Representing JULES models in XML

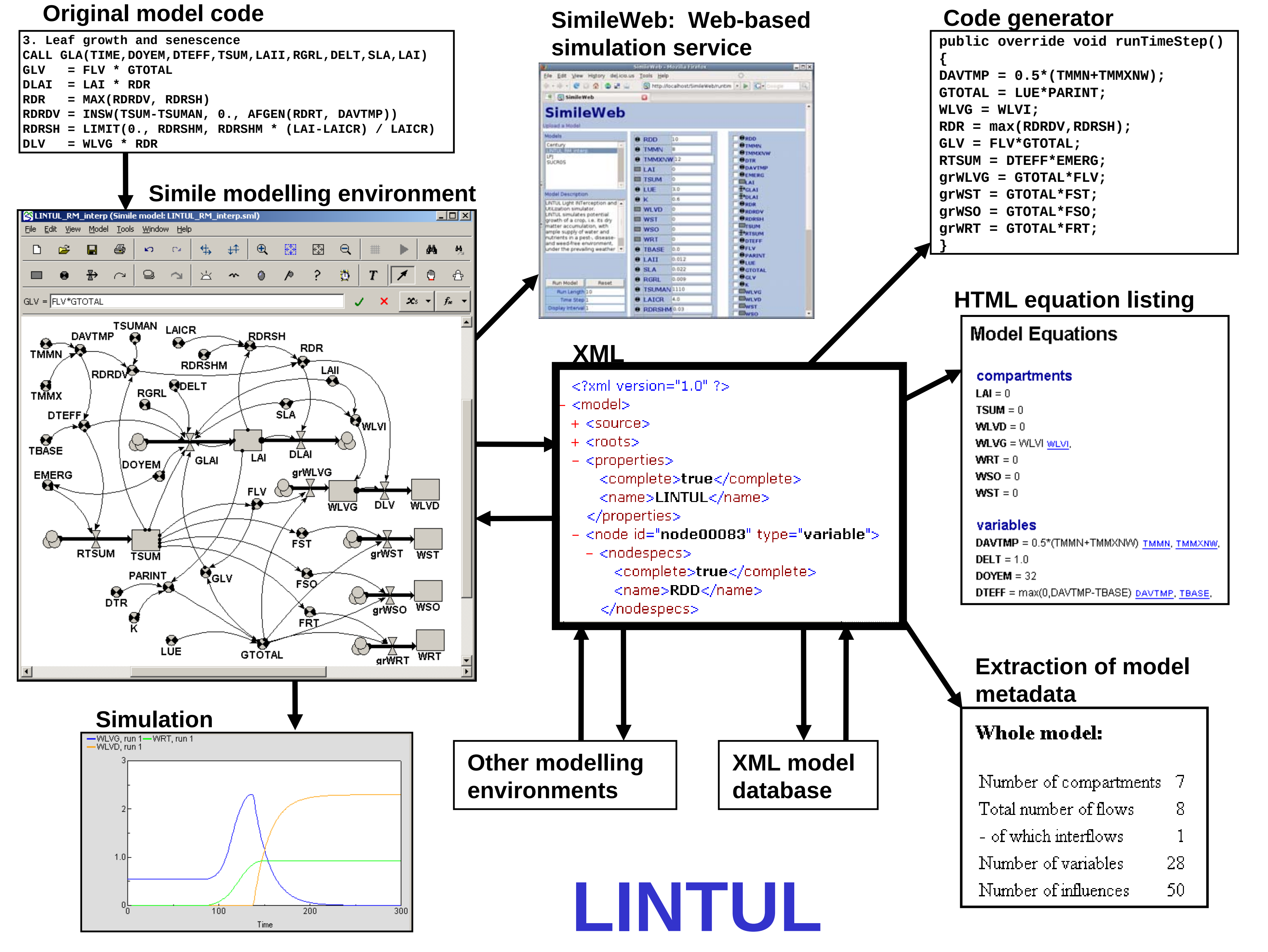
Robert Muetzelfeldt

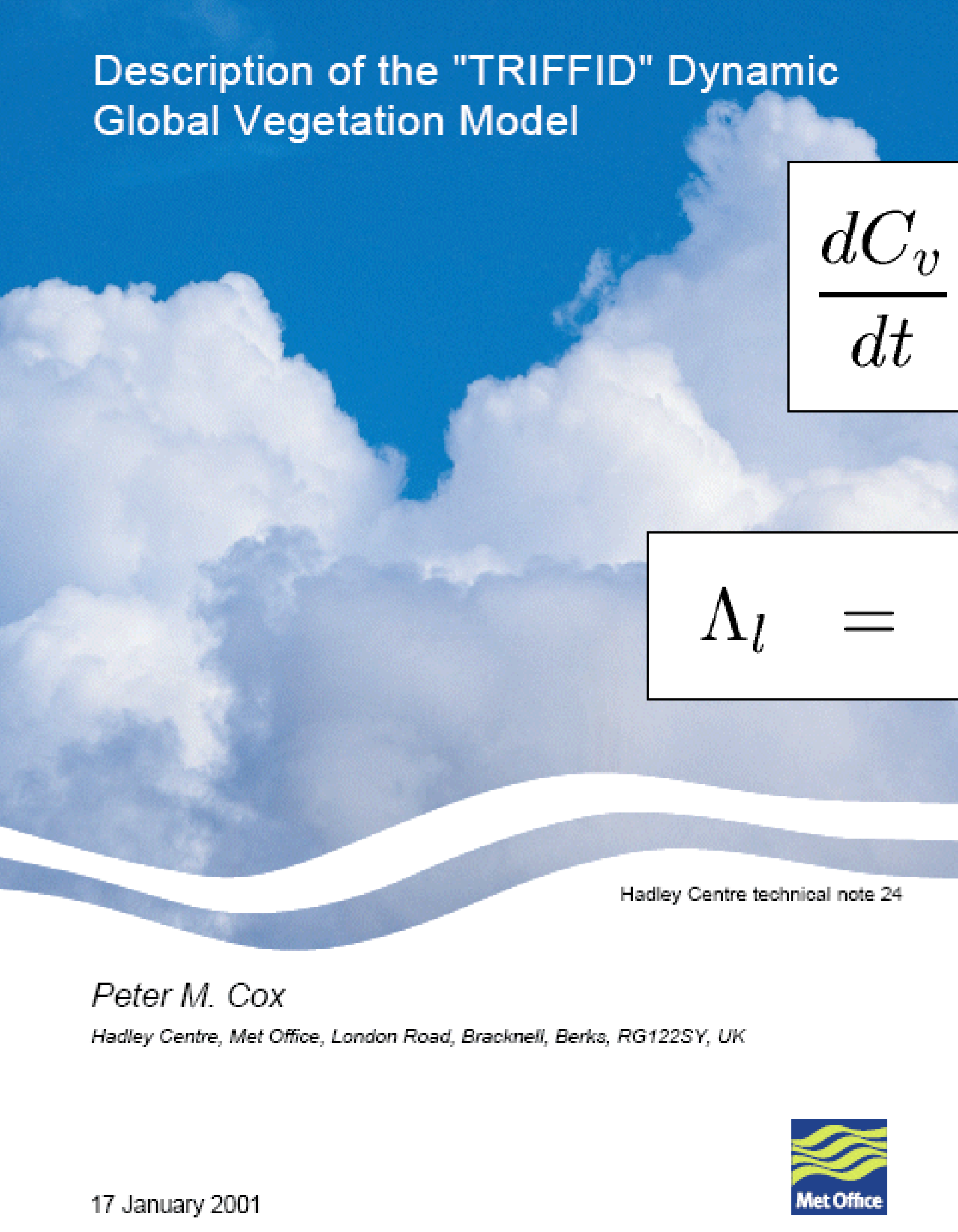
University of Edinburgh
School of Informatics
and
Simulistics Ltd

robertm@ed.ac.uk

The challenge

Is it possible to replace the hand-coded JULES components (i.e. subroutines) with code generated automatically from model equations?


