GroIMP v1.4.2
Introduction and Overview

Michael Henke

Department Ecoinformatics, Biometrics and Forest Growth, University of Göttingen, Germany

International Summer School
”Modelling of Ecosystems by Tools from Computer Science”

Prague, 16.09.2013

GroIMP

Overview I

Paradigm(s)	multi-paradigm: imperative, object-oriented, rule-based, chemical
Appeared in	2003
Designed by	Ole Kniemeyer
Stable release	1.4.2 (June 06, 2013)
Typing discipline	static, strong, safe
Influenced by	GROGRA
Operating System	Cross-platform, (multi-platform - JVM)
License	GNU General Public License V.3
Usual filename extensions	gsz, rgg, xl
Available at	sourceforge.net

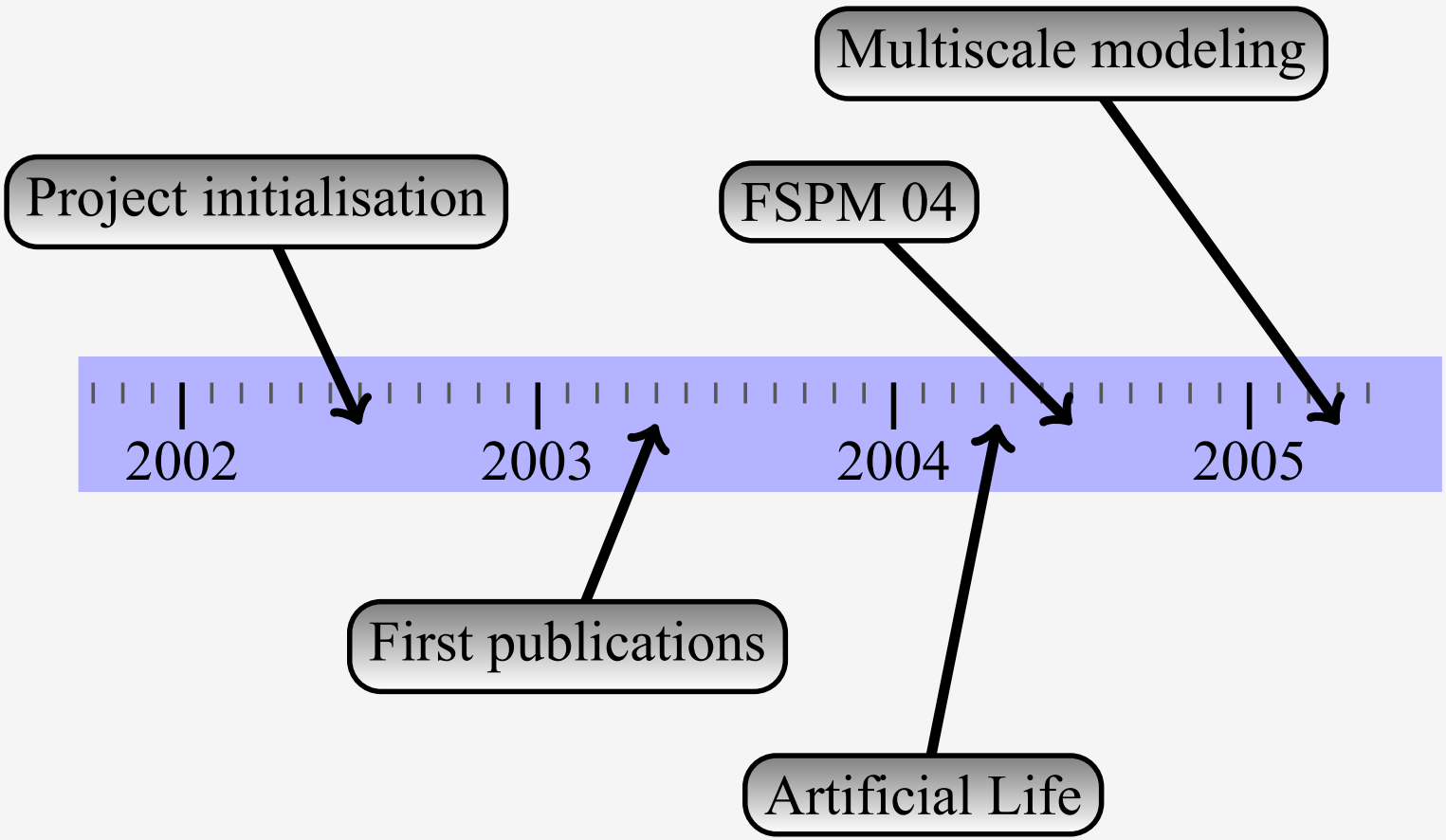
GroIMP
Overview II

Growth-grammar related **I**nteractive **M**odelling **P**latform

- Objectives
  - 3D modelling software specialized for FSPMs
- Target groups
  - Students (e.g. biology, agronomy, horticulture)
  - Scientists
  - Modeller
- Application fields
  - Educational tool
  - (Experimental) Research & Science
  - Decision-support tool
  - Industry

