Population ecology modelling with R

A Comparison of Object Oriented Approaches

Thomas Petzoldt[1] Karsten Rinke[2] Louis Kates[3]

[1]Institute of Hydrobiology
Technische Universität Dresden, Germany

[2]Limnological Institute
Universität Konstanz, Germany

[3]GKX Associates Inc.
Waterloo, ON, Canada

Second use-R Conference Vienna, 2006

TECHNISCHE UNIVERSITÄT DRESDEN

OOP in Ecological Modelling

Petzoldt, Rinke, Kates

Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

Outline

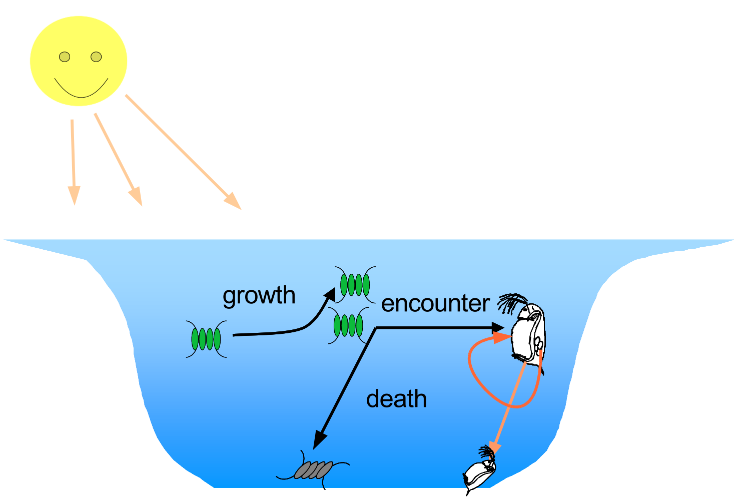
TECHNISCHE UNIVERSITÄT DRESDEN

OOP in Ecological Modelling

Petzoldt, Rinke, Kates

Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

A Basic Lake Model



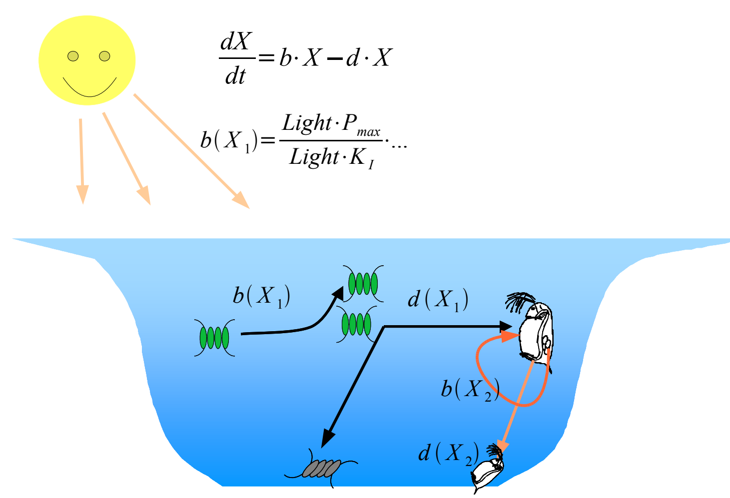
TECHNISCHE UNIVERSITÄT DRESDEN

OOP in Ecological Modelling

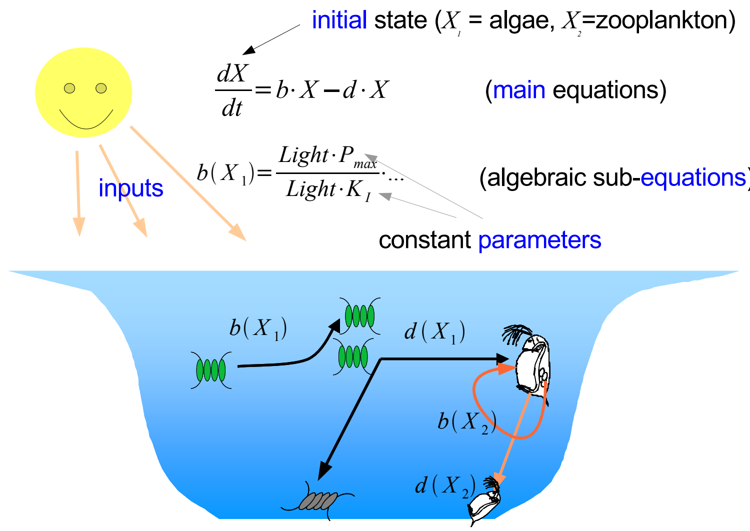
Petzoldt, Rinke, Kates

Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

A Basic Lake Model



A Basic Lake Model



R in Ecological Modelling

A great tool:

- Well suited to implement all types of models:
  - ODE (Lotka-Volterra …"complete Lakes")
  - Individual-based
  - Grid-Based, …

R in Ecological Modelling

A great tool:

- Well suited to implement all types of models:
  - ODE (Lotka-Volterra …"complete Lakes")
  - Individual-based
  - Grid-Based, …

Problems:

- Different types of models
- Different people, programming skills,
- Few time for science – no time for documentation.
- Incompatible spaghetti-code.

R in Ecological Modelling

A great tool:

- Well suited to implement all types of models:
  - ODE (Lotka-Volterra …"complete Lakes")
  - Individual-based
  - Grid-Based, …

Problems:

- Different types of models
