Ecological Modelling 319 (2016) 178–189

Contents lists available at ScienceDirect

Ecological Modelling

journal homepage: www.elsevier.com/locate/ecolmodel

Ecopath with Ecosim as a model-building toolbox: Source code capabilities, extensions, and variations

Jeroen Steenbeek[a,*], Joe Buszowski[a], Villy Christensen[b], Ekin Akoglu[c], Kerim Aydin[d], Nick Ellis[e], Dalai Felinto[b], Jerome Guitton[f], Sean Lucey[g], Kelly Kearney[h], Steven Mackinson[i], Mike Pan[b], Mark Platts[i], Carl Walters[b]

[a] Ecopath International Initiative Research Association, Mestre Nicolau 8, Bellaterra, Barcelona 08193, Spain
[b] Fisheries Centre, University of British Columbia, 2202 Main Mall, Vancouver, BC, Canada V6 T 1Z4
[c] Istituto nazionale di Oceanografia e di Geofisica Sperimentale, Trieste, Italy
[d] Alaska Fisheries Science Center, 7600 Sand Point Way N.E., Building 4, Seattle, WA 98115, USA
[e] Commonwealth Scientific and Industrial Research Organisation, Ecosciences Precinct, 41 Boggo Rd, Dutton Park, Brisbane, QLD 4102, Australia
[f] Université Européenne de Bretagne, Agrocampus Ouest, UMR985 Ecologie et santé des écosystèmes, Rennes Cedex, France
[g] NOAA Fisheries, Northeast Fisheries Science Center, 166 Water Street, Woods Hole, MA 02536, USA
[h] University of Washington, Joint Institute for the Study of the Atmosphere and Oceans (JISAO), Seattle, WA, USA
[i] Centre for Fisheries, Environment and Aquaculture Science, Pakefield Road, Lowestoft, Suffolk NR33 0HT, UK

ARTICLE INFO

Article history:
Available online 9 July 2015

Keywords:
Food web model
Ecopath with Ecosim
Ecosystem
Fisheries
Open source software
Model interoperability

ABSTRACT

Ecopath with Ecosim (EwE) is a widely applied food web model that is mostly known as desktop software for the Microsoft Windows platform. The freely available Microsoft .NET source code of EwE, however, provides a range of possibilities to use the model in different ways, to customize and extend the model, and to execute the model on different operating systems.

We provide an overview of the EwE source code, its philosophy, and its technical capabilities, targeted to model builders and advanced users of the EwE software. We showcase novel scientific applications of the EwE model that have been facilitated because of the modular approach of the system. We also present three additional versions of EwE that have been written in different computer languages for dedicated purposes. Lastly, we provide an entry point for users for obtaining and using the source code for custom purposes.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Ecopath with Ecosim (EwE) is a mainly aquatic food web model, the start of which was conceived more than 30 years ago (Polovina, 1984). The EwE modelling approach consists of three main, linked routines: (i) Ecopath, a static model that represents the energy mass-balance of a trophic web (Christensen and Pauly, 1993, 1992; Polovina, 1984); (ii) Ecosim, a time-dynamic model that obtains some of its initial parameters and assumptions from the Ecopath model to assess temporal dynamics of an ecosystem (Walters, 2000; Walters et al., 1997); and (iii) Ecospace, a spatial–temporal explicit model that applies the Ecosim equations over a grid of spatial cells where cells are connected through biomass exchange using diffusion and migration movement relationships (Walters et al., 1999). Made freely available as desktop software in 1992, two decades of continued development accompanied by a strong commitment to user support and training have turned the EwE approach into the most widely applied model of its kind (Christensen and Pauly, 1992; Christensen and Walters, 2011, 2004; Coll et al., 2008; Walters et al., 1997). EwE is predominantly used for ecological studies and Ecosystem Based Management, and in a recent case, to support an Environmental Impact Assessment (Christensen and Maclean, 2011; Canadian Environmental Assessment Agency, 2015; Link, 2010).

The EwE approach can be used, extended, and applied in an endless number of ways with even only rudimentary software programming experience. In this paper, we introduce a variety of applications that the EwE source code has facilitated, present an overview of novel applications of the model, introduce how the EwE source code can be used to customize applications, and present implementations of EwE in other programming languages, written for dedicated purposes.

* Corresponding author. Tel.: +34 685 319 597.
E-mail address: jeroen@ecopathinternational.org (J. Steenbeek).

http://dx.doi.org/10.1016/j.ecolmodel.2015.06.031
0304-3800/© 2015 Elsevier B.V. All rights reserved.

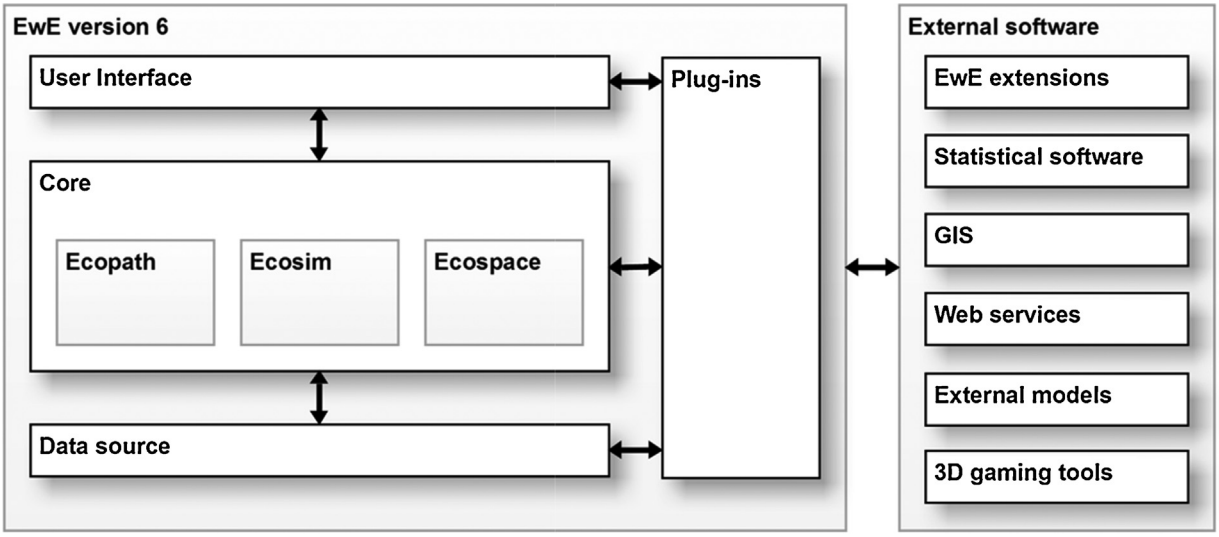


Fig. 1. The modular structure of the Ecopath with Ecosim source code, separating data access, computations, and user interface into exchangeable components. A plug-in system provides the means to extend the EwE model, and to connect to other programs external to the EwE software.

2. Source code organization

As outlined in Christensen and Lai (2007), the EwE modelling software underwent a structural overhaul and partial rewrite from 2005 to 2007. The Ecopath source code was rewritten using the Microsoft.NET framework for a few important reasons.

First, Visual Basic .NET, one of the programming languages under .NET, is structurally compatible with deprecated Microsoft Visual Basic 6, the former programming language of Ecopath. Visual Basic 6 was primarily selected because of its ease of programming, and its ability to deliver user-friendly interfaces. Largely retaining code structures in .NET made it possible to update the programming environment without having to do a complete overhaul of the computational core, while the .NET offered a similar but broader toolset to retain the familiar user interface.

Second, the .NET framework is a long-term technological investment by Microsoft, and has been promoted as the main platform for building applications for Windows since late 2000 (Microsoft Support, 2014). Because of this long-term commitment, there is no foreseen need to rewrite the EwE code again due to discontinued development tools. The .NET framework has attained a large user base worldwide, supported by a large volume of technical resources on the Internet, which aids programming efforts and may make EwE source code attractive for third-party developers.

Third, the .NET framework offers an extensive library of software building blocks. Any program written in .NET, in turn, can provide its own building blocks for reuse in other programs. A version of EwE built in .NET would thus become a programmers' toolbox for model extension and integration.

Fourth, Visual Basic .NET and other .NET framework programming languages are based on the Common Language Infrastructure (CLI), an open standard that defines a structure for high-level programming languages to function on different computer systems without having to adhere to any particular computer hardware architectures (ECMA International, 2012). This means that all .NET programs – if properly adhering to the CLI specifications – can be executed on any computer system for which a CLI runtime environment (such as Mono, http://www.mono-project.com/, and Portable.NET, http://www.gnu.org/projects/dotgnu/pnet.html) is available. CLI-compliance would open the possibility to run EwE on other operating systems, for instance on non-Windows parallel computing clusters. Recently, Microsoft released part of the .NET runtime environment code as open source, and pledged close collaboration with Mono to provide better support for cross-platform deployment of .NET software (Microsoft News Center, 2014). This holds potential for wider applications of EwE and other .NET tools.

