Integrating Ecopath with Ecosim into the MSP software - conceptual design

Jeroen Steenbeek

Ecopath International Initiative

Date:	2 February 2018
Version:	2.5
Status:	Final

https://doi.org/10.6084/m9.figshare.5849634

Ecopath
International
Initiative

Document revisions

Version	Status	Date	Author	Description
0.1	Initial	17 /10/2016	JS	First conceptual draft
0.2	Draft	14/11/2016	JS	Incorporated NHVT discussion topics
0.9	Draft	23/11/2016	JS	Completed text and figures
1.0	Revision	25/11/2016	JS	Incorporated NHTV review
2.0	Revision	3/12/2016	JS	Incorporated RWS review
2.2	Final	11/12/2016	JS	Approved, finalized
2.5	Published	2/2/2018	JS	Published on FigShare

Contents

Figures

Tables

1. Introduction

Ecopath International Initiative Research Organization (EII) has been commissioned to conceptualize the integration of the Ecopath with Ecosim (EwE) food web modelling approach into the Marine Spatial Planning (MSP) Challenge 2050 software.

The MSP Challenge 2050 serious game is an immersive, three-dimensional software system that educates stakeholders, planners and the general public about trade-offs and environmental consequences of planning decisions in the marine and coastal ecosystem over space and time (Stolte et al., 2013). The ecology component in the MSP software is a simplification of a EwE model of the North Sea based on Mackinson and Daskalov (2007). This food web model was recently reviewed and was deemed unsuitable to deliver useful ecological predictions, even for game purposes (Steenbeek, 2015). Rather than fixing identified shortcomings in the MSP software, it has been decided to replace the MSP ecology calculations with a direct link to EwE.

EII and a team at NHTV Breda University of Applied Sciences have been commissioned to conceptualize this linkage. This document provides the conceptual design of this linkage from the EwE point of view. A brief overview of EwE and its internal structure is given. Capabilities of EwE to incorporate MSP player input into its food web dynamics, and the mechanisms to provide game feedback back to MSP, are identified. The essential data flow to and from EwE is explained, and a conceptual communication protocol between MSP and EwE is provided.

NHTV will provide an accompanying document that conceptualizes the integration EwE into the MSP software from the MSP point of view.

2. Ecopath with Ecosim

2.1. EwE in a nutshell

Ecopath with Ecosim or EwE is a open source ecological/ecosystem modelling software suite to numerically represent the energy dynamics of a food web. EwE is used by approximately 7000 users worldwide, and is under active development through the Ecopath Research and Development Consortium. EwE is freely available from http://www.ecopath.org.

EwE has three main components: Ecopath, Ecosim, and Ecospace.

2.1.1. Ecopath

Ecopath is the mass-balanced snapshot of an ecosystem. Here, food web components are represented as functional groups, which are connected through predator-prey linkages. In Ecopath, all energy flows into, through and out of the system should be accounted for, yielding a mass-balanced static snapshot of an ecosystem. Ecopath requires a relatively low number of parameters for each functional group often readily available from traditional stock assessments, local studies, and online repositories such as FishBase (Christensen and Walters, 2004).

2.1.2. Ecosim

Ecosim is the time-dynamic model of EwE. Here, the Ecopath assumptions are played out over time, requiring only a few extra parameters related to species interaction dynamics. Time series of observed trends in catches, species biomasses, fishing mortality etc. can be included to either drive the Ecosim dynamics, or to validate its predictions in order to hind-case known trends. When an Ecosim model is fitted to time series it can be used for future scenario exploration. Ecosim can include changing environmental conditions and the functional responses of ecosystem components to these conditions. Ecosim is mostly used for policy exploration (Christensen and Walters, 2004; Heymans et al., 2016).

2.1.3. Ecospace

Ecospace is the spatial-temporal module of EwE, where the Ecosim dynamics are played out over a spatial grid of cells, and fishing effort and functional groups gravitate toward better conditions. Ecospace requires a set of extra input parameters related to species movement, and spatial fields that define the environment such as a bathymetry map, habitats, primary productivity, fisheries restrictions, threats and pressures. Ecospace was primarily designed for exploring different placement scenarios of protected areas and their impacts, but is increasingly used to explore the impact of environmental change and human perturbation on the ecosystem (Christensen and Walters, 2004; Christensen et al., 2014; Steenbeek et al., 2013). The majority of recent scientific advances in EwE model development have focused on Ecospace.

Ecospace offers mechanisms that could be used for the MSP Challenge 2050 to impact food web dynamics:

Habitat affinity

From to the very first release of Ecospace, functional groups in Ecospace can be distributed across the modelled area via affinity to habitats. In order to distribute functional groups, EwE modellers define habitats and their spatial distributions, and then define habitat preferences per functional group. This allows detailed control connecting species distributions to substrate.

Habitat affinity will be useful for the MSP game to represent the impact of artificial habitats.

Habitat foraging capacity model

The capacity model introduces functional group preferences for environmental conditions. Each environmental condition in EwE is entered as a map layer that represents distributions of, for instance, salinity, oxygen content, surface temperature, etc. over space. Different species are known to have different responses to these conditions, which is expressed in EwE through a response curve that defines the minimum, optimal, and maximum tolerances to the environmental condition. The capacity model accepts any number of environmental drivers, and functional groups can respond to any number of these drivers, giving modellers a large degree of freedom to represent environmental impacts in their model. The capacity model is explained in detail in Christensen et al. (2014). The capacity model can be used with habitat affinity.

The capacity model provides the flexibility and responsiveness needed to include environmental changes caused by MSP player actions into the ecology.

Fishing intensity and fishing restrictions

The Ecopath model can contain any number of fishing fleets, each valuing and targeting any number of functional groups. The temporal Ecosim simulations introduce variability in the employed fishing effort intensity per fleet, which serves to either reproduce historic trends or to explore possible future scenarios. The spatial-temporal module Ecospace then distributes the fishing effort, at the given fishing intensity, across the represented ecosystem area using a gravity model. In this gravity model, cost of fishing is balanced against the yield of target species where fishable.

Ecospace offers different means to control fishing.

- Marine Protected Areas can restrict fishing, per month, per fleet, for user-specified cells in the Ecospace map;
- Habitats not only define species' affinity with substrate, but can also pose limits to specific fishing gear. Users can define any number of habitat types in the Ecospace model, and assign habitat distributions across the Ecospace map;
- Total employed fishing effort can be controlled as a scalar value per fleet, determining the amount of fishing that Ecospace will employ across the Ecospace map.