- Different people, programming skills,
- Few time for science – no time for documentation.
- Incompatible spaghetti-code.

- Hack complete program to change only one parameter?
- Better write new code than re-use existing?

Workflow and requirements

Common tasks:

- Compare the same model with different data,
- Compare two different models with same data.

Typical application scenario:

- Load the model,
- Run the model,
- Create scenarios,
- Compare scenarios.

Requirements:

- Ease of application,
- Meaningful defaults,
- Storage of results and settings.

TECHNISCHE UNIVERSITÄT DRESDEN
OOP in Ecological Modelling
Petzoldt, Rinke, Kates
Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

Workflow and requirements

Common tasks:

- Compare the same model with different data,
- Compare two different models with same data.

Typical application scenario:

- Load the model,
- Run the model,
- Create scenarios,
- Compare scenarios.

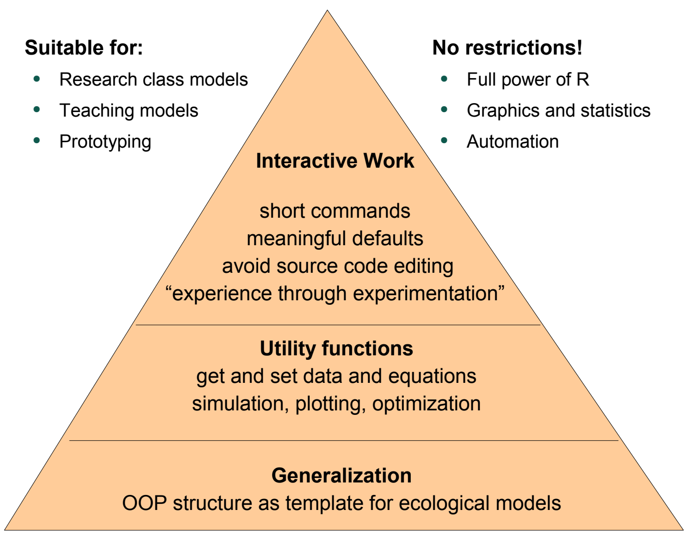
Requirements:

- Ease of application,
- Meaningful defaults,
- Storage of results and settings.

TECHNISCHE UNIVERSITÄT DRESDEN
OOP in Ecological Modelling
Petzoldt, Rinke, Kates
Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

Basic idea and goal

Provide a standard architecture and utility functions and propagate a common style.



TECHNISCHE UNIVERSITÄT DRESDEN
OOP in Ecological Modelling
Petzoldt, Rinke, Kates
Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

Approach: OOP template and package – simplify and unify ecological modelling with R

- Which OOP approaches are available?
- What is typical in ecological modelling?
- Provide an R Package with one selected OOP paradigm.

