

# Introduction to FLR

FLR Core Team

November 19, 2010

# Outline

## Introduction

# Outline

Introduction

Philosophy of FLR

# Outline

Introduction

Philosophy of FLR

What is FLR?

# Outline

Introduction

Philosophy of FLR

What is FLR?

Design of FLR

# Outline

Introduction

Philosophy of FLR

What is FLR?

Design of FLR

What's next ?

# Need for FLR

Schnute *et al.* (2007 and 1998) compared the number of software tools and languages currently available for stock assessments with the Babel tower myth:

# Need for FLR

Schnute *et al.* (2007 and 1998) compared the number of software tools and languages currently available for stock assessments with the Babel tower myth:

”After the people of Babel sought to build a tower to heaven, the Lord God devised a plan (Genesis 11: 4-7). ’Behold the people is one; and they all have one language; and this they began to do; and *now nothing will be restrained from them, which they have imagined to do....* Let us go down, and there confound their language, that they may not understand one another’s speech.’ Italics highlight the prospects for accomplishment with a common language, if the scientific community could ever agree on one”



# Need for FLR

Schnute *et al.* (2007 and 1998) compared the number of software tools and languages currently available for stock assessments with the Babel tower myth:

"After the people of Babel sought to build a tower to heaven, the Lord God devised a plan (Genesis 11: 4-7). 'Behold the people is one; and they all have one language; and this they began to do; and *now nothing will be restrained from them, which they have imagined to do....* Let us go down, and there confound their language, that they may not understand one another's speech.' Italics highlight the prospects for accomplishment with a common language, if the scientific community could ever agree on one"

"The cosmic plan for **confounding software languages** seems to be working remarkably well among the community of quantitative fishery scientists!"

# Goals

To **promote and generalize** the use of **good quality, open source, flexible software** in all areas of quantitative fisheries research and management advice, with a key focus on Management Strategies Evaluation.

# Goals

In detail, FLR aims to facilitate and promote research about:

# Goals

In detail, FLR aims to facilitate and promote research about:

- ▶ Stock assessment and provision of management advice

# Goals

In detail, FLR aims to facilitate and promote research about:

- ▶ Stock assessment and provision of management advice
- ▶ Data and model validation through simulation

# Goals

In detail, FLR aims to facilitate and promote research about:

- ▶ Stock assessment and provision of management advice
- ▶ Data and model validation through simulation
- ▶ Risk analysis

# Goals

In detail, FLR aims to facilitate and promote research about:

- ▶ Stock assessment and provision of management advice
- ▶ Data and model validation through simulation
- ▶ Risk analysis
- ▶ Capacity development & education

# Goals

In detail, FLR aims to facilitate and promote research about:

- ▶ Stock assessment and provision of management advice
- ▶ Data and model validation through simulation
- ▶ Risk analysis
- ▶ Capacity development & education
- ▶ Promote collaboration and openness in quantitative fisheries science



# Goals

In detail, FLR aims to facilitate and promote research about:

- ▶ Stock assessment and provision of management advice
- ▶ Data and model validation through simulation
- ▶ Risk analysis
- ▶ Capacity development & education
- ▶ Promote collaboration and openness in quantitative fisheries science
- ▶ Support the development of new models and methods

# Goals

In detail, FLR aims to facilitate and promote research about:

- ▶ Stock assessment and provision of management advice
- ▶ Data and model validation through simulation
- ▶ Risk analysis
- ▶ Capacity development & education
- ▶ Promote collaboration and openness in quantitative fisheries science
- ▶ Support the development of new models and methods
- ▶ Promote the distribution of new models and methods to a wide public.

# A brief history of FLR

# A brief history of FLR

- ▶ Started by FEMS EU project

# A brief history of FLR

- ▶ Started by FEMS EU project
- ▶ COMMIT & EFIMAS EU projects provided much of time and sweat

# A brief history of FLR

- ▶ Started by FEMS EU project
- ▶ COMMIT & EFIMAS EU projects provided much of time and sweat
- ▶ Presented to ICES WG Methods 2004

# A brief history of FLR

- ▶ Started by FEMS EU project
- ▶ COMMIT & EFIMAS EU projects provided much of time and sweat
- ▶ Presented to ICES WG Methods 2004
- ▶ FLCore version 1.0 - December 2005
  - ▶ FLQuant with 5 dimensions, no "iter"
  - ▶ Release often, release early. Bugs galore