Original model code
3. Leaf growth and senescence
CALL GLA(TIME,DOYEM,DTEFF,TSUM,LAII,RGRL,DELT,SLA,LAI)
GLV   = FLV * GTOTAL
DLAI = LAI * RDR
RDR   = MAX(RDRDV, RDRSH)
RDRDV = INSW(TSUM-TSUMAN, 0., AFGEN(RDRT, DAVTMP))
RDRSH = LIMIT(0., RDRSHM, RDRSHM * (LAI-LAICR) / LAICR)
DLV   = WLVG * RDR
Simile modelling environment
LINTUL_RM_interp (Simile model: LINTUL_RM_interp.sml)
File Edit View Model Tools Window Help
GLV = FLV*GTOTAL
TSUMAN
DAVTMP
LAICR
RDRSH
TMMN
RDR
RDRSHM
RDRDV
LAII
DELT
TMMX
RGRL
SLA
DTEFF
WLVI
TBASE
GLAI
LAI
DLAI
DOYEM
EMERG
grWLVG
FLV
WLVG
DLV
WLVD
RTSUM
TSUM
FST
grWST
WST
PARINT
GLV
FSO
DTR
grWSO
WSO
K
FRT
LUE
GTOTAL
grWRT
WRT
Simulation
WLVG, run 1
WRT, run 1
WLVD, run 1
3
2
1.0
0
0
100
200
300
Time
SimileWeb: Web-based simulation service
SimileWeb
Code generator
public override void runTimeStep()
{
DAVTMP = 0.5*(TMMN+TMMXNW);
GTOTAL = LUE*PARINT;
WLVG = WLVI;
RDR = max(RDRDV,RDRSH);
GLV = FLV*GTOTAL;
RTSUM = DTEFF*EMERG;
grWLVG = GTOTAL*FLV;
grWST = GTOTAL*FST;
grWSO = GTOTAL*FSO;
grWRT = GTOTAL*FRT;
}
XML
<?xml version="1.0" ?>
- <model>
+ <source>
+ <roots>
- <properties>
<complete>true</complete>
<name>LINTUL</name>
</properties>
- <node id="node00083" type="variable">
- <nodespecs>
<complete>true</complete>
<name>RDD</name>
</nodespecs>
HTML equation listing
Model Equations
compartments
LAI = 0
TSUM = 0
WLVD = 0
WLVG = WLVI WLVI,
WRT = 0
WSO = 0
WST = 0
variables
DAVTMP = 0.5*(TMMN+TMMXNW) TMMN, TMMXNW,
DELT = 1.0
DOYEM = 32
DTEFF = max(0,DAVTMP-TBASE) DAVTMP, TBASE,
Extraction of model metadata
Whole model:
Number of compartments 7
Total number of flows 8
- of which interflows 1
Number of variables 28
Number of influences 50
Other modelling environments
XML model database
LINTUL