All three controls offer potential for impacting the ecology layer through MSP player interactions. If sufficient in size, regulatory obstacles such as Natura 2000 sites can be used to form Marine Protected Areas; artificial habitats and man-made structures can become unfishable habitat; and total fishing effort can be used as a player control measure to fish or reprieve stocks.

2.2. EwE technical organization

The EwE software is written in .NET, a high-level framework of languages based on the Common Language Infrastructure (CLI) standard (ECMA International, 2012). Due to its modular design the EwE model has been deployed in many different ways (Figure 1). Although the EwE desktop application is only available for Windows Desktop, the operating system independent computational core has been deployed on non-Windows computational clusters (Steenbeek et al., 2016). The EwE plug-in system is extensively used to extend or share the food web computations with third-party analysis (Steenbeek et al., 2016). Last, due to its reliance on the CLI standard, the EwE code based can potentially communicate with, and can be directly integrated in, code written in any other CLI-dependent languages. This offers the potential to connect the EwE software directly to Python, the programming language used to implement the MSP software (Steenbeek, 2015).

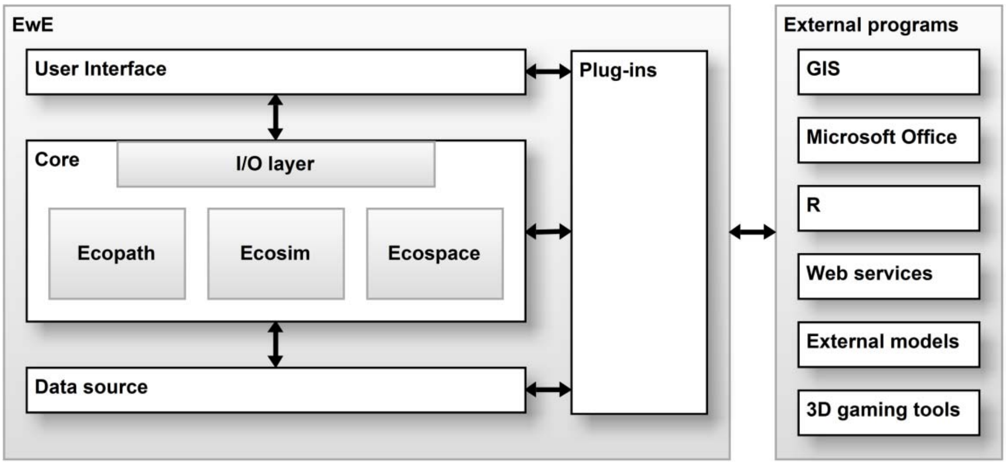

Figure 1 – The EwE internal organization (Steenbeek et al., 2016)

2.3. A note about EwE nomenclature

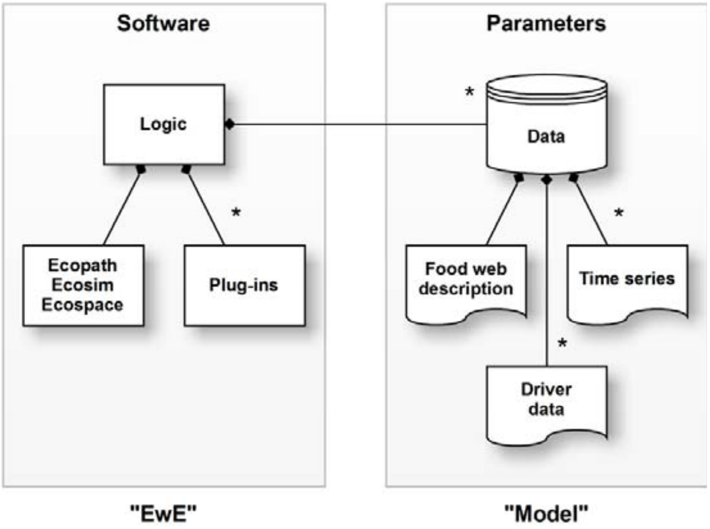

Figure 2 – EwE terminology. The software with its Ecopath, Ecosim and Ecospace logic is often simply referred to as "EwE", while the parameter sets representing an ecosystem is referred to as a "model".

The EwE approach is fully data-driven; hundreds of ecosystems worldwide have been represented via different parameterizations of the EwE software (Colléter et al., 2013, 2015) using the same equations. The EwE desktop software typically stores these parameterizations in a Microsoft Access file, where each file can represent a different ecosystem, or a different scope on the same ecosystem. Because of the separation of the EwE modelling logic and its data, EwE users tend to refer to the EwE software as "EwE", and to their data as a "model" (Figure 2). This is perhaps technically incorrect, but it is indicative of the organizational flexibility of the EwE approach.

3. MSP – EwE linkage

As described in Steenbeek (2015), the MSP Challenge 2050 contained a limited food web model that was loosely based on the Ecopath with Ecosim approach. For the MSP Challenge 2050 redesign phase it was decided to discontinue the internal food web model. EwE was selected to replace the ecological computations for a number of reasons:

- Model credibility: EwE is the most widely used ecological food web model in the world, with over 400 peer-reviewed models that have been fitted to time series (Colléter et al., 2015) and a high level of uptake in advisory bodies;
- Data availability: the MSP Challenge focuses on European seas for which many EwE models have been developed. These models can provide the ecological interactions for serious game scenarios;
- Model capabilities: Ecospace incorporates environmental variability and habitat suitability into the spatial food web calculations, which directly translate to the notion of human pressures in the MSP software;
- Model connectivity: the EwE software is compatible with the Common Language Runtime (CLR) as is Python, the language used to develop the MSP software. EwE has been linked in the past to server-client Python environments (Steenbeek et al., 2016);
- Separation: separating the food web calculations from the MSP game core means that both components can be tested and validated in isolation and together. This will bolster robustness of system results;
- Support: Europe has a large number of EwE expert modellers that can help organizing and fine-tuning models for MSP Challenge game scenarios.

Linkage of MSP and EwE is an exercise with clear technical and scientific components.

- Technically, the MSP – EwE linkage will be set up as a server-client architecture that must support all data exchange needed to start, run and terminate a MSP game, controlled by the MSP server. The conceptual structure of this architecture is presented in chapter 3.
- Scientifically, the MSP model and EwE model to cover the same domain at agreed resolutions in terms of time, area, and game inputs and outputs. The conceptual steps needed to prepare and optimize a EwE model for MSP game play are discussed in chapter 4.

