

Outline

1. Facts
2. a4a
3. Modular MSE
4. Examples
5. Comments

Fact

Since 2003 the European Union Member States have spent collecting data

≈ 75 million euros per year

from an average gross value landed per year of

7.1k million euros¹.

¹Source AER

Fact

Nevertheless the number of stocks which are managed under **scientific advise based on analytical assessments did not increase** that much.

The 'assessment for all' Initiative (a4a)

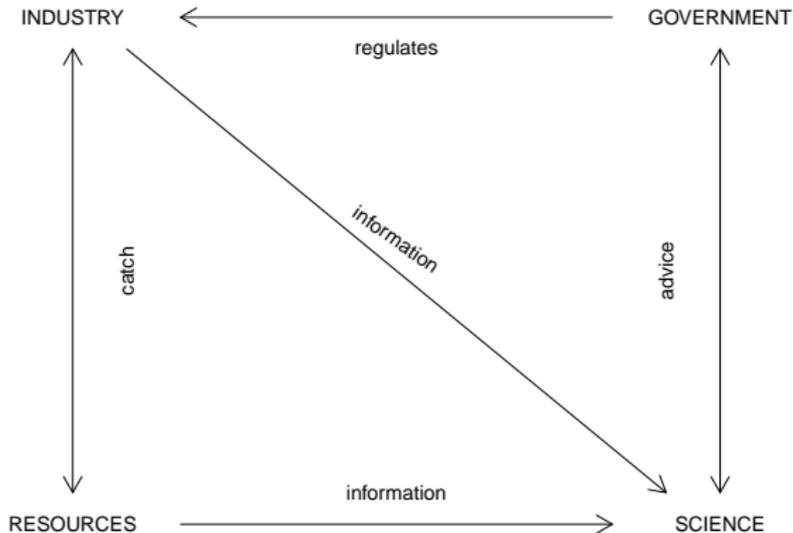
The European Commission Joint Research Centre (JRC) started its a4a Initiative, with the aim to:

- develop, test, and distribute methods to assess a large numbers of stocks and give advice to policy makers in an operational time frame:
 - stock assessment framework
 - **MSE algorithm**
- build the necessary capacity/expertise on stock assessment and advice provision

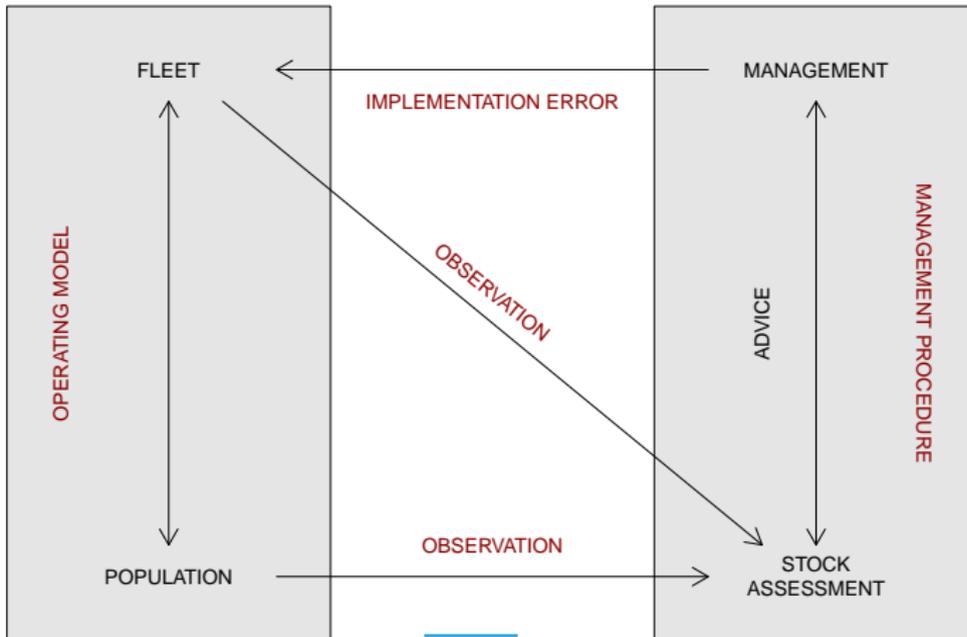
Modular MSE

What is a modular MSE and how does it help ?