Description of the "TRIFFID" Dynamic Global Vegetation Model

Hadley Centre technical note 24

Peter M. Cox

Hadley Centre, Met Office, London Road, Bracknell, Berks, RG122SY, UK

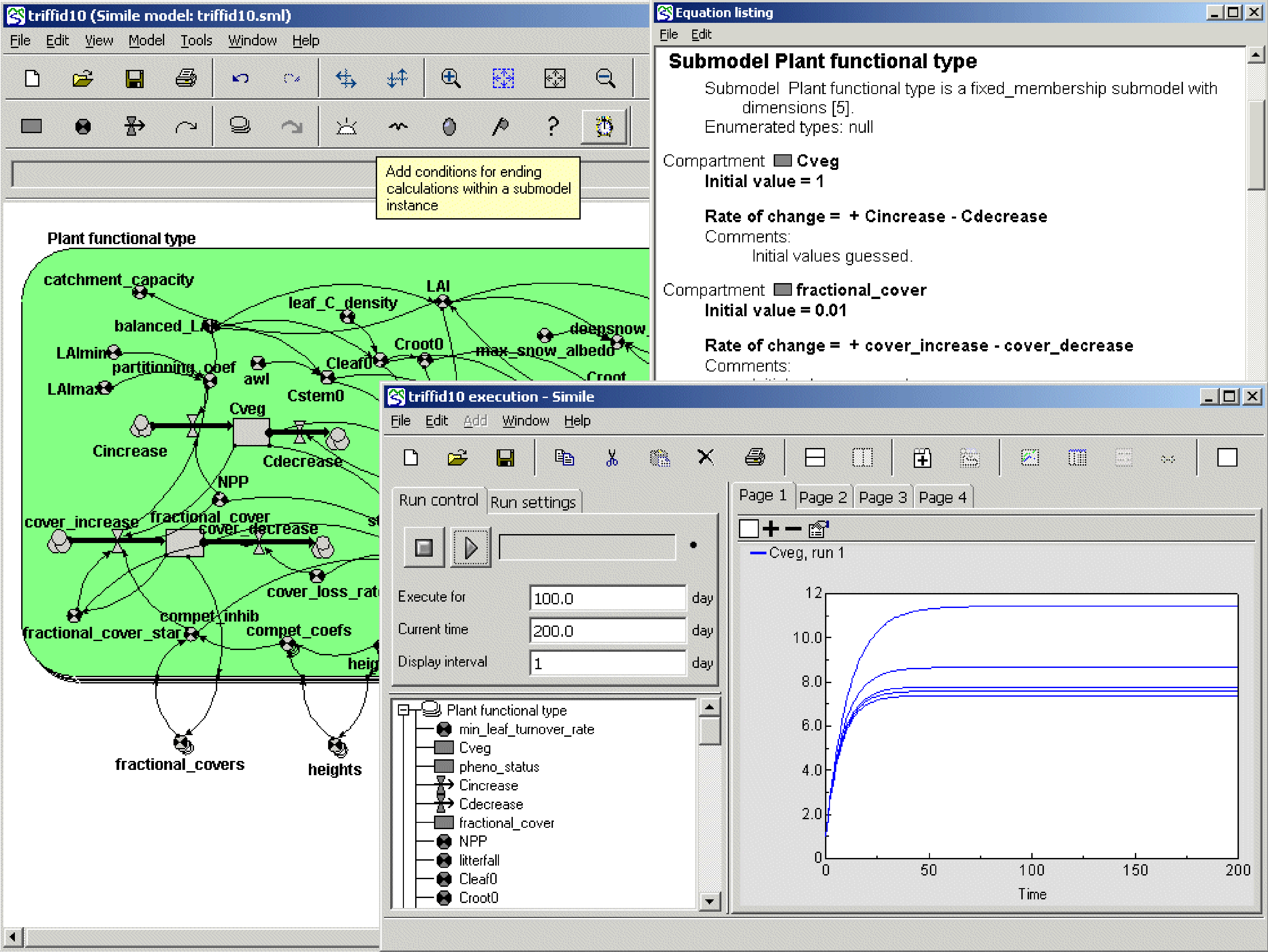
17 January 2001

Met Office

$$\frac{dC_v}{dt} = (1-\lambda)\,\Pi - \Lambda_l$$

$$\Lambda_l = \gamma_l\,\mathcal{L} + \gamma_r\,\mathcal{R} + \gamma_w\,\mathcal{W}$$

```
|              |-- sice_htf
|
|-- hydrol ---|
|              |-- sfsnow
|              |-- surf_hyd --|
|              |               |-- frunoff
|              |               |-- sieve
|              |               |-- pdm
|              |-- calc_baseflow
|              |-- soil_hyd --|
|              |               |-- hyd_con(_vg)
|              |               |-- darcy(_vg) --|
|              |               |                 |-- hyd_con(_vg)
|              |               |--gauss
|              |-- calc_fsat
|              |-- soil_htc --|
|              |               |-- heat_con
|              |               |-- gauss
|              |-- ice_htc
|              |-- soilmc
|              |-- soilt
|              |-- ch4_wetl
|-- veg2 --|
|           |-- tilepts
|           |-- phenol
|           |-- triffid --|
|           |              |-- vegcarb --|
|           |              |              |-- growth
|           |              |-- lotka --|
|           |              |            |-- compete
|           |              |
|           |              |-- soilcarb --|
|           |              |               |--decay
|           |-- tilepts
|           |-- sparm --|
|                        |-- pft_sparm
|-- veg1 --|
```


triffid10 (Simile model: triffid10.sml)
File Edit View Model Tools Window Help
Add conditions for ending calculations within a submodel instance
Plant functional type
catchment_capacity
LAI
leaf_C_density
balanced_LAI
deepsnow
LAImin
Croot0
max_snow_albedo
partitioning_coef
Cleaf0
awl
Croot
LAImax
Cstem0
Cveg
Cincrease
Cdecrease
NPP
cover_increase
fractional_cover
cover_decrease
cover_loss_rat
compet_inhib
fractional_cover_star
compet_coefs
heig
fractional_covers
heights
Equation listing
File Edit
Submodel Plant functional type
Submodel Plant functional type is a fixed_membership submodel with dimensions [5].
Enumerated types: null
Compartment Cveg
Initial value = 1
Rate of change = + Cincrease - Cdecrease
Comments:
Initial values guessed.
Compartment fractional_cover
Initial value = 0.01
Rate of change = + cover_increase - cover_decrease
Comments:
triffid10 execution - Simile
File Edit Add Window Help
Run control
Run settings
Execute for 100.0 day
Current time 200.0 day
Display interval 1 day
Plant functional type
min_leaf_turnover_rate
Cveg
pheno_status
Cincrease
Cdecrease
fractional_cover
NPP
litterfall
Cleaf0
Croot0
Page 1
Page 2
Page 3
Page 4
Cveg, run 1
12
10.0
8.0
6.0
4.0
2.0
0
0
50
100
150
200
Time