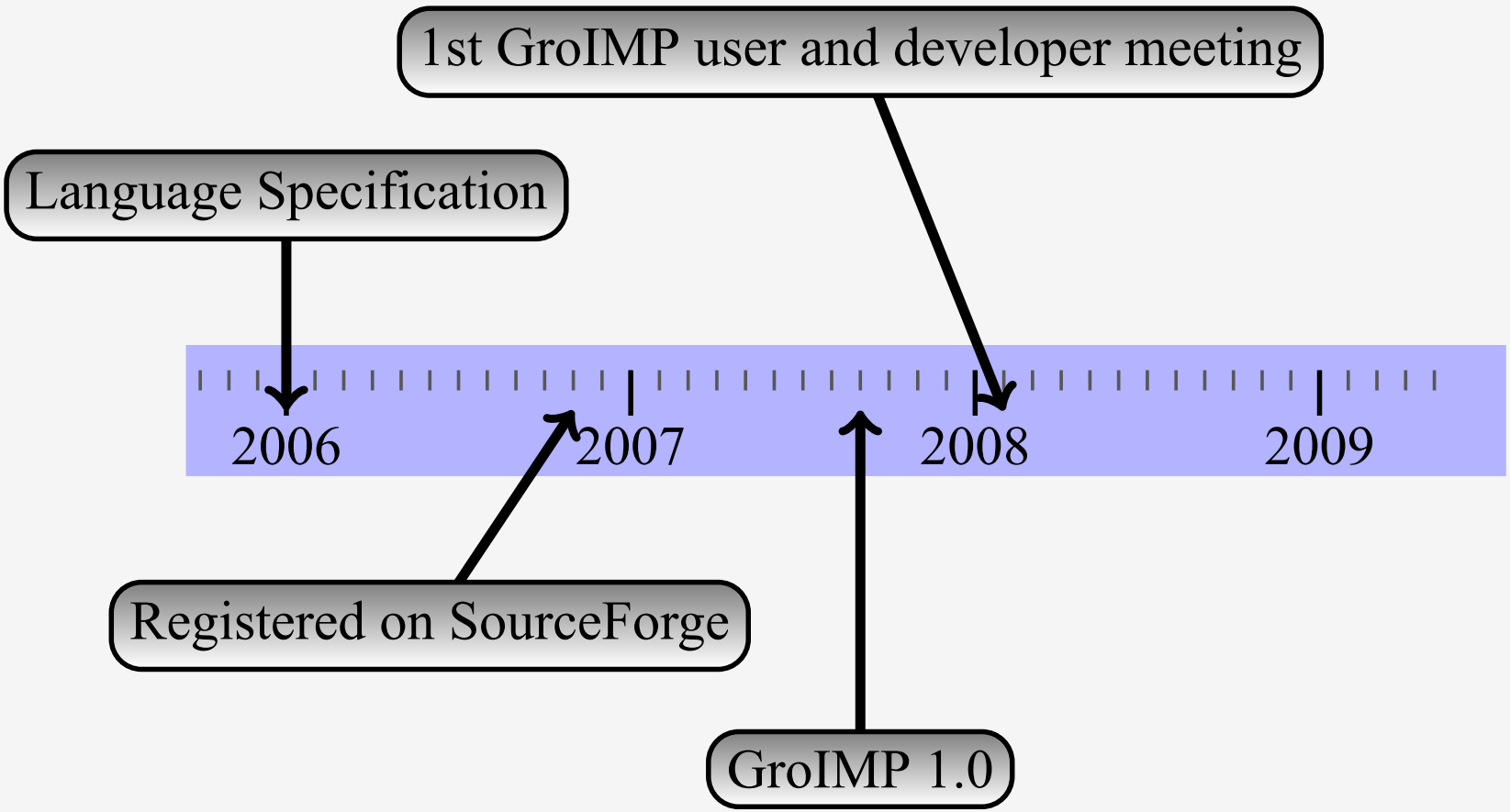
GroIMP

Timeline



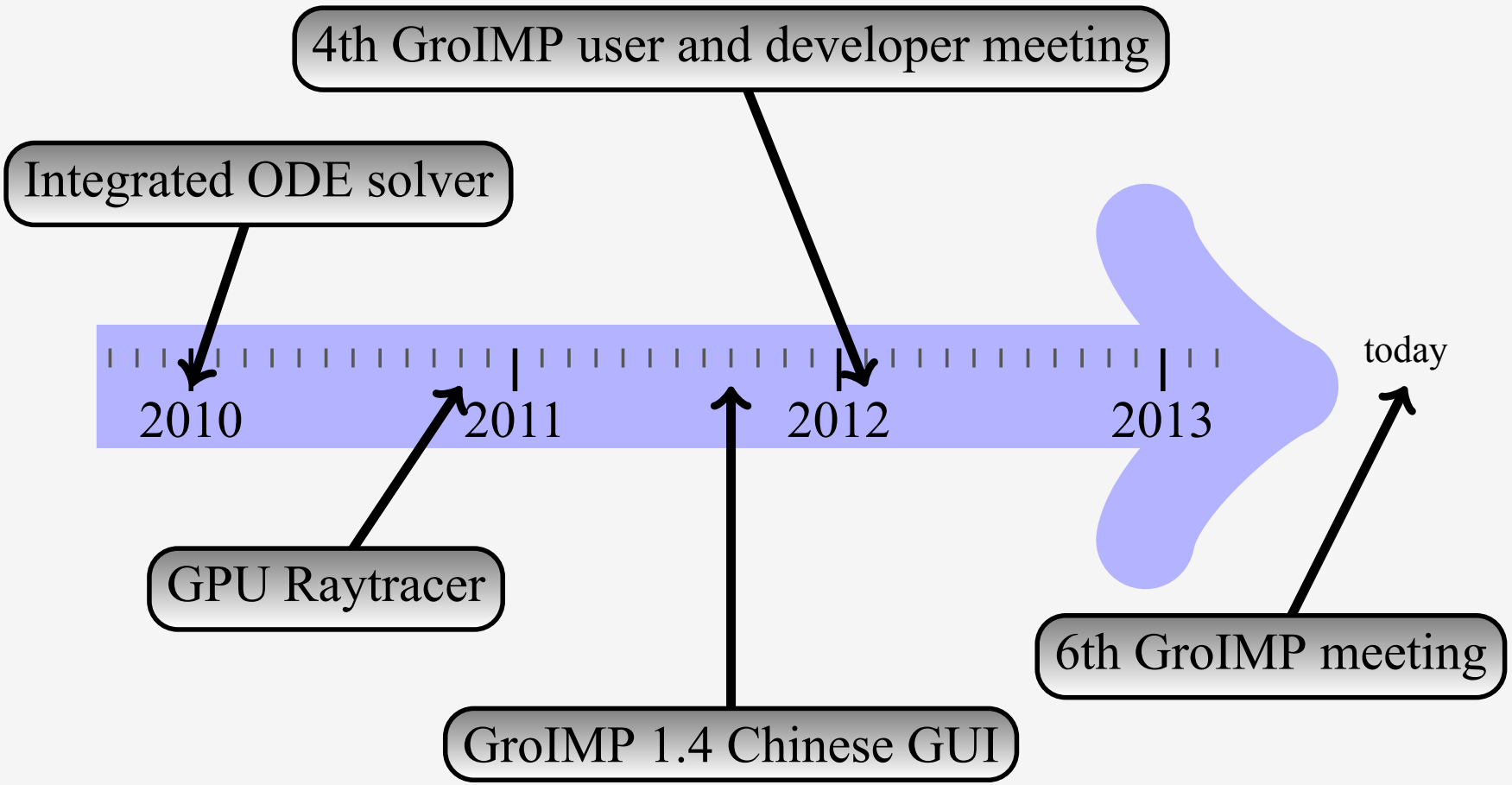
GroIMP

Timeline



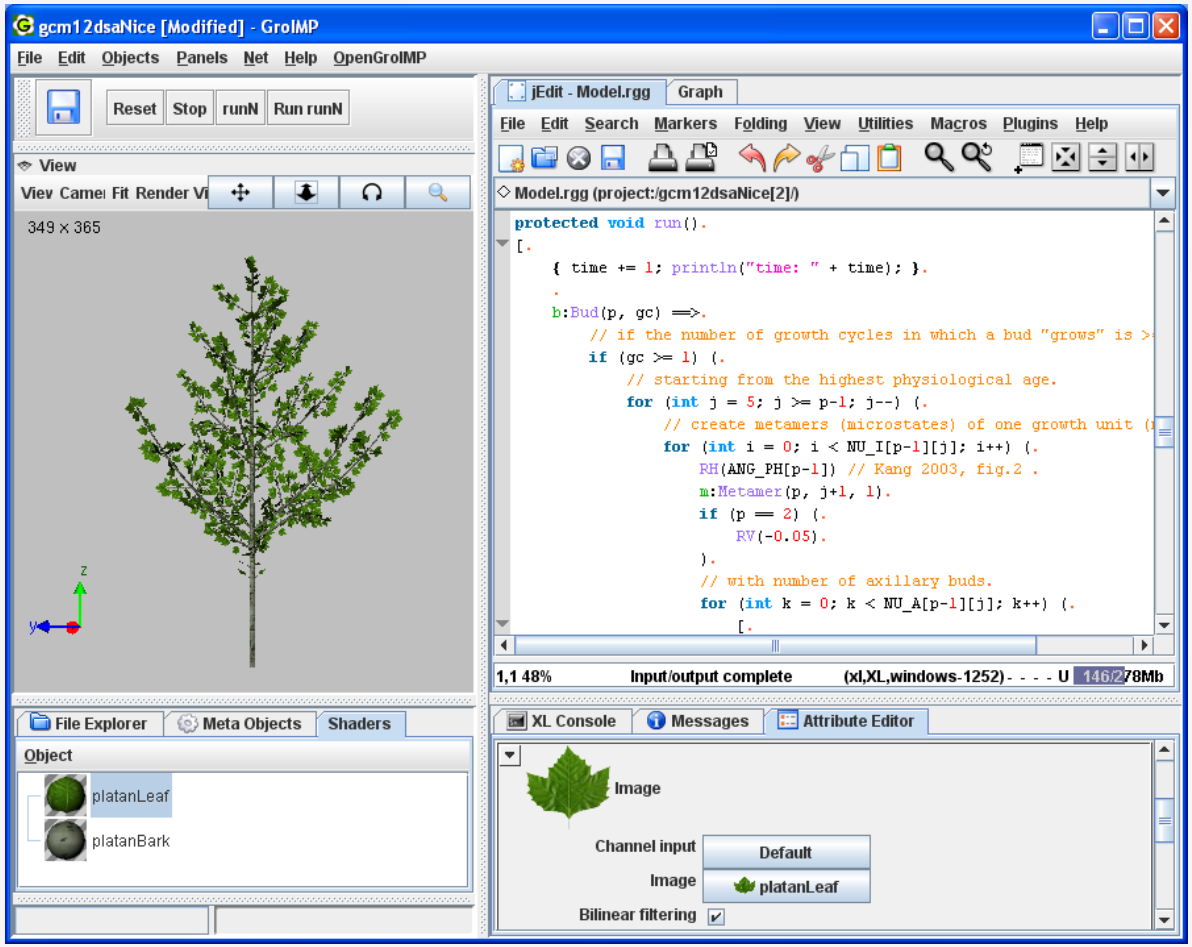
GroIMP

Timeline



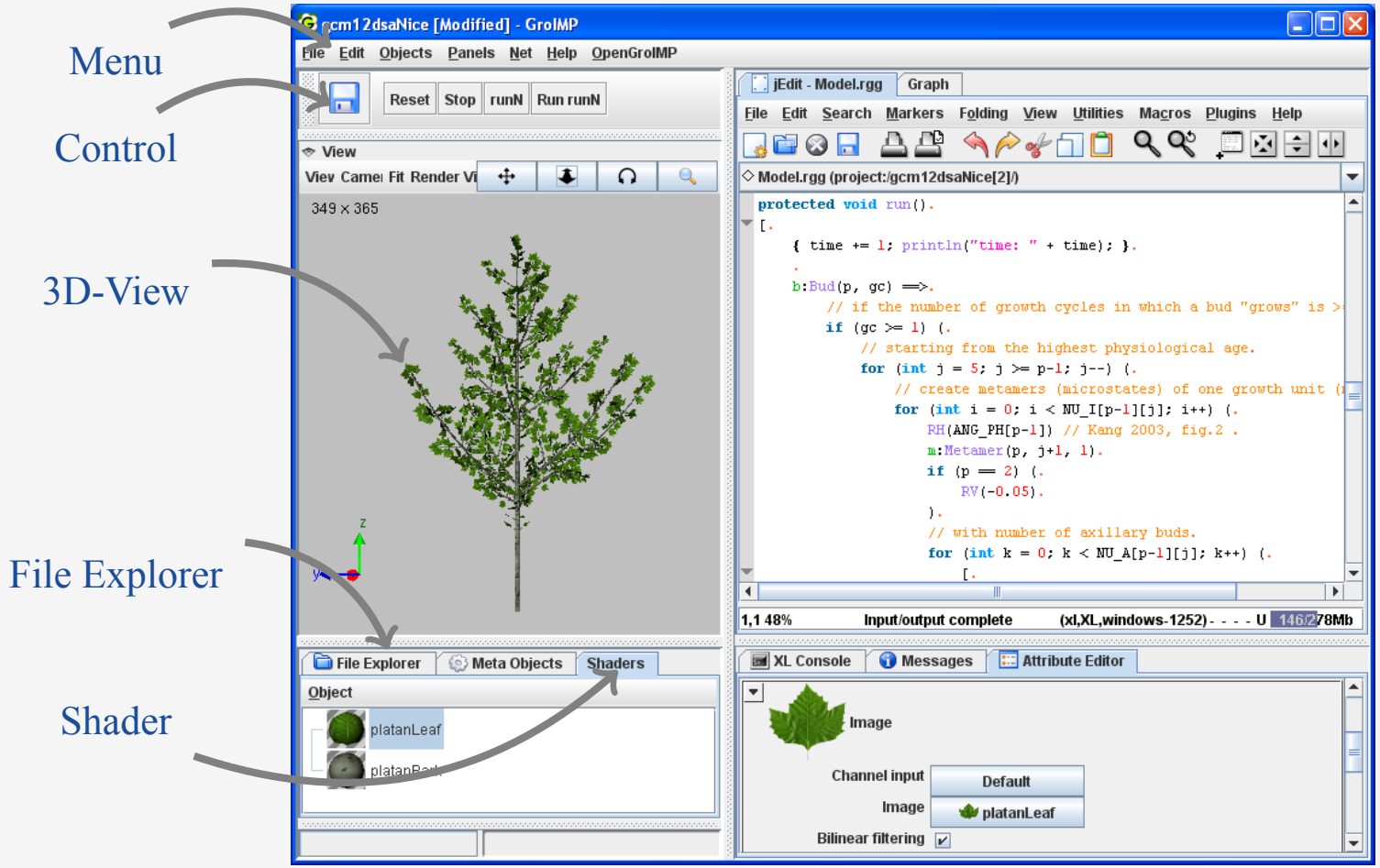
GroIMP

User interface



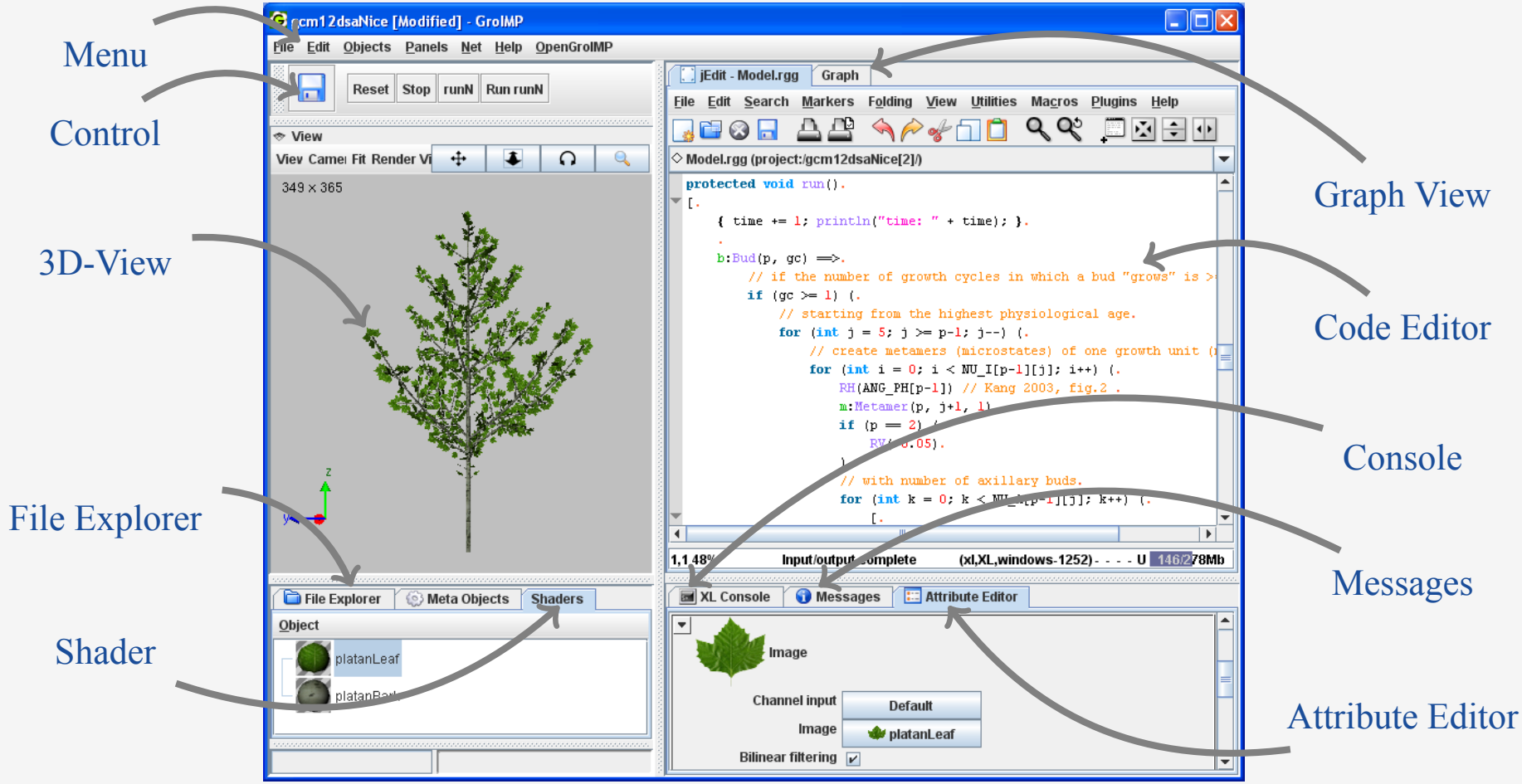
GroIMP

User interface

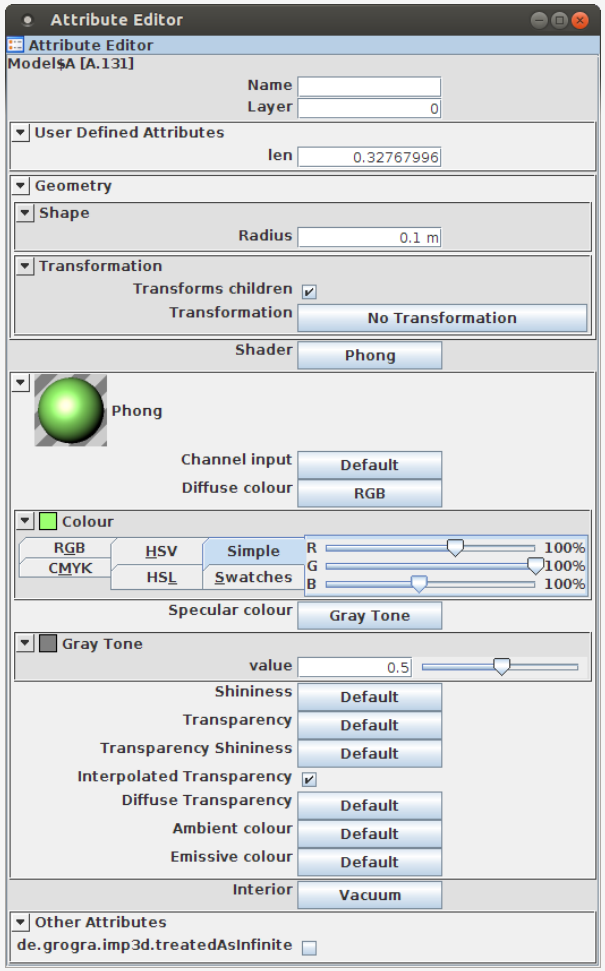


GroIMP

User interface

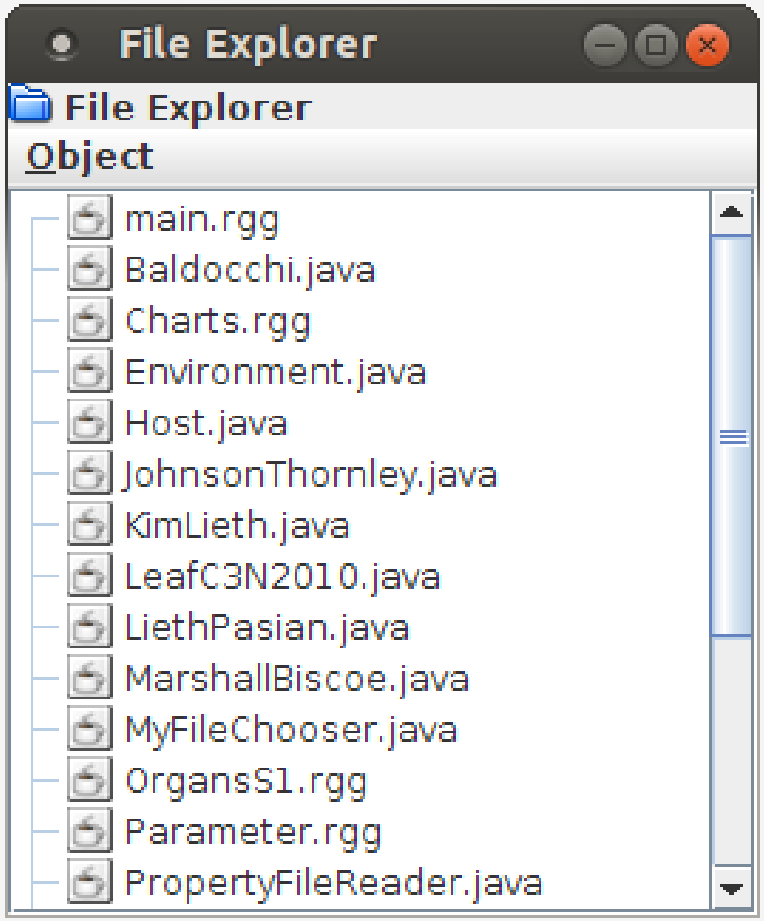


Attribute Editor



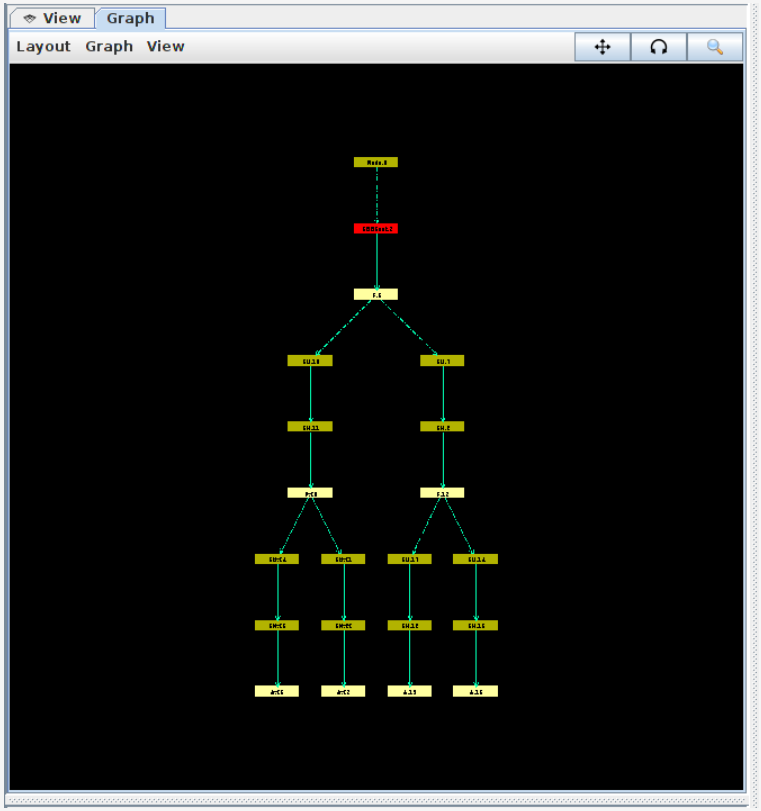
- The easy way to change attributes of an object.
- Attribute of a selected object are shown

File Explorer



- Projects may contain several RGG, XL and Java files. In addition, JAR files can be included.
- Plain text files (e.g. parameter files)