3.1. Server – Client organization

NHTV will connect the MSP server and EwE via an intermediate process called MEL (**M**SP **E**wE **L**ink). Within the MSP game, players can make management decisions that may affect the ecosystem. EwE is responsible for calculating ecosystem changes, and communicating these changes back to MEL (Figure 3).

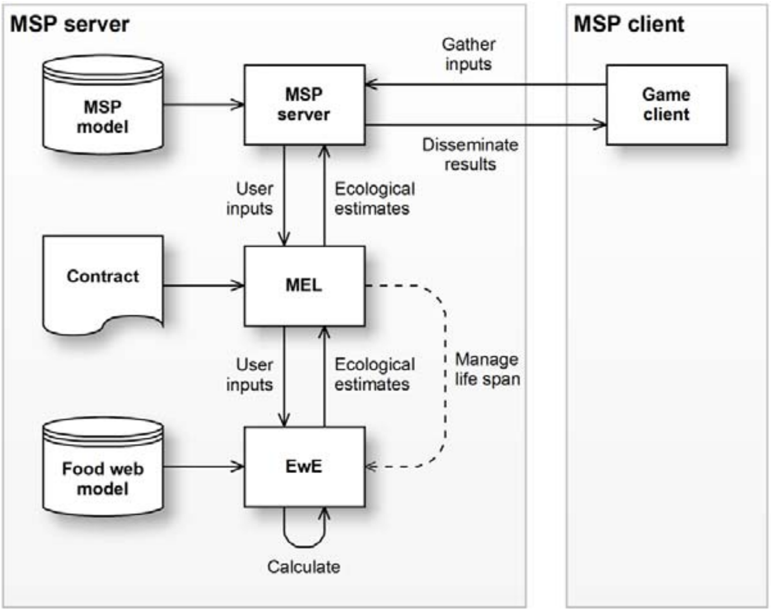

Figure 3 – Conceptual linkage of the MSP game server and EwE software

From the EwE point of view, MEL has a series of responsibilities:

3.1.1. Managing the life span of the ecological computations

MEL will control the execution of the ecological model during an MSP game session. This involves tasks such as starting, time stepping, and stopping the ecological model, but also restoring the ecological model run state in the eventuality of technical problems.

This aspect of MEL is detailed in the NHTV design documentation.

3.1.2. Ensuring that MSP and EwE communicate about an agreed set of data

The MSP game as planned for delivery in 2017 will be a data-driven engine that can be fed different parameter sets to represent different marine areas with different management challenges. As explained in paragraph 2.2, EwE works on a similar basis.

For a given game session it is essential that both MSP and EwE use data that have been fine-tuned for particular game play (see chapter 4). As part of this game preparation stage a data file called a data contract will be created which provides both MEL and EwE with the means to validate, at game start-up, whether both tools have the correct data loaded to play a specific MSP game.

The game contract must declare the following:

- Game area. MSP and EwE must operate onto the same area, with identical grid resolution. The spatial resolution of the grid will be determined by MSP game objectives, requiring a cell resolution fine enough to provide species distributions and representation of pressures, but coarse enough to ensure fast run times. A presumed target resolution of 100 x 100 cells could satisfy both requirements;

- Time step. Both the MSP client an Ecospace already operate at a temporal resolution of one month. This is sufficient for game play purposes, and will need no revisiting;
- Pressures. MSP and EwE must reflect and incorporate the same set of spatial and non-spatial pressures relevant to the game;
- Outputs. EwE must be able to provide ecological estimates meaningful to the MSP game play purposes.

3.1.3. Communicating player input to EwE

MEL will be responsible for receiving and combining user actions from the MSP server, and translating these actions into pressure layers that EwE can understand. MEL will send the pressures layers to EwE, and will request EwE to calculate ecological impacts. This aspect of MEL is further explained in the NHTV design documentation.

3.1.4. Communicating ecological estimates back to MSP

MEL will be responsible for receiving ecological estimates computed by the EwE model, and sending these back to the game server for further dissemination. This aspect of MEL is further explained in the NHTV design documentation.

3.2. Actions and pressures

During the flow of the MSP game users make management decisions which impact the ecology. In discussion with NHTV, two categories of player controls were identified.

1. Spatial management decisions require MSP players to explicitly designate affected areas as subject to a particular decision. With regards to impacting the ecology, relevant actions include infrastructure developments, shipping, closing an area for marine protection, etc. These decisions all take part within a restricted arena designated by players.
2. Non-spatial management decisions can be thought of as policies goals or thresholds, without a clear spatial component. This type of decision will have spatial impacts, but players instead will provide a target threshold number for the MSP complex to take into account or satisfy while performing its calculations. Non-spatial management decisions could be an overall reduction or increase of fishing effort of a given fleet, an incentive to create a certain number of jobs in a specific industry, etc. The impacts of these decisions will have a spatial component, but the decision itself is provided as a single number or set of numbers.

For the MSP ecological calculations all management decisions are referred to as actions. The NHTV documentation lists a long list of actions that the MSP game can and could support. MEL gathers the actions from the MSP game server, and translates the actions to a reduced list of pressures that may or may not affect the ecology. The pressures are sent to EwE, which will calculate the impact of the pressures onto the ecosystem. The resulting ecological estimates are sent back to MEL for further dissemination (Figure 4).

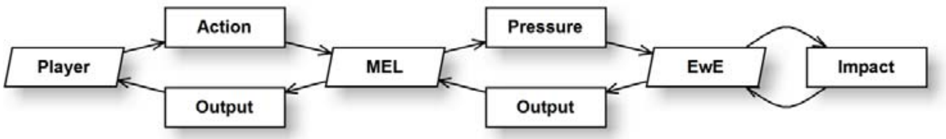

Figure 4 – Translation of player actions into pressures, which EwE uses to calculate ecological impacts and produce ecological outputs for dissemination back to MSP players.

3.2.1. Spatial actions

As explained above, MEL is responsible for collecting player actions and aggregating these into pressure layers. NHTV documentation provides a long list of spatial actions that the MSP must and could support to provide a complete gameplay experience. The effect of spatial actions can be categorized into a reasonably short list of pressures onto the ecology:

Table 1 – Spatial pressure layers to impact the ecological model. Each layer expresses the intensity of a pressure, per cell, across all user actions, for the entire MSP game area. More pressures can be added, depending on the MSP game purposes.