http://similette.com/models/triffid/triffid10v1.xml - mirs Internet Explorer

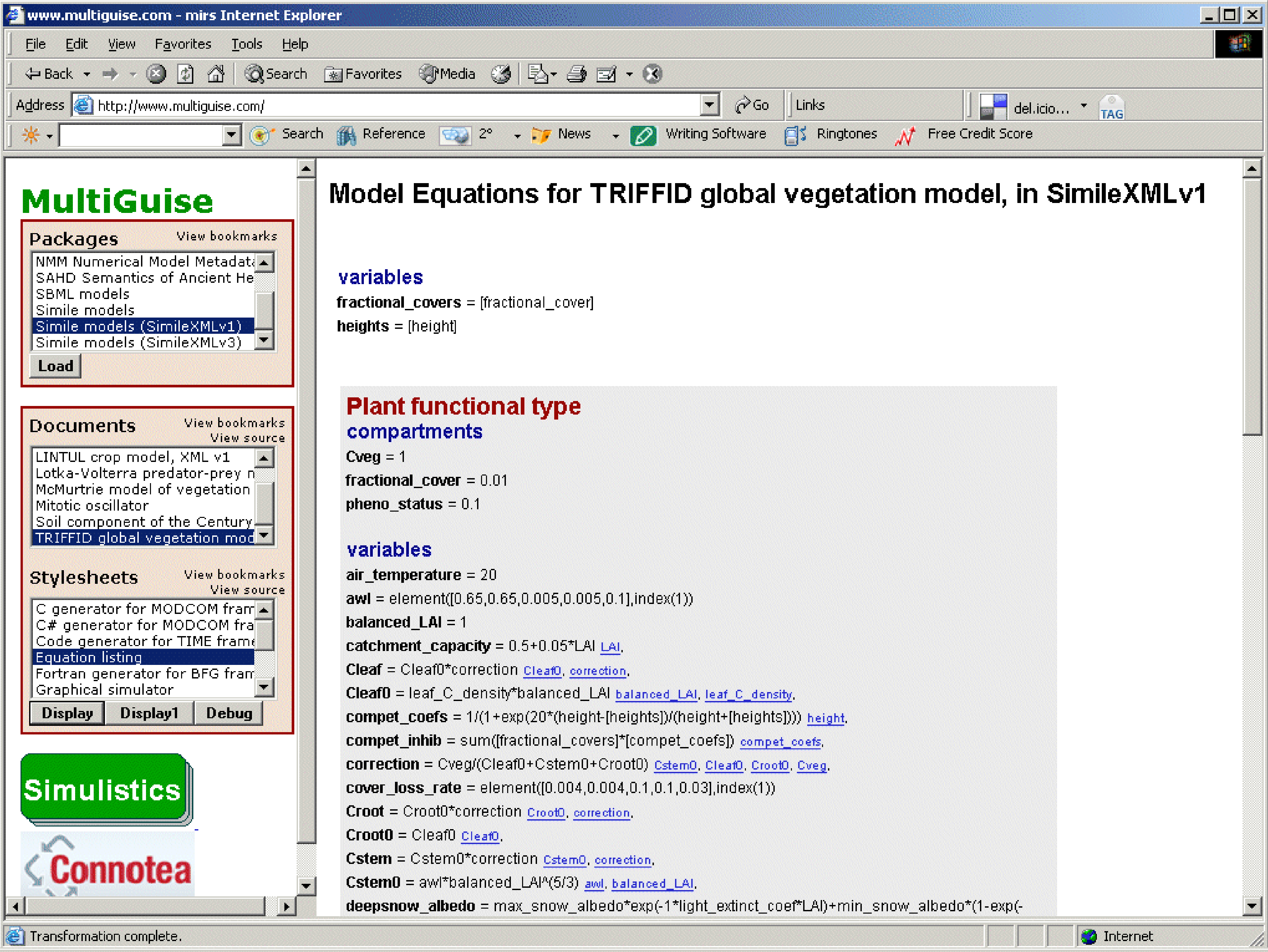
File Edit View Favorites Tools Help

Back Search Favorites Media

Address http://similette.com/models/triffid/triffid10v1.xml Go Links del.icio... TAG


```
<?xml version="1.0" ?>
- <model xmlns:m="http://www.w3.org/1998/Math/MathML">
  + <source>
  + <roots>
  - <properties>
      <complete>true</complete>
      <name>triffid10</name>
      <title>TRIFFID global vegetation model, in SimileXMLv1</title>
      <description>The Hadley Centre's TRIFFID model couples a photosynthesis model (Cox et al. (1998)) to a population
        model (in the ecological sense). The population model updates the fractional coverage and height of vegetation
        depending on the predicted photosyn- thesis. This population model therefore plays a large role in the dynamic
        properties of the land surface.</description>
    </properties>
  - <submodel id="top">
    + <graphics>
    + <nodes>
    + <arcs>
    - <submodel id="node00002">
      + <infos>
      + <graphics>
      - <nodes>
        + <clouds>
        - <compartments>
          - <compartment id="node00221">
            - <infos>
                <complete>true</complete>
                <name>Cveg</name>
              </infos>
            + <graphics>
            </compartment>
          - <compartment id="node00223">
            - <infos>
                <complete>true</complete>
                <name>pheno_status</name>
              </infos>
```


Internet



Model Equations for TRIFFID global vegetation model, in SimileXMLv1

variables

fractional_covers = [fractional_cover]

heights = [height]