2D Graph Panel

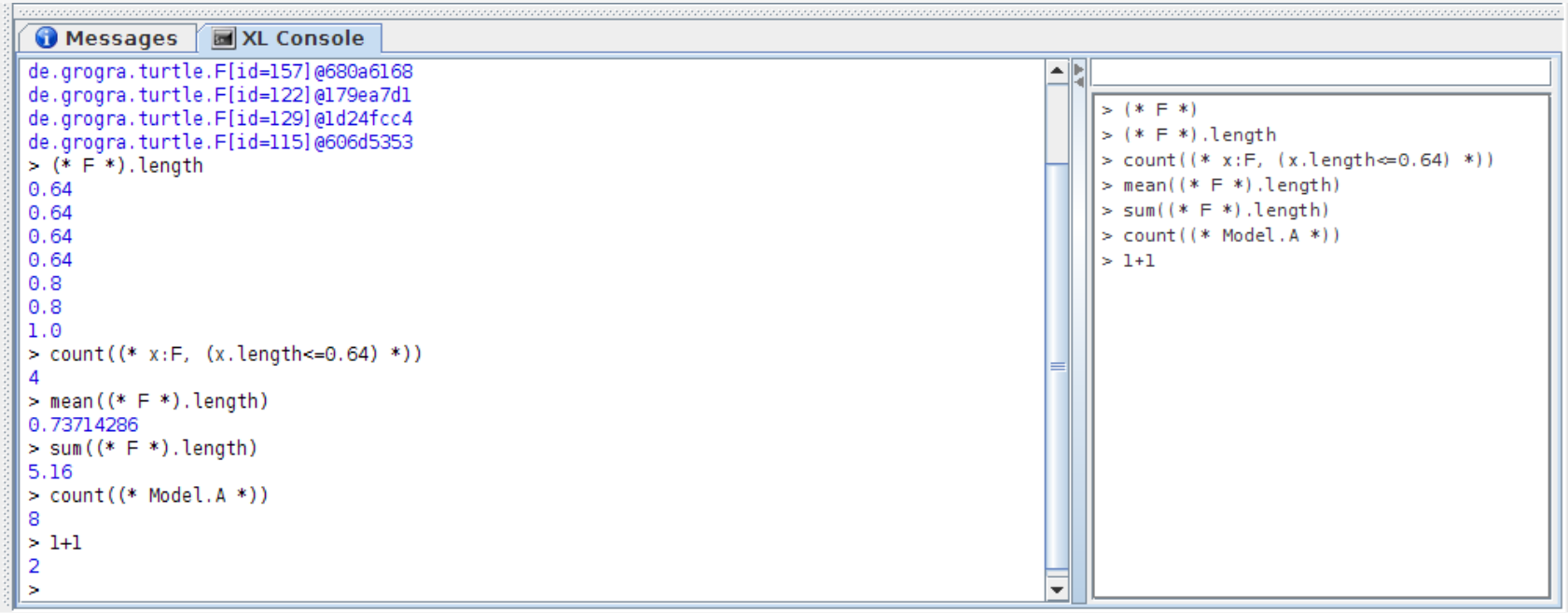


… Example, drawn in tree layout

- Graph structure produced by your model
- Several layout algorithm

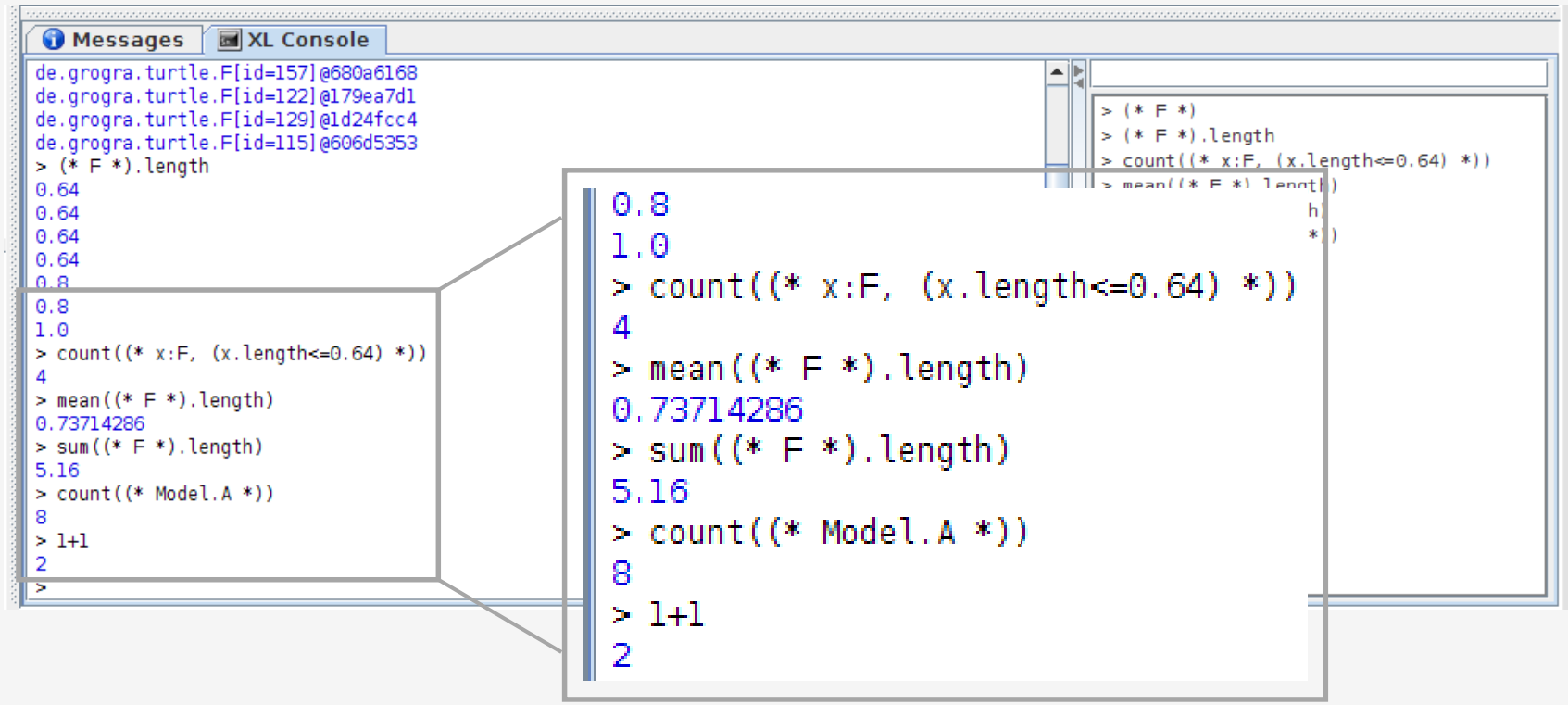
XL Console

Command line with History function



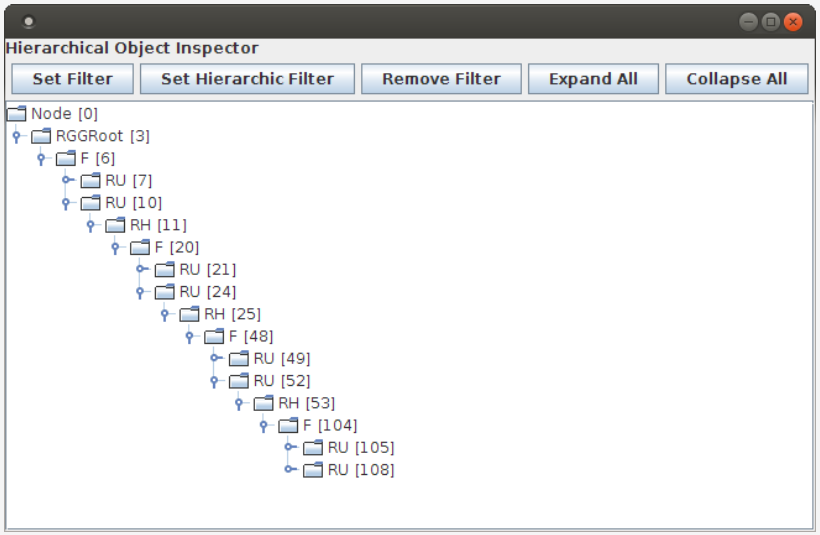
XL Console

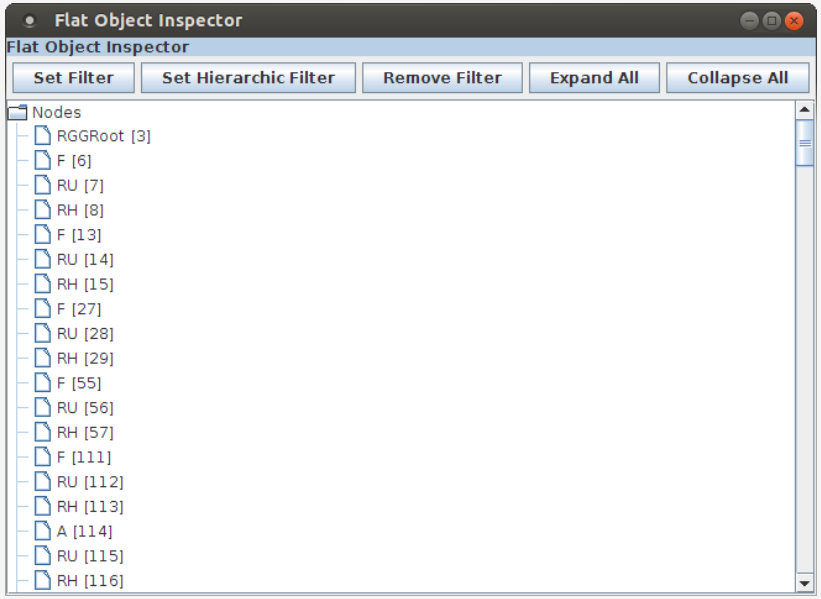
Command line with History function



Object Inspector

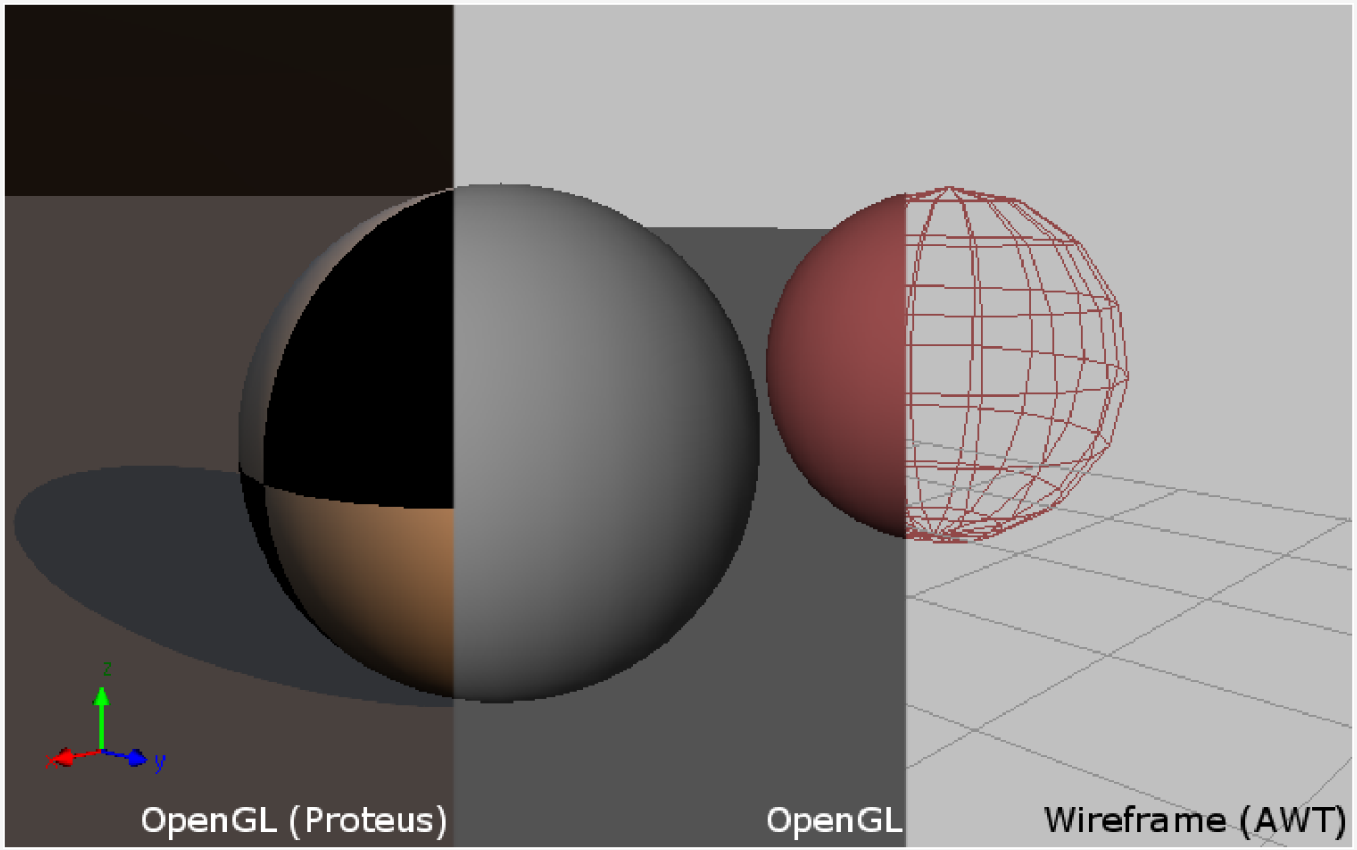
…hierarchical and flat





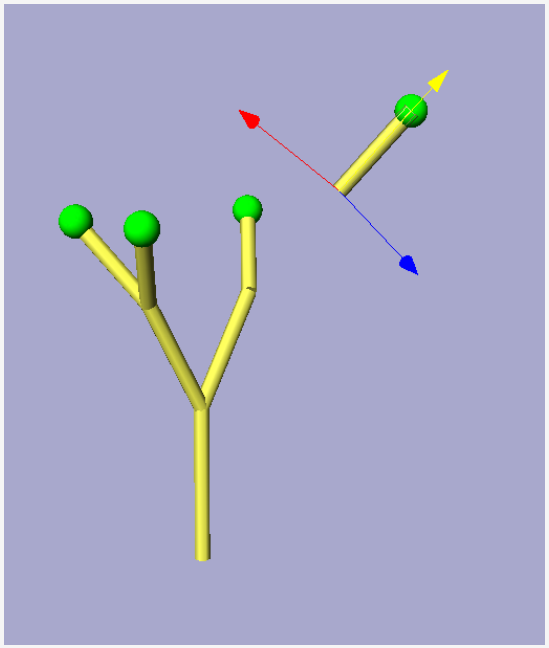
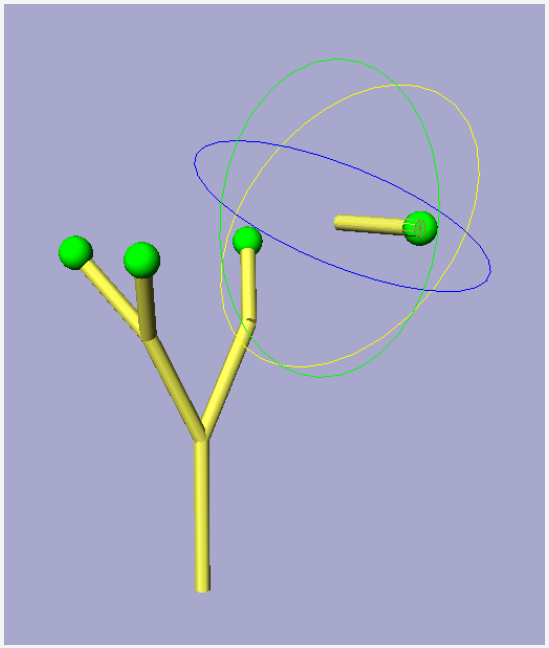
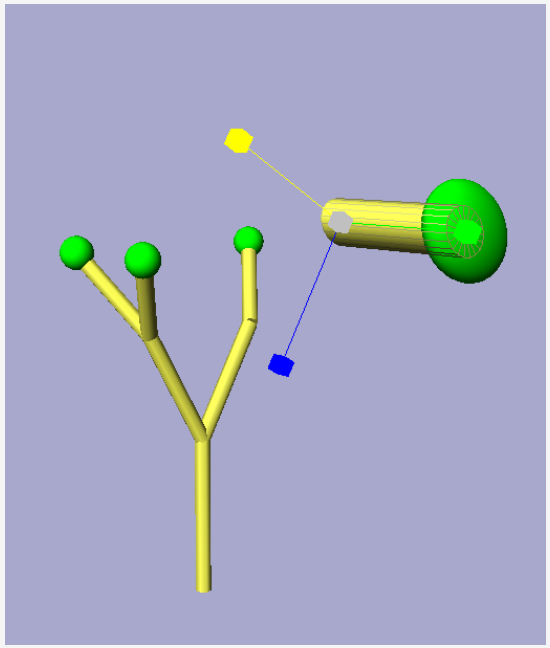
Fast node excess …+ filter options

Live 3D-View



Comparison of the different 3D views in GroIMP. From left to right: advanced OpenGL view, normal OpenGL view, software renderer.

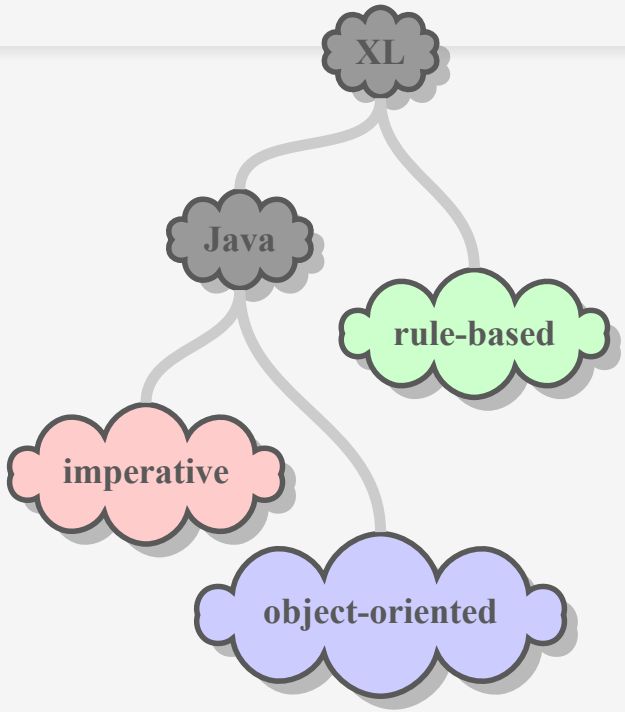
Interactive Live 3D-View

Objects can be interactively edited: moved, rotated, scaled
inserted and deleted

XL programming language

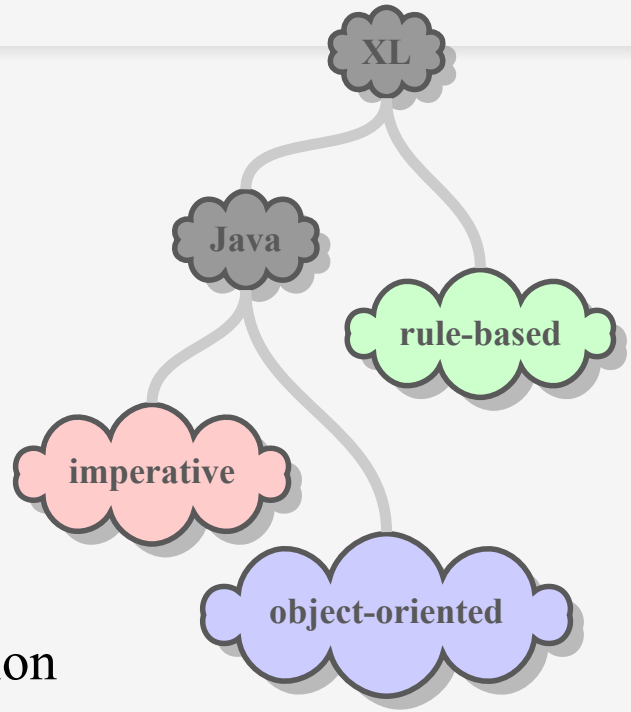
- e**X**tended **L**-system language
- Superset of Java, multiparadigm language



XL programming language

- eXtended L-system language
- Superset of Java, multiparadigm language

- Different types of rules
  - **L-system rule (==>)**
    ```
    A ==> A [B C];
    ```
  - **General graph rewriting rule (==>>)**
    more general rules for graph transformation
  - **Application rule (::>)**
    only parameters are updated
    (age, demand, biomass, length, etc.)

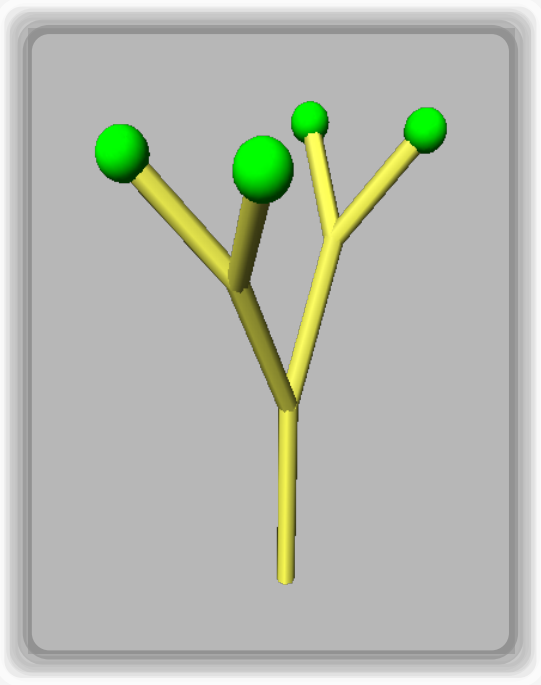


XL - Code example

```
1  import ...;
2
3  module A(float len) extends Sphere(0.1) {
4      {setShader(GREEN);}
5  }
6
7  protected void init () [
8      Axiom ==> A(1);
9  ]
10
11 public void run () [
12     A(x) ==> F(x)
13         [RU(30) RH(90) A(x*0.8)]
14         [RU(-30) RH(90) A(x*0.8)];
15 ]
```

XL - Code example

```
1  import ...;
2
3  module A(float len) extends Sphere(0.1) {
4      {setShader(GREEN);}
5  }
6
7  protected void init () [
8      Axiom ==> A(1);
9  ]
10
11 public void run () [
12     A(x) ==> F(x)
13         [RU(30) RH(90) A(x*0.8)]
14         [RU(-30) RH(90) A(x*0.8)];
15 ]
```



Integrated Query Language

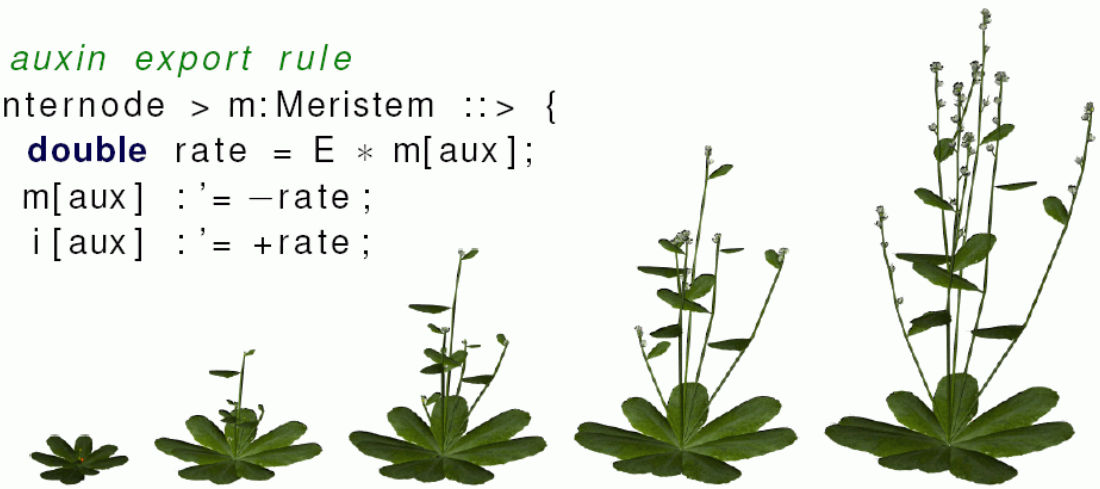
- **Aggregate operations** to collect / return specific values
  - `count`, `sum`, `min`, `max`, `mean`, `first`, `last`, `selectRandomly`, `exist`, etc.
- **Queries** to analyse the plant (graph) structure
  - Search for a specific pattern
  - Can be combined with aggregators, e.g., search for all leaves and sum up their area

```
1 > (* Leaf *)
2 > (* Leaf *).getArea()
3 > sum((* Leaf *).getArea())
```

ODE System

ODE system for easy specification of ordinary differential equations (ODEs) within rules, and their (separated) solution

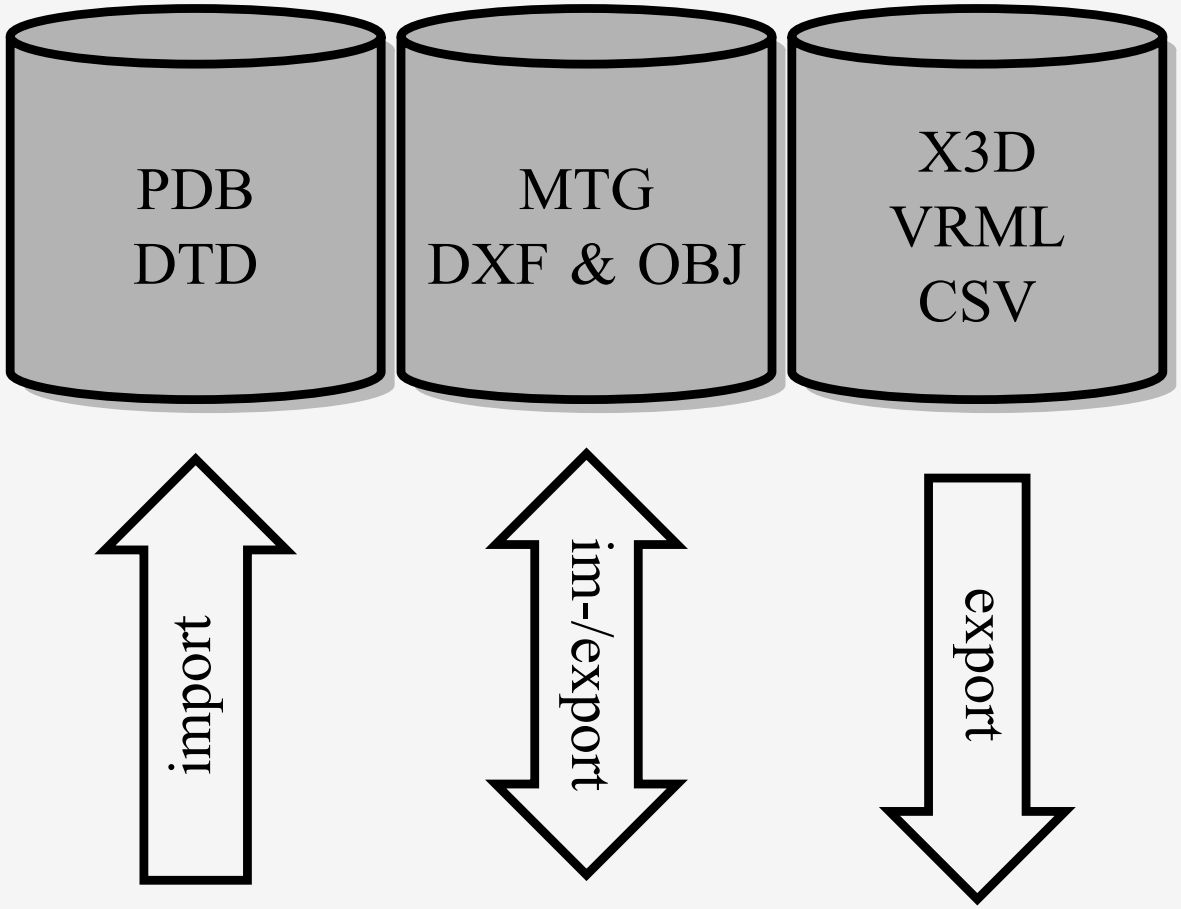
```
// cytokinin biosynthesis in root system
r:Roots ::> { r[cyt] :'= P - Q * r[aux]; }

// auxin export rule
i:Internode > m:Meristem ::> {
    double rate = E * m[aux];
    m[aux] :'= -rate;
    i[aux] :'= +rate;
}
```



(Evers 2012)

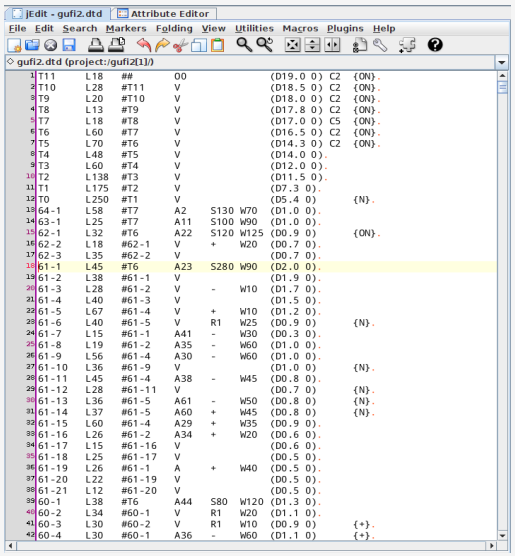
$\Longrightarrow$ Hemmerling 2012: Extending the Programming Language XL to Combine Graph Structures with Ordinary Differential Equations.