OOP in R

Several OOP systems in R:

- S3 : original class system of R,
- S4 : the new standard OOP system (Chambers, 1998); ensures method consistency,
- R.oo : a contributed OOP system (Bengtsson, 2003) based on S3: method consistency, references, documentation facility,
- proto : class-less (prototype-based) OOP (Kates & Petzoldt, 2005): intentionally lightweight, delegation (prototype form of inheritance), references.

Questions:

- Is there a best OOP system for ecological modelling?
- Does OOP kill performance?
- Does end user code depend on the OOP selected?

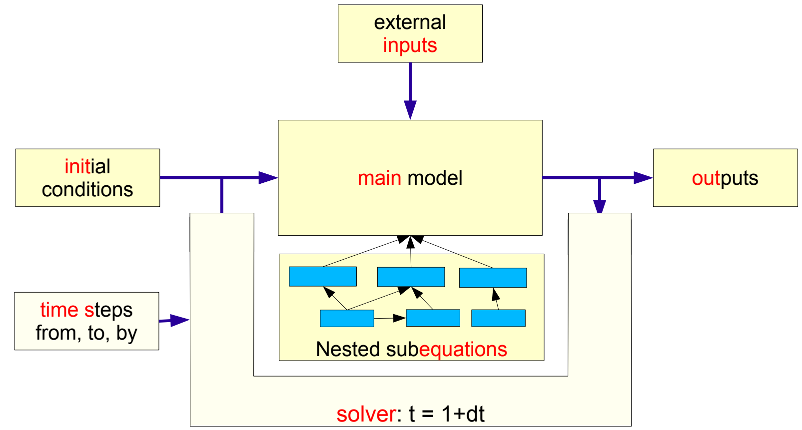
TECHNISCHE UNIVERSITÄT DRESDEN

OOP in Ecological Modelling

Petzoldt, Rinke, Kates

Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

Ecological models as state machine



TECHNISCHE UNIVERSITÄT DRESDEN

OOP in Ecological Modelling

Petzoldt, Rinke, Kates

Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

Characteristics of ecological models

- Tight relationship between methods (equations) and data
- Different types of data:
  - parameters (constants),
  - state variables,
  - input values,
  - time steps
- Different types of functional information
  - The main model
  - a set of (possibly nested) sub-models (sub-equations)
  - solvers, integrators, visualization (common within one model class)

TECHNISCHE UNIVERSITÄT DRESDEN

OOP in Ecological Modelling

Petzoldt, Rinke, Kates

Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

The SimObj model specification

simecol-package

S4 class: simObj	
main:	function
equations:	list of functions
parms:	data
times:	data
init:	data
inputs:	data
solver:	character
out:	data

- Generics:
  - `sim`, `plot`, `print`
  - get/set -functions
- Solvers
  - `lsoda`-wrapper, `rk4`
  - `iteration`
  - ...
- Utility functions
  - `approxTime`
  - `neighbours`

Implementation: S4 version

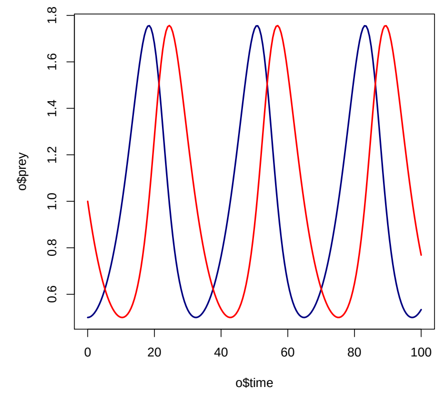
of the Lotka-Volterra model

```
lv <- new("OdeModel",
  main = function (time, init, parms)
    x <- init
    with(as.list(parms), {
      dx1 <-   b * x[1] - e * x[1] * x[2]
      dx2 <- - d * x[2] + e * x[1] * x[2]
      list(c(dx1, dx2))
    })
  ,
  ##          birth encounter death
  parms  = c(b=0.2, e=0.2, d=0.2),
  times  = seq(0, 100, 1),
  init   = c(prey=0.5, predator=1)
)
```

S3, S4, R.oo, proto: The model objects are quite similar.