Plant functional type

compartments

Cveg = 1

fractional_cover = 0.01

pheno_status = 0.1

variables

air_temperature = 20

awl = element([0.65,0.65,0.005,0.005,0.1],index(1))

balanced_LAI = 1

catchment_capacity = 0.5+0.05*LAI LAI,

Cleaf = Cleaf0*correction Cleaf0, correction,

Cleaf0 = leaf_C_density*balanced_LAI balanced_LAI, leaf_C_density,

compet_coefs = 1/(1+exp(20*(height-[heights])/(height+[heights]))) height,

compet_inhib = sum([fractional_covers]*[compet_coefs]) compet_coefs,

correction = Cveg/(Cleaf0+Cstem0+Croot0) Cstem0, Cleaf0, Croot0, Cveg,

cover_loss_rate = element([0.004,0.004,0.1,0.1,0.03],index(1))

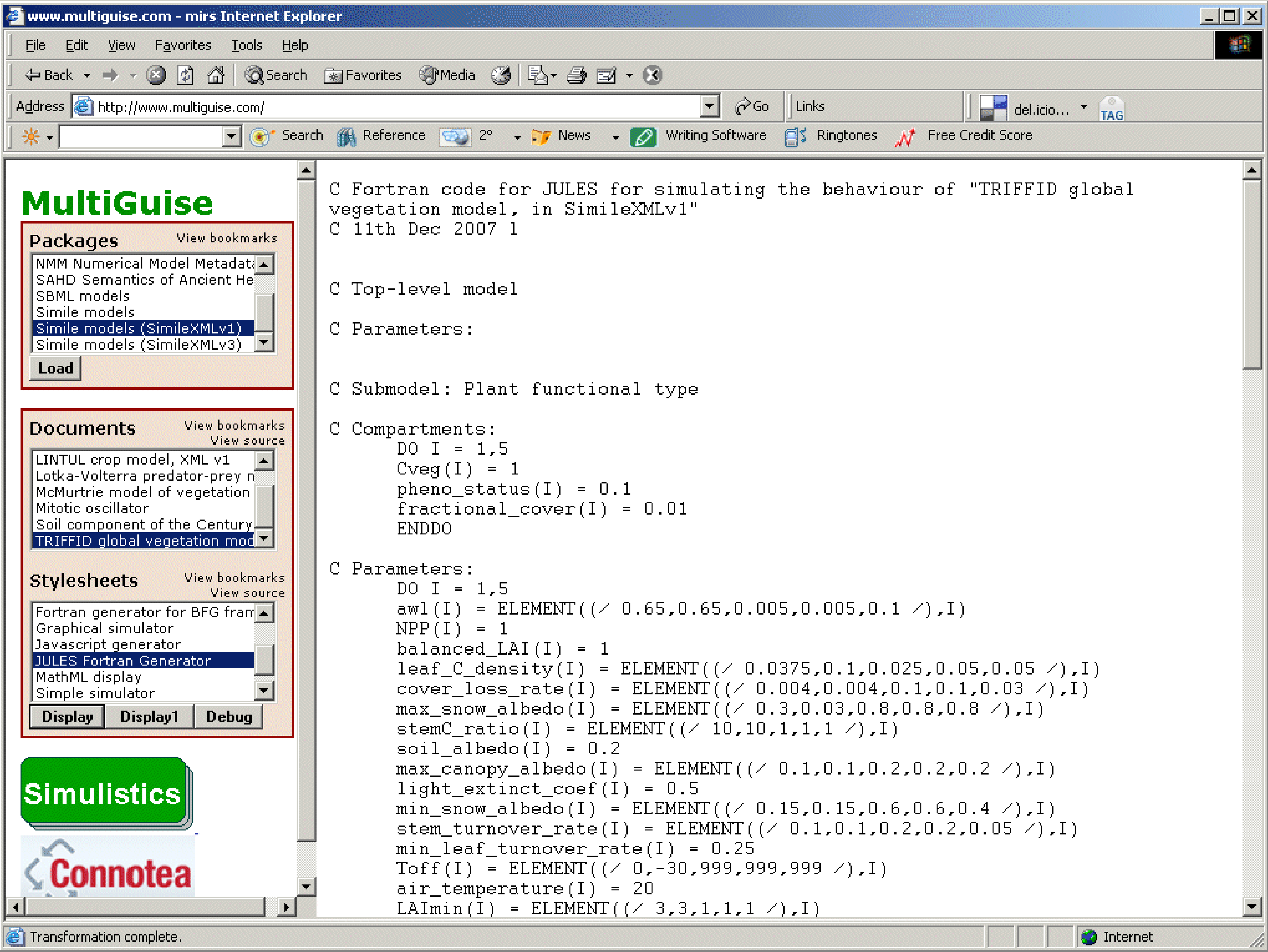
Croot = Croot0*correction Croot0, correction,