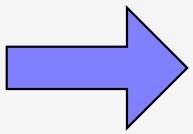
Import- and Export formats

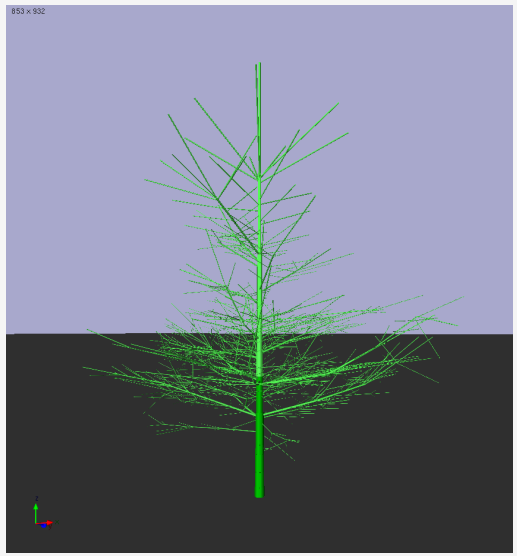


Descriptive tree data - DTD Editor

- Plain text file (Example file:)
- Every line represents a shoot
- Whole file represents a plant or a part of a plant
  - Used primarily for trees
  - General: All branching structures







http://www.uni-forst.gwdg.de/~wkurth/cb/html/workshop07/ws07_derer.pdf

http://www.uni-forst.gwdg.de/~wkurth/dtd_code.pdf

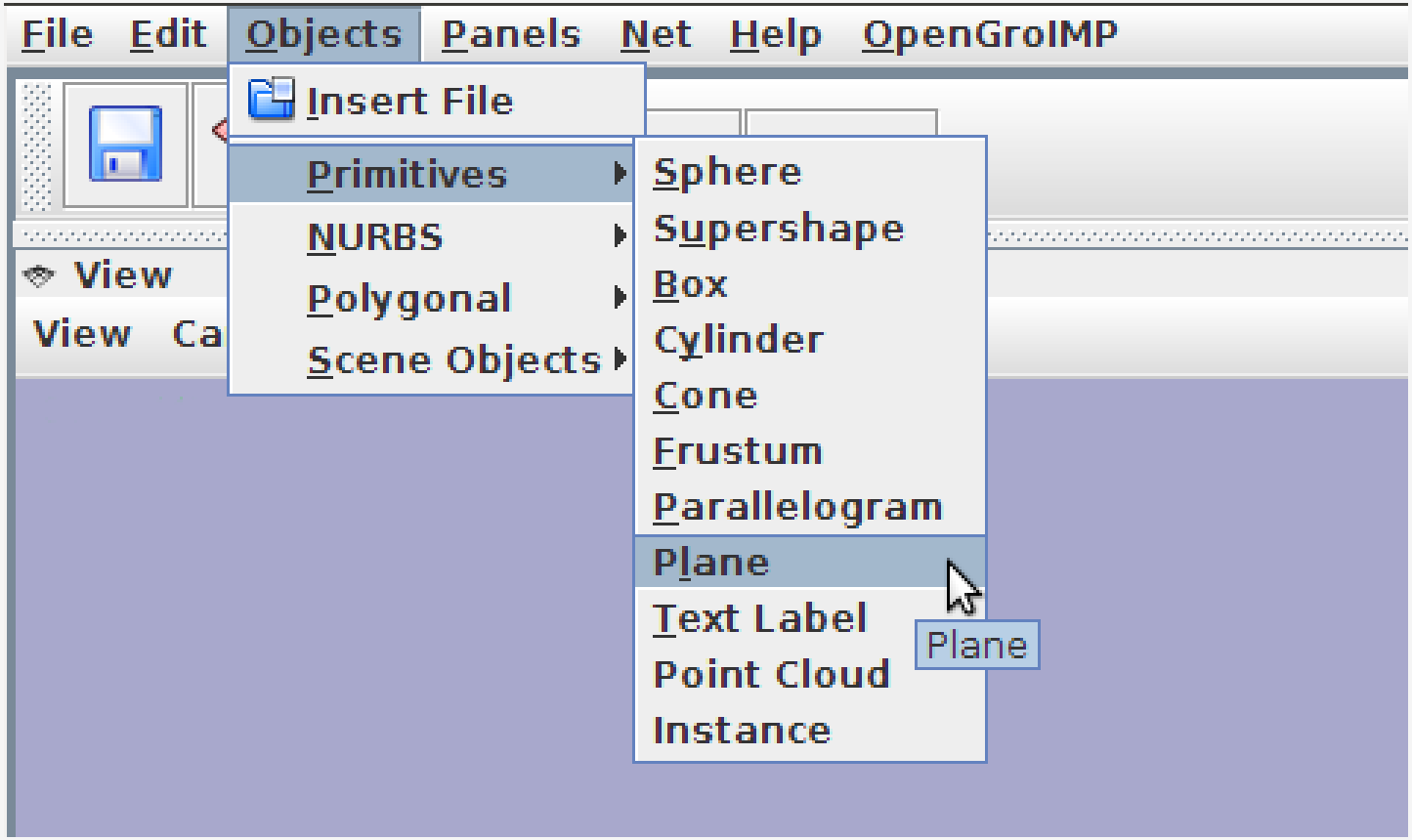
Simple File export

$\Longrightarrow$ Example: *Gallery/Technics/print_in_file2.gsz*

```
1  import java.io.*;
2
3  PrintWriter tmpfile;
4
5  protected void init () {
6    // create the file
7    tmpfile = new PrintWriter(new FileWriter("/home/../out1.csv"));
8    [  Axiom ==> A(1);  ]
9  }
10
11 public void run () [
12   a1:A(x) ==> F(x) [RU(30) RH(90) a2:A(x*0.8)] [RU(-30) RH(90) a3:A(x*0.8)]
13   {
14     tmpfile.println("" + a1.len + " " + a2.len + " " + a3.len);
15     //write data to the file
16     tmpfile.flush();
17   };
18 ]
19
20 // close the file
21 public void end(){
22   tmpfile.close();
23 }
```

Objects

Available primitives objects, NURBS, Polygonals and LightNodes

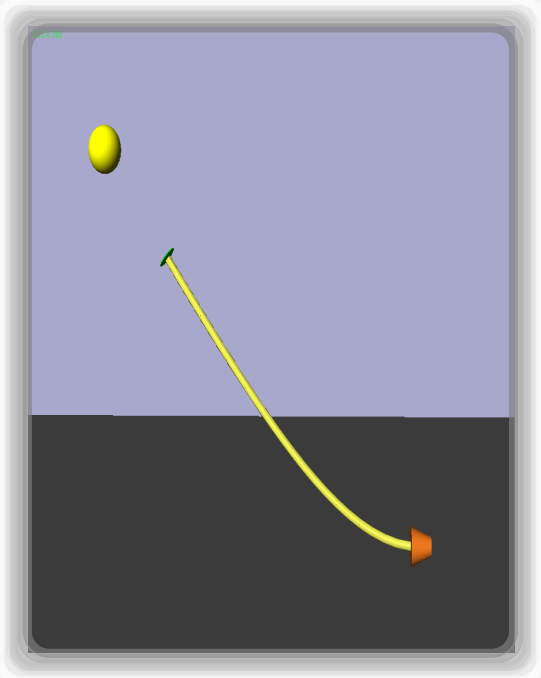


Tropism Nodes

- growth in response to an environmental stimulus
- Commands: RD, RP, RN
- e.g. Phototropism: Shoots bends towards a light source

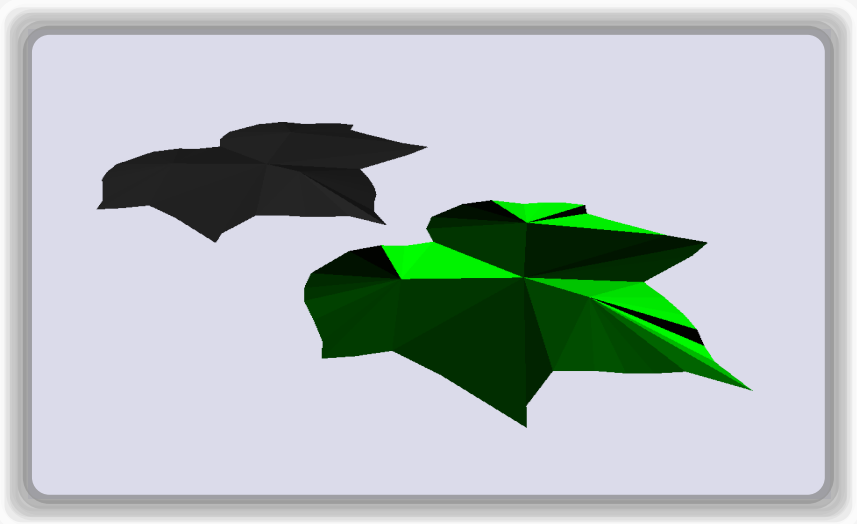
```
1 Sphere sun = new Sphere(0.25).(setShader(YELLOW),
        setTransform(5,0,5));
2
3 ...
4
5 public void run () [
6     top_bud ==> RN(sun, 0.2) F(0.2) top_bud;
7 ]
```

Point-Tropism(*point/direction, power*)



Heliotropism

Algorithm Switch Shader



Allows us to define tow different textures for an object. One is used by the light model (raytracer) and defines the optical properties of the object for light absorption, reflection etc., the second one is used for visualisation in the 3D view.

```
1  Shader leafTexture = shader("leaf");
2  Phong leafShader = new Phong().(
3      setDiffuse(new RGBColor(reflectance,reflectance,reflectance)),
4      setDiffuseTransparency(new RGBColor(transmittance,transmittance,
           transmittance)));
5  ...
6
7  leaf.setShader(new AlgorithmSwitchShader(leafTexture , leafShader));
```

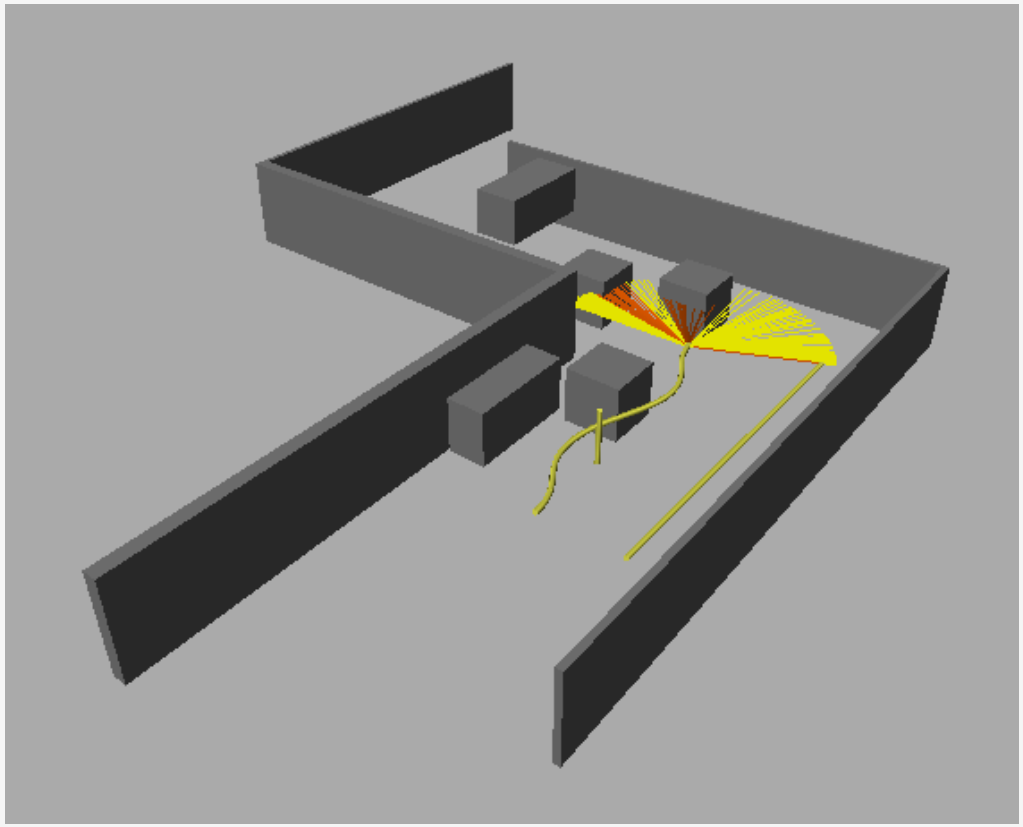
Billboard generator

Integrated tool to generate billboards.

(Hemmerling 2010)

Intersection avoidance mechanism

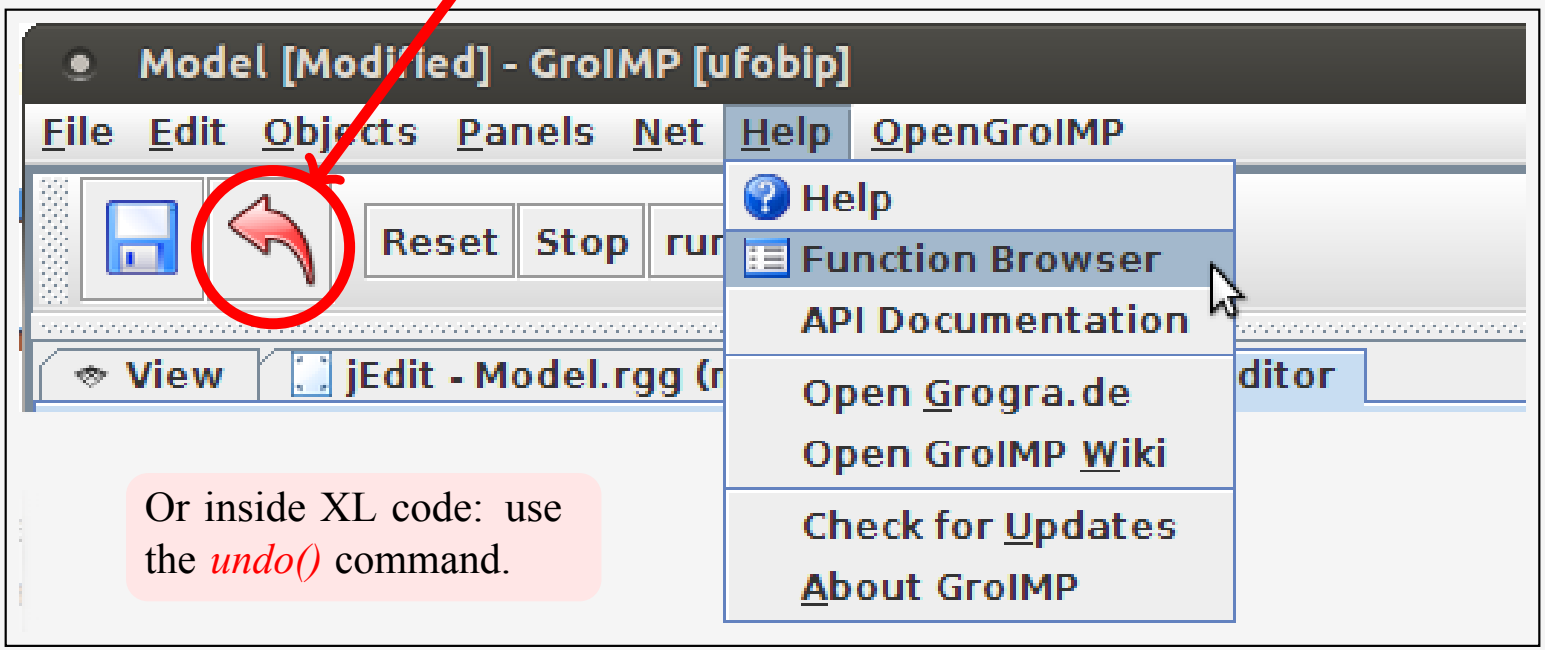
The class AvoidIntersection uses rays and intersection tests to obtain a new direction for growth.