# A brief history of FLR

- ▶ Started by FEMS EU project
- ▶ COMMIT & EFIMAS EU projects provided much of time and sweat
- ▶ Presented to ICES WG Methods 2004
- ▶ FLCore version 1.0 - December 2005
  - ▶ FLQuant with 5 dimensions, no "iter"
  - ▶ Release often, release early. Bugs galore
- ▶ FLCore version 1.4 - 2007
  - ▶ Stable, full of treats and joy



## FLR 1.4 - *The Golden Jackal*



# A brief history of FLR

- ▶ 2007-2009: The Silk Road to version 2
  - ▶ New FLQuant with 6 dimensions: uncertainty in structure
  - ▶ Rewrite of most methods
  - ▶ Extension of methods available
  - ▶ New classes: FLModel
  - ▶ Stronger use of class inheritance
  - ▶ Overhaul of man pages
  - ▶ Simplification of package map

# A brief history of FLR

- ▶ 2007-2009: The Silk Road to version 2
  - ▶ New FLQuant with 6 dimensions: uncertainty in structure
  - ▶ Rewrite of most methods
  - ▶ Extension of methods available
  - ▶ New classes: FLModel
  - ▶ Stronger use of class inheritance
  - ▶ Overhaul of man pages
  - ▶ Simplification of package map
- ▶ FLCore version 2.0 - January 2009
  - ▶ 2.2: actual stable version
  - ▶ 2.3: under development

## FLR 2.2 - *Swordfish Polka*



# Mission statement

The FLR project provides a **platform for quantitative fisheries science** based on the R statistical language. The guiding principles of FLR are:

# Mission statement

The FLR project provides a **platform for quantitative fisheries science** based on the R statistical language. The guiding principles of FLR are:

- ▶ **openness** - through community involvement and the open source ethos

# Mission statement

The FLR project provides a **platform for quantitative fisheries science** based on the R statistical language. The guiding principles of FLR are:

- ▶ **openness** - through community involvement and the open source ethos
- ▶ **flexibility** - through a design that does not constrain the user to a given paradigm

# Mission statement

The FLR project provides a **platform for quantitative fisheries science** based on the R statistical language. The guiding principles of FLR are:

- ▶ **openness** - through community involvement and the open source ethos
- ▶ **flexibility** - through a design that does not constrain the user to a given paradigm
- ▶ **extendibility** - through the provision of tools that are ready to be personalized and adapted.



# Mission statement

FLR's framework **facilitates and promotes collaboration** within and across disciplines, e.g. biological, ecological, statistical, mathematical, economic, and social. In particular it ensures that new modelling methods are widely available, so that alternative fisheries management strategies and procedures can be evaluated for their robustness to uncertainty before implementation.

# Mission statement

FLR's framework **facilitates and promotes collaboration** within and across disciplines, e.g. biological, ecological, statistical, mathematical, economic, and social. In particular it ensures that new modelling methods are widely available, so that alternative fisheries management strategies and procedures can be evaluated for their robustness to uncertainty before implementation.

FLR is distributed with an **open source** license and encourages all packages to be distributed under open source licenses in order **to promote transparency and technology transfer** between disciplines and researchers.

# GNU project (<http://gnu.org>)

”Free software is a matter of liberty, not price.”

**free = free speech**

**free != free beer**

# Collaboration and Open Source

”I think the real issue about adoption of open source is that **nobody can really ever 'design' a complex system.** That's simply not how things work: people aren't that smart - nobody is. And what open source allows is to not actually 'design' things, but let them **evolve**, through lots of different pressures in the market, and having the end result just **continually improve**”

Linus Torvalds

# Development of FLR

FLR is a **collaborative development project**, where distinct scientists work simultaneously on code, documentation, etc.

# Development of FLR

FLR is a **collaborative development project**, where distinct scientists work simultaneously on code, documentation, etc.

- ▶ Development is managed through R-Forge  
(<http://r-forge.r-project.org/projects/flr/>)

# Development of FLR

FLR is a **collaborative development project**, where distinct scientists work simultaneously on code, documentation, etc.

- ▶ Development is managed through R-Forge  
(<http://r-forge.r-project.org/projects/flr/>)
- ▶ Packages on repository  
(`install.packages(repos="http://flr-project.org/R")`)

# Development of FLR

FLR is a **collaborative development project**, where distinct scientists work simultaneously on code, documentation, etc.

- ▶ Development is managed through R-Forge (<http://r-forge.r-project.org/projects/flr/>)
- ▶ Packages on repository (`install.packages(repos="http://flr-project.org/R")`)
- ▶ Documentation on wiki (<http://www.flr-project.org>)



# Development of FLR

FLR is a **collaborative development project**, where distinct scientists work simultaneously on code, documentation, etc.