#	Spatial pressure	Units	Description	Implies
1	Noise	[0, 1]	Average noise levels per cell. Noise will most likely be calculated on a logarithmic scale, increasing rapidly in lower levels and gain decreases with higher noise levels.	
2	Bottom disturbance	[0, 1]	Amount of cell bottom area that is disturbed through construction, sedimentation, etc. Bottom disturbance will be calculated on a linear scale.	
3	Surface disturbance	[0, 1]	Amount of cell surface area that is disturbed through construction, sedimentation, etc. Surface disturbance will be calculated on a linear scale.	
4	Artificial substrate	[0, 1]	Amount of cell area covered by man-made habitat. Substrate cover will be calculated on a linear scale.	Protection
5	Protection	Yes/No	States if a cell is closed for fishing	

The NHTV documentation describes the translation of spatial actions translate into pressures. The EwE model will only need to respond adequately to the spatial pressures; it will not need to know which player actions generated specific spatial pressures. To add to game play realism it has been discussed that user actions may go through different phases during the life span of the action, thus generating different amounts of different pressures over time. For instance, a wind park will see a construction phase with bottom and surface disturbance and high noise levels for a few game years. Once construction is completed, the disturbances will dissipate and noise levels will be reduced. If implemented, MEL will be responsible for adding these dynamics to the pressure calculations.

For simplicity reasons all pressure layers will be expressed as either presence/absence (or true/false) values per cell, or intensity per cell on a scale from 0 (no pressure) to 1 (maximum pressure), for each cell.

All pressure layers should be complete; there will be no notion of “missing values” in the pressure layers. If a specific action does not apply to a given cell, the intensity of that pressure will be 0 (in the case if a scalar pressure), or false (in the case of a presence/absence pressure).

Note that the MEL – EwE connection must be designed with flexibility in mind: adding new pressures should be a straight-forward task if required for a particular game play scenario. Concerns to include pressures such as pollution and marine litter, for instance, have already been expressed.

3.2.2. Non-spatial actions

As explained earlier, EwE may need to respond to non-spatial pressures that represent policy goals or thresholds that the model should honour. One such pressure identified in Steenbeek (2015) is the intensity of fishing per fleet, which users can increase or reduce to affect the ecology without having to specify a desired effort distribution: the EwE model will allocate available fishing effort where most logical (Table 2).

Table 2 – Non-spatial pressures to impact the ecological model. More pressures can be added, depending on the MSP game purposes.

#	Non-spatial pressure	Units	Description	Implies
1	Fishing intensity	[0, ∞⟩	Scalar to initial fishing pressure, per fleet	

Just as with spatial pressures, the MEL – EwE connection should offer sufficient flexibility to include other non-spatial pressures if required for a specific game scenario.

3.3. Ecological impacts

After the pressures layers collected and sent to EwE by MEL, the ecological model will need to assess their impacts on the food web and fisheries, after which ecological outputs are aggregated and sent back to MEL. This section provides a brief breakdown of the main tasks of the ecological model.

3.3.1. Calculating ecological impacts

Table 3 summarizes how the six pressures identified in section 3.2 will be incorporated by EwE using the mechanisms to drive the food web model dynamics from section 2.1.3.

Note that the ecological impacts described here are based on the previous assessment of spatial and non-spatial pressures. Since the list of these pressures can be extended, the ecological impact system can also be extended with ease if necessary.

Noise

Although noise is known to have a wide range of effects on marine species (see, for instance, synthesis papers such as Bailey et al., 2010; Thomsen et al., 2006), for gameplay purposes it will probably sufficient to implement avoidance as the main response. Noise will apply to only a select number of fish and mammal species with auditory capabilities, and these species are generally able to move away from the source of the noise.

The EwE mechanism to trigger avoidance is a local reduction in cell suitability for sensitive functional groups via the capacity model. Noise pressure intensities are fed into a dedicated environmental driver layer, to which sensitive species respond with functional response curves.

Bottom and surface disturbance

These disturbances generally cause severe deterioration of local conditions for almost all marine species due to increased and lasting turbidity and sedimentation. Stationary and slow-moving species will experience an increase in mortality, while faster moving species will be able to flee the disturbance.

The EwE mechanism to reduce local conditions due to intensity of disturbances is the capacity model. Bottom and surface intensities are fed into two respective environmental driver layers, to which sensitive functional groups respond with functional response curves. Functional groups with low dispersal rates (Christensen and Walters, 2004; Walters et al., 1999) will not be able to move out of the way, and will experience local mortalities.

Artificial substrate

Establishment of artificial substrate has two main effects. First, it will provide spawning habitat, shelter, and protection from predation for a number of functional groups in the ecosystem. Second, sufficient artificial substrate cover will impose noticeable fishing limitations.

In EwE, both effects can be represented at once by using habitat affinity (see 2.1.3). New, dedicated habitat(s) will receive artificial substrate pressure layer(s) data, where cell values express the fraction of a cell area covered by artificial substrate. Through the traditional Ecospace habitat affinity setup benefiting functional groups should be given a strong affinity to the artificial habitat(s) to stimulate a general improvement in local conditions, with subsequent biomass gains over time.

To implement protection from fishing, fleets can be assigned varying levels of access to the artificial substrate habitat(s).

It may be necessary to keep certain species away from artificial substrate. The traditional Ecospace habitat affinity setup may have sufficient impact to see the effect on affected species. Often, the avoidance is more likely a response to noise (such as wind turbines, Bailey et al., 2010) than the presence of the actual structures. MEL can add noise around wind turbines to ensure this impact.

Protection

Areas designated as off-limits for fishing through the protection pressure can directly be represented in EwE through Marine Protected Areas. For every type of marine protection in the MSP game the EwE model should have a matching MPA with relevant fishing regulations in place. The content of the protection pressure layer can be directly fed into the correct MPA map layer in Ecospace.

Fishing intensity

As indicated previously in this report, Ecospace contains a fishing effort gravity model that evaluates the trade-off between the costs of sailing out to fish against the likelihood of catching valuable targeted species, in areas where fishing is permitted and possible (Christensen and Walters, 2004). MSP game players can be given the means to reduce or increase fishing pressure for one or more fleets. This pressure is non-spatial; the fishing intensity is a scalar value from zero to infinity, where 1 represents fishing at nominal (read: Ecopath baseline) level. The pressure will be directly integrated as a scalar factor on the amount of effort that Ecospace can disperse for its calculations.

Table 3 – Overview how identified MSP pressures will be received by EwE, and which impact mechanisms should be used to respond to pressures.