$\Longrightarrow$ Example: *Gallery/Technics/smart_line.gsz*

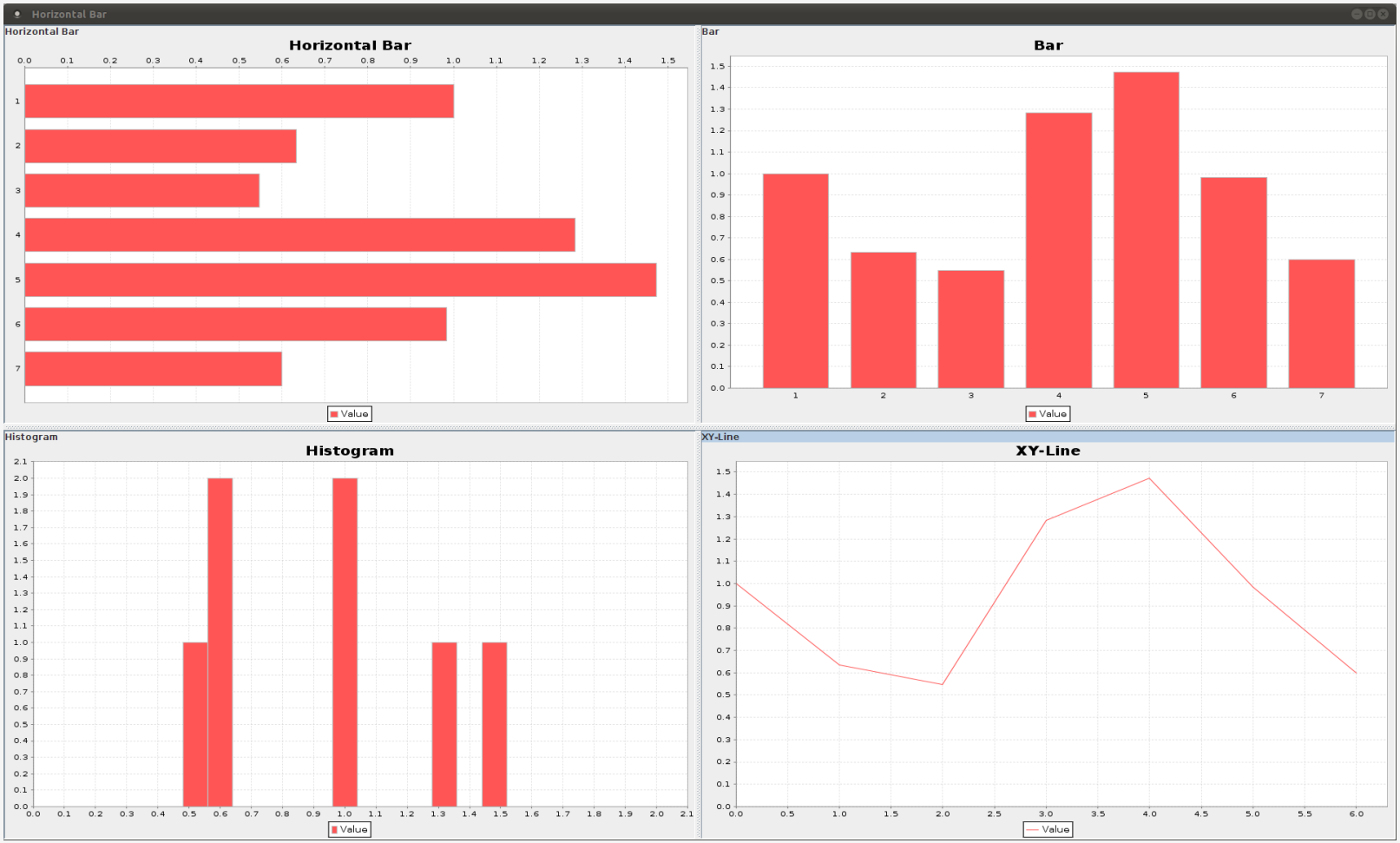
Undo

Undo derivation steps.



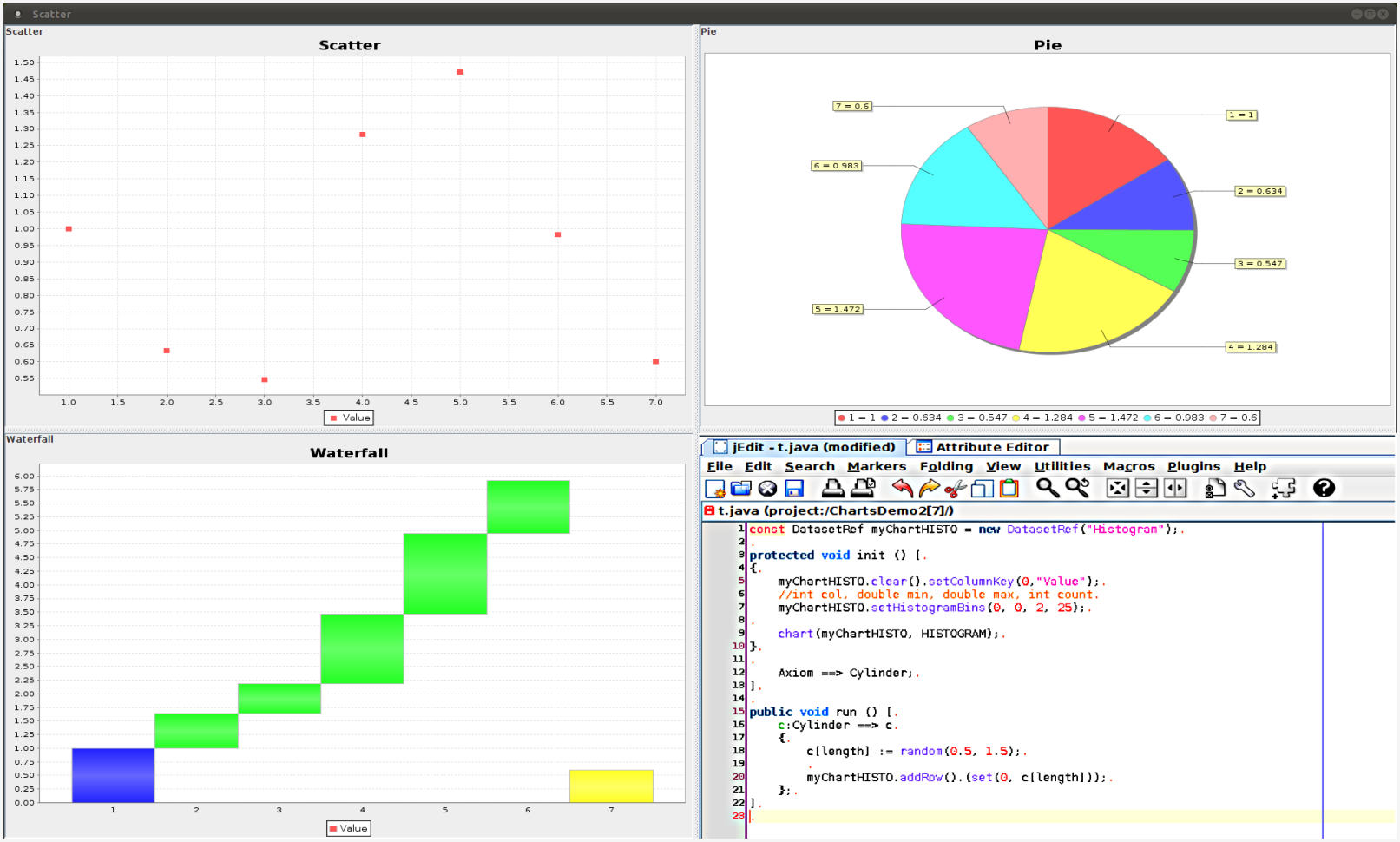
A helpful tool, e.g., when you search for certain criteria of your model. After each step you can check you structure against you conditions. If they are not fulfilled, undo the last step, change you parameters and try again.

Chart demo



$\Longrightarrow$ Example: *Gallery/Technics/ChartsDemo2.gsz*

Chart demo



$\Longrightarrow$ Example: *Gallery/Technics/ChartsDemo2.gsz*

Analysis Functions

- Usable for all on XL-based structures, e.g. RGG-models, models from DTD/LSY/SSY-files
- Export interfaces:
  *.txt, *.html, *.xml, *.csv, *.xls, *.pdf
- Functions based on the Grogra analysis functions
- Some analysis functions give a list for further processing
  e.g. number of daughter shoots, shoots branching positions, length and angles, …
- It‘s a small overview of the analysis functions:
  - Current 16 analysis functions
  - Topological & distribution analysis, analysis …

Analysis Functions

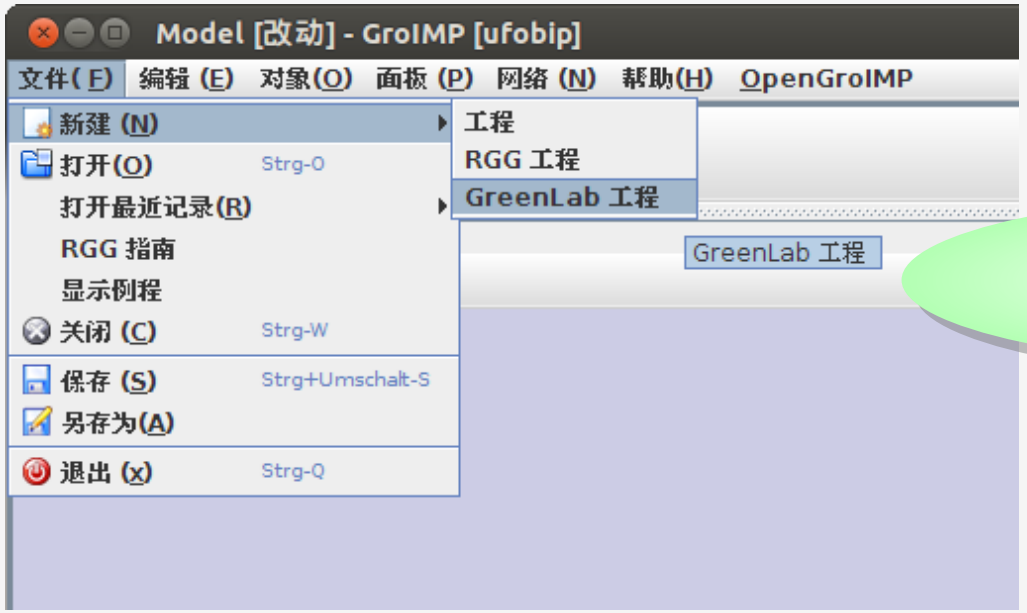
- Usable for all on XL-based structures, e.g. RGG-models, models from DTD/LSY/SSY-files

View | GROGRA Functions

data Analysis

#	shoot-ID	diameter	av.length	max.length	no. of longe...	length sum	volume sum	n sum	order
1	6	0.10000000...	3.36159920...	3.36159992...	216	15.8096027...	0.12416827...	0.0	0
2	20	0.10000000...	2.36159992...	2.36159992...	216	7.40479898...	0.05815715...	0.0	1
3	48	0.10000000...	1.56159996...	1.56159996...	216	3.30240011...	0.02593698...	0.0	2
4	104	0.10000000...	0.92159992...	0.92159992...	216	1.33119988...	0.01045522...	0.0	3
5	216	0.10000000...	0.40959995...	0.40959995...	216	0.40959995...	0.00321699...	0.0	4
6	209	0.10000000...	0.40959995...	0.40959995...	209	0.40959995...	0.00321699...	0.0	4
7	97	0.10000000...	0.92159992...	0.92159992...	202	1.33119988...	0.01045522...	0.0	3
8	202	0.10000000...	0.40959995...	0.40959995...	202	0.40959995...	0.00321699...	0.0	4
9	195	0.10000000...	0.40959995...	0.40959995...	195	0.40959995...	0.00321699...	0.0	4
10	41	0.10000000...	1.56159996...	1.56159996...	188	3.30240011...	0.02593698...	0.0	2
11	90	0.10000000...	0.92159992...	0.92159992...	188	1.33119988...	0.01045522...	0.0	3
12	188	0.10000000...	0.40959995...	0.40959995...	188	0.40959995...	0.00321699...	0.0	4
13	181	0.10000000...	0.40959995...	0.40959995...	181	0.40959995...	0.00321699...	0.0	4
14	83	0.10000000...	0.92159992...	0.92159992...	174	1.33119988...	0.01045522...	0.0	3
15	174	0.10000000...	0.40959995...	0.40959995...	174	0.40959995...	0.00321699...	0.0	4
16	167	0.10000000...	0.40959995...	0.40959995...	167	0.40959995...	0.00321699...	0.0	4
17	13	0.10000000...	2.36159992...	2.36159992...	160	7.40479898...	0.05815715...	0.0	1
18	34	0.10000000...	1.56159996...	1.56159996...	160	3.30240011...	0.02593698...	0.0	2
19	76	0.10000000...	0.92159992...	0.92159992...	160	1.33119988...	0.01045522...	0.0	3

Multilingual user interface



English, Chinese

work in progress / to do

Slovak, French, German

Batch mode

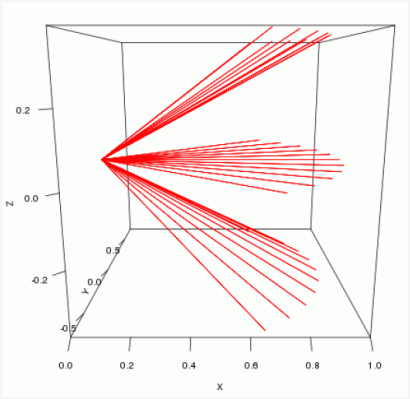
Run GroIMP headless to realise repetitive simulations like automated scenario test.

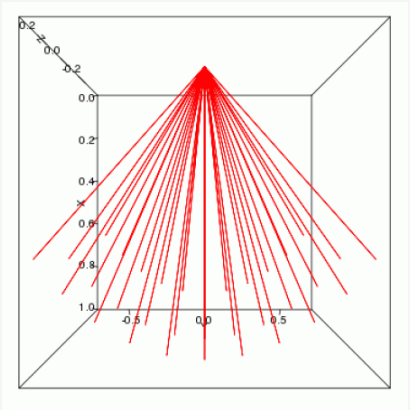
```
1 java -cp ... de.grogra.pf.boot.Main --project-tree --headless Model.gsz
```

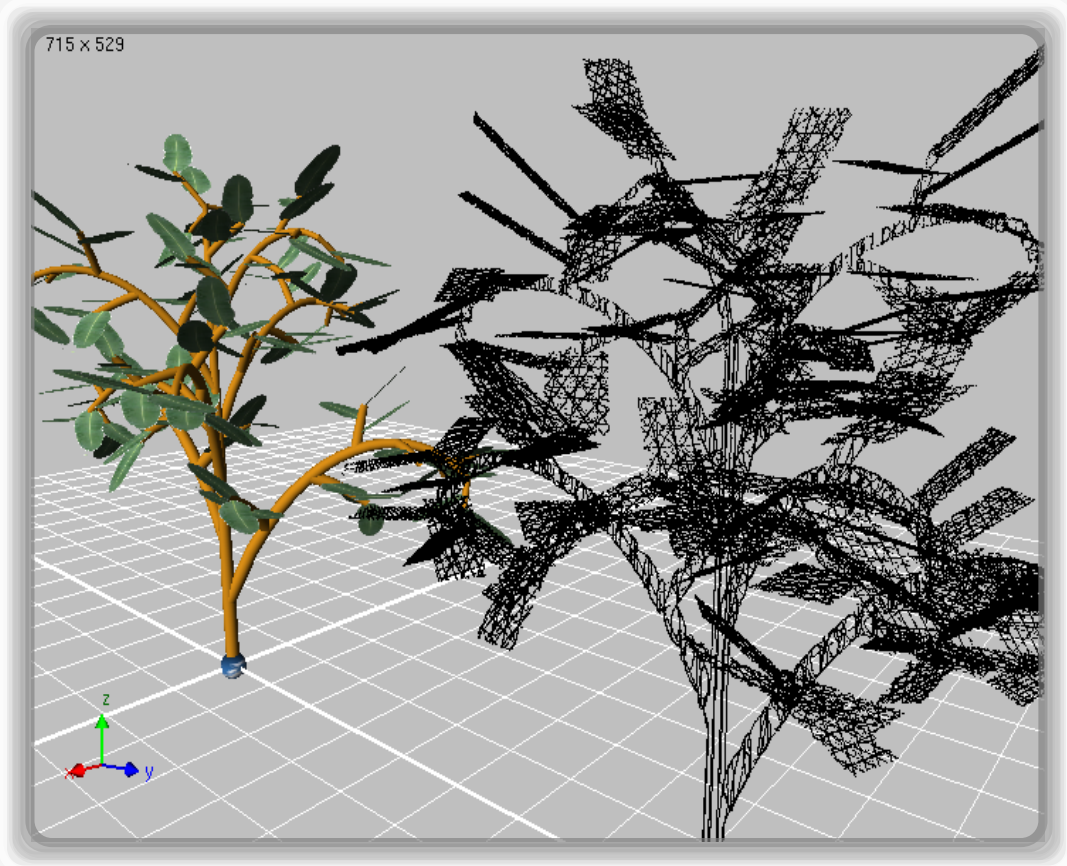
$\Longrightarrow$ Examples: *Gallery/Technics/renderHeadlessDemo2.gsz*

Virtual Laser Scanner

Produced point clouds



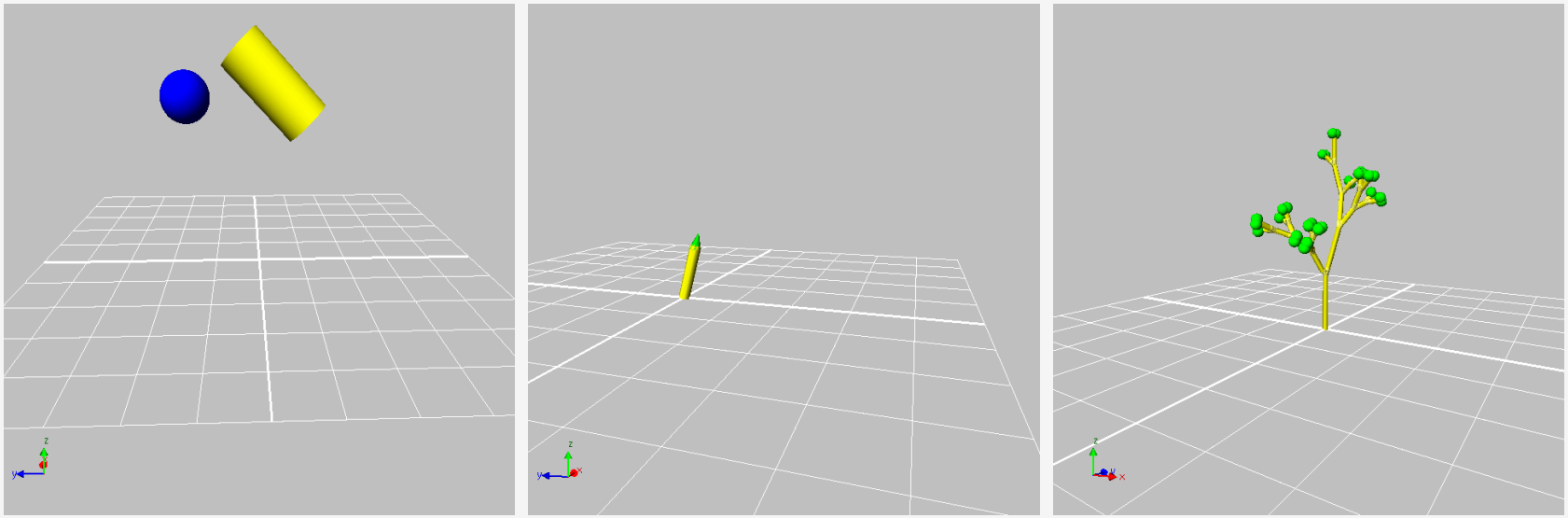




$\Longrightarrow$ Etard 2011: Virtual Laser Scanner for GroIMP

Physics Engine

Robotropism: Distributed Control of Plant
Control of the spatial shape of plants through small artificial controllers attached to the plan.