The last major benefit of the CLI architecture of .NET is its technical compatibility with over 20 other CLI-based programming languages within and outside the .NET framework, such as C# (C sharp), F# (F sharp), IronPython, IronRuby, C++/CLI, Silverfrost FTN95, Visual Basic.NET, J-script.NET, and Eiffel. The CLI foundation provides high-level interconnectivity between code written in many languages, both for scientific and non-scientific programming purposes.

Because of the factors outlined above, the Ecopath source code was ported to Visual Basic .NET. The original Ecopath, Ecosim and Ecospace data structures and algorithms were preserved, but were wrapped in an object-oriented structure of classes that was designed with reusability and extensibility in mind. The EwE source code was conceptually reorganized, physically separating computational logic, database access, and user interfaces into separate, replaceable, and extendable modules (Fig. 1). The user interface was redesigned. It retains familiar elements but provides a more intuitive workflow. Further, a plug-in system was added to allow third-party programmers to write code modules for EwE without having to change the original EwE source code (Christensen and Lai, 2007). EwE recognizes and integrates plug-in modules into its execution flow and its desktop user interface, allowing for seamless extension of the software (Fig. 2).

The reorganization of the EwE source code and implementation in .NET has opened novel ways to run, integrate, and extend the EwE model.

2.1. Running the EwE model

EwE is most widely known as desktop software available for the Windows platform. This desktop software is written in .NET, and is composed of logical building blocks that can be freely re-used and incorporated in other programs. As such, with some programming experience, it is quite feasible to build custom programs that use EwE code components for their own purposes.

For instance, one can construct small programs to automate EwE-related tasks that would be tedious to perform in the desktop software, such as repeatedly tweaking model input

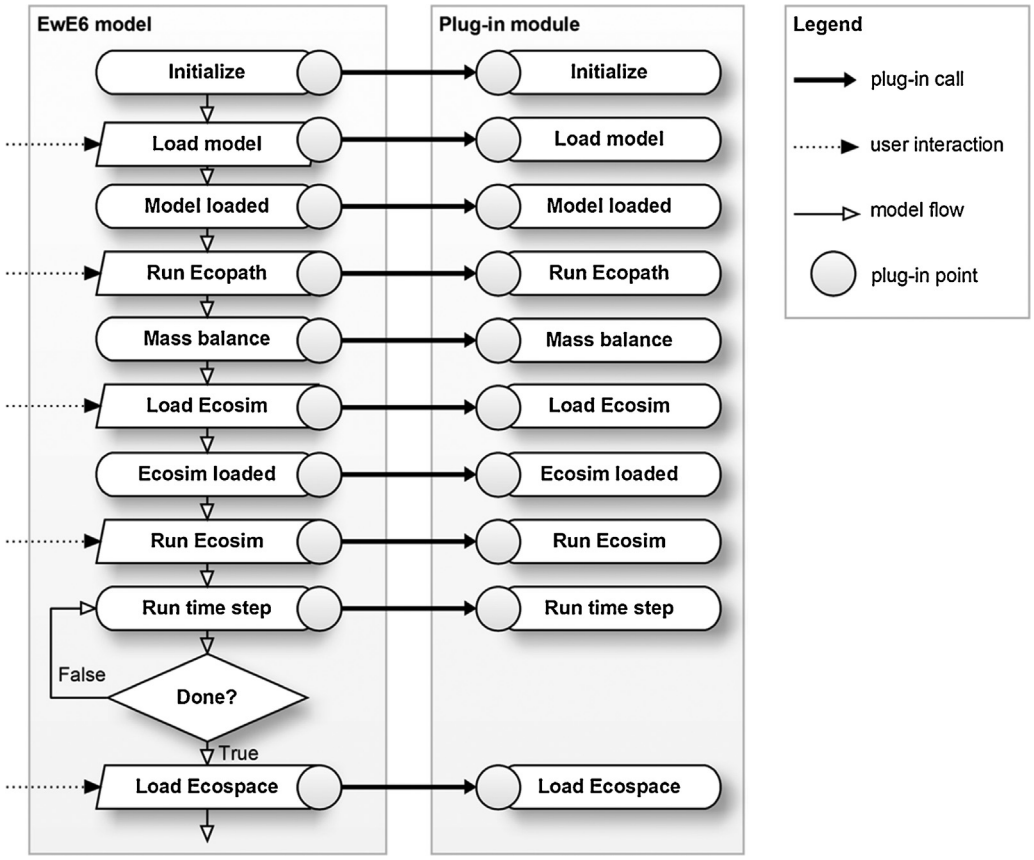


Fig. 2. An example how plug-in modules are seamlessly included in the execution flow of the EwE model.

parameters and gathering model results, or performing sensitivity analyses.

Using a similar approach, programmers can decide to present the EwE computational models using alternative user interfaces for specific purposes, catering to specific audiences. Programmers can deploy their EwE-based programs on any operating system for which a CLI runtime environment is available. However, it is important though to note two limitations of the EwE software regarding CLI compliance:

- Although the EwE computational modules are entirely CLI-compliant, the EwE desktop software utilizes user interface components that are tied to Windows, and are not CLI-compliant. This means that the EwE computational models can be executed on other operating systems than Windows, but that the EwE desktop software is, at the time of writing, limited to execution on the Windows operating system only.
- For historical reasons, the EwE software relies on the Microsoft Access database system for reading and writing its parameterization files. Connectivity to this database system is not supported in the CLI standards, and this EwE functionality is thus limited to the Windows operating system only. However, the EwE source code contains provisions to read model parameterization data from other types of file formats that can be used on any operating system.

2.2. Integrating the EwE model

Any programming language based on the CLI standards can incorporate the building blocks offered by the EwE source code, directly accessing the model functionality.

If programmers are not able to use a CLI compliant programming language – and can thus not directly integrate the EwE building blocks onto their own code – then the .NET framework offers numerous communication protocols that can be used to establish bi-directional contact between a user's code and the EwE computations.

2.3. Extending the EwE model

Historically, extending the EwE model with custom logic by third parties required making modifications to the EwE source code, which needed to be carefully reintegrated whenever the EwE source code was updated to a newer version by EwE core developers. This was a rather inefficient system needing a different approach in the redesigned version of EwE. In response, a plug-in system was added to the EwE source code to allow third-party developers to build additions to the EwE software independently of the EwE source code. Plug-in modules for EwE are .NET application modules that are automatically detected when EwE code is launched, and that are integrated into the EwE model execution flow (Fig. 2).

This plug-in system is conceptually very simple. The EwE source code contains a long list of plug-in points that are made available at key locations in the EwE source code. Typical plug-in points capture (i) model events such as loading and unloading of data, or start and end of computation cycles; or (ii) user interface events such as population of navigational elements or population of particular user interfaces.

Programmers can build plug-in modules that connect to any number of available plug-in points, with full access to model data, from .NET and EwE building blocks. When the EwE model launches, it searches and indexes all plug-in modules in the EwE program

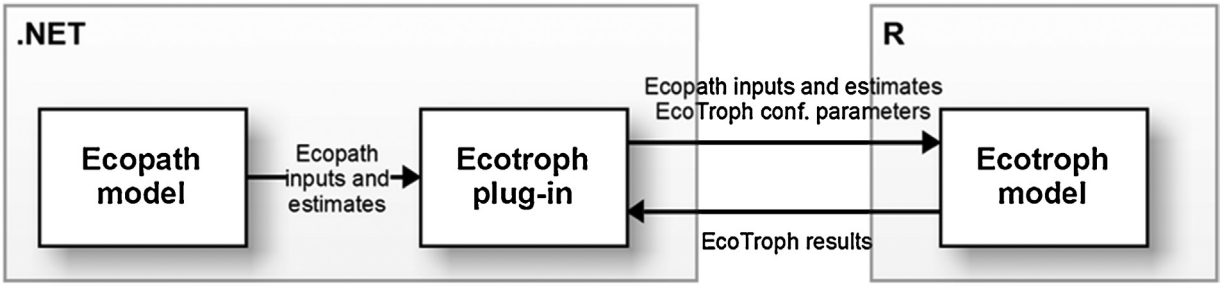


Fig. 3. Schematic overview how the EcoTroph R model integrates into the EwE .NET code. An EcoTroph plug-in to EwE collects Ecopath model inputs and estimages, and collects EcoTroph configuration parameters. The EcoTroph is then executed in R. Results produced by the R model (text and PDF images) are collected by the EcoTroph EwE plug-in, and are displayed in its user interface.

directory. When the EwE software encounters a particular plug-in point during its execution time, all previously found plug-in modules are checked to see if they provide a connection to that particular plug-in point, and if so, the external logic in the plug-in module is invoked.

