Sensitive Growth Grammars Specifying Models of Forest Structure, Competition and Plant-Herbivore Interaction

Winfried Kurth and Branislav Sloboda

University of Göttingen, Institute of Forest Biometry and Informatics,
Büsgenweg 4, D-37077 Göttingen, Germany
wkurth@gwdg.de
http://www.uni-forst.gwdg.de/~wkurth/plant.html

Abstract. Sensitive growth grammars are systems of rewriting rules (extended Lindenmayer systems) with graphical interpretation and with "sensitive functions" designed to allow a feedback from the created virtual 3-D structures to the subsequent rule-application process. Thus it is possible to combine morphological (genetically fixed) growth rules with environmental impact and with functions depending on the competitive situation of individuals in the framework of a precisely-specified model of plant growth. The dynamics of stand development in such models results from purely local rule application. Preliminary results are shown for three different applications in forest-ecosystem modelling: (a) Creation of irregular stand structures, (b) simulation of competitive effects on crown radius development and resulting stand dynamics, and (c) modelling the interaction between trees and herbivores, including the energy budgets of the individual plants and animals. The software system GROGRA, designed to interpret sensitive growth grammars, enables visualisation of the resulting spatial structures and provides analysis tools and data interfaces to other software.

1 Introduction

Simulation models which are able to reproduce and to predict forest structure and stand dynamics have found increasing attention in recent years [21]. Among different types of ecosystems, natural forests are characterized by a distinctly high degree of spatial heterogeneity and complex structure. Moreover, changing the spatial structure is the main method used by foresters to manipulate the development of forest stands and of individual trees [3].

Given the large number of different models in the literature, there arises the need for short and precise model specifications. Whereas simple models of whole-stand dynamics, disregarding spatial structure, can often be expressed in terms of one or several equations, models involving spatial details usually need some computer source code, written in some standard programming language, for their full specification. However, classical source code does usually involve many technical constructions distracting attention from the essentials of the model, and cannot be understood easily by users who are not computer scientists. Furthermore, the requirements of genericness and modular design of software [2] are often violated by ad-hoc models implemented by biologists or agronomists who lack specific training in software development.

Proceedings of the IUFRO 4.11 Congress "Forest Biometry, Modelling and Information Science", Greenwich (UK), June 25-29, 2001 (in press)

A way to overcome these difficulties is the design of a higher-level model specification language, adapted to the particular needs of tree and stand simulation and spatial interaction. When model specifications written in this language can be read and interpreted by a generic software, there will be no need to modify and re-compile the source code of the software each time some assumptions or relations in the model are changed. Instead, only the specifications made in the high-level language are to be modified, and these can be made open to "informed users" other than computer scientists (Figure 1). Comparisons of different models are easier if the basic software with its technical details remains the same.

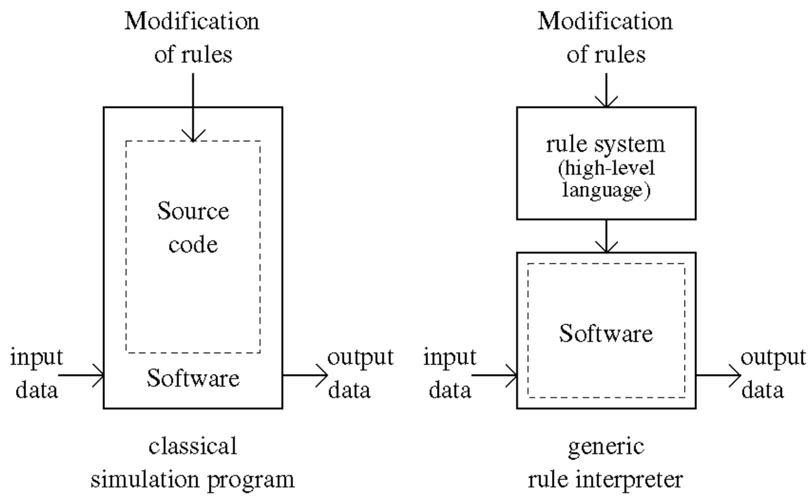

Fig. 1. Comparison of the architecture of a classical simulation model (left), where each modification requires a rewriting of the software source code, with a generic software shell for an advanced model-specification language (like extended L-systems; right)

A candidate formalism which could adopt the role of such a high-level specification language are cellular automata (CA). A CA is specified by a transition function which determines the state of a cell in a grid from the previous states of the same cell and its neighbours. CA have been used for a lot of ecological models [6], including forest models. However, their inherent preferential treatment of certain directions, spatial and temporal scales restricts their use.

Another quite general specification language for biological growth processes is the rule-based language of L-systems (Lindenmayer systems, named after the botanist Aristid Lindenmayer, 1925-1989). Originally devised to resemble growth rules of simple, filamentous plants [16], numerous extensions have been added to the formalism since then (see overviews [23, 10]). However, the main field of application of this language is still the specification of architecture of individual plants. In this paper, we want to explore the possibilities of sensitive growth grammars (extended L-systems) for spatial simulations at a lower scale of resolution. In our examples, the architecture of single trees will be highly simplified, but their spatial arrangement, competition and interaction with mobile herbivores will be taken into account. In the same way as for L-system based models of architectural development of single trees, the dynamics of development will result from purely local rule application. That means, no global curve of self-thinning or other aggregated description of stand growth is taken as input. Instead, the overall behaviour of the model will result as an emergent property from local rules (cf. [5]).