#	Pressure	EwE input variable	EwE mechanism to calculate impacts
1	Noise	Environmental driver layer	Include functional response for sensitive species
2	Bottom disturbance	Environmental driver layer	Include functional response for sensitive species
3	Surface disturbance	Environmental driver layer	Include functional response for each sensitive species
4	Artificial substrate	Habitat layer	Creates habitat affinity for sensitive species Imposes fishing restrictions for all fleets
5	Protection	MPA layer	Imposes fishing restrictions for specific fleets
6	Fishing intensity	Fishing effort multiplier	Increase or decrease the total amount of fishing effort per fleet

3.3.2. Providing ecological estimates

EwE will deliver its ecological estimates back to MEL in the form of output layers. The decision which ecological estimates EwE must deliver back to MEL will largely depend on MSP game play purposes, area characteristics, and ecological model structure.

A base set of output categories is suggested here (see also Table 4)

Fishing efforts

MSP game players may want to know where their fishing efforts have been concentrating. This requires EwE to provide fishing effort distributions to MEL, where decisions can be made about aggregations.

It is presumed that fishing effort distributions should be available on a per-fleet basis. It may also desirable to aggregate fishing effort per country, for instance. Possible effort aggregations will depend on the purpose of an MSP game.

Species biomass

MSP game players may want to know how individual functional groups are faring in the ecological layers. EwE will provide this information for each functional group.

It is presumed that biomass distributions should be available on a per-group basis. It may also desirable to aggregate biomasses in different manners (such as for all commercial species, for threatened species, etc). Possible biomass aggregations will depend on the purpose of an MSP game.

Biomass indicator

Perhaps game players may benefit from seeing generally accepted biodiversity indicators such as the Shannon or Simpson diversity indices (Jost, 2006). This will largely depend on MSP game play purposes and foreseen audiences.

Table 4 – Proposed EwE layer output categories for MEL

#	Output layer	Description	Predefined
1	Fishing effort	Fishing intensity and distribution, as heat maps, per fleet / gear type, per country	All fleets
2	Aggregated fishing efforts	Aggregated fishing intensity and distribution, as heat maps	(depends on game purpose)
2	Biomasses	Relative distribution of functional biomass, log scale	All functional groups
3	Aggregated biomasses	Aggregations of the relative distributions of organism biomasses, log scaled	(depends on game purpose)
5	Biodiversity indicator	A generally accepted biodiversity indicator such as the Shannon Diversity index or the Simpson Diversity Index	(depends on game purpose)

4. EwE model preparation for MSP game play

The infrastructure to link MEL and EwE described so far will have a conceptual layout as shown in Figure 5. This diagram also introduces two new tools that will be needed for the last stage of MSP game preparation: fine-tuning EwE and its data for integration in an MSP game.

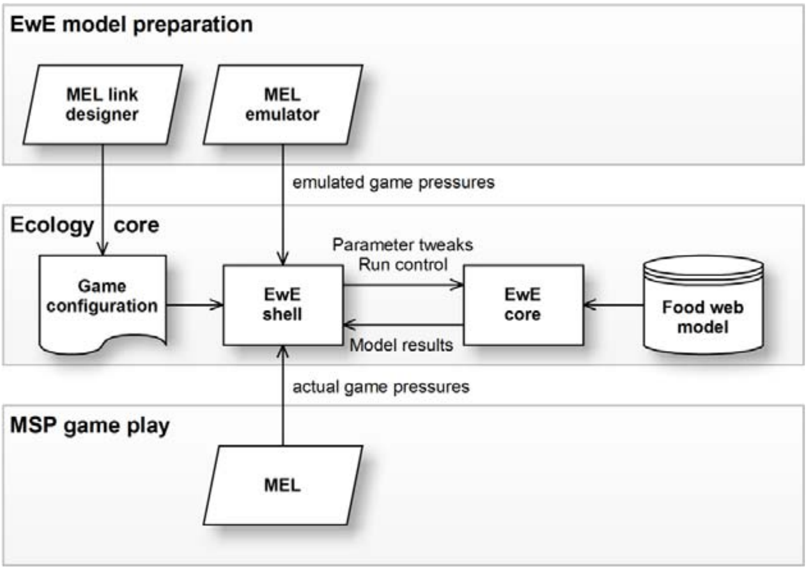

Figure 5 – Layout of the tool set that will be needed to link EwE to MSP via MEL. The ecological core consists of the EwE computational core and a food web model. The computational core is controlled via a new process called "EwE shell" which interacts with MEL. To support the EwE model preparation phase, when a EwE model is adjusted for MSP game play, EwE modellers will need two tools, "MEL link designer" to configure how pressure data originating from MEL will be used in the EwE model, and "MEL emulator" to test how EwE will behave under hypothetical (and possibly extreme) MSP game pressures. During MSP game play, the MEL emulator is replaced with a link to the actual MEL software, delivering actual MSP game pressures.

Note that it should not be needed to build a EwE model from the ground up, as many EwE models are already available that have been fitted to time series (Colléter et al., 2015). It is recommended to only consider including EwE models into the MSP games that have been fitted to time series, as their functioning and structure has been proven to be robust and representative of reality.

4.1. Updating EwE model structure

There are many spatially explicit EwE models available for European seas, including the areas for which MSP games are planned. However, these EwE models have not been developed with MSP game play purposes in mind and may require adjusting. Just as the MSP game needs proper parameterizing for a specific game play scenario, the underlying ecological model will need similar fine tuning to incorporate relevant features, and to respond as expected to MSP pressures.

A series of model changes can be foreseen. The chosen functional group structure may be too complex for MSP game play purposes, or species have been aggregated into functional groups while MSP game purposes require these species to require specific focus as separate functional groups. Fishing fleets may not have been specified at the administrative level required for MSP, and may need splitting or merging. Configuration of Marine Protected Areas and Habitats may require adjustments to be able to implement the impact of required pressures. Most certainly, the capacity model will need to be extended with new functional responses to implement proper responses of the EwE model to MSP pressures.

4.2. Fine-tuning EwE model behaviour for MSP game play

Besides updating the EwE model structure, EwE modellers will have to ensure that the EwE model responds as expected when perturbed through a wide range of possible and often extreme MSP user actions.

Full game play testing, where all components of the MSP – EwE complex are connected and parameterized, constitutes as an essential and final stage in MSP game development. However, testing the EwE responses through the full MSP – EwE configuration is not a practical choice for fine-tuning the EwE model for a few reasons.

First, if EwE modellers can only test the ecological model through the MSP complex, EwE modellers must run a finished and fully parameterized version of the MSP software on their computers. Aside from posing significant hardware demands, this also provides a bottleneck in the MSP game preparation pathway. Second, it will be difficult for EwE modellers to explore the responses of a EwE models trough a wide range of perhaps even unrealistic and extreme pressures. Third, due to workflow differences in both tools, the MSP software system cannot connect directly to the EwE desktop software which is the best available workbench for EwE modellers to fine-tune their data.