With this structure, plug-in modules can be built, maintained, and distributed independently of the EwE source code. This provides great flexibility and independence to third-party developers for extending the functionality of the EwE model for their research needs.

3. New scientific uses of the EwE model

The technical reorganization and re-implementation of the EwE source code in the Microsoft .NET environment has led to a wide range of novel ecological modelling research, of which we wish to present just a few below.

3.1. EcoTroph in R

EcoTroph (ET) is a modelling approach articulated around the idea that an ecosystem can be represented by its biomass distribution across trophic levels, i.e., the biomass trophic spectrum (Gascuel, 2005). The ecosystem functioning is viewed as a biomass flow surging up the food web from lower to higher trophic levels, due to predation and ontogenetic processes (Gascuel et al., 2011; Gascuel and Pauly, 2009). Thus, the ecosystem biomass present at a given trophic level can be estimated from two simple equations, one describing biomass flow, the other their kinetics (which quantifies the velocity of biomass transfers). The EcoTroph model is commonly used as a complementary approach to the Ecopath model (Gasche et al., 2012).

The restructured EwE .NET source code enabled a direct linkage between the EcoTroph and Ecopath models: EcoTroph is implemented as an R package (Colleter et al., 2013) that is linked to the EwE desktop software via an EwE plug-in (Fig. 3). The plug-in acts as an intermediary between the two models by providing bi-directional data exchange. The plug-in gathers EwE model data and EcoTroph configuration parameters as defined by the user, and runs the EcoTroph model through R functions. Results of the R model are captured and displayed within the plug-in as text (R-produced output data) or as figures (R-produced PDFs).

This separation of logic allows the R package to be developed and used independently of the EwE source code and the EwE plug-in.

The EcoTroph plug-in source code is provided with the EwE desktop source code. The EcoTroph R package can be obtained from the R repository (http://cran.r-project.org/web/packages/EcoTroph).

3.2. Northern Prawn Fishery integrated model

Dichmont et al. (2013) developed an integrated model to investigate the effects of spatial closures on conflicting objectives addressing biodiversity, benthic impacts, ecosystem function, key species at risk to fishing, economics and sustainability. The approach drew on expertise from a range of models. The models comprised a suite of fishery management models and fleet and stock operating models defined on a regional scale (Fig. 4, top row), coupled to fine-scale models in which the details of the spatial closures were implemented and fine-scale impacts were calculated (Fig. 4, lower rows). The models were embedded within an overarching framework, thus allowing simulations to be run as a single integrated model. Similar versions of this framework have also been applied in management of the effects of trawling (Ellis et al., 2008) and of coastal water quality (Dutra et al., 2010).

Ecospace was a crucial part of this integrated model, because of quantified ecosystem impacts at the fine spatial scale of marine organisms that was the focus the project. Therefore, an integration was required in which Ecospace used the effort distribution supplied by a fleet dynamics sub-model, and an ecological risk assessment model, in turn, used biomass distributions as predicted by Ecospace (Fig. 4).

In typical usage, EwE would be run as a stand-alone program, and other models would supply their outputs to it by means of static files. However, this would be prohibitive for integration within the framework where a bi-directional linkage between modelled components was required. Using the source code of EwE, the authors programmed a client–server architecture in which the framework (the client) communicated with an instance of a running EwE model (the server). At the beginning of an execution time step of the framework, effort distributions in Ecospace were overwritten by desired effort distributions from the fleet dynamics sub-model. Ecospace then used these distributions for its computations. The resulting computed spatial distributions of marine organism biomasses were then sent back to the framework for consideration by other models.

To ensure that both the framework and EwE were working on the same time step, a software synchronization technique commonly referred to as 'wait and pulse' was used (e.g., Drayton et al., 2003): at each time step, the framework waits for EwE to continue one time step, after which EwE waits for the framework to catch up, effectively taking turns.

3.3. Uncertainty in management strategy evaluations

Arguably, one of the strongest criticisms of Ecosystem models in general, with Ecopath with Ecosim being no exception,

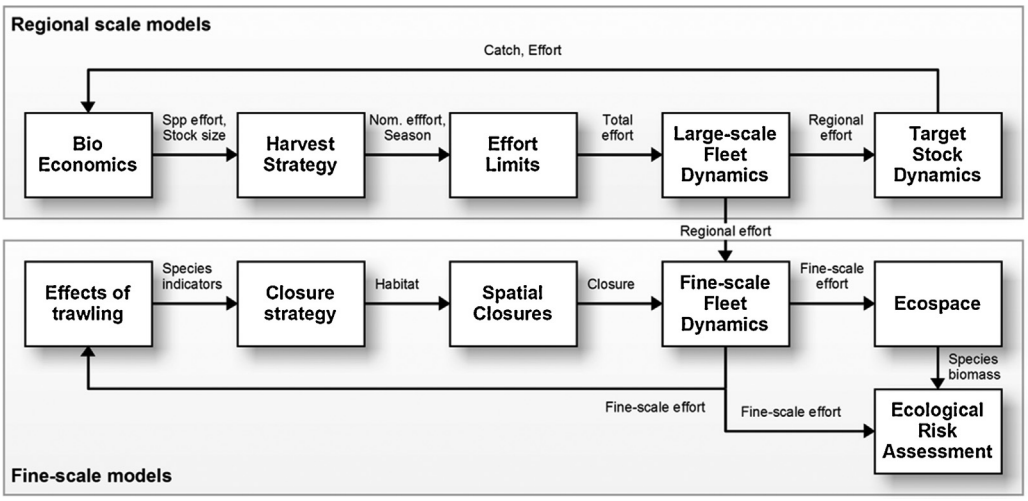


Fig. 4. Schematic diagram of the Northern Prawn Fishery integrated model. Each box represents a separate sub-model, with data links between the models indicated by labelled arrows. Ecospace, using the fine-scale effort provided by the fleet dynamics sub-model, provides spatially resolved biomasses to the ERA sub-model (adapted from Dichmont et al., 2013).

is the considerable uncertainty inherent to model parametrisation, and consequently, propagated to model results (e.g., Plaganyí and Butterworth, 2004). This uncertainty can have a considerable impact on decision-making processes when choosing between several alternative management strategies (Butterworth et al., 1997; Walters et al., 2008; Walters and Green, 1997). Since eliminating model parameterization uncertainty is not always feasible, the next best possible course of action is to expose it—enabling decisions to be made that are robust to these uncertainties.

By allowing the user to specify observation, implementation, and stock assessment error, the management strategy evaluation (MSE) facility that in incorporated with EwE takes a vital step towards handling uncertainties in the fishery system. Using the facilities provided by the restructured EwE source code, an alternate MSE module is being developed at the Centre for Environment, Fisheries and Aquaculture Science (CEFAS) that enables the specification of uncertainty in additional biological parameters governing the dynamics of the ecosystem. Greater flexibility is also provided for specification of harvest control rules, allowing for the limits of the levels of fishing to be contingent on the state of another (conservation) species.

Using these input improvements, the MSE plug-in uses the EwE Monte Carlo routines (Christensen and Walters, 2004) to sample and collect a range of alternate feasible Ecopath parameter configurations. These parameter sets are written to disk, and can be loaded and explored in the Ecopath model. Then, the MSE can be used to run a random subset of alternate model parameterizations through Ecosim whilst controlling Ecosim effort based on quotas derived from the harvest control rules. Run results are collected and can be further analysed using dedicated statistical software (Fig. 5). The MSE plug-in is currently in a testing phase, is being tested at training courses, and is scheduled for distribution during 2015.

3.4. Spatial temporal data framework

A long-standing shortcoming of the Ecospace model was its limited facility to consider varying environmental conditions (Steenbeek et al., 2013). With an increasing demand for consideration of environmental variability into spatial–temporal food-web dynamics, a spatial temporal data framework was constructed.

Built as a series of EwE plug-in components, this framework is a flexible, extensible set of geospatial data processing tools that provide Ecospace with the ability to interact with time series of maps external to the EwE model. Time series of maps can be read from an open-ended number of sources (files, web services, other spatially explicit models, etc.) to vary the content of driver maps in the Ecospace model. Ecospace, in turn, can provide its time-varying maps to any number of destinations in a range of geospatial data formats (Steenbeek, 2012; Steenbeek et al., 2013).

