Restriction areas.* These either block pathing for all or some ship types. An example of restriction geometry would be the landmass layer which blocks pathing completely for all ship types. Ship traffic separation areas, aquaculture and wind farms are other examples of restriction areas taken into consideration when pathing.

As described above, there is specific metadata behind each port. It was a game design decision to keep the number of vessels "generated" by each port configurable per game session, and allowing the game master to tweak the number of vessels per port before the game session starts. This way different scenarios can easily be configured.

SEL Ship Navigation Considerations

In order for SEL to find realistic paths, we defined the following common ship navigation considerations:

1. *Freedom of navigation and basic economics.* Under international maritime law, ship captains can, in principle, choose their own paths. Ideally, they would choose the most direct and thus most efficient path.
2. *IMO route adherence.* Shipping companies have been following predetermined routes for safety reasons for over a hundred years. Nowadays, traffic separation schemes and shipping routes are the responsibility of the IMO, regulating congested sea areas in the world. An IMO designated route has a strong legal status and the benefit of increased safety in particularly busy areas. We take some exceptions to these rules into account (see final point) but other than those the simulated ships will follow IMO routes.
3. *Obstructions.* Certain obstructions do not allow specific vessels to enter specific areas, notably:
 - a. Specific ships are not allowed to go through human-made structures (notably wind farms, oil and gas installations).

- b. Ship restrictions may be in place, for example for traffic separation, during wind farm construction, or no-shipping zones.
 - c. A shallow area can represent an obstruction to bigger cargo and tanker vessels requiring deeper waters. This is implemented through a type-specific penalty system. Larger ships will have a larger penalty than smaller ships. We cannot treat the shallowest waters (0 - 20 metres) as no-go zones as ships will need to cross these depths to get to certain ports.
4. *Ship type differences.* Although generally each ship wants to take the most direct route possible, there are noticeable differences between each ship type:
- a. Tankers and cargo vessels will try to follow the defined (IMO) shipping lanes as long as they do not create too big a detour.
 - b. Ferry ships will take the most direct route regardless of whether there are shipping lanes it can use.
 - c. Maintenance ships construct and maintain offshore man-made structures. They are the only type allowed to go to and cross over any offshore energy areas (notably wind farms). Maintenance ships are normally smaller, will always take the most direct route, and always originate from the port closest to the port with maintenance facilities.

SEL'S ARCHITECTURE

With all requirements described, in this section we explain the approach we took for building SEL. We specify how each simulated month the simulation finds the paths for all ships to generate the desired outputs.

Pathfinding

The simulation uses a heuristic-based, graph-searching algorithm to find paths between two points on the internal graph (A*) (Hart et al. 1968). There are three main steps to take the data provided by MSP Challenge and transform it into a usable structure for pathfinding.

Step 1: Connection Graph Setup

SEL internally builds a complex graph from the geometry data received from the game server. Port information is added as graph vertices. Similarly, all the geometry points defined in the shipping lane layers are added as graph vertices. All shipping lane connections between points are added as edges on the graph. The rest of the geometry is converted into restriction areas, forming rules that are reviewed when generating the rest of the connectivity graph.

We created a separate layer invisible to players with a set of points in a grid pattern on navigable areas. These grid points create more graph vertices for populating the graph and supporting the pathfinding algorithm in finding alternative paths when required. Moreover, they define the alternative path's resolution, important for defining the degree of resolution for our heat maps.

To simulate the different ship navigation behaviours we implemented a system of rulesets working with restriction zones. Depending on the ruleset configuration we can force specific shipping routes to be very strict and have ships always take them if possible, or let them be more flexible to allow ships to take the shortest paths available. As a result of the A* heuristic function that is used the different routes that SEL calculates are usually sub-optimal in terms of distance travelled, but closer to the real-world results.

Step 2: Route Calculation

Once the input data is set up for the graph we start connecting the entire graph together by creating more edges. SEL loops over every vertex in the graph and connects it to the closest neighbours that are within a certain direction, as long as there is no blocking restriction geometry in between.

To give an example, for every vertex in our graph a connection is made to the closest navigable vertex north, east, south and west of it, as long as there is no blocking restrictions in between. These edges that are created are marked as being *implicit* edges, as opposed to the *explicit* edges which are shipping lanes defined by the data. We uphold this difference between *implicit* and *explicit* edges for the pathfinding algorithm. In the pathfinding implementation travelling over *implicit* edges incurs a configurable cost penalty. This cost penalty influences the pathing in such a way that we can control how likely the ships are to adhere to *explicit* edges (official shipping lanes, e.g. IMO routes) by increasing and decreasing the penalty.

