WBDev

- WinBUGS Development Interface
  - – Possible to code your own compiled functions and univariate distributions.
  - – These are then used as
    - x <- myfunction(a,b,c)
    - y ~ dmydistribution(a,b,c)
  - – Because they are precompiled, they are fast to compute (compared to equivalent constructions within BUGS syntax).

WBDev

- Go to WB development site from the WinBUGS website.
- Download WBDev
  - Save the file (**wbdev.exe**) to *any* location on your computer. The saved file is a self-extracting archive that when double-clicked (and subsequently unzipped) will deposit a file called **wbdev_01_09_04.txt** into your

    Program Files/WinBUGS14 directory.
  - To install the patch open the **wbdev_01_09_04.txt** file from within **WinBUGS 1.4** and follow the instructions at the top of the file.
  - After installation, documentation for this version of the WBDev interface can be found in the files:

    Program Files/WinBUGS14/WBDev/Docu/WBDev_distributions.pdf and

    Program Files/WinBUGS14/WBDev/Docu/WBDev_functions.pdf

WBDev

- Download *BlackBox Component Builder* from the following web-page:

  http://www.oberon.ch/blackbox.html

- Install ‘BlackBox’ by double-clicking on the Setup.exe icon and following the instructions. The software should be installed into the new directory

  Program Files/BlackBox.

WBDev

- Go to the Program Files/WinBUGS14 directory; then press Ctrl+A to select all files and sub-directories within the WinBUGS14 directory. Press Ctrl+C to copy them.
- Go to the Program Files/BlackBox directory and then press Ctrl+V to paste them here.
- Now your copy of BlackBox should include the *full* functionality of WinBUGS 1.4 (including WBDev interface) within it.

WBDev

- There are templates for making your own functions and distributions – follow the instructions through the example there.
- Only some parts (in blue) of the templates are meant to be changed.
- Language: Component Pascal, but some previous programming experience in any (PROCEDURAL) language will help.

WBDev

Example: Poisson process model with piecewise constant intensity
(K unknown change points and K+1 unknown levels). The BUGS code could be:

```
model{
 for(year in 1:N){
 T[year] <- year+1850
 D[year]~dpois(mu[year])
 mu[year] <- Piecewiseconst(year,K,levels[],points[])
 }
 loglevels[1] ~ dnorm(0,0.001);  log(levels[1]) <- loglevels[1];
 for(k in 2:K+1){ levels[k] ~  dgamma(a[k],b); a[k] <- levels[k-1]*b }
 b ~ dgamma(0.01,0.01)

 theta ~ dgamma(1,1); thetaplus <- theta+0.1 # to ensure 0< E(pts)<10
 for(k in 1:K){pts[k] ~ dexp(thetaplus)}
 points[1] <- pts[1]
 for(k in 2:K){points[k]<-points[k-1]+pts[k] }
}
```

WBDev

- Describe the new function for BlackBox in a file ”functions.odc” under WBDev/Rsrc/

```
s <- "Piecewiseconst"(s, s, v, v)       "WBDevPiecewiseconst.Install"
END
```

- This specifies its name, input arguments and output.

WBDev

- Write the Component Pascal code of the new function in a file ”Piecewiseconst.odc” under WBDev/Mod/. (Use function templates to modify them). Compile it in BlackBox. Quit BlackBox and start it again.

- Now the BUGS model should run with this new function in BlackBox.
- Also possible to run from R, by using R2WinBUGS and an appropriate file path to BlackBox given as an argument.