The framework holds potential for future applications of Ecospace. Including varying conditions in the habitat capacity foraging model (Christensen et al., 2014) combines food web and

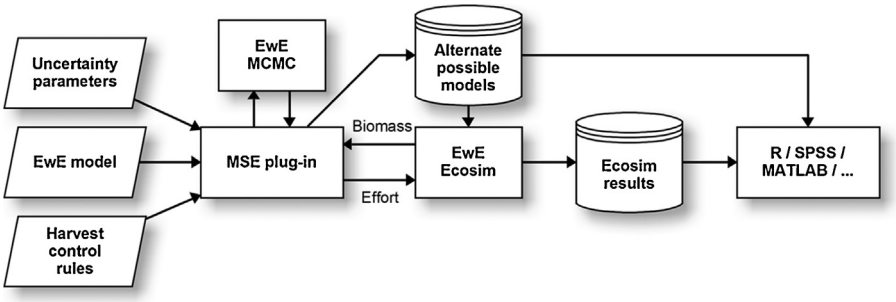


Fig. 5. Schematic overview of the MSE plug-in workflow. Using the Monte Carlo (MCMC) routine of EwE, a model, and uncertainty parameters, the plug-in samples a series of alternative balanced EwE models. Then Ecosim is executed upon these alternate models, where the MSE plug-in uses biomass estimates and harvest control rules to determine effort in Ecosim. Run results are gathered for further analysis in statistical software.

species distribution modelling within a single tool, allowing for integration of functional responses of organisms to varying environmental conditions within the food web dynamics.

Additionally, the framework is designed to simplify technical access to geo-spatial data sources, thus reducing the technical complexity of bi-directional data exchange between Ecospace and other spatially explicit models.

3.5. Global ocean model

Christensen et al. (2015) performed a study that aims to improve understanding how the combined effects of climate change and fisheries impact life in the global oceans, and how this impacts derived ecosystem services. A spatially explicit EwE model was constructed encompassing components from primary producers to top predators, and fishing, at the global ocean scale. The model was fitted to historical fisheries data, and was used for forecasting using various environmental driver datasets from dedicated models.

For this, the study draws heavily on recent scientific additions to the EwE software such as the habitat foraging capacity model (Christensen et al., 2014) to determine species niches in response to environmental conditions, and the spatial temporal data framework (Steenbeek et al., 2013) to enable time-varying environmental conditions as delivered by, e.g., the atmosphere-ocean circulation model COBALT (Stock et al., 2014). However, complex spatially explicit EwE models of this magnitude – with high numbers of modelled components each responding to a number of environmental drivers spread across high numbers of spatial cells – can be troublesome to execute on a regular Windows PC. Execution times can become relatively long (up to four minutes per Ecospace time step for this particular model). Moreover, Windows memory fragmentation may cause EwE to run out of available working space at unforeseen moments because models of this magnitude require large consecutive blocks of available memory to perform their computations.

The platform-independence of the EwE source code provided a solution for these potential performance problems. With minimal programming effort a console application – a program without graphical user interface – was built that integrated the EwE computational core. The standard EwE Windows-bound Access database was replaced by a generic text-based database. The Windows-bound graphical user interface was entirely omitted, and was replaced by run instructions provided in a text file. The program produced output maps and an execution log file to track progress and to aid troubleshooting. The resulting software, model database and environmental driver files where then placed on a Linux 64-core parallel computing cluster (Fig. 6). The multiple cores enabled Ecospace to perform its spatial computations in parallel for each group and fleet, while the hardware itself provided ample working memory for the spatial temporal model. The model was executed using the Mono runtime environment, which was used as a replacement for the .NET runtime framework on Linux (Christensen et al., 2015).

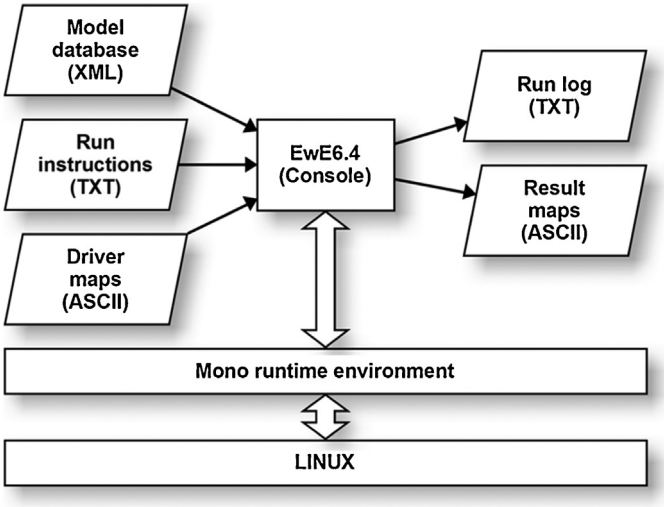


Fig. 6. Schematic overview of how EwE runs in a console application on the Linux operating system. All Windows-dependent components of EwE have been either replaced (using a text-based model database instead of the regular Microsoft Access database, and using a text file with run instructions instead of an interactive user interface), or have been omitted. The Mono runtime environment executes the EwE console application on Linux. Using the model database, run instructions and COBALT-derived driver maps, EwE produces a run log and result maps with spatial distributions of various quantities such as biomass, catch, and fishing effort.

3.6. OceanViz

In 2007, with the EwE code restructuring mostly completed, the EwE core development team embarked on an exercise to connect the Ecopath and Ecosim models to a 3D game engine as a visually appealing aid for laypersons to decision making.

The first product to emerge was OceanViz, an interactive underwater visualization for displaying the amount of marine life in the sea, driven by live food web models (Fig. 7) (Christensen and Lai, 2011). The OceanViz visualizations were built in the open-source 3D graphics and animation software Blender (http://www.blender.org) with its built-in Game Engine (Felinto and Pan, 2013). A plug-in for Ecopath with Ecosim seamlessly embeds the OceanViz visualizations within the EwE desktop software. Bi-directional communication between the plug-in and Blender is established via commonplace internet communication technology (Internet Sockets, see Makofske et al., 2004). The container plug-in again acts as intermediary between EwE and the visualizations: biomass estimates from Ecopath and Ecosim are intercepted by the plug-in, and are relayed to the Blender visualizations for display. In turn, when elements in the 3D visualizations are clicked by the user, this selection is relayed back to the EwE plug-in where additional information can be displayed (Fig. 7).

A second product that is currently under development is OceanSummits, a client–server system that uses the OceanViz capabilities to allow a group of stakeholders to interactively explore policy management alternatives using an ecosystem model (Christensen and Lai, 2011). Key to this effort is that a central, server-side version of EwE distributes its model results to a group of client computers. On these client machines, the regular graphs and charts of EwE are replaced by OceanViz environment visualizations, accompanied by traffic light plots and a minimal set of graphs. Summit participants can change management options in a collaborative setting, which are then taken into account by the ecosystem model, and reflected in the summit room via OceanViz (Fig. 8).

Through the open architecture of the EwE source code it was feasible to replace the familiar EwE desktop user interface with a 3D game environment, and reroute interaction with the model via a client–server architecture to the computers of multiple, simultaneously active participants of policy exploration sessions.

3.7. Plug-ins

As illustrated in this section, the EwE plug-in system allows for seamlessly extending the EwE model and user interface, and a range of scientific and technical plug-ins has been released or is currently under development (Table 1). Their functionality can be divided into four categories: (i) analysis plug-ins extract model results to