Each edge also stores information about what ship types are allowed to use it and with which direction. The restriction zones the edge crosses influences the types of ships allowed to cross the edge. By default all ships are allowed to path over all edges, but this is changed when an edge is created over a restriction zone that only allows a subset of ship types. The edge copies the allowed ship types from the restriction zone it crosses. The directionality setting restricts in what direction the edge can be crossed and can be set to *unidirectional* (only from start to end) or *bidirectional* (either way) for every edge. This mimics IMO traffic separation schemes.

For finding paths, we query the constructed graph using an A* algorithm that takes into account the edges the ship type can cross and respects the directionality of the edge. Depending on the ship type configuration *implicit* edges are penalised by using a cost multiplier for crossing that edge. Additionally, restriction geometry can specify cost multipliers to make ships only cross the geometry when alternatives are either not found or significantly more costly. A usage example of this restriction geometry penalty is the bathymetry layer. The 0 - 20 metre depth bathymetry layer specifies a large cost penalty for crossing the layer by large ships. This causes large ships to avoid coastal areas that are shallow unless they need to cross it to get to a harbour.

During the pathfinding calculation stage we cache all created routes. Before we calculate a new path, we check if there is a path that already matches our requirements of source, destination, ship type and directionality. For instance routes

from point A to B can be reused for paths from point B to A provided that they do not contain any *unidirectional* edges and allow for the required ship type.

Step 3: Storing the Connection Graph.

The connection graph generated in Step 1 and the paths generated in Step 2 are stored for use in future calculations. By keeping the data, we can significantly reduce the required calculation time for months that do not influence any layers that affect shipping. The graph and routes are completely discarded and recalculated when one of the layers taken into account for the shipping simulation changes.

Heat Map Generation

Generating heat maps is an important step for our implementation. This takes the abstract data of shipping intensities over a route to data that can be visualised in the form of a heat map. The process consists of three steps.

First, SEL generates an unbounded raster of intensity values. This starts with a two-dimensional array of a size equal to the final output image initialised with 0 values. For every route that was calculated the algorithm walks the edges that make up the route. We project the edge onto the raster using a line rasterisation method (Bresenham 1965). When we use this method, every cell the edge crosses has the cell's value increased by the intensity of that route. The rasterised data obtained (Figure 2) contains very sharp results of the actual intensity values for each pixel on the simulated raster.

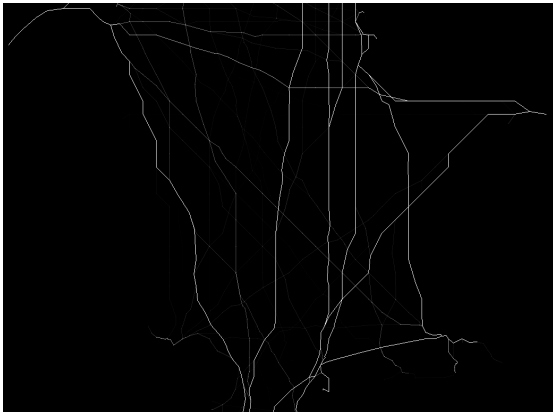


Figure 2: Unbounded intensity map

Second, SEL creates a raster mask defining what areas are inaccessible to each ship type. These images are either fully black or fully white, where all pixels covered by an unpassable restriction zone are white (Figure 3). This mask serves a purpose in the next step, ensuring that we do not blur ship traffic over areas where ships are not allowed to go, for example over land.

Third, SEL blurs out the values to the intended display range. It needs to flatten the unbounded grid values in our heat map to values that we can represent as an image. We use a modified Gaussian convolution matrix as an image filtering technique to spread the intensity values that exceed the chosen maximum (Fisher, Wolfart, and Wiley 1996). If the heat map is configured to contain e.g. max 10 intensity

and the unbounded heat map has values of 50, that value should be smeared out over adjacent pixels (Figures 4 and 5).