- ▶ Development is managed through R-Forge (<http://r-forge.r-project.org/projects/flr/>)
- ▶ Packages on repository (`install.packages(repos="http://flr-project.org/R")`)
- ▶ Documentation on wiki (<http://www.flr-project.org>)
- ▶ Funding comes from a number of EU projects (FEMS, COMMIT, EFIMAS, Fisboat, UNCOVER)

# Development of FLR

FLR is a **collaborative development project**, where distinct scientists work simultaneously on code, documentation, etc.

- ▶ Development is managed through R-Forge (<http://r-forge.r-project.org/projects/flr/>)
- ▶ Packages on repository (`install.packages(repos="http://flr-project.org/R")`)
- ▶ Documentation on wiki (<http://www.flr-project.org>)
- ▶ Funding comes from a number of EU projects (FEMS, COMMIT, EFIMAS, Fisboat, UNCOVER)
- ▶ FLR Core Team, aka "*Les misérables*", code, maintenance, tests, documentation, etc.

# Cut the crap, what is FLR?

**A working environment  
for quantitative fisheries analysis in R!**

# Cut the crap, what is FLR?

- ▶ Extendable toolbox for implementing bio-economic simulation models of fishery systems

# Cut the crap, what is FLR?

- ▶ Extendable toolbox for implementing bio-economic simulation models of fishery systems
- ▶ Tools used by managers (hopefully) as well as scientists

# Cut the crap, what is FLR?

- ▶ Extendable toolbox for implementing bio-economic simulation models of fishery systems
- ▶ Tools used by managers (hopefully) as well as scientists
- ▶ With many applications including:

# Cut the crap, what is FLR?

- ▶ Extendable toolbox for implementing bio-economic simulation models of fishery systems
- ▶ Tools used by managers (hopefully) as well as scientists
- ▶ With many applications including:
  - ▶ Fit stock-recruitment relationships,
  - ▶ Model fleet dynamics (including economics),
  - ▶ Simulate and evaluate management procedures and HCRs,
  - ▶ More than just stock assessment (VPA, XSA, ICES uptake)
  - ▶ etc....

# R and FLR

Why do we use R?



# R and FLR

Why do we use R?

- ▶ Existing platform for statistical modelling

# R and FLR

Why do we use R?

- ▶ Existing platform for statistical modelling
- ▶ Good graphics capabilities

# R and FLR

Why do we use R?

- ▶ Existing platform for statistical modelling
- ▶ Good graphics capabilities
- ▶ Multi-platform

# R and FLR

Why do we use R?

- ▶ Existing platform for statistical modelling
- ▶ Good graphics capabilities
- ▶ Multi-platform
- ▶ Open source

# R and FLR

Why do we use R?

- ▶ Existing platform for statistical modelling
- ▶ Good graphics capabilities
- ▶ Multi-platform
- ▶ Open source
- ▶ Links with compiled languages like fortran, C / C++ code for speed

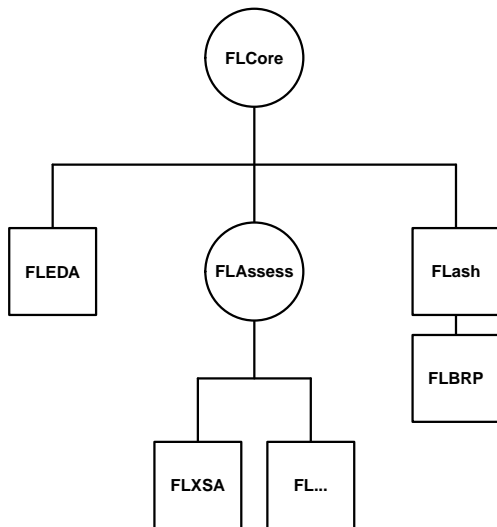
# R and FLR

Why do we use R?

- ▶ Existing platform for statistical modelling
- ▶ Good graphics capabilities
- ▶ Multi-platform
- ▶ Open source
- ▶ Links with compiled languages like fortran, C / C++ code for speed
- ▶ Easily extendable in the form of 'packages'

# Packages

## FLR packages' development model



# Object oriented programming with S4



# Object oriented programming with S4

- ▶ A programming language model organized around "objects" rather than "actions"

# Object oriented programming with S4

- ▶ A programming language model organized around "objects" rather than "actions"
- ▶ Uses R S4 classes

# Object oriented programming with S4

- ▶ A programming language model organized around "objects" rather than "actions"
- ▶ Uses R S4 classes
- ▶ Everything is an object of a particular class