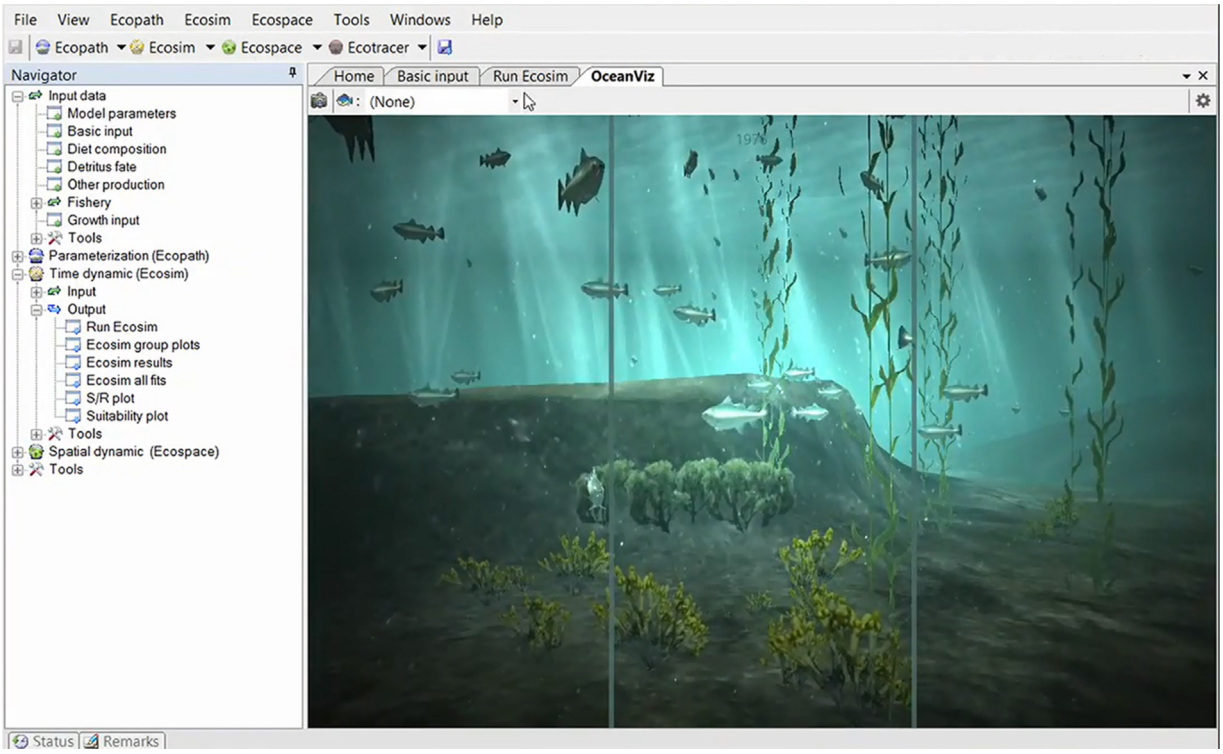


Fig. 7. The OceanViz visualizations hosted as a plug-in within the EwE desktop software. Here time is divided into three panels, each panel reflecting a different year of biomasses as predicted by Ecosim, the temporal module of EwE. A group called "90+ snook" in this prototype is selected, and is highlighted in the 3D visualizations in the same colour that EwE will use in its charts for this same group.

perform additional modelling or meta-analysis. This analysis can be performed with the .NET code base or outside as previously illustrated; (ii) automation plug-ins perform repeated tasks can be manually performed in the desktop software but that are lengthy, repetitive, and likely error-prone; (iii) engine plug-ins affect the behaviour of EwE. They can override or extend EwE computations, add new functionality, or add means to exchange data to and from EwE; and (iv) user interface plug-ins add visual elements to the front-end of the desktop software to enhance its usability.

Due to lack of funding, there is no central repository with third-party plug-ins for the EwE software. For now, plug-ins are either

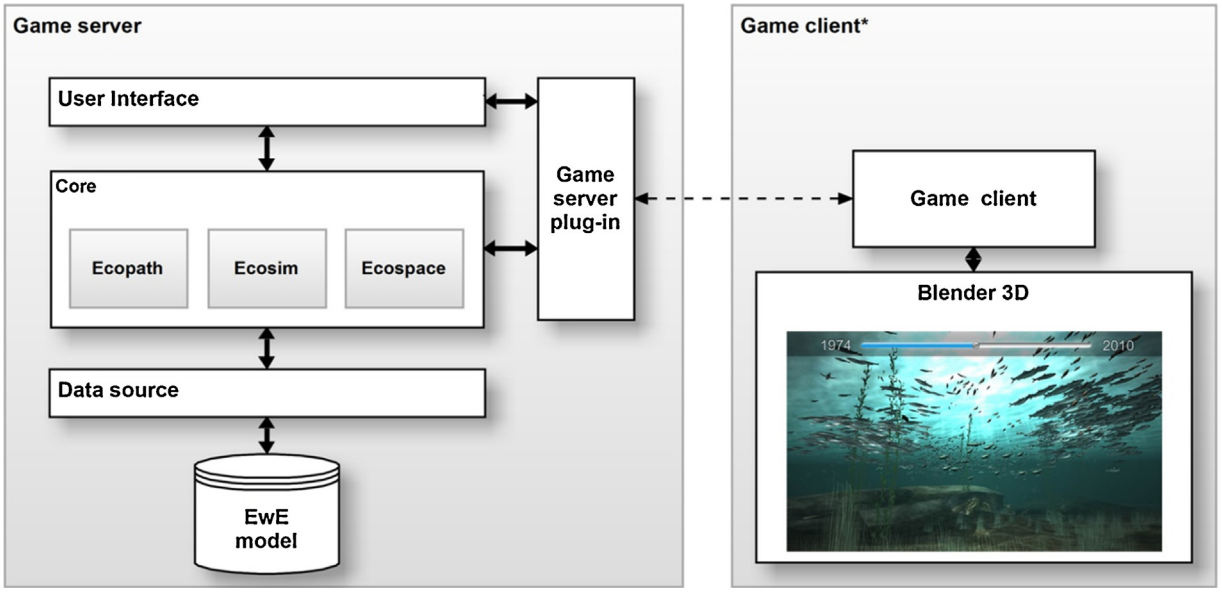


Fig. 8. Conceptual client–server architecture of the Ocean Summits system. The EwE food web model executes on a central server and hosts a model that represents the ecosystem for a summit. A game server plug-in provides a local game moderator interface, and provides connectivity between the running EwE model and game client applications that execute on individual client PCs. At each client PC, a participant of the summit can view model results. Participants can also make adjustments to a limited set of model input parameters that are then relayed back to the server, where they are processed by EwE. Resulting model outputs are distributed to the client PCs.

Table 1
An overview EwE plug-ins (* indicates approaches under development), their developer, category, EwE components they integrate with, and a brief summary. Plug-ins are listed by four categories: (i) analysis plug-ins meta-analyse EwE results; (ii) automation plug-ins performed lengthy tasks; (iii) engine plug-ins change EwE behaviour and functionality; and (iv) user interface plug-ins enhance the EwE usability.

Plug-in	Developer	Category	Ecopath	Ecosim	Ecospace	Other	Summary
Biodiversity indicators*	IRD, ICM	Analysis	√	√	√	√	Taxonomy-based analysis for Ecopath, Ecosim, Ecospace and Monte Carlo
Database-derived base maps	EII	Automation			√		Generate base map layers for an Ecospace scenario from geospatial data sources
Dynamic habitat modification	UBC	Automation			√		Adds time-dynamic variations in bathymetry in Ecospace
Ecopath model from Ecosim	ICM, CSIRO, DFO	Automation	√				Generate an Ecopath model from an Ecosim time step
Ecopath flow diagram	UBC	User interface	√				Flow diagram for Ecopath
Ecosim flow diagram*	NOAA	User interface		√			Flow diagram for Ecosim
EcoTroph	AO	Analysis	√	√			EcoTroph analysis in R, linked to the desktop user interface
FishBase taxon search*	EII	Engine	√				FishBase taxonomy search engine
Management Strategy Evaluation	CEFAS	Analysis		√			Parameter uncertainty testing for management advice
Multi-Sim	DFO, EII	Automation		√			Automatically perturb forcing functions and collect Ecosim results
Network analysis-R	SAMS	Analysis	√				Network Analysis, coded in R
Network analysis	SAMS	Analysis	√	√			Network Analysis, coded in .NET
OceanViz	UBC, EII	User interface				√	Visualization of Ecopath and Ecosim results in Blender, a 3D game engine
Reading of GIS shape files*	EII	Automation			√		Provide transparent access to ESRI shape files
Remarks extractor	UBC	User interface	√	√	√	√	Present all remarks and references scattered throughout a model in one table
Results-extractor	CEFAS	Engine		√			Extract results from Ecosim, offering a range of aggregation and filter options
Sea Around Us taxon search*	UBC	Engine	√				Sea Around Us Project taxonomy database search engine
Shape grids	UBC	User interface	√	√	√		Add a gridded overview of forcing functions, mediation functions, and time series
Ecospace spin-up	UBC	Engine			√		Add a configurable spin-up time to stabilize Ecospace dynamics prior to a run
Temporal–spatial dynamic data	UBC, EII	Engine	√		√		Geospatial data access framework for Ecospace
Value chain	UBC	Analysis	√	√			Socio-economic analysis for Ecopath and Ecosim
WoRMS taxon search*	EII	Engine	√				World Register of Marine Species taxonomy search engine

Developers: AO (Agrocampus Ouest, France); CEFAS (Centre for Environment; Fisheries and Aquaculture Science, UK); CSIRO (Commonwealth Scientific and Industrial Research Organisation, Australia); DFO (Department of Fisheries and Oceans, Canada); EII (Ecopath International Initiative, Spain); ICM (Instituto de Ciencias del Mar, CSIC, Spain); IRD (Institut de recherche pour le développement, France); NOAA (National Oceanic and Atmospheric Administration, USA); SAMS (Scottish Association for Marine Science, UK); UBC (University of British Columbia, Canada).

added to the EwE either install package or can be obtained directly from the developer.

4. How to obtain the Ecopath source code

The EwE source code is freely available upon request. Instructions for obtaining the source code from the EwE SVN source code repository, development system requirements, and required programming tools can be found in the Ecopath development web portal and wiki (http://sources.ecopath.org/trac/Ecopath), see also Table A1.