Figure 3: Restriction map used in shipping rasterisation

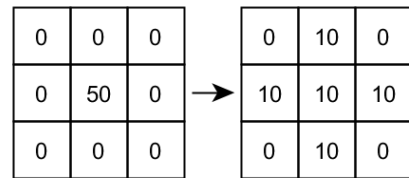


Figure 4: Gaussian convolution filter example and how the intensity values are distributed to adjacent cells

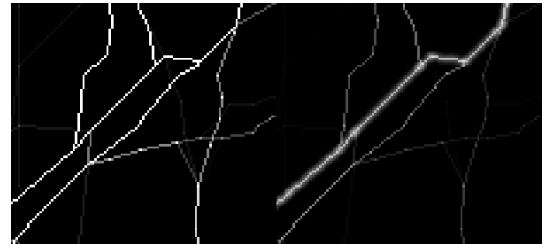


Figure 5: Unbounded heat map (left) with Gaussian convolution filter applied (right)

To increase the accuracy of the Gaussian blur, our implementation takes into account the restriction mask from the previous step to know where it is allowed to put intensity. When a pixel is marked as unavailable in the mask, the blur kernel weights are adjusted to compensate, ensuring that we do not lose intensities from moving them around.

Key Performance Indicators

The key performance indicators that SEL calculates are derived from different data already present after calculating the routes. Three main KPI categories are calculated:

1. The number of ships a port produced in the simulated month. We derive the number of ships of a certain type that a port produces from the input data defined by the scenario. This value is the actual number of ships that are sent over a particular route to a destination, and provides an insight into port development.
2. Per-port routing efficiency percentage. For each port we examine each route, and divide the length of the route and by the rectilinear distance between origin and

destination. This single value fed back is the average efficiency of all routes from the particular port.

3. The amount of ships travelled over a shipping lane. For every lane we track which source geometry it belongs to. Source geometry is only defined for *explicit* shipping lanes. SEL goes over all available routes and all of the edges that make up that route. If an edge is an *explicit* shipping lane, then SEL adds the route intensity to it. The sums of these intensities are incorporated into each shipping lane's metadata.

PERFORMANCE, OPTIMISATION, VALIDATION

In this section we evaluate SEL and the challenges of keeping the simulation running as fast as possible while offering a wide variety of player options and maintaining accuracy to keep the simulation believable. To check and increase SEL's accuracy, we compared the SEL generated maps to shipping intensity data we acquired for three regions: Firth of Clyde, North Sea and Baltic Sea.

Firth of Clyde

The Firth of Clyde is a relatively small sea basin to the west of Glasgow, Scotland. Marine Scotland provided us with real-world AIS data and resulting heat maps for this marine region. In Figure 6 we compare the Marine Scotland heat map to SEL's generated heat map as viewed from within MSP Challenge. The MSP Challenge play area for the Firth of Clyde is just the Clyde estuary, so all the shipping intensity outside was not considered for the simulation, but it is possible to clearly observe the similarities.

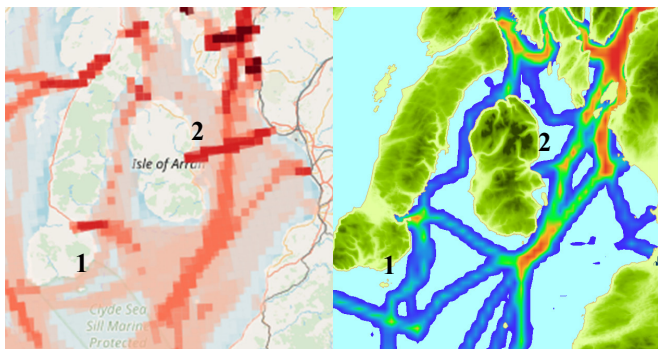


Figure 6: Firth of Clyde - Marine Scotland left, SEL right

We note the following differences between the two maps:

- There is a small island (1 in Figure 6) in the lower left quadrant of the image where the real-world data shows shipping intensity going north of the island, while SEL outputs the intensity further south. This is a result of the resolution of the pathing graph.
- The real-world data shows there is a line between the Isle of Arran (2 in Figure 6) and the Scottish mainland which is very busy. In SEL, this line is completely absent. Real-world maps show there is a ferry route at that exact intensity line. This is because the provided source data did not include this particular ferry route's number of ships per month or geometry.

- SEL noticeably distributes shipping over more separate lines than we see on the real-world heat map. This is a result of our pathing algorithm implementation.

Over the course of 2017, several Marine Scotland MSP professionals were involved in this implementation and evaluated the developments and final results. They deemed the results close enough to reality and representative enough for the region to render it useful for MSP Challenge sessions oriented towards education, training and stakeholder engagement. In early 2018, the simulation was applied in two MSP Challenge sessions with a total of 21 participants, both successful in their respective objectives of stakeholder engagement and higher education.