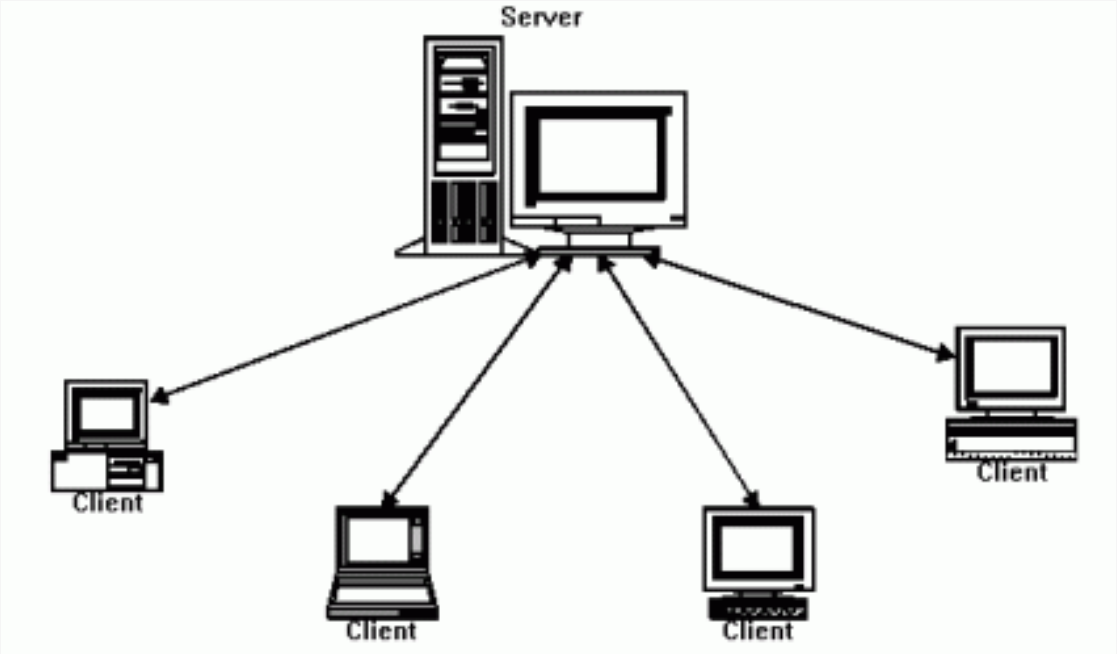


$\Longrightarrow$ Masters 2010:

http://www.uni-forst.gwdg.de/~rhemmer/workshop/XL10/Robotropism_v1.pdf

Open-GroIMP Specification

HTTP server: The Open-GroIMP plugin allows to call GroIMP from other applications via the network.

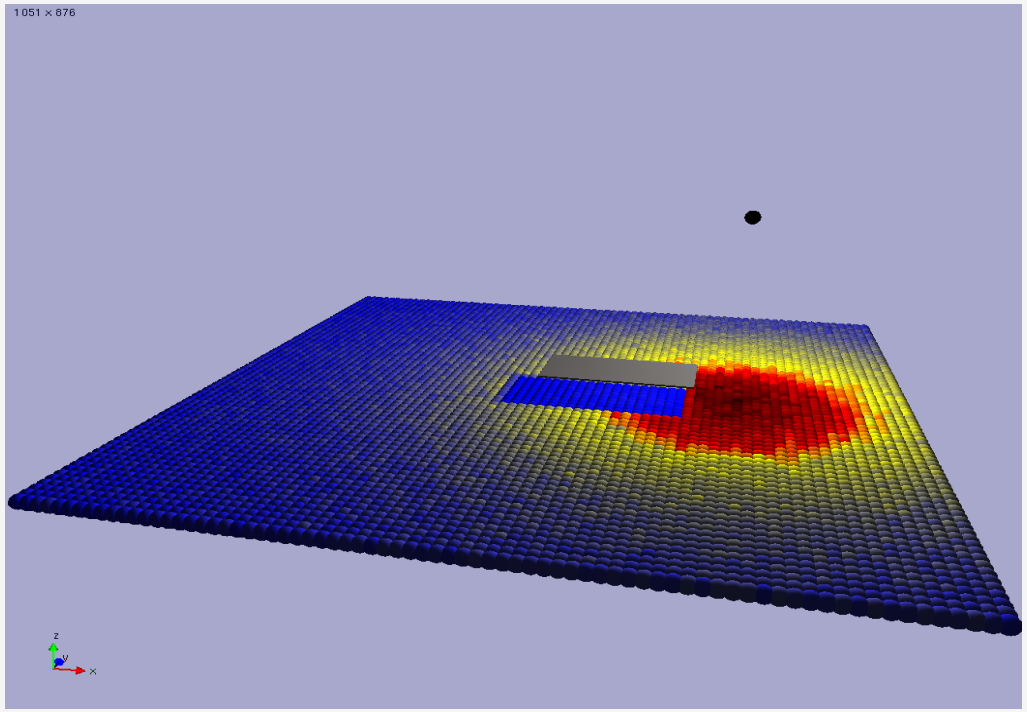


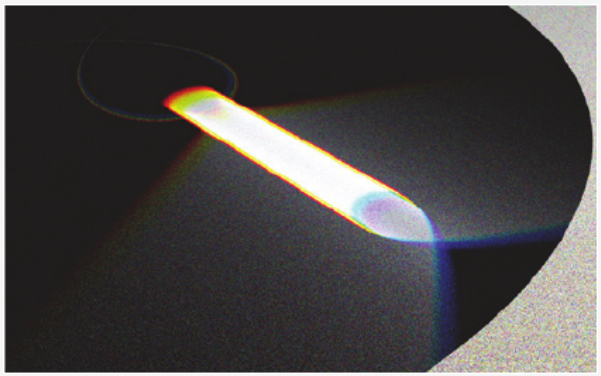
http://wwwuser.gwdg.de/~groimp/grogra.de/documentation/Specification.txt

http://wwwuser.gwdg.de/~groimp/grogra.de/documentation/Description.txt

Integrated Light Model

- External raytracer (POV-Ray)
- Built-in raytracer (Twilight) & stochastical pathtracer; HDR
- High performance light model (GPU Flux)
  for pathtracer, bidirectional pathtracer and spectral rendering

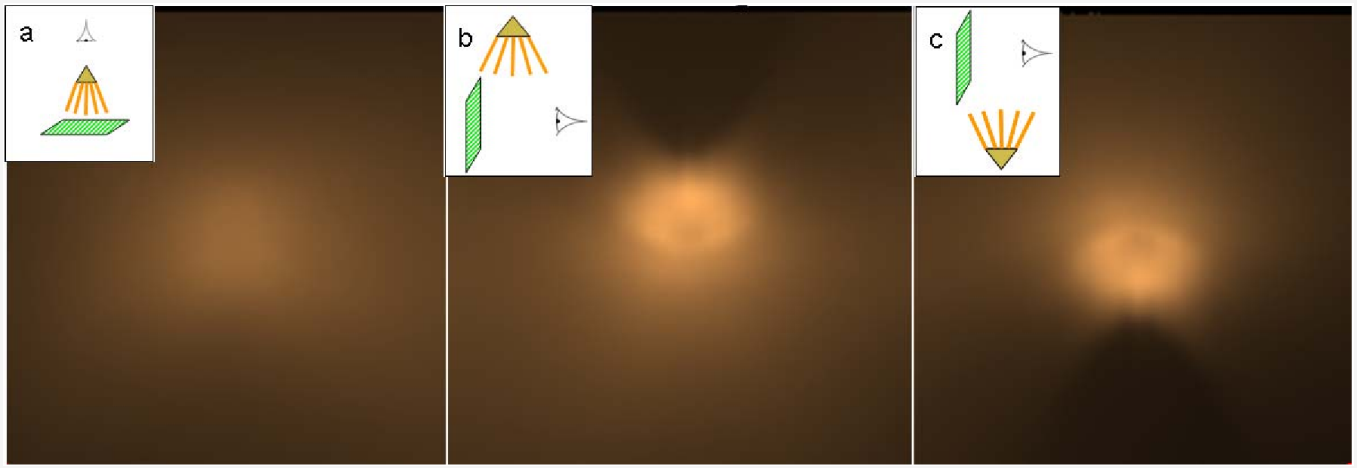




Supports full spectral rendering (van Antwerpen 2011)

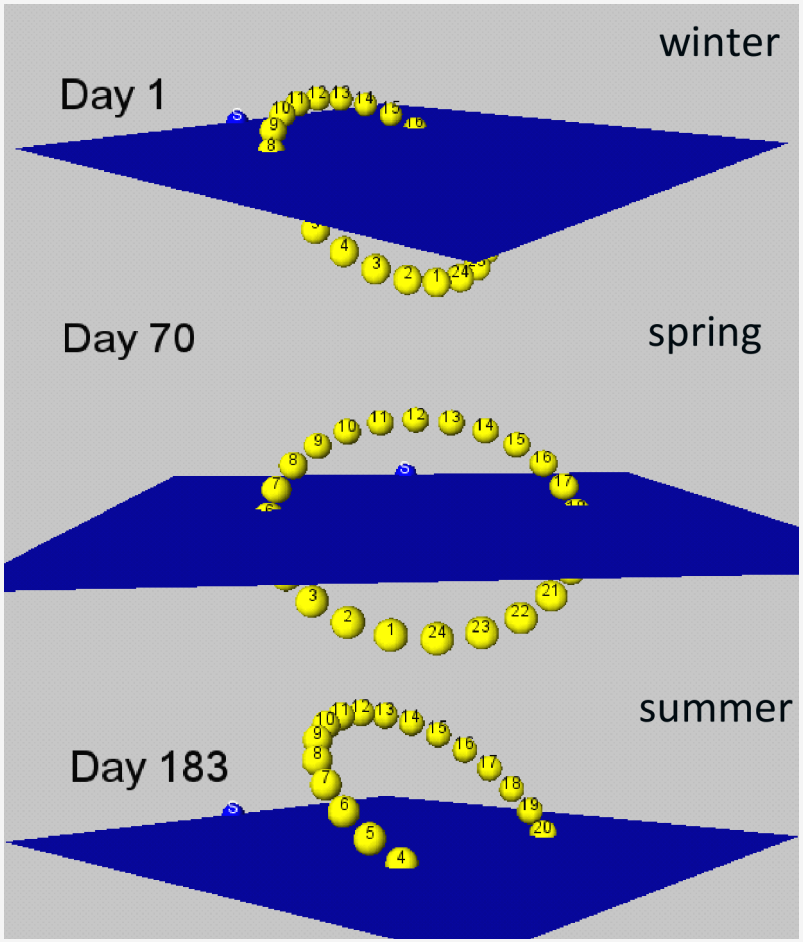
Different Light Sources

- Direct and indirect light sources
- Light nodes: point-, spot light
- Sky object (SunSkyLight shader based on Sunflow)
- Individually defined light sources: SON-T, LED
  Specific light spectrum



$\Longrightarrow$ Buck-Sorlin et al. 2010: Modelling of spatial light distribution in the greenhouse: Description of the model.

Sun model

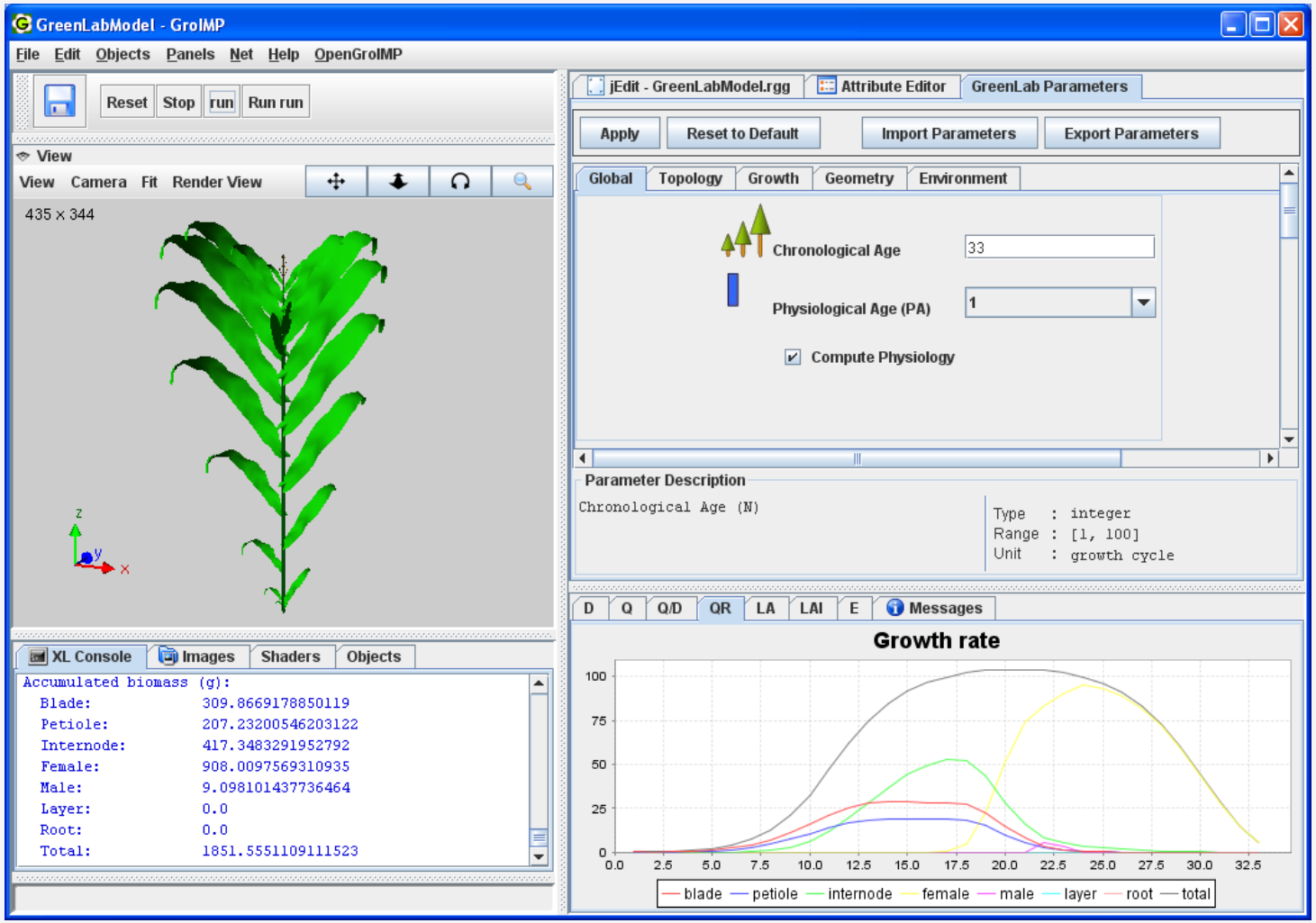


- input: latitude, day of year, time of day [h]
- simulation of direct sunlight as a directional light source (parallel rays), which changes position, thereby adjusting direction

- simulation of diffuse skylight with sky object (Preetham et al. 1999)

$\Longrightarrow$ Buck-Sorlin et al. PMA09: Modelling of spatial light distribution in the greenhouse: Description of the model.

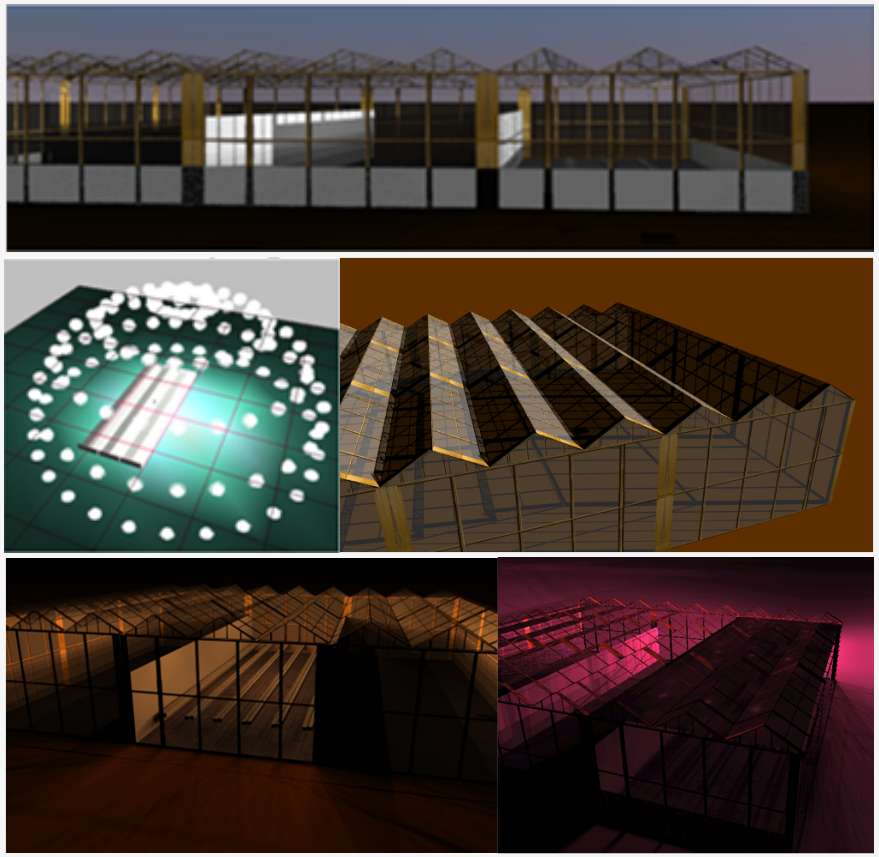
GreenLab/XL model

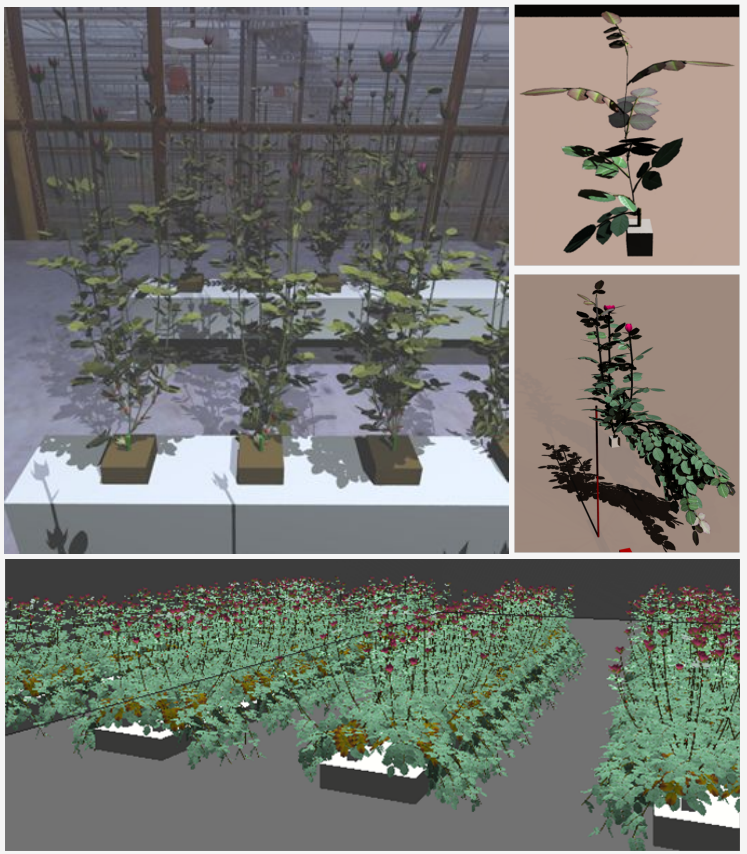


Main attention / Features of the GreenLab/XL model

- GUI aided input method for parameter input
- Make it “user save”
  - Control of inputs (general)
    - Number fields
    - Choice boxes
    - Check boxes
  - Check input ranges
  - Deactivate not used fields
  - Measurement units
- Description window (parameter description, type, unit, range)
- Provide im- and export of GreenLab parameter files (“*sci*”)
- Support AMAPSymbol files (“*smb*”) for shape objects
- Support of textures