As a solution, a simple plug-in for the EwE desktop software must be built that emulates connectivity to MEL during fine-tune stage. This tool, “MEL emulator”, will use the full Ecology core shown in Figure 5, but instead of requiring a live link to MEL and the MSP software, the MEL emulator allows EwE modellers to imitate pressure data to affect the ecological model as if this were to originate from MSP user input. With the MEL emulator, EwE modellers can provide simulated pressure test data for validating model behaviour, responsiveness, and correctness as if directly connected to MSP game play. The MEL emulator can also be made to intercept and validate EwE output data.

4.3. Parameterizing the interactions with MEL

Once the technical link between MEL and EwE has been established there should be no need for further code development. Once technical development is completed, definition of specific gameplay scenarios, which requires fine-tuning the MSP game, a EwE model, and the linkage between MEL and EwE, should be a pure data issue.

A simple tool, “MEL link designer”, is needed as another plug-in to the EwE desktop software where EwE modellers can specify:

- Mappings of MEL pressures to EwE driver mechanisms. Since the list of MEL pressures is in principle open ended, EwE modellers must be able to flexibly define, as part of the game setup, how pressures connect to EwE impact mechanism (see section 2.1.3).
- Mappings of EwE model estimates to MEL outputs. Similarly, since MEL can request any number of output layers of varying content, EwE modellers must be able to flexibly design how ecological estimates are aggregated as output layers to satisfy game play requirements.

These parameterizations will be stored in the EwE model Access database for a particular MSP game, where the EwE shell (see Figure 5) will be able to find and use them for communicating with MEL.

5. Estimates

This last section provides rough time estimates for all developments identified in this document. Additionally, a few risks will be summarized here with mitigation measures where possible.

5.1. Technical developments

All activities related to development, testing, and documentation of new code modules, and integration with existing code modules, are summarized in Table 5. For each activity a rough time estimate is provided, as well as reliance on internal (EwE) and external (MSP / MEL) components.

EwE shell

Development of the EwE shell will take the largest share of time. The EwE shell will be the contact point for MEL, and will function as the mediator between MEL and the EwE computational models. The EwE shell will be responsible for receiving MEL commands, controlling the life span of the EwE model, feeding incoming pressure information into the correct EwE mechanisms, running the model, and extracting output data. Especially the communication to the multi-threaded Ecospace model will take some time.

MEL emulator

This light-weight tool will communicate with the EwE shell during the EwE model preparation phase, and acts as a substitute for a live link to MEL during MSP game play. The MEL emulator can have a simple user interface that allows EwE modellers to send MEL commands to EwE shell. The EwE desktop software will respond to the simulated MEL commands to show EwE modellers the ecological impacts of feigned pressure data.

To limit development time, the MEL Emulator will have a simple user interface. Feigned spatial pressure data will most likely be read from files, while non-spatial pressure data will be entered as single values in the Mel emulator user interface. Received output layers will be written to disk.

Available MEL commands are discussed in Appendix A.

MEL link designer

This second light-weight tool allows an EwE modeller to configure how EwE shell must interpret and apply pressures originating from MEL, and how EwE shell will provide Ecospace ecological assessments (= model output) back to MEL in the form of output layers. This game configuration will be stored within each Ecopath Access model database that is used for a MSP game.

MEL link designer will have a simple user interface, and will handle easy-to-test information. Development time will be limited.

Test plan

This plan provides step-by-step testing instructions for the ecology core to ensure its proper functioning. The plan includes test data, descriptions of input conditions, and expected outputs.

EwE test model

A test model will be needed to validate the MEL – EwE linkage during the development phase. Ideally two models should be used, a simple and a more complex model. The simple model should respond directly and with obvious results to pressure perturbations, while the more complex model can be used to test the generation of complex output layers.

To limit development time, the already available Anchovy Bay model could be used and updated to serve as the simple test model. This spatially explicit model of a fictional ecosystem in a fictional location has been developed by Villy Christensen for EwE training purposes. The Anchovy Bay model is shipped for free with the EwE desktop software since version 6.5, and there are no limitations to its use.

For the complex test model the North Sea model would be a logical candidate. This model is currently updated at the Centre for Environment, Fisheries and Aquaculture Science (CEFAS) in Lowestoft (UK). The model may not be ready on time for the MEL – EwE developments; in that case another model may have to be used.

Quality assurance and testing

Testing is a continuous activity during the entire development cycle by the developers. However, to ensure an impartial and unbiased assessment, independent testing will be needed according to the test plan.

Documentation

Two types of documents will need to be written during the development phase:

- Design documentation will detail technical and conceptual decisions made during development, will discuss the layout of code components, and will perform the basis for long-term code maintenance. This documentation is intended for code developers.
- Usage documentation will provide point-and-click instructions for the various tools developed during this phase, and will provide installation instructions. This documentation is intended for MSP game technicians and for EwE modellers tasked to update a model for MSP game play.

Table 5 – Technical developments needed to implement the EwE - MEL linkage

#	Activity	Duration	Internal dependencies	External dependencies
1	Develop EwE shell	4 weeks	EwE 6 source code	MEL linkage requirements as outlined in this document
2	Develop MEL emulator	1 week	#1	MEL programming model (see Appendix A)
3	Develop MEL link designer	1 week	#1	
4	Develop test plan	1 week		
5	Develop EwE test model	1 week		
6	Quality Assurance & testing	3 weeks	#1-5	
7	Write documentation	2 weeks		

5.2. Scientific developments

All scientific efforts related to updating, fine-tuning and testing an Ecospace model for actual game play are listed in Table 6. Note that this is a very coarse assessment, for which no local EwE experts were consulted, and no specific MSP game play scenarios were specified.

Model restructuring

The time needed to adjust the structure of an existing EwE model for specific MSP game play, as described in 4.1, will largely depend on the complexity of the selected EwE model and its compatibility with MSP gameplay needs. Normally, restructuring an existing model is not a very difficult task, as arduous input data collection for the various model components has already been performed.

A EwE model should be updated by an expert EwE modeller with good working knowledge of Ecospace, and familiar with the ecological characteristics of the modelled area.

Model fine-tuning

The amount of time needed to fine-tune a EwE model for specific game play, as described in 4.2, is hard to estimate without any specific model and game play scenario in mind. The process may involve a certain amount of trial and error to dampen or amplify impacts of pressures to prevent unrealistic runaway events and ecosystem crashes.

Model fine-tuning should be performed under the guidance of a EwE modeller with good working knowledge of Ecospace, preferably familiar with the ecological characteristics of the modelled area.