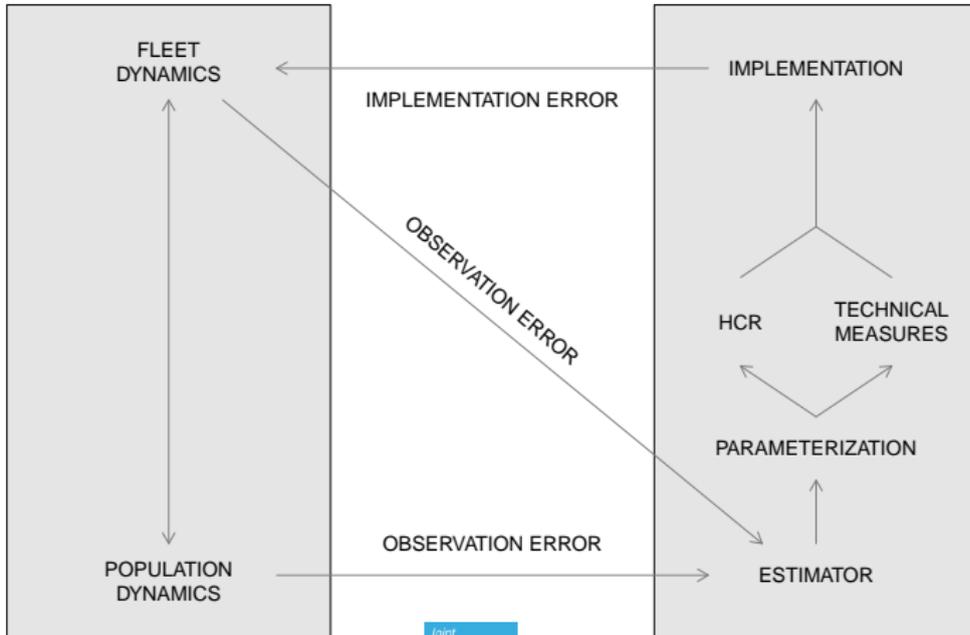
Modular MSE: The management cycle



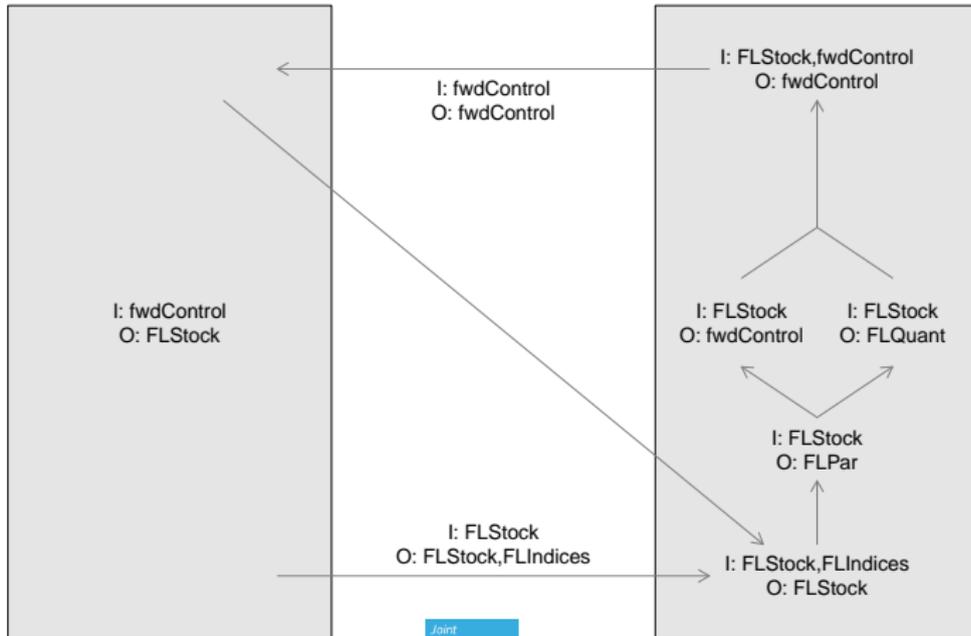
Modular MSE: overview



Generalizing and modularizing the a4a MSE



Coding contract: I/O



Coding contract: the mseCtrl class

```
showClass("mseCtrl")  
  
## Class "mseCtrl" [package "mse"]  
##  
## Slots:  
##  
## Name:    method    args  
## Class: function  list  
##  
## Known Subclasses: "FLoem", "FLiem"
```

Coding a module (HCR example)

```
fixedF.hcr <- function(stk, ftrg, genArgs, tracking) {  
  ay <- genArgs$ay  
  if (!is(ftrg, "FLQuant"))  
    ftrg <- FLQuant(ftrg, dimnames = list(iter = dimnames(stk@catch)$iter))  
  ctrl <- getCtrl(c(ftrg), "f", ay + genArgs$management_lag, dim(ftrg)[6])  
  list(ctrl = ctrl, tracking = tracking)  
}  
  
ctrl.hcr <- mseCtrl(method = fixedF.hcr, args = list(ftrg = 0.3))
```