Besides the code that is needed to recompile the desktop version of Ecopath with Ecosim, the EwE sources package contains a wide range of examples that show how to perform common tasks with EwE. These examples, together with the actual source code of EwE, provide a comprehensive roadmap to becoming at ease with programming with EwE—presuming that one is already familiar with the.NET programming languages.

The online EwE code documentation (http://webservice.ecopath.org/Ecopath/Help/Index.aspx) that is automatically updated daily from comments in the EwE source code, and provides a detailed description of all code elements of EwE.

If technical or scientific assistance is required, users can turn to the Ecopath user forms (http://www.ecopath.org/forum) or the Ecopath social network portal (http://www.facebook.com/eweconsortium) to find advice. For one-on-one assistance of the EwE core development team, one can seek support from the Ecopath Research and Development Consortium (http://www.ecopath.org/support).

4.1. How to contribute to EwE

EwE is an open-source, peer-reviewed program licensed under the GNU General Public License version 2. Third-party programmers are encouraged to submit plug-ins or code change suggestions to the Ecopath Research and Development Consortium, who will review the submission for scientific and technical merit under a user support contract. Once accepted, changes will be incorporated into the EwE install package and source code repository, and will be available to the EwE user community upon the next public release of the EwE desktop software. Note that EwE code changes, and plug-ins built upon the EwE code, automatically inherit the EwE software license, and require their source code to be made publicly available once a plug-in is made publicly available. For more details see http://www.gnu.org/licenses/gpl-2.0.html. For more details about submitting code to EwE please see the EwE development Wiki (http://sources.ecopath.org/trac/Ecopath/wiki/PlugIns).

Table 2
An overview of core EwE capabilities that have been adapted by the different versions of EwE.

EwE capabilities	.NET	Fortran	MATLAB	R
Ecopath (mass-balance)	√	√	√	√
Max. number of functional groups	∞	∞	∞	∞
Max. number of fishing fleets	∞	1	∞	∞
Age-structure (multi-stanza) representations of differentiation in diet, predation risk, and fishing impacts	√	√	√	√
Pedigree (input parameter confidence)	√		√	√
Ecosim (temporal dynamics)	√	√		√
Sub-month time steps	√	√		
Environmental forcing	√	√		√
Mediation	√			
Type II functional response effects (handling time, foraging time adjustment)	√	√		√
Type III functional response (switching)	√			√
Bout foraging and shared arenas	√			√
Time series driving	√	√		
Time series fitting	√			
Max. number of time steps	∞	∞		∞
Ecospace (spatial–temporal dynamics)	√			
Spatial distribution of biomass	√			
Spatial distribution of effort	√			
Environmental forcing	√			
Represent variable habitat quality	√			
Fishing closures through Marine Protected Areas	√			

5. EwE in other programming languages

The EwE models are not only available in .NET. Different research groups have reprogrammed parts of the EwE approach for specific scientific purposes as described below. Table 2 provides a comparison of core scientific and technical capacities between these versions.

5.1. EwE in Fortran

The aim of having Ecopath with Ecosim in Fortran (EwE-F) was to enable flexible incorporation of EwE in Fortran-based modelling schemes (Akoglu et al., 2014, 2015). Considering that most of the physical and biogeochemical oceanographic models are written in Fortran, EwE-F was primarily developed to facilitate its coupling with these models in order to set-up holistic (end-to-end) representations of marine ecosystems. However, the potential of EwE-F goes beyond this primary goal, since it permits integration and bi-directional coupling of multiple Fortran models within one single modelling scheme (Akoglu et al., 2014, 2015).

EwE-F is coded as two standalone components representing two core routines of EwE, i.e., Ecopath-F and Ecosim-F. This latter includes foraging arena and age-structured (multi-stanza) representations. However, given that EwE-F was developed for facilitating coupling and not as an alternative to EwE, Ecopath-F does not include, for example, capabilities of EwE-like network analysis, while Ecosim-F lacks both time series fitting and the Monte Carlo parameter search routine. Moreover, Ecospace is not included in EwE-F. Therefore, for analysis and calibration of the food web model users might want to utilize the desktop version of EwE, while using EwE-F for coupling purposes.

The source code of EwE-F is hosted at http://www.bitbucket.org and freely available. It comes with a detailed user's manual describing how to compile and run EwE-F as well as how to prepare EwE models for simulation by EwE-F. EwE-F can run on various computing environments without modification if a Fortran 95/2003 compliant compiler is available. The basic input/output is via delimited text files and EwE-F utilises HDF5 database format to exchange information from Ecopath-F to Ecosim-F (Table A2).

5.2. Ecopath in MATLAB

The MATLAB version of Ecopath was originally written in order to allow automated, repeated calculation of the Ecopath mass-balance algorithm without the overhead of the EwE Graphical User Interface. It has since developed into a suite of functions that focus on the use of ensembles of Ecopath models to incorporate parameter uncertainty into Ecopath-derived simulations and analyses (Kearney, 2013).

To date, this version of Ecopath has been used in a few lines of research. It was developed alongside, and is incorporated into the source code of the Kearney ecosystem model (Kearney et al., 2013, 2012), an end-to-end ecosystem model that couples physics, biogeochemistry, and an Ecosim-like food web model. While the process equations in that food web model differ slightly from the functional forms used by Ecosim, the underlying concept is the same, with initial conditions and process parameter constraints derived from the mass-balance output of an Ecopath model.

In the process of developing the above ecosystem model, it was found that small variations in initial Ecopath-derived parameters could lead to unexpectedly large variations in some biomass trajectories (more so than in a traditional Ecosim model, due to the much higher temporal resolution of lower trophic level processes). However, the uncertainty values associated with Ecopath input data is often high, up to or beyond an order of magnitude of the point value, meaning that these alternative output trajectories were all well within the realm of possibility. To incorporate this uncertainty into results, the authors developed an ensemble parameterization approach. This approach draws Ecopath parameters from probability distributions, based on initial point estimates and data pedigree values (i.e. uncertainty quantified based on data source), checking that the resulting parameter combinations meet the Ecopath mass-balance requirement. The result is an ensemble of balanced Ecopath models, each encompassing a different set of plausible input parameters.

In addition to serving its original purpose, the resulting ensemble-generation code has proven useful to incorporate uncertainty into a variety of Ecopath-derived ecosystem analyses. For example, it can be used to examine the potential uncertainty associated with ecological network indices. It also allows users to quickly search for balanced Ecopath parameter sets based on unbalanced starting data.

The source code for the MATLAB implementation of Ecopath is hosted at https://github.com/kakearney/ecopathlite-pkg. A brief tutorial for using the included functions is provided in the README.md file of this repository, and full documentation of use is provided in the header of each .m file, following the standard MATLAB help format. The current version of the code has been tested in MATLAB releases R2012a through R2014b; it should run in releases back to R2007a. The MATLAB Statistics Toolbox is required (Table A3).

5.3. EwE in R

The R package, Rpath, was developed to provide a platform-independent version of the core routines, in order to (i) implement EwE in a streamlined, transparent codebase suitable for code review within a management and/or legal framework; (ii) integrate results from EwE into the full suite of tools available for analysis in R; (iii) give advanced users direct access to core functional response forms, both for predator/prey and fishery interactions; and (iv) for Ecosim simulations in particular, maximize run speed (including

the use of parallelization) for analyses that require large number of runs.

The Ecopath balancing routine is implemented in approximately 200 lines of R code. Ecosim initialization is performed in R code, while the core Ecosim run routines are in a C++ plugin library, integrated into the R front end using Rcpp (Eddelbuettel et al., 2011) in order to allow for fast run times. A pedigree-based Monte Carlo routine, Ecosense (Aydin et al., 2005), is included with the code. Ecosense uses a Bayesian synthesis approach (Givens et al., 1993), which combines multiple sources of evidence and uncertainty linked by a deterministic model to generate a range of plausibly constrained ecosystems without requiring an initial Ecopath balance. Ecosense has been used to explore implications of a wide range of functional response types (Gaichas et al., 2012) and alternate hypotheses governing ecosystem structure (Aydin and Mueter, 2007; Gaichas et al., 2011).

One advantage of the R platform is that it allows access to the full range of developed R statistical and graphics tools. This allows, for example, a range of statistical distributions contained within R for input parameters rather than single values. It also allows for the user to implement complex fisheries control rules in R (for example, if bycatch of species A shuts down a fishery for species B, or fishing rate reduces past a fix reference point) or use existing R tools to find policy optima or reference points.

Rpath is currently in beta testing with plans for a summer 2015 release on the Comprehensive R Archive Network (CRAN, see Table A4).

5.4. EwE in C++

For sake of completeness it should be mentioned that previous efforts to implement EwE in C++ have been discontinued as a separate effort. The EwE-C++ code has been merged with RPath.