North Sea

The North Sea is a much larger sea basin in Europe, known for its heavy traffic. We acquired total shipping intensity data and heat maps for this sea basin concerning the period July 2016 - July 2017 from the Havbase website. In Figure 7 we compare the Havbase heat map to SEL's generated heat map as viewed from within MSP Challenge.

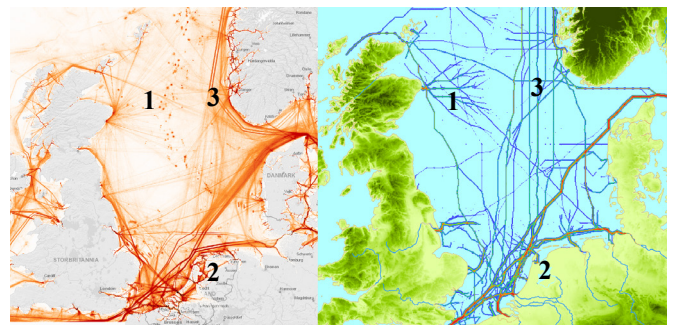


Figure 7: North Sea - Havbase left, SEL right

We note the following differences between the two maps:

- Due to the large amount of energy facilities, SEL generates a large amount of maintenance ships travelling to and from them. This is particularly visible to the right of Scotland, with its many oil and gas installations (1 in Figure 7). In this case this seems to be quite similar to the real world. However, SEL also does this for wind farms. While this is in itself realistic, we do not see these maintenance intensities in the same manner and with the same ports of origin in the real-world data.
- In the northern part of the Netherlands, at the Den Helder port (2 in Figure 7), there is a routing error that causes ships to go around the island of Texel before faring into the sea basin. This is an issue caused by the way that we treat bathymetry-based cost penalties.
- Like in the Firth of Clyde, SEL distributes shipping intensities much more on separate lines than can be seen on the real-world map (3 in Figure 7). This is again a result of our pathing algorithm implementation.

Over the course of 2017 and 2018, we worked extensively with several maritime professionals within the NorthSEE partnership to get to this implementation. They generally deem the results valuable, although they also tend to quickly

point out the aforementioned differences with the real world. Since 2018 we have applied the simulation in 15 MSP Challenge sessions with a total of 215 participants, all successful in their objectives of stakeholder engagement, planning support and higher education.

Baltic Sea

The Baltic Sea is an even larger sea basin in North-Eastern Europe, officially spanning the Kattegat, Baltic Proper, Bothnian Sea and Gulf, Gulf of Riga, and Gulf of Finland marine regions. In this case the Baltic Marine Environment Protection Commission - Helsinki Commission (HELCOM) provided us with real-world AIS data and resulting heat maps concerning every month of 2016, split over our (and more) ship types. This was as yet the most complete dataset we were able to obtain for an area this large. In Figure 8, we compare the HELCOM heat map to SEL's generated heat map as viewed from within MSP Challenge.

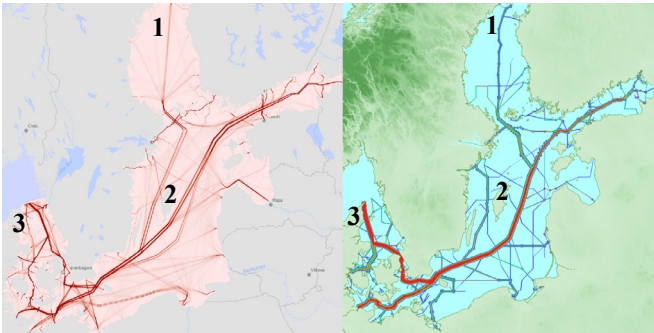


Figure 8: Baltic Sea - HELCOM left, SEL right

We note the following differences between the two maps:

- Missing data from ports in the northern Bothnian Gulf (1 in Figure 8) created a noticeable difference in ship dispersion.
- The large island of Gotland (roughly in the middle of image) allows ships to go past both sides of the islands (2 in Figure 8). To avoid congested areas, most of the times ships will take alternative routes in the real world. In this case, that would mean they divert to going north of the island to avoid congested areas at a cost of a (slightly) longer route. In our model alternative routes are not considered when the area reaches a specific density threshold.
- There is a significant difference in the position of shipping intensity at the Kattegat entry and exit area in the west (3 in Figure 8). Again this is attributed to the fact that our model does not include congestion.