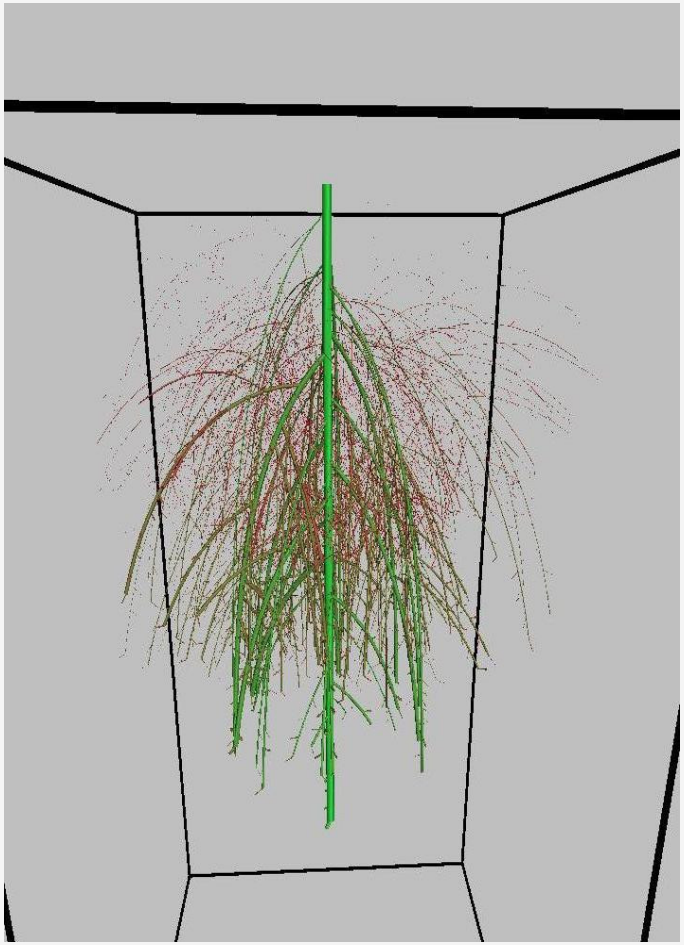
Cut-rose FSPM

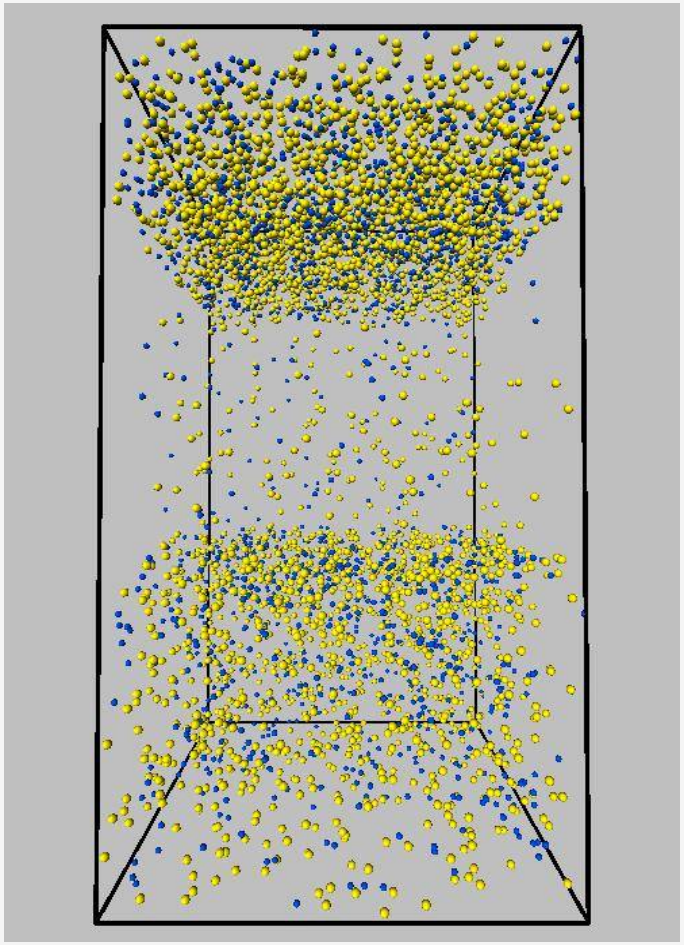




(Buck-Sorlin *et al.* 2009, 2011)

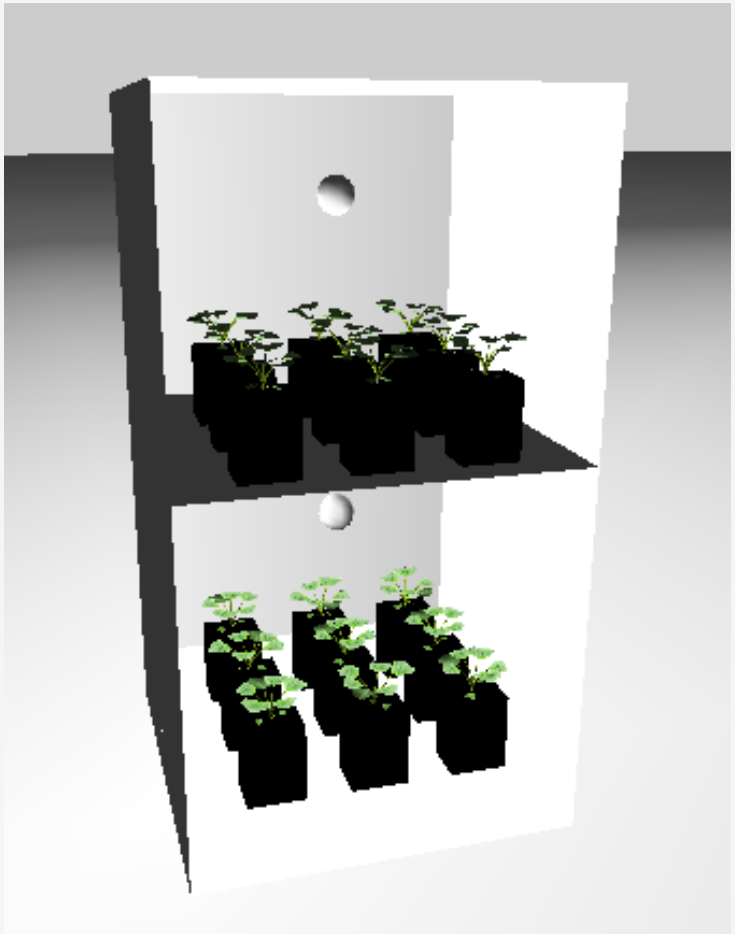
Soil / Root FSPM





(Henke, Sarlikioti, Pagès, Buck-Sorlin, Kurth; unpubl.)

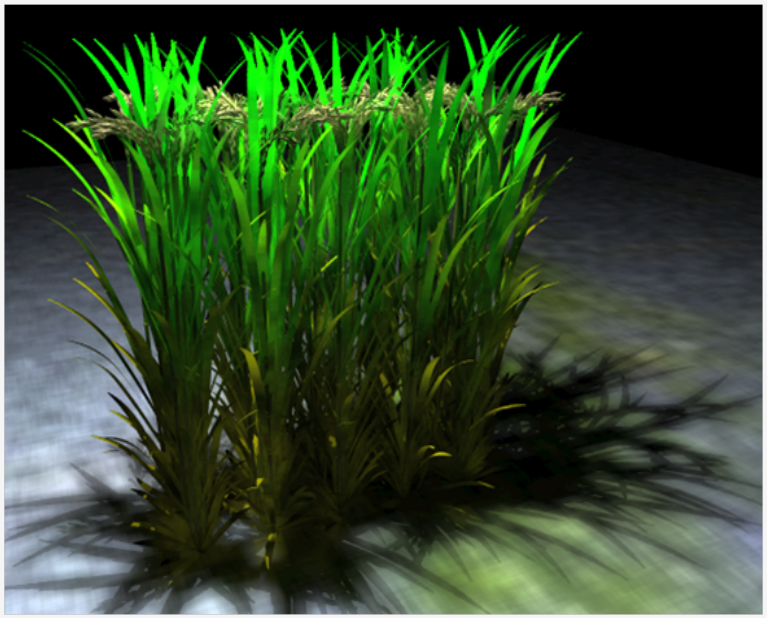
Temperature sensitive rapeseed FSPM

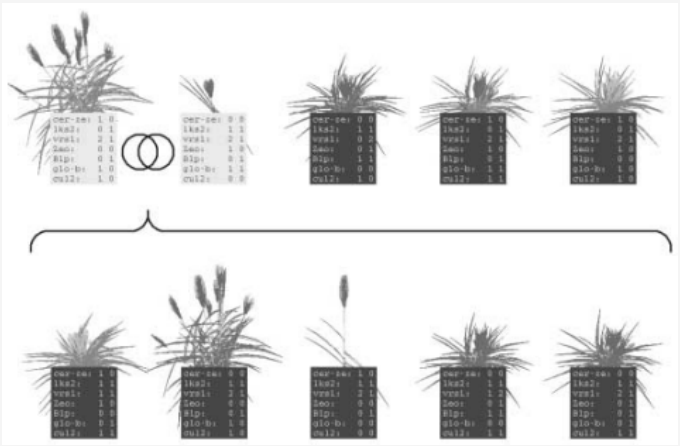


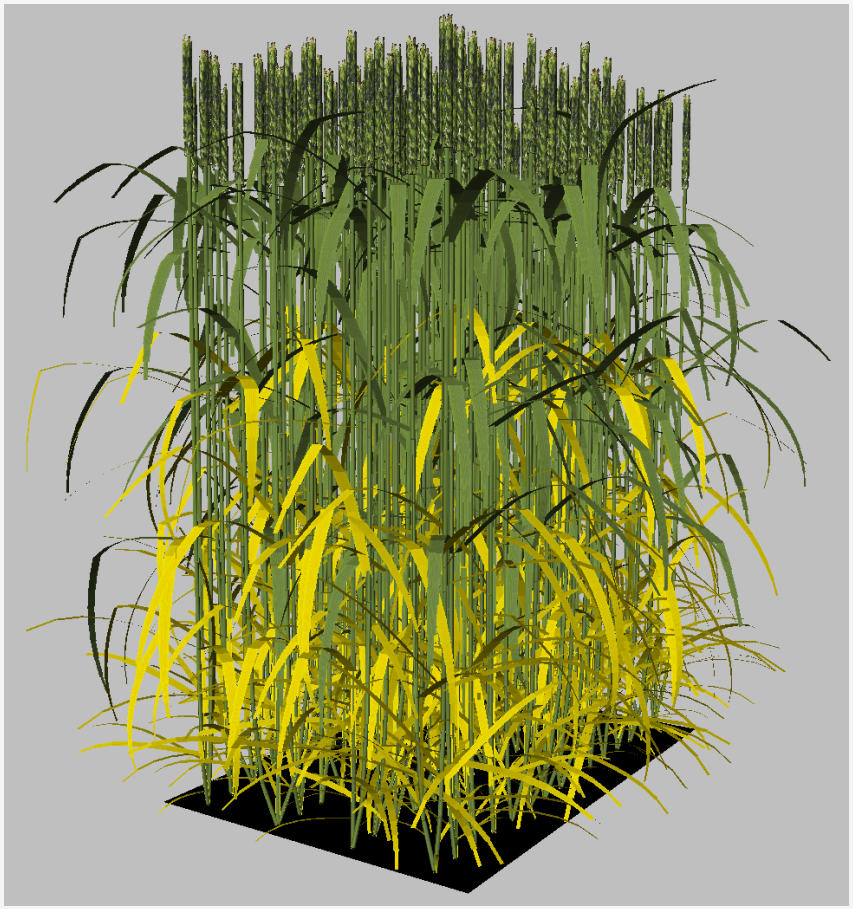
(Tian, Wu, Henke, Zhou, Buck-Sorlin; unpubl.)

Three different temperature regimes are applied to young rapeseed plants in a simulated climate chamber.

Rice, barley breeder, wheat FSPMs







(rice: Xu *et al.* 2011, barley: Buck-Sorlin *et al.* 2005, wheat: Evers *et al.* 2010)

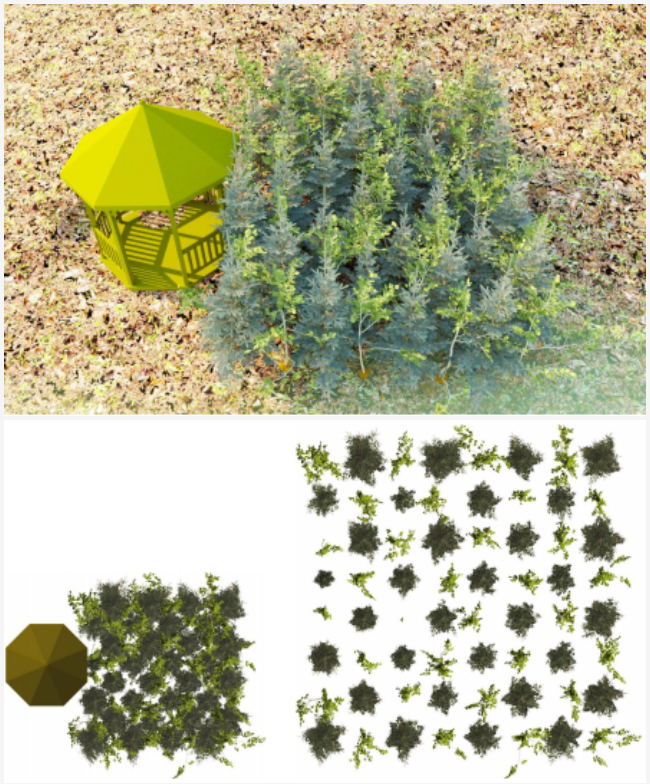
Arabidopsis, rapeseed, tomato FSPMs

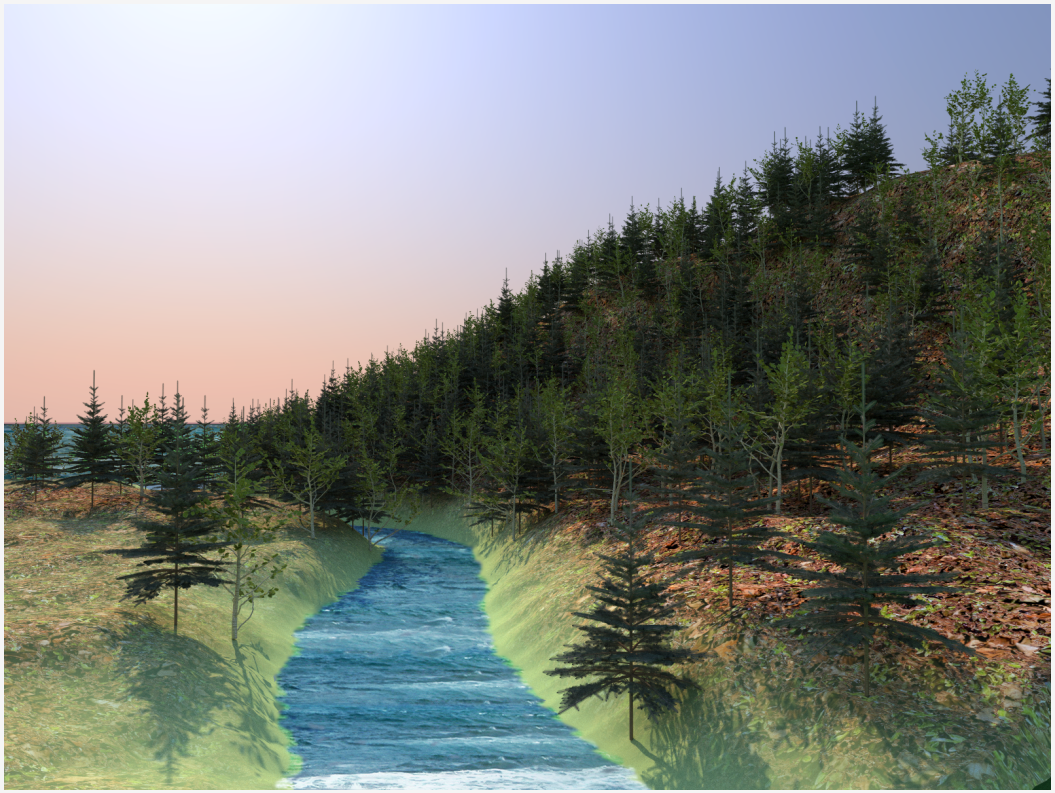
(*Arabidopsis*: Evers *et al.* 2011, rapeseed: Groer *et al.* 2007, tomato: Buck-Sorlin *et al.* unpubl.)

Modelling Park Landscapes

(Rogge & Moschner 2007, für Stiftung Branitzer Park, Cottbus.)

Tree competition (spruce and beech FSPMs)





(Hemmerling *et al.* 2008)

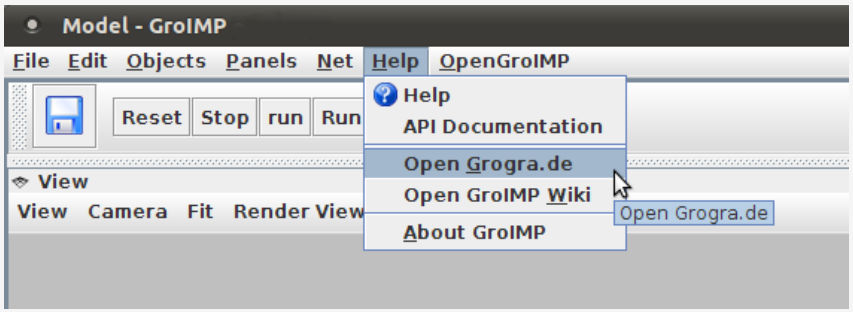
Where to find Help?