6. What's next

There has been significant demand for food web models to deliver spatially explicit results, especially in the context of climate change. In addition, food web models are increasingly applied in new areas such as environmental impact assessment. We expect that this trend will continue, and as such, we foresee an increasing interest in models that can provide quantitative predictions in complex integrated assessments that build on interoperating models from a range of disciplines. EwE modelling toolbox will be a key element in these capabilities.

However, this poses technical challenges to how we design our models beyond their scientific capabilities.

As a model development community, we can rise to the technical challenge by factoring interoperability into the very design of our tools. Using standard programming platforms that provide interconnectivity to other languages; using standard data formats and providing connectivity via common communication protocols; making the code of our models freely accessible; and providing clear documentation to code and data formats: all are important steps toward technical interoperability. Although these adaptations are common paradigms in commercial software development, they are rarely applied to ecological modelling software (Steenbeek et al., 2013).

Up to 2011, the EwE approach was conceptualized, developed, and maintained by one single institute. The Ecopath Research and Development Consortium was established to share this responsibility with the users of the approach, turning EwE in a truly peer-reviewed, open source, community-driven scientific effort. With a diverse development community behind the approach, it will be a significant challenge to build upon EwE core functionalities without turning the approach into a 'Frankenmodel' (Mackinson et al., 2009; Rose, 2012): let EwE do what EwE is good at, and connect to other specialist models if such specialized science needs to be integrated. The very code structure of EwE was designed with this concept in mind, and we hope that this philosophy will set a precedent for other, future modelling tools to follow suit.

Acknowledgements

Ecopath with Ecosim development is managed by the technical committee of the Ecopath Research and Development Consortium: Ekin Akoglu, Joe Buszowski, Villy Christensen, Nick Ellis, Jerome Guitton, Mark Platts, Frank Poulsen, Jeroen Steenbeek, and Carl Walters. We thank Marta Coll for her input to this manuscript.

EwE-F has been developed by team work of Ekin Akoglu, Simone Libralato, Baris Salihoglu, Temel Oguz and Cosimo Solidoro and results from the collaborative effort of IMS-METU (Institute of Marine Sciences, Middle East Technical University, Turkey) and OGS (Istituto Nazionale di Oceanografia e di Geofisica Sperimentale, Italy). The EwE-F development benefited in part from the support of the TRIL (Training and Research in Italian Laboratories) programme of the Abdus Salam International Centre for Theoretical Physics (ICTP, Trieste).

EcoTroph has been developed by Paul Gatti, Mathieu Colleter, and Jérôme Guitton under the supervision of Didier Gascuel.

The Ecopath .NET development team wishes to acknowledge the Lenfest Ocean Program (www.lenfestocean.org) for funding the reimplementation of the EwE software and the conceptualization of OceanViz until 2009. We wish to thank all EwE users and Ecopath Research and Development Consortium members who, through commissioning developments and applications, have been securing the future of the EwE approach since 2009.

The NPF integrated modelling framework was conceived by Francis Pantus, who in collaboration with Nick Ellis and Andrzej Welna, developed the first prototypes. The mature version was designed by Sharon Tickell with assistance on linkage to EwE by Ricardo Pascual.

Appendix A.

Tables A1–A4.

Table A1
Basic requirements of EwE in .NET.

Requirement	Type
Operating system	Windows (EwE desktop version) Mac OS X and Unix-like (EwE computations)
Disk space	250 MB of free disk space
Memory	4 GB recommended
Additional software	Microsoft Access database drivers (2007 or newer) Visual Studio/Visual Basic .NET Express (2010 or newer) TortoiseSVN or similar .NET Framework 4 (Windows) Mono/Portable.NET (Windows, Mac OS X and Unix-like)
License	GNU General Public License version 2
Homepage	http://www.ecopath.org http://sources.ecopath.org/trac/Ecopath
Source code access	Upon request
Technical contact	ewedevteam@gmail.com

Table A2
Basic requirements of the Fortran version of EwE.

Requirement	Type
Operating system	Windows, Mac OS X and Unix-like (Linux, BSD)
Disk space	50 MB
Memory	256 MB
Additional software	HDF5 version 1.8.11 or above
Compiler	Fortran 95/2003 standards-compliant compiler (e.g., GNU Fortran Compiler, Intel® Fortran Compiler, Oracle® Solaris Studio)
License	GNU General Public License version 2
Homepage	http://www.bitbucket.org/ewe-f
Source code access	Publicly available
Technical contact	ewefortran@gmail.com

Table A3
Basic requirements of the MATLAB version of Ecopath.

Requirement	Type
Operating system	Windows, Mac OS X and Unix-like (Linux, BSD)
Disk space	~1.5 GB for MATLAB installation 804 kB for ecopathlite code (2.2 MB for full clone with history)
Memory	1024 MB, 2048 MB recommended (for MATLAB)
Additional software	MATLAB (R2007a or later) MATLAB Statistics Toolbox
License	MIT License
Homepage	https://github.com/kakearney/ecopathlite-pkg
Source code access	Free
Technical contact	kakearney@gmail.com

Table A4
Basic requirements of the R version of EwE.

Requirement	Type
Operating system	Windows, Mac OS X and Unix-like (Linux, BSD)
Disk space	60 MB
Memory	2 GB
Additional software	R version 3.1.2 and newer
License	GNU General Public License version 2 (GPL-2)
Homepage	http://cran.r-project.org/
Source code access	Free
Technical contact	Sean.Lucey@NOAA.gov or Kerim.Aydin@NOAA.gov

References

Akoglu, E., Libralato, S., Oğuz, T., Solidoro, C., 2015. EwE-F 1.0: An implementation of Ecopath with Ecosim in Fortran 95/2003 for coupling. Geosci. Model Dev.

Akoglu, E., Libralato, S., Salihoğlu, B., Cossarini, G., Lazzari, P., Oğuz, T., Solidoro, C., 2014. Two-way coupling of EwE in Fortran with an intermediate complexity NPZD model. In: Steenbeek, J., Piroddi, C., Coll, M., Heymans, J.J., Villasante, S., Christensen, V. (Eds.), Ecopath 30 Years Conference Proceedings: Extended Abstracts P. 189, Fisheries Centre Research Reports. Fisheries Centre, University of British Columbia, p. 237.

Aydin, K., Mueter, F., 2007. The Bering Sea—a dynamic food web perspective. Deep Sea Res., II: Top. Stud. Oceanogr. 54, 2501–2525, http://dx.doi.org/10.1016/j.dsr2.2007.08.022 (Effects of Climate Variability on Sub-Arctic Marine Ecosystems A GLOBEC Symposium GLOBEC-ESSAS Symposium on "Effects of climate variability on sub-arctic marine ecosystems").

Aydin, K.Y., McFarlane, G.A., King, J.R., Megrey, B.A., Myers, K.W., 2005. Linking oceanic food webs to coastal production and growth rates of Pacific salmon (Oncorhynchus spp.), using models on three scales. Deep Sea Res., II: Top. Stud. Oceanogr. 52, 757–780, http://dx.doi.org/10.1016/j.dsr2.2004.12.017 (Linkages between coastal and open ocean ecosystems. Linkages between open ocean and coastal systems).

Butterworth, D.S., Cochrane, K.L., De Oliveira, J.A.A., 1997. Management procedures: a better way to manage fisheries? The South African experience. Glob. Trends Fish. Manage. 20, 83–90.

Canadian Environmental Assessment Agency, 2015. Roberts Bank Terminal 2 Project—Environmental Impact Statement (Environmental Impact Statement). Canadian Environmental Assessment Agency, Roberts Band, Delta, BC.

Christensen, V., Coll, M., Buszowski, J., Cheung, W.W.L., Frölicher, T., Steenbeek, J., Stock, C.A., Watson, R., Walters, C.J., 2015. The global ocean is an ecosystem: simulating marine life and fisheries. Global Ecol. Biogeogr. 24, 507–517, http://dx.doi.org/10.1111/geb.12281

Christensen, V., Coll, M., Steenbeek, J., Buszowski, J., Chagaris, D., Walters, C.J., 2014. Representing variable habitat quality in a spatial food web model. Ecosystems 1–16, http://dx.doi.org/10.1007/s10021-014-9803-3

Christensen, V., Lai, S., 2011. Scenario development for decision making. In: Christensen, V., Maclean, J. (Eds.), Ecosystem Approaches to Fisheries: A Global Perspective. Cambridge University Press, Cambridge, pp. 304–314.

Christensen, V., Lai, S., 2007. Ecopath with Ecosim 6: the sequel. Sea US Proj. Newsl., 43.

Christensen, V., Maclean, J., 2011. Ecosystem approaches to fisheries: a global perspective. Cambridge Univ Pr.

Christensen, V., Pauly, D., 1993. Trophic Models of Aquatic Ecosystems. The WorldFish Center.

Christensen, V., Pauly, D., 1992. ECOPATH II—a software for balancing steady-state ecosystem models and calculating network characteristics. Ecol. Model. 61, 169–185.