TECHNISCHE UNIVERSITÄT DRESDEN
OOP in Ecological Modelling
Petzoldt, Rinke, Kates
Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

A short example




```
> library(simecol)
> data(lv)
> parms(lv)
 k1  k2  k3
0.2 0.2 0.2

> lv <- sim(lv) # pass-back modification
> plot(lv)
> o  <- out(lv)

> plot (o$time, o$prey,     col="navy", lwd=2, type="l")
> lines(o$time, o$predator, col="red",  lwd=2)
```

TECHNISCHE UNIVERSITÄT DRESDEN
OOP in Ecological Modelling
Petzoldt, Rinke, Kates
Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

A slightly more complex example . . .

. . . with sub-equations:

```
model <- list(
  main = function (equations, x)
  {
    dx1 <- f2(x[1], 0.1, 10)
  },
  equations = list(
    f1 = function(x, K)    K - x,
    f2 = function(x, r, K) r * x * f1(x, K)
  ),
  times  = seq(0, 10, 0.1),
  init   = c(x=0.5)
)
```

TECHNISCHE UNIVERSITÄT DRESDEN
OOP in Ecological Modelling
Petzoldt, Rinke, Kates
Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

More complex models:

Problems with scoping rules

- Lexical scoping in R
- Sub-equations assembled in a common structure (a list)
- How can these functions see each other ?
- Two possible approaches:
  - A) pass the whole object (or parts of it) down to the called function,
  - B) provide all necessary functions and data within a local environment.

A) Object Passing

```
eqA <- list(
  f1 = function(eq, x, K)    K - x,
  f2 = function(eq, x, r, K) r * x * eq$f1(eq, x, K)
)
solverA <- function(eq) {
  eq$f1(eq, 3, 4) + eq$f2(eq, 1, 2, 3)
}
solverA(eqA)
```

TECHNISCHE UNIVERSITÄT DRESDEN
OOP in Ecological Modelling
Petzoldt, Rinke, Kates
Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

A) Object Passing

```
eqA <- list(
  f1 = function(eq, x, K)    K - x,
  f2 = function(eq, x, r, K) r * x * eq$f1(eq, x, K)
)
solverA <- function(eq) {
  eq$f1(eq, 3, 4) + eq$f2(eq, 1, 2, 3)
}
solverA(eqA)
```


B) Temporary Environment

```
eqB <- list(
  f1 = function(x, K)    K - x,
  f2 = function(x, r, K) r * x * f1(x, K)
)
solverB <- function(eq) {
  eq <- putInEnv(eq, environment()) # a little trick
  f1(3,4) + f2(1,2,3)
}
solverB(eqB)
```

TECHNISCHE UNIVERSITÄT DRESDEN
OOP in Ecological Modelling
Petzoldt, Rinke, Kates
Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

Benchmarks . . . are more or less subjective

TECHNISCHE UNIVERSITÄT DRESDEN
OOP in Ecological Modelling
Petzoldt, Rinke, Kates
Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

Benchmarks . . . are more or less subjective

. . . and here is one:

Model	Size	nested	S3	S4	R.oo	proto	simecol	
Lotka-Volterra	small	no	3.5	3.6	3.6	3.9	3.7	(*a*)
Extended Lotka-Volterra	small	yes	4.8	4.8	4.9	5.1	4.8	(*b*)
DEB (bioenergetic Daphnia model)	medium	yes	2.8	2.8	2.9	3.0	2.7	(*c*)

Benchmarks . . . are more or less subjective

. . . and here is one:

Model	Size	nested	S3	S4	R.oo	proto	simecol	
Lotka-Volterra	small	no	3.5	3.6	3.6	3.9	3.7	(a)
Extended Lotka-Volterra	small	yes	4.8	4.8	4.9	5.1	4.8	(b)
DEB (bioenergetic Daphnia model)	medium	yes	2.8	2.8	2.9	3.0	2.7	(c)

Performance:

of OOPs quite equal (with ecological models !)