Croot0 = Cleaf0 Cleaf0,

Cstem = Cstem0*correction Cstem0, correction,

Cstem0 = awl*balanced_LAI^(5/3) awl, balanced_LAI,

deepsnow_albedo = max_snow_albedo*exp(-1*light_extinct_coef*LAI)+min_snow_albedo*(1-exp(-


www.multiguise.com - mirs Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.multiguise.com/
Go Links del.icio... TAG
Search Reference 2° News Writing Software Ringtones Free Credit Score
MultiGuise
Packages View bookmarks
NMM Numerical Model Metadat
SAHD Semantics of Ancient He
SBML models
Simile models
Simile models (SimileXMLv1)
Simile models (SimileXMLv3)
Load
Documents View bookmarks View source
LINTUL crop model, XML v1
Lotka-Volterra predator-prey n
McMurtrie model of vegetation
Mitotic oscillator
Soil component of the Century
TRIFFID global vegetation mod
Stylesheets View bookmarks View source
Fortran generator for BFG fram
Graphical simulator
Javascript generator
JULES Fortran Generator
MathML display
Simple simulator
Display Display1 Debug
Simulistics
Connotea
C Fortran code for JULES for simulating the behaviour of "TRIFFID global vegetation model, in SimileXMLv1"
C 11th Dec 2007 1
C Top-level model
C Parameters:
C Submodel: Plant functional type
C Compartments:
DO I = 1,5
Cveg(I) = 1
pheno_status(I) = 0.1
fractional_cover(I) = 0.01
ENDDO
C Parameters:
DO I = 1,5
awl(I) = ELEMENT((/ 0.65,0.65,0.005,0.005,0.1 /),I)
NPP(I) = 1
balanced_LAI(I) = 1
leaf_C_density(I) = ELEMENT((/ 0.0375,0.1,0.025,0.05,0.05 /),I)
cover_loss_rate(I) = ELEMENT((/ 0.004,0.004,0.1,0.1,0.03 /),I)
max_snow_albedo(I) = ELEMENT((/ 0.3,0.03,0.8,0.8,0.8 /),I)
stemC_ratio(I) = ELEMENT((/ 10,10,1,1,1 /),I)
soil_albedo(I) = 0.2
max_canopy_albedo(I) = ELEMENT((/ 0.1,0.1,0.2,0.2,0.2 /),I)
light_extinct_coef(I) = 0.5
min_snow_albedo(I) = ELEMENT((/ 0.15,0.15,0.6,0.6,0.4 /),I)
stem_turnover_rate(I) = ELEMENT((/ 0.1,0.1,0.2,0.2,0.05 /),I)
min_leaf_turnover_rate(I) = 0.25
Toff(I) = ELEMENT((/ 0,-30,999,999,999 /),I)
air_temperature(I) = 20
LAImin(I) = ELEMENT((/ 3,3,1,1,1 /),I)
Transformation complete.
Internet

Initial section

```
C Submodel: Plant functional type
C Compartments:
      DO I = 1,5
      Cveg(I) = 1
      pheno_status(I) = 0.1
      fractional_cover(I) = 0.01
      ENDDO
C Parameters:
      DO I = 1,5
      awl(I) = ELEMENT((/ 0.65,0.65,0.005,0.005,0.1 /),I)
      NPP(I) = 1
      balanced_LAI(I) = 1
      leaf_C_density(I) = ELEMENT((/ 0.0375,0.1,0.025,0.05,0.05 /),I)
      max_snow_albedo(I) = ELEMENT((/ 0.3,0.03,0.8,0.8,0.8 /),I)
      stemC_ratio(I) = ELEMENT((/ 10,10,1,1,1 /),I)
      soil_albedo(I) = 0.2
      air_temperature(I) = 20
      LAImin(I) = ELEMENT((/ 3,3,1,1,1 /),I)
      LAImax(I) = ELEMENT((/ 9,9,4,4,4 /),I)
      leaf_drop_rate(I) = 20
      . . . . . . . .
      ENDDO

C Submodel: Soil carbon
C Compartments:
      Csoil = 1
C Parameters:
      soil_temperature = 10
      soil_moisture = 0.5
      specific_soil_resp = 5*10^ -9
      wilting_soil_moisture = 0.2
      saturation_soil_moisture = 1.0
```