Coding the a4a sca estimator

```
sca.sa <- function(stk, idx, genArgs, update = TRUE, dfm = c(0.75, 0.75), ...) {  
  args <- list(...)  
  if (update)  
    args$fmodel <- defaultFmod(stk, dfm = dfm)  
  args$stock <- stk  
  args$indices <- idx  
  if (is.null(args$fit))  
    args$fit <- "MP"  
  tracking <- args$tracking  
  args$tracking <- NULL  
  fit <- do.call("sca", args)  
  stk <- stk + fit  
  tracking["conv.est", ac(range(stk)["maxyear"] + 1)] <- fit@fitSumm["maxgrad",  
    ]  
  list(stk = stk, tracking = tracking)  
}  
  
ctrl.est <- mseCtrl(method = sca.sa)
```

Advantages of abstraction

Process	Full feedback	Data limited
Observation model	catch-at-age, survey	catch length frequencies
Estimator	stock assessment	$\bar{L}_{current}$
Parametrization	MSYBtrigger	L_{opt}
HCR	ICES, 40/20	$\delta_{future} = \frac{\bar{L}_{current} - L_{opt}}{L_{opt}}$
Technical measures	MPA (changes F@age)	MPA (changes \bar{L}_{catch})
Implementation	$TAC = f(C_{past} HCR)$	$TAC = f(C_{past} HCR)$
Implementation error	Overcatch	Overcatch

Table: Comparative example of full feedback and data limited MSEs

Full feedback model examples



**WORKSHOP ON NORTH SEA STOCKS
MANAGEMENT STRATEGY EVALUATION
(WKNSMSE)**

Please note: Section 4 was added to this report 17 June 2019

VOLUME 1 | ISSUE 12

ICES SCIENTIFIC REPORTS 



JRC SCIENCE FOR POLICY REPORT

Scientific, Technical and Economic
Committee for Fisheries (STECF)

-

Multiannual Plan for the fisheries
exploiting demersal stocks in the
Adriatic Sea
(STECF-19-02)

Data limited stocks examples

ICES REPORT WKLIFE VII 2017

ICES Advisory Committee

ICES CM 2017/ACOM:43

Report from the Workshop on the
Development of Quantitative Assessment
Methodologies based on Life-history traits,
exploitation characteristics, and other relevant
parameters for stocks in categories 3–6

2–6 October 2017

Lisbon, Portugal

Received: 28 July 2017 | Revised: 22 July 2018 | Accepted: 27 July 2018
DOI: 10.1111/wlfe.12216



ORIGINAL ARTICLE

WILEY 

Trade-offs for data-limited fisheries when using harvest strategies based on catch-only models

Jessica C. Walsh¹ | Cöllin Minto² | Ernesto Jardim³ | Sean C. Anderson^{4,5} |
Olaf P. Jensen⁶ | Jamie Afflerbach⁷ | Mark Dickey-Collas^{8,9} | Kristin M. Kleisner¹⁰ |
Catherine Longo¹¹ | Giacomo Chato Osio³ | Elizabeth R. Selig^{12,13} | James T.
Thorson¹⁴ | Merrill B. Rudd⁴ | Katherine J. Papacostas^{12,15,16} | John N.
Kittinger^{17,18} | Andrew A. Rosenberg¹⁹ | Andrew B. Cooper^{1,20}

Comments about modular approach

- Make it simpler to implement and share methods,
- Reduces the current workload,
- Improves readability, replicability, etc,
- Improves communication !

Comments about modular MSE

If MSEs are easier to implement and more "transferable", we should be able to **escape** the yearly turn-the-crank approach and focus on a longer term perspective.

We can take more time **doing research and thinking** about the stock(s) we're giving advice !

Comments about development

- **R**: well established language and development environment, "easy" parallelization;
- **FLR**: well established data structures, methods and visualization;
- **OOP/S4**: as much as possible;
- **modules**: standardize and define I/O to make it possible to plugin new methods;
- **dev/test/apply** loop;
- **bugs**: add unit tests as much as possible (some hick ups here ...);
- **dev**: small number of developers (4), github;
- **document**: arghhh ...;
- **management**: informal, strategic, funded by the JRC;

Thank you !

<ernesto.jardim@ec.europa.eu>