Over the course of 2017 and 2018, we worked extensively with several maritime professionals from HELCOM within the context of the Baltic LINes partnership to get to this implementation. Similar to the NorthSEE partners, they generally deem the results valuable, although they also tend to quickly point out the aforementioned differences. Since 2018 we have applied the simulation in three MSP Challenge sessions with a total of 66 participants, all successful in their objectives of stakeholder engagement and planning support again.

SEL Performance

As can be imagined, the processing times of the three implementations differ highly. For the Firth of Clyde dataset (Figure 6) it takes around 4 seconds for the initial processing step to complete, resulting in around 3,000 routes. For the heaviest Baltic Sea dataset (Figure 8) this initial processing step takes around 17 seconds to complete, resulting in over 33,000 routes.

Each simulation step after the first takes less time since we can re-use data that we previously processed. When one or more of the *dynamic* data layers have been changed that SEL uses, subsequent steps for the Firth of Clyde take around 2.5 seconds and for the Baltic Sea around 13 seconds to complete. When no data layers are changed, the simulation time is reduced to less than one second for any of the three implementations.

The two parts of the simulation that currently take up the most time are building the pathing graph and calculating the routes, approximately 30% and 50% of total processing time, respectively when measured on the Baltic Sea data. The simulation time that we are currently able to achieve is slightly above MSP Challenge targets. Optimising the simulation, while keeping the same level of accuracy for the results, is an ongoing task. Still, the performance we are able to achieve is currently not a bottleneck in any of the sessions we are running.

A performance improvement that we can still implement with the current SEL architecture is to locally rebuild the pathing graph. Currently, if a data layer changes that SEL uses the entire graph is discarded and rebuilt. When we use the new graph, all paths are recalculated to ensure the routes are still valid. Instead of rebuilding the entire graph, we could invalidate a smaller portion of the graph and only rebuild that area. This has the potential to drastically lower the rebuild times of the graphs. Determining which paths need to be rebuilt is still a challenge as a change might open up shortcuts that were not possible before.

CONCLUSION

In this paper we have discussed how we created a convincing shipping simulation which can perform its calculations in an acceptable period of time. The simulation we outlined works with, and responds to, dynamic data fed into it by the MSP Challenge server. We presented how we approached the implementation of shipping simulation as a graph-based pathfinding problem, and how we rasterised this graph data to build a convincing heat map by use of several techniques, including a modified Gaussian blur, for use within the game environment.

We implemented this simulation in three marine regions: Firth of Clyde, North Sea, and Baltic Sea. In all three implementations we worked with shipping professionals to understand the shipping logic and obtain shipping intensity data from the region. We determined that with these three

highly diverse regions the simulation is able to run within a small timeframe (4 - 17 seconds per simulated month).

We applied all three regions in a total of 20 formal MSP Challenge sessions successfully reaching their objectives of stakeholder engagement, planning support, and higher education. We incorporated feedback obtained during each application to improve the simulations for later sessions, and identify even further improvement potential. The provided outputs are nonetheless very suitable for representing ship navigation behaviour, and for keeping players engaged and thinking about (in)direct impacts of their plans on shipping.

We continue to improve the accuracy of the simulation. The first improvement that we will address concerns how we treat bathymetry cost penalties. Currently the cost penalties are incurred every time a ship crosses the line between deep and shallow water, while no cost is incurred as long as the ship is within shallow water. This is a naive approach, but works well for a large part of the shipping routes. There are a couple of routes where this nonetheless leads to unrealistic path segments. If we constantly incur cost penalties while a ship is within shallow waters, this might improve the accuracy of the paths. Moreover, as seen in the Baltic Sea shipping intensity data, there are several inaccuracies that can be attributed to our simulation not taking congestion into account. Implementing congestion into the simulation to have ships avoid congested areas is another accuracy improvement with high potential.

ACKNOWLEDGEMENTS

The research leading to these results acknowledges the contribution of the NorthSEE (2016-2020), Baltic LINes (2016-2019), SIMCelt (2016-2018) and SP-MSP (2016-2019) projects, which received funding from the European Union's Interreg North Sea Region (NSR) and Baltic Sea Region (BSR) programmes (both part of the European Regional Development Fund), the European Maritime and Fisheries Fund (EMFF), and the Erasmus+ programme, respectively.