- GroIMP Project Page:

grogra.de

- GroIMP Wiki page:

http://sourceforge.net/apps/mediawiki/groimp/index.php?title=Main_Page



- E-Mail: info(at)grogra.de

Where to find Help?

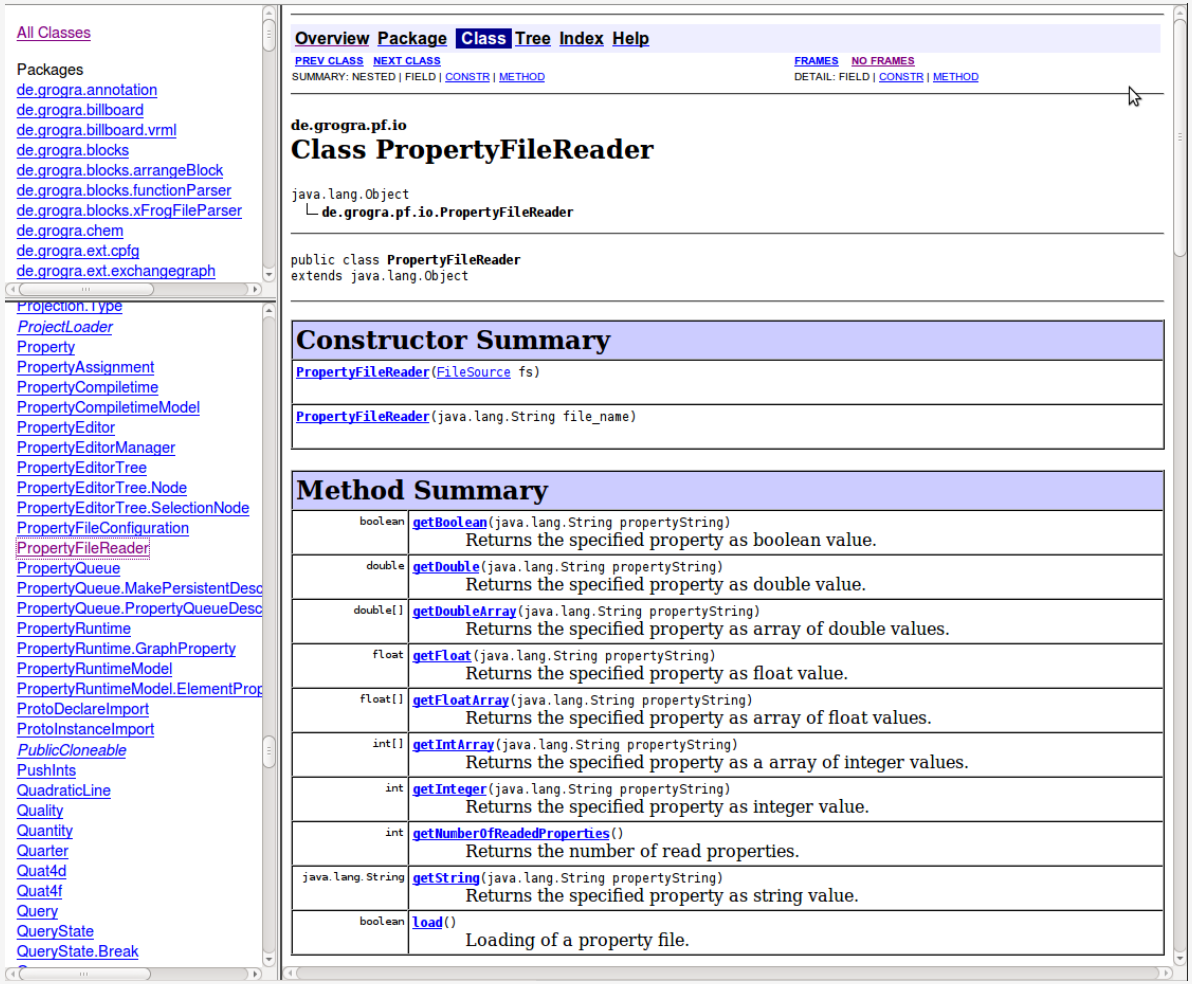
- Lectures:
  Struktur-Funktions-Modelle auf ökophysiologischer Basis (german)
- Video tutorial:
  coming soon (english & german)
- Courses and tutorials:
  Step by step models
- Example Gallery:
  At grogra.de
- Literature:
  Publications W.Kurth
  Publications grogra.de

Where to find Help?

- General XL:
  - List of turtle commands (german)
  - Overview of XL syntax (german)
- Dissertations:
  - Kniemeyer, 2008
  - Hemmerling, 2012
- Master and bachelor thesis:
  - Selected students' theses

GroIMP API

Overview over the whole code of GroIMP



All Classes

Packages
de.grogra.annotation
de.grogra.billboard
de.grogra.billboard.vrml
de.grogra.blocks
de.grogra.blocks.arrangeBlock
de.grogra.blocks.functionParser
de.grogra.blocks.xFrogFileParser
de.grogra.chem
de.grogra.ext.cpfg
de.grogra.ext.exchangegraph

Projection.Type
ProjectLoader
Property
PropertyAssignment
PropertyCompiletime
PropertyCompiletimeModel
PropertyEditor
PropertyEditorManager
PropertyEditorTree
PropertyEditorTree.Node
PropertyEditorTree.SelectionNode
PropertyFileConfiguration
PropertyFileReader
PropertyQueue
PropertyQueue.MakePersistentDesc
PropertyQueue.PropertyQueueDesc
PropertyRuntime
PropertyRuntime.GraphProperty
PropertyRuntimeModel
PropertyRuntimeModel.ElementProp
ProtoDeclareImport
ProtoInstanceImport
PublicCloneable
PushInts
QuadraticLine
Quality
Quantity
Quarter
Quat4d
Quat4f
Query
QueryState
QueryState.Break

Overview Package Class Tree Index Help

PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

de.grogra.pf.io

Class PropertyFileReader

java.lang.Object
└ **de.grogra.pf.io.PropertyFileReader**

public class **PropertyFileReader**
extends java.lang.Object

Constructor Summary

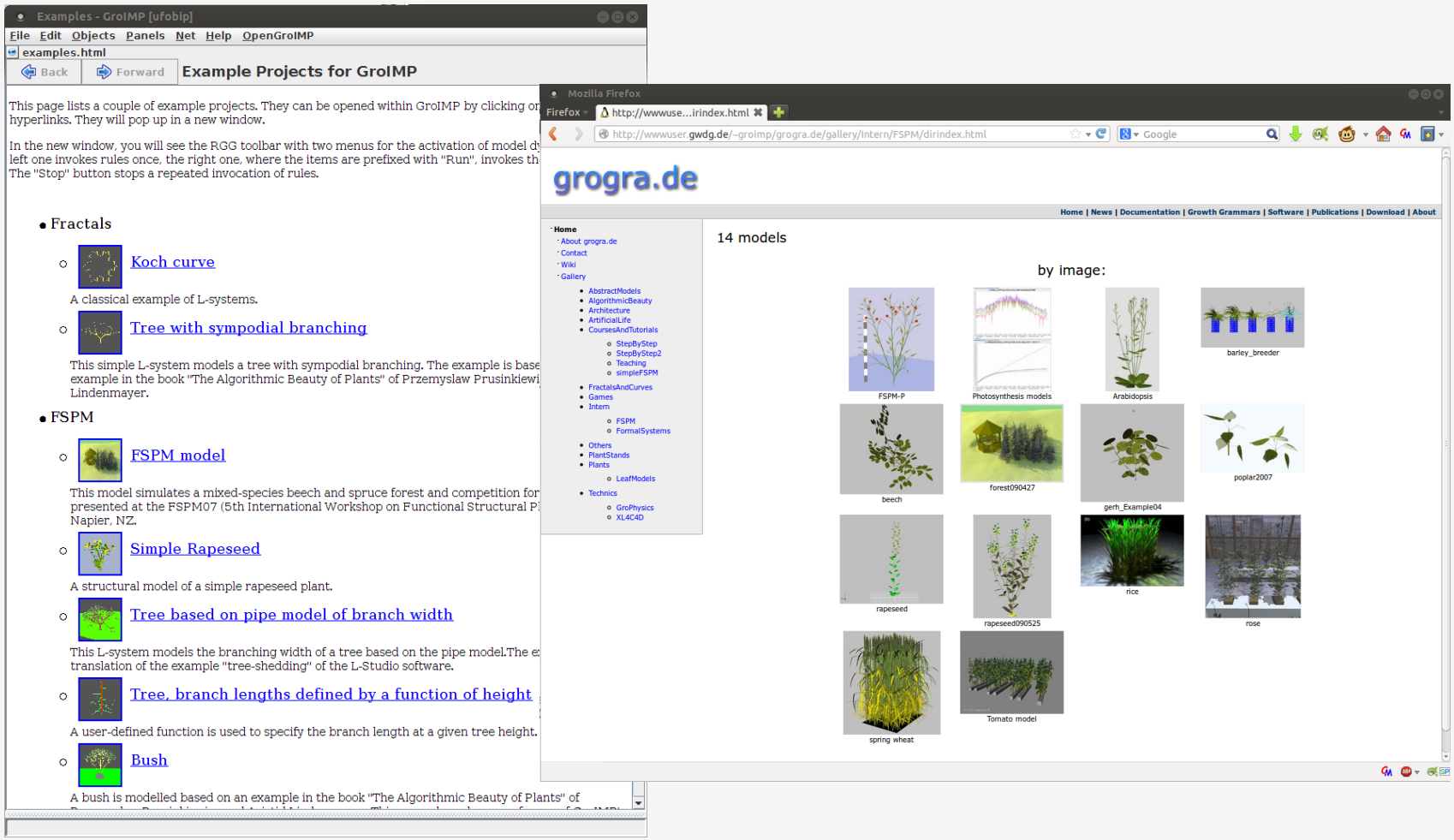
PropertyFileReader(FileSource fs)
PropertyFileReader(java.lang.String file_name)

Method Summary

boolean	getBoolean(java.lang.String propertyString) Returns the specified property as boolean value.
double	getDouble(java.lang.String propertyString) Returns the specified property as double value.
double[]	getDoubleArray(java.lang.String propertyString) Returns the specified property as array of double values.
float	getFloat(java.lang.String propertyString) Returns the specified property as float value.
float[]	getFloatArray(java.lang.String propertyString) Returns the specified property as array of float values.
int[]	getIntArray(java.lang.String propertyString) Returns the specified property as a array of integer values.
int	getInteger(java.lang.String propertyString) Returns the specified property as integer value.
int	getNumberOfReadedProperties() Returns the number of read properties.
java.lang.String	getString(java.lang.String propertyString) Returns the specified property as string value.
boolean	load() Loading of a property file.

GroIMP Example Gallery

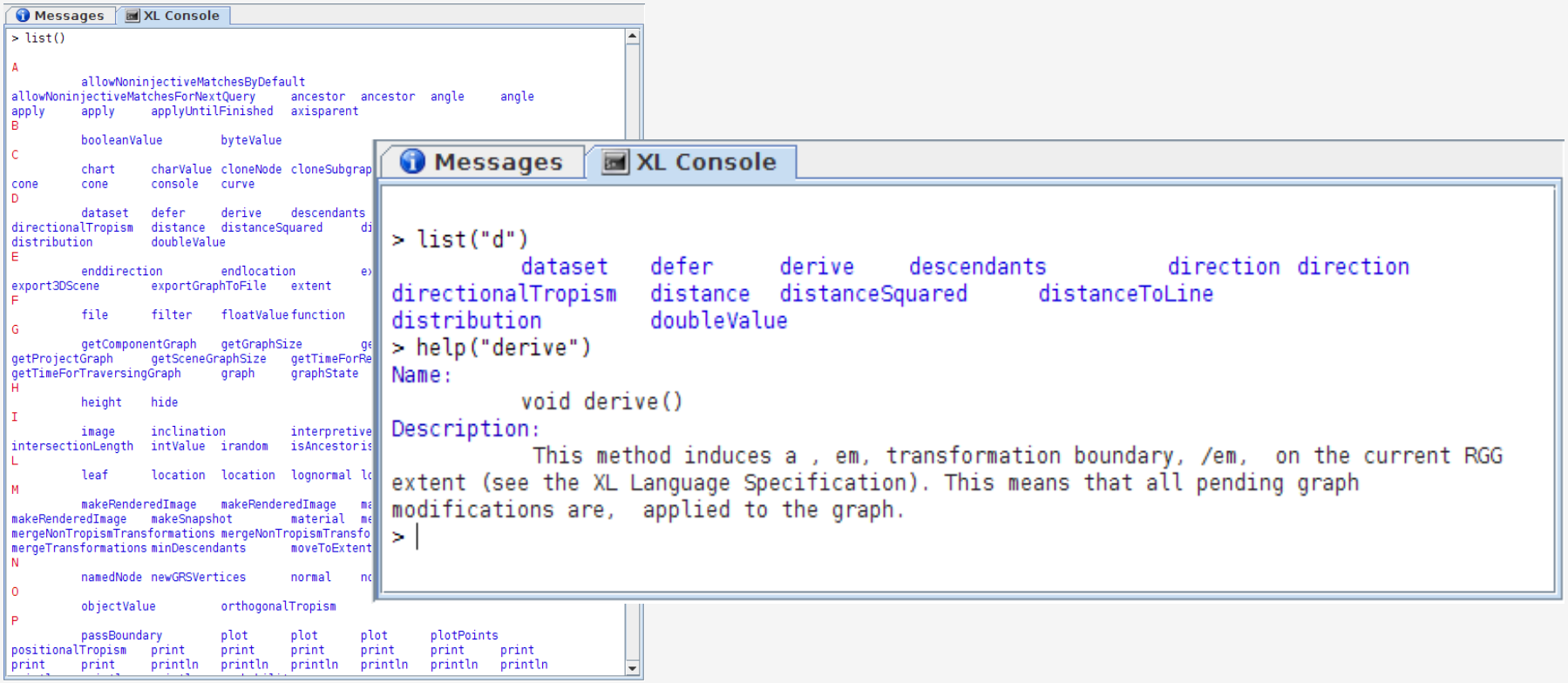
Included as ”Show Examples” or on the net with nearly 400 examples.



GroIMP Command Observer

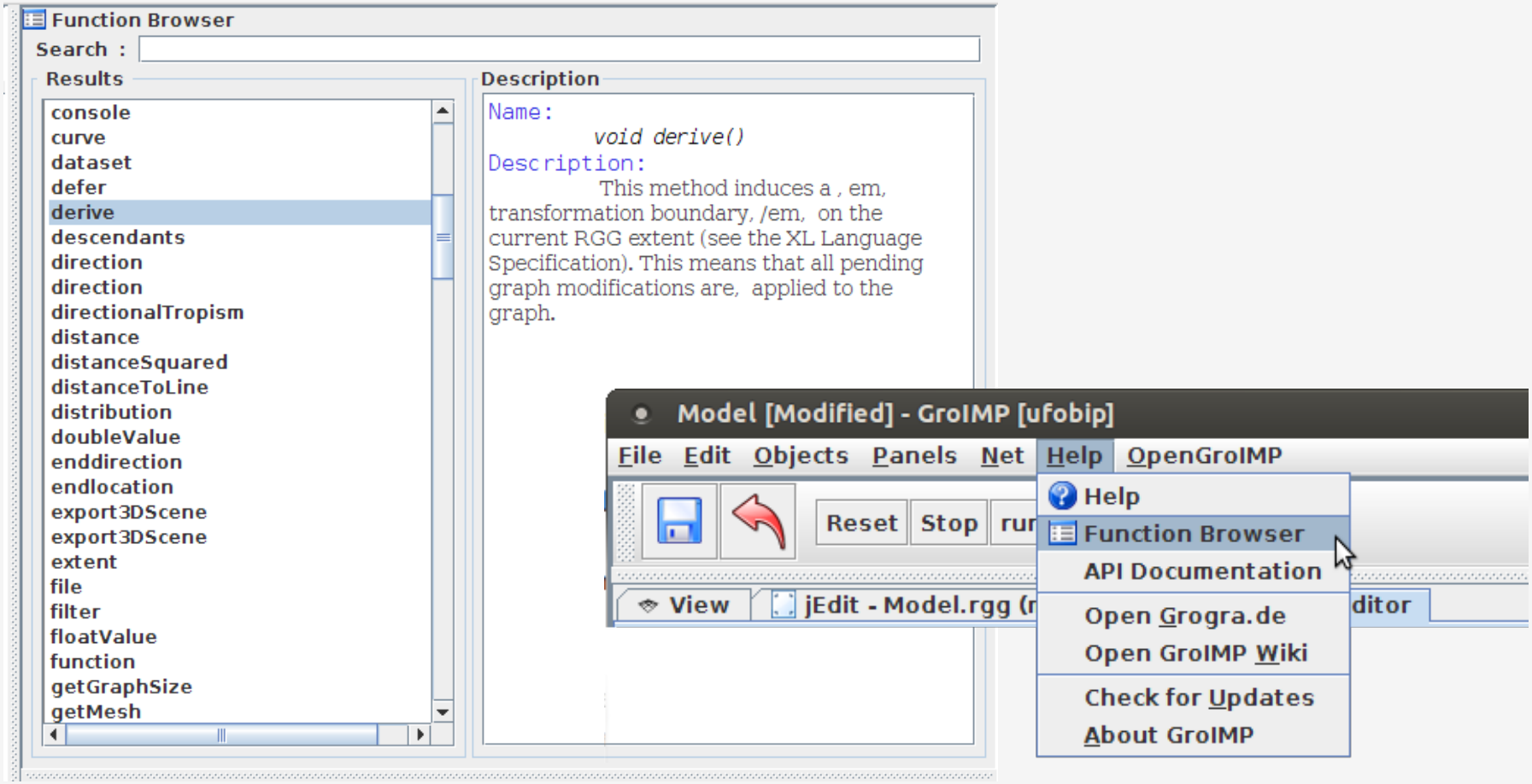
Overview and description of GroIMP commands via XL Console

list()	lists all available commands
list("key")	lists all available commands starting with the specific key
help("key")	prints out the help of the command starting with the specific key



GroIMP Function Browser

…or use the Function Browser



The Developer Team

Contact

Katarína Smoleňová	:	ksmolen@uni-goettingen.de
Michael Henke	:	mhenke@uni-goettingen.de
Yongzhi Ong	:	yong@gwdg.de
Winfried Kurth	:	wk@informatik.uni-goettingen.de

Invitation and Call for Papers

International Summer School

"Modelling of Ecosystems by Tools from Computer Science"

combined with the 6th GroIMP user and developer meeting and with a workshop

"Functional-structural plant models in XL"

Date: 16 – 20 September 2013

Start:	16 Sept., 9:00 h
End:	20 Sept., ca. 16:00 h
Workshop:	17 Sept., 14 – 17 h
GroIMP user and developer meeting:	17 Sept., 17 – 18 h

Location:

Czech University of Life Sciences (CULS / ČZU), Faculty of Forestry and Wood Sciences, Praha-Suchdol,Prague, Czech Republic

WWW:

http://www.uni-forst.gwdg.de/~wkurth/ssc_prague.pdf

Thank you for your attention!

www.grogra.de