Please note that it may be needed to further restructure a model during fine-tuning if a structural weakness is encountered that does not allow the model to behave as needed (Figure 6).

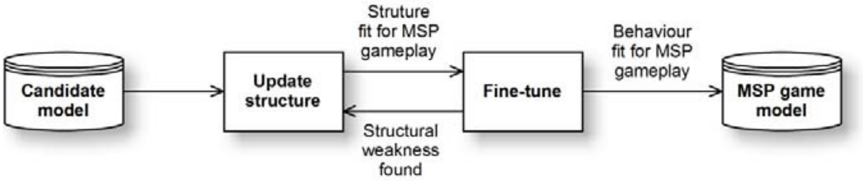
Candidate model
Update structure
Struture fit for MSP gameplay
Structural weakness found
Fine-tune
Behaviour fit for MSP gameplay
MSP game model

Figure 6 – The process of adjusting a EwE model for MSP game play. It may be necessary to revert back to model restructuring if structural weaknesses are found.

Test plan

This plan provides step-by-step testing instructions for the ecological model to ensure its proper functioning in the MSP complex. The plan includes test data, descriptions of input conditions, and expected outputs.

Testing

Even though the EwE model behaviour will be extensively assessed during restructuring and fine-tuning, to ensure an impartial and unbiased assessment, independent testing will be needed according to the test plan.

Documentation

This documentation is intended for support future EwE modellers to further update and fine-tune a EwE model for game play purposes. Structural and model parameter changes made to the original model, with their motivations, need to be described to provide insights in the workings of the final gameplay model.

Table 6 – Scientific developments needed to adjust a EwE model for MSP game play

#	Activity	Duration	Internal dependencies	External dependencies
1	Update model structure	2 weeks	MSP game play needs	Existing model structure
2	Fine-tune model for MSP	2 weeks	#1	
3	Develop test plan	1 week	EwE shell structure	Pressure definitions
4	Test game play	2 weeks	#1-4	
5	Write documentation	1 week		

6. Summary

The Maritime Spatial Planning (MSP) Challenge 2050 is a powerful serious game about the complexities of planning a socio-economic ecosystem and the ecology that it relies on. The MSP software is currently being redesigned and restructured to overcome limitations of its initial design. As a part of this restructuring, the computational model that represents the marine ecosystem in the MSP software will be replaced with a direct link to an existing food web model, Ecopath with Ecosim (EwE).

To realize this integration, technical and scientific developments will be needed.

First, MSP and EwE programs will need to be linked. This will require a number of new software developments. "EwE shell" will be a new tool that interfaces between the EwE computational core and the MSP software. Since EwE shell is a generic tool, it will need configuration instructions how to translate incoming data to EwE, and outgoing data back to MSP. These configuration instructions will be generated through "MEL link designer", a new plug-in for the EwE desktop software that will need developing. Last, in order to test and fine-tune the behaviour of a EwE model for MSP gameplay purposes, a required third tool "MEL emulator" will provide a test bench for simulating MSP data.

Second, when the technical pathways have been implemented, largely scientific efforts are needed to adjust an existing EwE models to become suitable for a specific MSP game. EwE models may need restructuring to represent the food web components needed for a specific MSP game play scenario; and EwE models may need fine-tuning to respond to environmental perturbations initiated by MSP players.

This document provides a conceptual overview of each development, provides recommendations where possible, identifies possible future developments, and provides time estimates for essential tasks to link the MSP serious game to the Ecopath with Ecosim food web modelling approach.

References

Bailey, H., Senior, B., Simmons, D., Rusin, J., Picken, G., and Thompson, P.M. (2010). Assessing underwater noise levels during pile-driving at an offshore windfarm and its potential effects on marine mammals. Mar. Pollut. Bull. *60*, 888–897.

Christensen, V., and Walters, C.J. (2004). Ecopath with Ecosim: methods, capabilities and limitations. Ecol. Model. *172*, 109–139.

Christensen, V., Coll, M., Steenbeek, J., Buszowski, J., Chagaris, D., and Walters, C.J. (2014). Representing Variable Habitat Quality in a Spatial Food Web Model. Ecosystems 1–16.

Colléter, M., Valls, A., Guitton, J., Morissette, L., Arreguín-Sánchez, F., Christensen, V., Gascuel, D., and Pauly, D. (2013). EcoBase: A Repository Solution to Gather and Communicate Information from EwE Models (Fisheries Centre Research Reports No. 21 (1)) University of British Columbia. Vanc. Can.

Colléter, M., Valls, A., Guitton, J., Gascuel, D., Pauly, D., and Christensen, V. (2015). Global overview of the applications of the Ecopath with Ecosim modeling approach using the EcoBase models repository. Ecol. Model. *302*, 42–53.

ECMA International (2012). Standard ECMA-335 - Common Language Infrastructure (CLI) 6th edition (June 2012).

Heymans, J.J., Coll, M., Link, J.S., Mackinson, S., Steenbeek, J., and Christensen, V. (2016). Best practice in developing, balancing, fitting and using Ecopath with Ecosim food-web models for ecosystem-based management. Ecol. Model. *331*, 173–184.

Jost, L. (2006). Entropy and diversity. Oikos *113*, 363–375.

Mackinson, S., and Daskalov, G. (2007). An ecosystem model of the North Sea to support an ecosystem approach to fisheries management: description and parameterisation. Cefas Sci. Ser. Tech. Rep. *142*, 195.

Steenbeek, J. (2015). MSP Challenge 2050 food web review (Barcelona: Ecopath International Initiative).

Steenbeek, J., Coll, M., Gurney, L., Mélin, F., Hoepffner, N., Buszowski, J., and Christensen, V. (2013). Bridging the gap between ecosystem modelling tools using geographic information systems: driving a food-web model with spatial-temporal primary production data. Ecol. Model. *263*, 139–151.

Steenbeek, J., Buszowski, J., Christensen, V., Akoglu, E., Aydin, K., Ellis, N., Felinto, D., Guitton, J., Lucey, S., Kearney, K., et al. (2016). Ecopath with Ecosim as a model-building toolbox: Source code capabilities, extensions, and variations. Ecol. Model.

Stolte, W., Keijser, X., de Kluijver, A., and Nolte, A.J. (2013). A conceptual pressure-response and a simple foodweb model for the Maritime Spatial Planning Challenge 2050 serious game.

Thomsen, F., Lüdemann, K., Kafemann, R., and Piper, W. (2006). Effects of offshore wind farm noise on marine mammals and fish. Biola Hambg. Ger. Behalf COWRIE Ltd *62*.