TECHNISCHE UNIVERSITÄT DRESDEN
OOP in Ecological Modelling
Petzoldt, Rinke, Kates
Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

Benchmarks . . . are more or less subjective

. . . and here is one:

Model	Size	nested	S3	S4	R.oo	proto	simecol	
Lotka-Volterra	small	no	3.5	3.6	3.6	3.9	3.7	(a)
Extended Lotka-Volterra	small	yes	4.8	4.8	4.9	5.1	4.8	(b)
DEB (bioenergetic Daphnia model)	medium	yes	2.8	2.8	2.9	3.0	2.7	(c)

Performance:

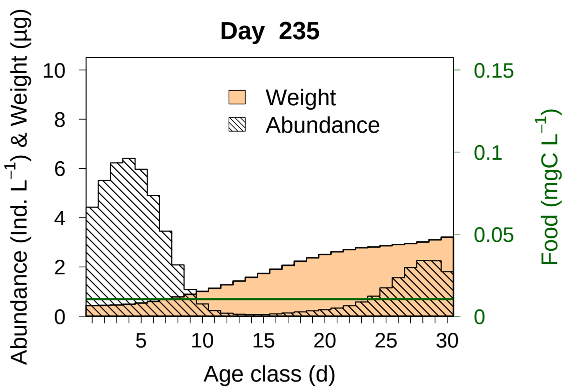
of OOPs quite equal (with ecological models !)

Reason:

- OOP used only to structure models.
- Excessive use of OOP features not necessary.
- Time consuming parts: variable assignments and numerics.

A practical application

Demographically structured population dynamics model of *Daphnia*



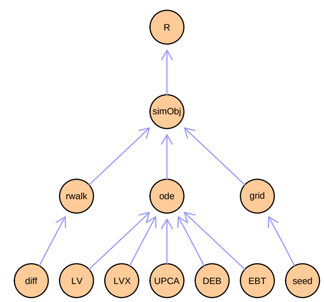
This model consists of two parts:

individual level: bioenergetic approach (differential equations)

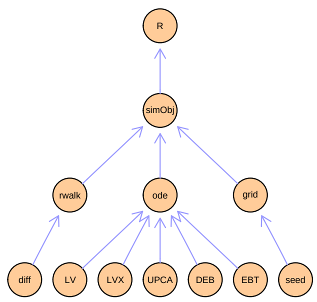
population level: discrete age-structure (cohort-based)

details, see Rinke & Vijverberg (2005)

Conclusion: Use R – and OOP



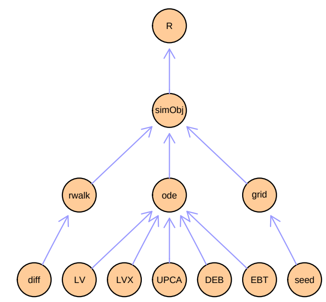
Conclusion: Use R – and OOP



- It's more important to use OOP at all than *the right* OOP.

TECHNISCHE UNIVERSITÄT DRESDEN
OOP in Ecological Modelling
Petzoldt, Rinke, Kates
Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

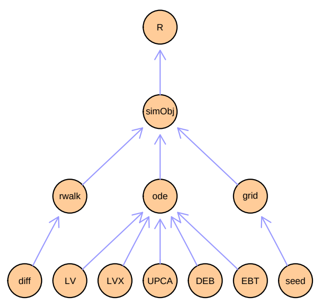
Conclusion: Use R – and OOP



- It's more important to use OOP at all than *the right* OOP.
- OOP helps to structure ecological models.
  R provides all mechanisms necessary.

