Using the Language XL for Structural Analysis

Ole Kniemeyer [1,*] Jan-Anton Dérer [1] Reinhard Hemmerling [1]
Gerhard Buck-Sorlin [2] Winfried Kurth [1]

[1] BTU Cottbus, Department of Computer Science, Chair for Practical Computer Science / Graphics Systems, P.O.Box 10 13 44, 03013 Cottbus, Germany
[2] Wageningen UR, Crop and Weed Ecology Group
Haarweg 333, 6709 RZ Wageningen, The Netherlands
[*] corresponding author, okn@informatik.tu-cottbus.de

Keywords: Structure analysis, plant modelling, querying language, L-systems, GroIMP

The analysis of plant structures is an important issue in functional-structural plant modelling, especially in the context of parameterization and validation. This holds equally for structures resulting from measurements of real plants and for modelled structures, i.e., the outcome of virtual plant simulations. Such an analysis has to consider both the topology of the structure and the values of parameters of its constitutive entities (e.g., geometry-related parameters, internal state). For example, one may be interested in the number of internodes of growth units as a function of their age (a purely topological property) or in the length ratio of consecutive growth units, which also includes geometric information.

The AMAPmod software [1] is an example of a sophisticated program which has been specifically designed for plant structure analysis. Existing structures (results of measurements or virtual plant simulations) are read in, the user then extracts the desired information using the querying language AML, and results can be visualized as a 3D-model or via several types of plots.

We present another analysis technique based on the programming language XL within the open-source modelling environment GroIMP [2]. Although XL has been designed as an extension of L-systems, i.e., for the implementation of functional-structural plant models, it turns out to be equally suitable for the analysis of plant structures by virtue of its graph query facilities. These queries allow the search for occurrences of patterns in the current structure which is represented as a graph. The advantage of this approach is that the same language can be used to model a virtual plant, to analyze the results, and to compare it with measured data. The analysis can be invoked interactively within GroIMP's XL console, or it can be part of the source code of the model itself for automatization purposes. As an example, the statement

```
plot(int age => double count((* g:GU, (g.age == age) *)), 0:10);
```

draws a plot of the total number of growth units `GU` as a function of their age, and

```
statistics(((* g:GU h:GU, (g.order == 0) *), h.len/g.len));
```

computes mean value, standard deviation and skewness of the ratio of lengths of consecutive growth units of branching order 0. The examples make use of XL's graph queries of the form `(*...*)` and aggregate methods like `count`, `statistics` and `plot`. We will give a concrete demonstration of its usage using a model of a young beech tree, where measured tree data is analyzed in order to parameterize the model, which in turn is analyzed for validation.

References

[1] C. Godin, Y. Guédon, E. Costes, and Y. Caraglio. Measuring and analysing plants with the AMAPmod software. In M. Michalewicz, editor, *Plants to Ecosystems: Advances in Computational Life Sciences*, pages 53–84. CSIRO Publishing, Australia, 1997.

[2] O. Kniemeyer. *Design and Implementation of a Graph Grammar Based Language for Functional-Structural Plant Modelling*. PhD thesis, BTU Cottbus, 2007. (forthcoming, see http://www.grogra.de).