Walters, C.J., Pauly, D., and Christensen, V. (1999). Ecospace: Prediction of Mesoscale Spatial Patterns in Trophic Relationships of Exploited Ecosystems, with Emphasis on the Impacts of Marine Protected Areas. Ecosystems *2*, 539–554.

Appendix A. MEL – EwE application programming interface (API)

MEL and EwE will be tightly integrated in code. MEL will be written in Python, and EwE is written in .NET languages. Both Python and .NET are languages of the Common Language Runtime (CLR) family, which allow for direct code integration via IronPython, Python.NET, or a similar communication library. A thin wrapper application, henceforth called 'EwE shell', will be used to: (i) control the EwE life span; (ii) translate MEL API calls to EwE model parameter tweaks; and (iii) gather EwE output that needs to be returned to MEL (Figure 7).

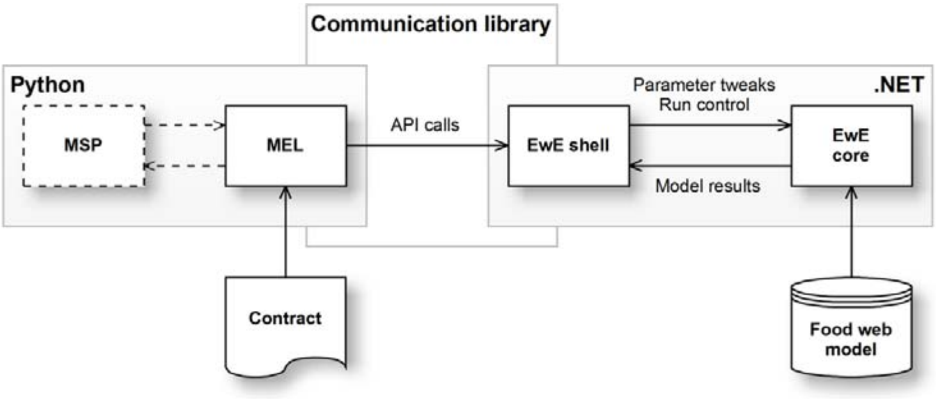

Figure 7 – Conceptual view of the linkage between MEL (python) and EwE (.NET) via an intermediate communication library (most likely IronPython). The interaction between MEL commands and the EwE computational core - which contains all EwE modelling logic - is managed by a thin shell that controls EwE parameter values and EwE computational time stepping, and gathers EwE model predictions to return to MEL.

A simple Application Programming Interface or API has been conceptualized to allow MEL to control the execution of the food web model during a MSP game session. The API commands to start; run; and stop the food web model are listed below.

API call: Configuration

Load a EwE model, validate whether MSP and EwE model use the same domain, and provide MEL with the Ecospace depth map and a definition of output layers.

Note that the definition of output layers is part of the MSP game setup. The definitions returned by this API call merely provide MEL with the means to validate whether the selected EwE model is compatible with the MSP game; the MEL-EwE API does not allow for specification, addition, and removal of ecological output layers once a MSP game session has started.

Syntax

```
void Configuration(model, mode, timestep,
       lon, lat, size, ncols, nrows,
       pressurelayers, depth, outcomelayers)
```

Parameters

Parameter	In/out	Type	Description
model	In	string	EwE model file name to load
mode	In	string	The mode parameter will direct the EwE shell which outcome layers to produce
timestep	In	integer	The number of time steps per year
longitude	In	float	Longitude of the corner of the top-left cell of the grid area, expressed in decimal degrees (WGS84 projection)
lattitude	In	float	Longitude of the corner of the top-left cell of the grid area, expressed in decimal degrees (WGS84 projection)
size	In	float	Cell size of the grid, in decimal degrees
ncolumns	In	integer	Number of columns in the grid
nrows	In	integer	Number of rows in the grid
pressurelayers	In	array	Array of [Name, float], providing the name and datatype of each pressure layer. See Datatype enumerator below for possible types.
depth	Out	array	Array of float[,] containing the Ecospace depth grid
outcomelayers	Out	array	Array of String[], providing the name of each outcome layer. Output layers always contain floating point values.

Internal processing

The EwE shell will attempt to load the EwE model, and will validate whether the EwE model configuration matches the game contract. Next, the EwE shell will load the required Ecosim and Ecospace scenarios, and will validate whether the resulting EwE model configuration matches the values provided by the other input parameters.

If any validation fails an appropriate exception is thrown (see Exceptions, below)

API call: Startup

Run the first relevant Ecospace time step, and return the content of outcome layers to start the game.

Syntax

```
void Startup(outcomelayers)
```

Parameters

Parameter	In/out	Type	Description
outcomelayers	Out	array	Array of [Name, float] name and values of outcome layers

Internal processing

This call will execute the first relevant Ecospace time step. Transparently to MEL, Ecospace may be configured as part of the MSP game setup to run a fixed number of time steps as a model *spin-up* or *burn-in* period to stabilize food web dynamics and distributions.

API call: Run

Perform a single Ecospace time step, integrating values in the provided pressure layers, and returning ecological estimates.

Syntax

```
void Run(pressurelayers, outcomelayers)
```

Parameters

Parameter	In/out	Type	Description
pressurelayers	In	array	Array of [Name, float] name and content of pressure layers
outcomelayers	Out	array	name and values of outcome layers

Internal processing

EwE will calculate spatial food web dynamics, increasing its internal time clock by one month. When the end of the Ecospace run time is encountered, the Ecospace clock will be set back by one month to avoid running out of run time memory. This will have no effect on results.

API call: Terminate

Close the EwE session.

Syntax

```
void Terminate()
```

Internal processing

This will terminate the Ecospace run, close Ecospace and Ecosim scenarios, unload the EwE model, and free up memory resources used by the EwE computational core.

Datatype enumerator

Enumerated type defining pressure layer datatypes, with the following values:

- GridLog
- GridLinear
- ScalarInt
- ScalarFloat
- ScalarBool

Exceptions

The following exceptions will be thrown by the EwE shell if an MEL API call ran into trouble:

EwEConfigurationException

The EwE shell detected that input parameters do not match expected values or definitions. Possible exception details are

- Out of bounds (parameter name + value)
- Mismatching Configuration
- Unknown mode
- File not found
- File invalid

EwE run exception

The EwE shell encountered an issue when running a model time step. Possible exception details are:

- Invalid type
- Model error

EwE terminate exception

The EwE shell encountered a problem when shutting down. Possible exception details are:

- Clean up error