REFERENCES

- Bresenham, J. E. 1965. "Algorithm for Computer Control of a Digital Plotter." *IBM Systems Journal* 4, No. 1, 25-30.
- European Union. 2014. "Directive 2014/89/EU of the European Parliament and of the Council of 23 July 2014 Establishing a Framework for Maritime Spatial Planning." *Official Journal of the European Union* 2014, 135-45.
- Ferreira, D.C.; R.C. Marques; and M.I. Pedro. 2018. "Explanatory Variables Driving the Technical Efficiency of European Seaports: An Order- α Approach Dealing with Imperfect Knowledge." *Transportation Research Part E: Logistics and Transportation Review* 119, 41–62.
- Fisher, R.B.; E. Wolfart; and J. Wiley. 1996. "Hypermedia Image Processing Reference." In *Hypermedia Image Processing Reference*, 156–60. John Wiley & Sons Ltd.
- Hart, P.E.; N.J. Nilsson; and B. Raphael. 1968. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." *IEEE Transactions on Systems Science and Cybernetics* 4, No. 2, 100-107.
- Hutchinson, K.; H.J.G. Warmelink; W. Boode; C. Santos; and I.S.

- Mayer. 2018. "An Offshore Energy Simulation through Flow Networks: CEL within the MSP Challenge 2050 Simulation Game Platform." In *The 32nd Annual European Simulation and Modelling Conference*. Ghent, Belgium: EUROSIS.
- Itoh, H.; M. Numano; and E. Pedersen. 2003. "Modelling and Simulation of Sea Traffic and a Visualization-Based Collision Avoidance Support System." In *Modsim 2003: International Congress on Modelling and Simulation*, edited by D. A. Post, 1996–2001. Modelling and Simulation Society of Australia and New Zealand.
- Mayer, I.S.; Q. Zhou; X. Keijser; and L. Abspoel. 2014. "Gaming the Future of the Ocean: The Marine Spatial Planning Challenge 2050." In *Serious Games Development and Applications*, 150–62. Lecture Notes in Computer Science. Springer, Cham.
- Mayer, I.S.; Q. Zhou; J. Lo; L. Abspoel; X. Keijser; E. Olsen; E. Nixon; and A. Kannen. 2013. "Integrated, Ecosystem-Based Marine Spatial Planning: Design and Results of a Game-Based, Quasi-Experiment." *Ocean & Coastal Management* 82 (Supplement C): 7-26.
- Nilsson, H.; J. van Overloop; R.A. Mehdi; and J. Pålsson. 2018. "Transnational Maritime Spatial Planning in the North Sea: The Shipping Context." Research Report NorthSEE Project.
- Steenbeek, J.; G. Romagnoni; J. Bentley; J.J. Heymans; N. Serpetti, M. Gonçalves; C. Santos, et al. 2019, under review "Combining Ecosystem Modelling and Serious Gaming to Aid Transnational Management of Marine Space."
- Stolte, W.; X. Keijser; A. de Kluijver; and A. J. Nolte. 2013. "A Conceptual Pressure-Response and a Simple Food Web Model for the Maritime Spatial Planning Challenge 2050 Serious Game." 1208167-000-ZKS-0007. Delft: Deltares.
- Wolfrum, R. 2008. "Freedom Of Navigation: New Challenges." In *Freedom of Seas, Passage Rights and the 1982 Law of the Sea Convention*, edited by Myron H. Nordquist, Tommy Koh, and John Norton Moore, 79-102. Brill.

WEB REFERENCES

- www.balticlines.eu
- www.havbase.no
- www.imo.org/en/About/Conventions/ListOfConventions/Pages/default.aspx
- maps.helcom.fi
- www.marin.nl/web/Facilities-Tools/Simulators/Simulator-Sales/Vessel-Traffic-Service-VTS-Simulators.htm
- marinescotland.atkinsgeospatial.com/nmpi
- www.mspchallenge.info
- www.northsee.eu
- www.simcelt.eu
- stmvalidation.eu/projects/stm-validation/

BIOGRAPHIES

PHIL DE GROOT, WILCO BOODE and **CARLOS SANTOS** are all part of the Breda University of Applied Sciences (BUas) games and media research and development team *Cradle*, functioning as core developer, lead designer, and technical manager of the MSP Challenge Simulation Platform, respectively. **HARALD WARMELINK** is a senior project leader at BUas for MSP Challenge and other games & media R&D projects. **IGOR MAYER** is professor of serious and applied gaming at BUas and functions as the MSP Challenge director.