Dynamic section

```
Cincrease = (1-partitioning_coef)*NPP
cover_decrease = cover_loss_rate*fractional_cover_star
IF leaf_mortality>2*min_leaf_turnover_rate THEN dpdt = -
   1*leaf_drop_rate ELSE dpdt = leaf_drop_rate*(1-pheno_status)
IF soil_moisture_fract>optimum_soil_moisture_fract THEN moisture_mult
   = 1-0.8*(soil_moisture_fract-optimum_soil_moisture_fract) ELSE
   moisture_mult = IF soil_moisture_fract <= wilting_soil_moisture_fract THEN
   moisture_mult = 0.2 ELSE moisture_mult = 0.2+0.8*(soil_moisture_fract-
   wilting_soil_moisture_fract)/(optimum_soil_moisture_fract-
   wilting_soil_moisture_fract)
DO I=1,5
correction(I) = Cveg(I)/(Cleaf0(I)+Cstem0(I)+Croot0(I))
ENDDO
microbial_respiration = specific_soil_resp*Csoil*temp_mult*moisture_mult
DO I=1,5
Cstem(I) = Cstem0(I)*correction(I)
Cleaf(I) = Cleaf0(I)*correction(I)
Croot(I) = Croot0(I)*correction(I)
litterfall(I) = Cstem(I)*stem_turnover_rate(I)+Cleaf(I)*leaf_turnover_rate(I)+
Croot(I)*root_turnover_rate(I)
live_stemC(I) = Cstem(I)/stemC_ratio(I)
height(I) = live_stemC(I)/(0.01*LAI(I))
ENDDO
Cdecrease = litterfall
DO I=1,5
IF I<=2 THEN roughness_length_momentum(I) = 0.05*height(I) ELSE
   roughness_length_momentum(I) = 0.1*height(I)
roughness_length_scalars(I) = 0.1*roughness_length_momentum(I)
ENDDO
```

Integration section

```
DO I=1,5
Cveg(I) = Cveg(I) + timestep*( +Cincrease(I) -Cdecrease(I) )
pheno_status(I) = pheno_status(I) + timestep*( +dpdt(I) )
fractional_cover(I) = fractional_cover(I) + timestep*(
   +cover_increase(I) -cover_decrease(I) )
ENDDO

Csoil = Csoil + timestep*( +total_litterfall -
   microbial_respiration)
```

Assessment

Feasible to re-cast at least some JULES 'modules' in XML.

Main constraint at the moment is the nature of the JULES implementation.

Re-implementation in an Modelling Framework would be a huge improvement.
Could retain much of the existing 'science' code.

Modelling Frameworks

In the UK Earth Systems community:
- FLUME (son of UM)
- GENIE
- SoftIAM
and abroad:
- ESMF
- PRISM / OASIS
- and many more...

FLUME, Genie and SoftIAM are realisations of BFG: the Bespoke Framework Generator

“BFG isolates the science that a model performs from the code used to control it and couple it with other models.”

“This is useful as it promotes the idea of scientists concentrating on the science rather than the computer science.”

“It also allows the flexible deployment of the coupled model onto different targets with no change to the model code.”

All model metadata and specification of compositions is in XML.

A possible future

JULES 'modules' (aka 'subroutines'?) become BFG 'models', and JULES becomes a BFG 'composition'.

Individual models can then be:

EITHER edited to make them BFG-compliant Fortran;

OR re-cast in XML, and used to generated BFG-compliant Fortran.

```
!Root structural pools (kg structural DM m-2).
!Differential equations (kg structural DM m-2 day-1):
 DMrt1 = plantvr * ( IXrt1 - OXrt1 )
 DMrt2 = plantvr * ( OXrt1_2 - OXrt2 )
 DMrt3 = plantvr * ( OXrt2_3 - OXrt3 )
 DMrt4 = plantvr * ( OXrt3_4 - OXrt4 )
 OXrt1 = OXrt1_2 + OXrt1_deg
 OXrt2 = OXrt2_3 + OXrt2_deg
 OXrt3 = OXrt3_4 + OXrt3_deg
 OXrt4 = OXrt4_li + OXrt4_deg
 Mrt1 = INTEG ( DMrt1, Mrt1ic )
 Mrt2 = INTEG ( DMrt2, Mrt2ic )
 Mrt3 = INTEG ( DMrt3, Mrt3ic )
 Mrt4 = INTEG ( DMrt4, Mrt4ic )
```

ACSL

Hurley Pasture Model fragment

FST

LINTUL fragment

```
* 2.9 Growth of plant organs and translocation
 ASRQ = FSH * (ASRQLV*FLV + ASRQST*FST + ASRQSO*FSO) + ASRQRT*FRT
 TRANSL = INSW(DVS-1., 0., WST * DVR * FRTRL)
 GTW = (GPHOT - MAINT + CONVL*TRANSL*CFST*30./12.) / ASRQ
 GRT = FRT * GTW
 GLV = FLV * FSH * GTW
 GST = FST * FSH * GTW - TRANSL
 GSO = FSO * FSH * GTW

* 2.11 Dry matter production
 WRT = INTGRL(WRTI, GRT)
 WLVG = INTGRL(WLVI, RWLVG)
 RWLVG = GLV - DLV
 WLVD = INTGRL(WLVDI, DLV)
 WST = INTGRL(WSTI, GST)
 WSO = INTGRL(WSOI, GSO)
```