# Object oriented programming with S4

- ▶ A programming language model organized around "objects" rather than "actions"
- ▶ Uses R S4 classes
- ▶ Everything is an object of a particular class
- ▶ Objects have:

# Object oriented programming with S4

- ▶ A programming language model organized around "objects" rather than "actions"
- ▶ Uses R S4 classes
- ▶ Everything is an object of a particular class
- ▶ Objects have:
  - ▶ members (data) and

# Object oriented programming with S4

- ▶ A programming language model organized around "objects" rather than "actions"
- ▶ Uses R S4 classes
- ▶ Everything is an object of a particular class
- ▶ Objects have:
  - ▶ members (data) and
  - ▶ methods (functions associated with it that act on member data)

# Object oriented programming with S4

- ▶ A programming language model organized around "objects" rather than "actions"
- ▶ Uses R S4 classes
- ▶ Everything is an object of a particular class
- ▶ Objects have:
  - ▶ members (data) and
  - ▶ methods (functions associated with it that act on member data)
- ▶ Inheritance used to extend and create new classes (FLSR inherits from FLModel)

# Object oriented programming with S4

- ▶ A programming language model organized around "objects" rather than "actions"
- ▶ Uses R S4 classes
- ▶ Everything is an object of a particular class
- ▶ Objects have:
  - ▶ members (data) and
  - ▶ methods (functions associated with it that act on member data)
- ▶ Inheritance used to extend and create new classes (FLSR inherits from FLModel)
- ▶ Classes can be members of other classes (most FLR classes include FLQuants as members)



# Design principles

# Design principles

- ▶ Classes to represent different elements of fisheries systems

# Design principles

- ▶ Classes to represent different elements of fisheries systems
- ▶ 'physical' objects (e.g. FLStock class represents a fish stock)

# Design principles

- ▶ Classes to represent different elements of fisheries systems
- ▶ 'physical' objects (e.g. FLStock class represents a fish stock)
- ▶ 'methodological' objects (e.g. FLBRP class containing methods to calculate BRP)

# Design principles

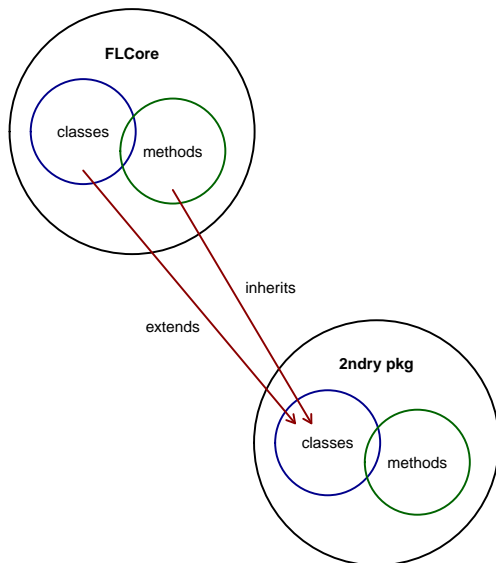
- ▶ Classes to represent different elements of fisheries systems
- ▶ 'physical' objects (e.g. FLStock class represents a fish stock)
- ▶ 'methodological' objects (e.g. FLBRP class containing methods to calculate BRP)
- ▶ Link objects to create simulations - Lego blocks (MSE example)

# Design principles

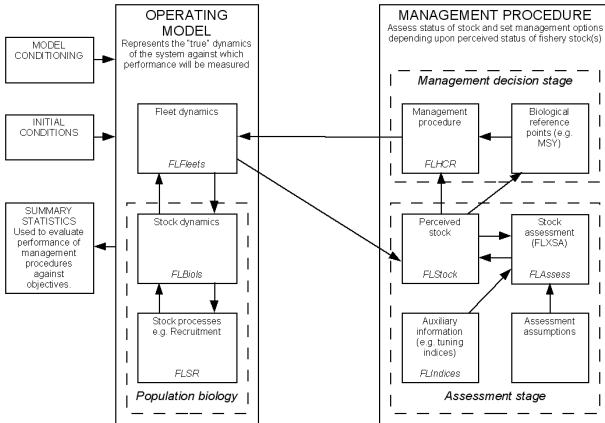
- ▶ Classes to represent different elements of fisheries systems
- ▶ 'physical' objects (e.g. FLStock class represents a fish stock)
- ▶ 'methodological' objects (e.g. FLBRP class containing methods to calculate BRP)
- ▶ Link objects to create simulations - Lego blocks (MSE example)
- ▶ Learning curve: trade off between flexibility and simplicity (no black boxes and no handle turning)

# FLR & S4

## FLR hierarchical model



## MSE - The Lego block approach





## What's next ?