Christensen, V., Walters, C.J., 2011. Progress in the use of ecosystem models for fisheries management. In: Christensen, V., MacLean, J. (Eds.), Ecosystem Approaches to Fisheries: A Global Perspective. Cambridge University Press, Cambridge, pp. 189–205.

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

Colleter, M., Guitton, J., Gascuel, D., 2013. An Introduction to the EcoTroph R package: Analyzing Aquatic Ecosystem Trophic Network. The R Journal 5 (1), 98–107.

Coll, M., Bundy, A., Shannon, L.J., 2008. Ecosystem modelling using the Ecopath with Ecosim approach. In: Megrey, B.A., Moksness, E. (Eds.), Computers in Fisheries Research. Springer, Dordrecht, Netherlands, pp. 225–291.

Dichmont, C.M., Ellis, N., Bustamente, R.H., Deng, R., Tickell, S., Pascual, R., Lozano-Montes, H., Griffiths, S., 2013. Evaluating marine spatial closures with conflicting fisheries and conservation objectives. J. Appl. Ecol. 50, 1060–1070, http://dx.doi.org/10.1111/1365-2664.12110

Drayton, P., Albahari, B., Neward, T., 2003. C# in a Nutshell. O'Reilly Media Inc, Cambridge, MA.

Dutra, L., de la Mare, W., Perez, P., Ellis, N., Tickell, S., Cannard, T., Thebaud, O., Pascual, P., Bayliss, P., Moeseneder, C., et al., 2010. Healthy Waterways Management Strategy Evaluation: Scoping Study for the Development of a "Catchment-to-coast" MSE in SE Queensland—Phase 2—Final Report. CSIRO Marine and Atmospheric Research, Brisbane.

ECMA International, 2012. Standard ECMA-335 [WWW Document]. ⟨http://www.ecma-international.org/publications/standards/Ecma-335.htm⟩ (accessed 8.11.12).

Eddelbuettel, D., François, R., Allaire, J., Chambers, J., Bates, D., Ushey, K., 2011. Rcpp: Seamless R and C++ integration. J. Stat. Softw. 40, 1–18.

Ellis, N., Pantus, F., Welna, A., Butler, A., 2008. Evaluating ecosystem-based management options: effects of trawling in Torres Strait, Australia. Cont. Shelf Res. 28, 2324–2338, http://dx.doi.org/10.1016/j.csr.2008.03.031, Marine resources, biophysical processes, and environmental management of a tropical shelf seaway; Torres Strait, Australia.

Felinto, D., Pan, M., 2013. Game Development with Blender, first ed. Cengage Learning, Boston, MA.

Gaichas, S.K., Aydin, K.Y., Francis, R.C., 2011. What drives dynamics in the Gulf of Alaska? Integrating hypotheses of species, fishing, and climate relationships using ecosystem modeling. Can. J. Fish. Aquat. Sci. 68, 1553–1578, http://dx.doi.org/10.1139/f2011-080

Gaichas, S.K., Odell, G., Aydin, K.Y., Francis, R.C., 2012. Beyond the defaults: functional response parameter space and ecosystem-level fishing thresholds in dynamic food web model simulations. Can. J. Fish. Aquat. Sci. 69, 2077–2094, http://dx.doi.org/10.1139/f2012-099

Gasche, L., Gascuel, D., Shannon, L., Shin, Y.-J., 2012. Global assessment of the fishing impacts on the Southern Benguela ecosystem using an EcoTroph modelling approach. J. Mar. Syst. 90, 1–12, http://dx.doi.org/10.1016/j.jmarsys.2011.07.012

Gascuel, D., 2005. The trophic-level based model: a theoretical approach of fishing effects on marine ecosystems. Ecol. Model. 189, 315–332, http://dx.doi.org/10.1016/j.ecolmodel.2005.03.019

Gascuel, D., Guénette, S., Pauly, D., 2011. The trophic-level-based ecosystem modelling approach: theoretical overview and practical uses. ICES J. Mar. Sci. J. Cons. 68, 1403–1416, http://dx.doi.org/10.1093/icesjms/fsr062

Gascuel, D., Pauly, D., 2009. EcoTroph: modelling marine ecosystem functioning and impact of fishing. Ecol. Model. 220, 2885–2898, http://dx.doi.org/10.1016/j.ecolmodel.2009.07.031, Selected Papers from the Sixth European Conference on Ecological Modelling—ECEM'07, on Challenges for Ecological Modelling in a Changing World: Global Changes, Sustainability and Ecosystem Based Management, November 27–30, 2007, Trieste, Italy.

Givens, G.H., Raftery, A.E., Zeh, J.E., 1993. Benefits of a Bayesian approach for synthesizing multiple sources of evidence and uncertainty linked by a deterministic model. Rep. Int. Whal. Comm. 43, 495–500.

Kearney, K.A., 2013. A Matlab-based Implementation of the Ecopath Food Web Model. GitHub Inc [WWW Document] ⟨https://github.com/kakearney/ecopathlite-pkg⟩ (accessed 1.28.15).

Kearney, K.A., Stock, C.A., Sarmiento, J.L., 2013. Amplification and attenuation of increased primary production in a marine food web. Mar. Ecol. Prog. Ser. 491, 1–14.

Kearney, K.A., Stock, C., Aydin, K., Sarmiento, J.L., 2012. Coupling planktonic ecosystem and fisheries food web models for a pelagic ecosystem: description and

validation for the subarctic Pacific. Ecol. Model. 237-238, 43–62, http://dx.doi.org/10.1016/j.ecolmodel.2012.04.006
Link, J., 2010. Ecosystem-based Fisheries Management: Confronting Tradeoffs. Cambridge University Press, Cambridge.
Mackinson, S., Beecham, J., Aldridge, J., 2009. Frankenstein models? Coupling Ecopath with Ecosim to biogeochemical models. MEECE Mar. Ecosyst. Evol. Chang. Environ. Newsl. 2, 5.
Makofske, D.B., Donahoo, M.J., Calvert, K.L., 2004. TCP/IP Sockets in C#: Practical Guide for Programmers. Morgan Kaufmann Publishers, San Franciso CA.
Microsoft News Center, 2014. Microsoft takes.NET open source and cross-platform, adds new development capabilities with Visual Studio 2015,.NET 2015 and Visual Studio Online. Microsoft News Cent. [WWW Document] ⟨http://news.microsoft.com/2014/11/12/microsoft-takes-net-open-source-and-cross-platform-adds-new-development-capabilities-with-visual-studio-2015-net-2015-and-visual-studio-online/⟩ (accessed 1.18.15).
Microsoft Support, 2014. Microsoft.NET Framework Support Lifecycle Policy FAQ. Microsoft Support [WWW Document] ⟨http://support.microsoft.com/gp/Framework_FAQ⟩.
Plaganyí, E.P., Butterworth, D.S., 2004. A critical look at the potential of Ecopath with Ecosim to assist in practical fisheries management. Afr. J. Mar. Sci. 26, 261–287.
Polovina, J.J., 1984. Model of a coral reef ecosystem. Coral Reefs 3, 1–11.
Rose, K.A., 2012. End-to-end models for marine ecosystems: are we on the precipice of a significant advance or just putting lipstick on a pig? Sci. Mar. 76, 195–201.
Steenbeek, J., 2012. Bridging the Gap Between Ecosystem Modelling Tools Using Geographic Information Systems. Manchester Metropolitan University, Manchester (M.Sc).
Steenbeek, J., Coll, M., Gurney, L., Mélin, F., Hoepffner, N., Buszowski, J., 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, http://dx.doi.org/10.1016/j.ecolmodel.2013.04.027
Stock, C.A., Dunne, J.P., John, J.G., 2014. Global-scale carbon and energy flows through the marine planktonic food web: an analysis with a coupled physical–biological model. Prog. Oceanogr. 120, 1–28, http://dx.doi.org/10.1016/j.pocean.2013.07.001
Walters, C., 2000. Impacts of dispersal, ecological interactions, and fishing effort dynamics on efficacy of marine protected areas: how large should protected areas be? Bull. Mar. Sci. 66, 745–757.
Walters, C.J., Christensen, V., Pauly, D., 1997. Structuring dynamic models of exploited ecosystems from trophic mass-balance assessments. Rev. Fish Biol. Fish. 7, 139–172.
Walters, C.J., Green, R., 1997. Valuation of experimental management options for ecological systems. J. Wildl. Manage. 61, 987–1006, http://dx.doi.org/10.2307/3802096
Walters, C.J., Martell, S., Christensen, V., Mahmoudi, B., 2008. An Ecosim model for exploring Gulf of Mexico ecosystem management options: implications of including multistanza life-history models for policy predictions. Bull. Mar. Sci. 83, 251–271.
Walters, C.J., Pauly, D., 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.