TECHNISCHE UNIVERSITÄT DRESDEN
OOP in Ecological Modelling
Petzoldt, Rinke, Kates
Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

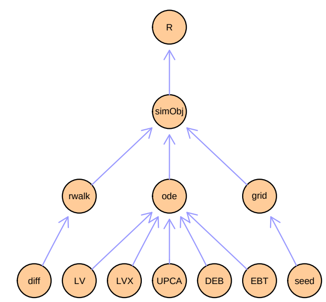
Conclusion: Use R – and OOP

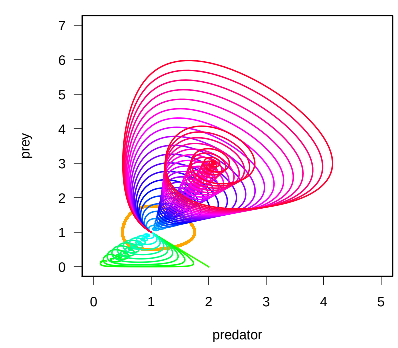


- It's more important to use OOP at all than *the right* OOP.
- OOP helps to structure ecological models.
  R provides all mechanisms necessary.
- The proposed OOP structure works –
  without and with simecol.

TECHNISCHE UNIVERSITÄT DRESDEN
OOP in Ecological Modelling
Petzoldt, Rinke, Kates
Motivation
Problem
Workflow
Basic idea
Approach
OOP in R
State Machine
What's typical?
simObj
Implementation
Example I
Example II
Scoping
Nesting
Benchmark
Application
Conclusions

Conclusion: Use R – and OOP





- It's more important to use OOP at all than *the right* OOP.
- OOP helps to structure ecological models.
  R provides all mechanisms necessary.
- The proposed OOP structure works –
  without and with simecol.

Bengtsson, H., 2003: The R.oo package – object-oriented programming with references using standard R code. *In*: K. Hornik, F. Leisch, & A. Zeileis (eds.), Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vienna, Austria, http://www.maths.lth.se/help/R/R.oo/.

Chambers, J. M., 1998: Programming with Data: A Guide to the S Language. Springer-Verlag, New York.

Kates, L. & T. Petzoldt, 2005: The R Proto Package. Package vignette of the CRAN proto package and http://hhbio.wasser.tu-dresden.de/projects/proto/.

Rinke, K. & J. Vijverberg, 2005: A model approach to evaluate the effect of temperature and food concentration on individual life-history and population dynamics of *Daphnia*. Ecological Modelling 186: 326–344.

Additional slides for discussion.

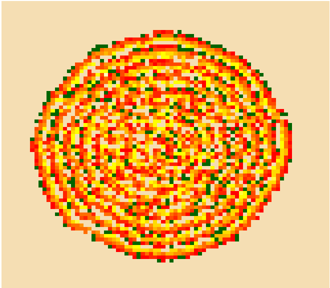
Pass the equation object down where it is needed.

```
sim <- function(obj) {
  x <- matrix(NA, length(obj$times), length(obj$init))
  x[1,] <- obj$init;
  dt    <- diff(obj$times)
  for (i in 2:length(obj$times)) {
    x[i,] <- x[i-1,] + obj$main(obj$equations, x[i-1, ]) * dt[i-1]
  }
  obj$out <- x
  obj
}

model <- list(
  main = function (equations, x)
  {
    dx1 <- equations$eq1(equations, x[1], 0.1, 10)
  },
  equations = list(
    eq1 = function(this, x, r, K) r * x * this$f(x, K),
    f   = function(x, K) (K - x)),
  times   = seq(0, 10, 0.1),
  init    = c(x=0.5)
)

model <- sim(model)
plot(model$times, model$out[,1], type="l")
```


Stochastic cellular automaton



```
source("http://www.simecol.de/data/ca.R")
times(CA) <- c(to=80)
sim(CA, animate=TRUE, col=mycolors(20), axes=F)
```

```
putInEnv <- function(eq, e) {
  ## clone, very important to avoid "interferences"!!!
  eq <- as.list(unlist(eq))
  lapply(eq, "environment<-", e)
  nn <- names(eq)
  for (i in 1:length(eq)) {
    assign(nn[i], eq[[i]], envir = e)
  }
  eq
}

eqB <- list(
  f1 = function(x, y)    x + y,
  f2 = function(a, x, y) a * f1(x, y)
)
solverB <- function(eq) {
  eq <- putInEnv(eq, environment())
  f1(3,4) + f2(1,2,3)
}
```