2 Sensitive Growth Grammars

An L-system consists of a set of symbols, a start symbol α and a set of *replacement rules*, each of the form "symbol $\rightarrow$ string of symbols". Additionally, there is a *geometrical interpretation* of the strings (i.e., a semantics) translating strings of symbols into structures in 3D-space. Usually, this interpretation is given by the conventions of *turtle geometry* [1]: Some symbols are used as commands for a virtual device (the turtle) which is able to move (command "f"), to produce cylindric elements while moving (command "F"), to rotate (command "RL") or to change internal parameters used for the next elements to be produced (commands "L" for length, "D" for diameter, "N" for leaf or needle mass, "P" for colour and more; see [9] for details). Before entering a pair of brackets [...], the current state of the turtle (including position, spatial orientation and all other internal parameters) is put on a stack, and is resumed after the corresponding closing bracket has been passed.

The rules of an L-system are applied in parallel to all symbols of a string at time t in order to get a new string at time t+1. This rewriting process is normally iterated several times. Thus one gets a (potentially infinite) sequence of turtle command strings s_0, s_1, s_2, ..., where s_{t+1} is obtained from s_t by application of the rules, and $s_0 = \alpha$. The string s_t is interpreted in terms of turtle geometry, resulting in a geometrical model of a single plant or stand at time t. Proceeding in discrete time steps, we obtain a developmental sequence of geometrical structures. In models of individual plants, the time step often corresponds to 1 year or even to shorter periods of growth, whereas in our application examples the time step will represent a period of several years of stand development. Figure 2 shows the strings and geometrical structures resulting from the application of a very simple, classical L-system describing the growth of a branching system. The start symbol is "a". The L-system contains only two rules: a $\rightarrow$ F [RU45 b] a and b $\rightarrow$ F b. The symbols a and b, standing for apical buds of main and lateral branches, respectively, are normally not interpreted by the turtle. In our picture, we have visualized the corresponding buds by ovals. Such a visual interpretation can be specified by using an additional sort of rules, so-called interpretive rules [9]. – Horizontal arrows in Fig. 2 stand for rule application, vertical arrows for interpretation of the strings by the turtle. The L-system contains one symbol with a parameter: RU, with the subsequent number specifying the rotation angle in degrees. Parameters can also be attached to other symbols like a and b [22]. – Several further extensions to classical L-systems will be mentioned in the subsequent examples where they are needed. Together, these extensions have lead to the notion of "stochastic, sensitive growth grammar" [9].

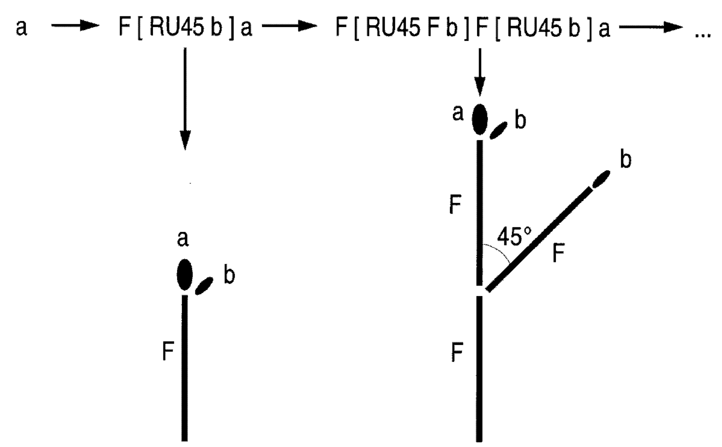

Fig. 2. Development of geometrical structures specified by the growth grammar
`a → F [ RU45 b ] a , b → F b` (see text)

3 Specification of Irregular Stand Structures

A first application of the grammar formalism at stand level is the specification of patterns of tree positions, disregarding dynamic aspects. Models of stand structure have often used stochastic approaches like point processes [17] or heuristic algorithms [15]. Extended L-systems provide a framework for a transparent specification of such models. Let us first focus on tree positions only, disregarding all other attributes like height, diameter, tree species etc. The simplest grammar rule for an irregular stand generates a random pattern of tree seedlings:

`stand → &(n) < [` *move_to_random_position* `seedling ] >`,

where `n` is the number of seedlings dispersed on the stand area and & a repetition operator which iterates the string enclosed in < ... > *n* times. Movement to a random position within, let us say, a rectangular area of extensions x_{extens} and y_{extens} can be specified by declaration of a uniformly-distributed random variable,

`\var rvar uniform 0 1`,

which takes values between 0 and 1, and by the turtle command sequence

`+ f(rvar*x_extens) RL90 f(rvar*y_extens) - .`

Here, "+" and "-" are abbreviated versions of the rotation commands `RU90` and `RU-90`, resp., bringing the turtle from vertical to horizontal mode of movement and vice versa. `RL90` enforces a rotation to the left by 90 degrees. When this string is inserted into the above replacement rule instead of "*move_to_random_position*", the result will be a random distribution of seedling positions with uniform distribution of *x* and *y* coordinates (i.e., a "Poisson forest"). However, when growing up, close neighbours will normally be outcompeted. To get a pattern where a minimum distance between trees is respected, we can modify the grammar by specifying the height of the juvenile tree as a random attribute of the symbol "seedling", e.g. `seedling(juv_h)` with

```
\var juv_h uniform 5 20,
```

and by introducing a sensitive function `dist(hg)` which gives back the distance to the next tree not smaller than the height (*hg*) of the considered seedling itself. By the conditional rule

```
(dist(hg) > inhib_r) seedling(jh) → tree(height),
```

we can specify that the development of the seedling to a mature tree will only take place if there is no higher competitor inside a circle with radius "*inhib_r*". This results in an arrangement of mature trees where positions are random, but no two trees are closer than the minimal distance *inhib_r*. Figure 3 shows a typical resulting pattern of seedlings (without minimal distance; upper part) and the corresponding pattern of mature trees after one additional step of rule application (with minimal distance; lower part). The complete grammar with 10 declarations and 7 rules is shown in Table 1.

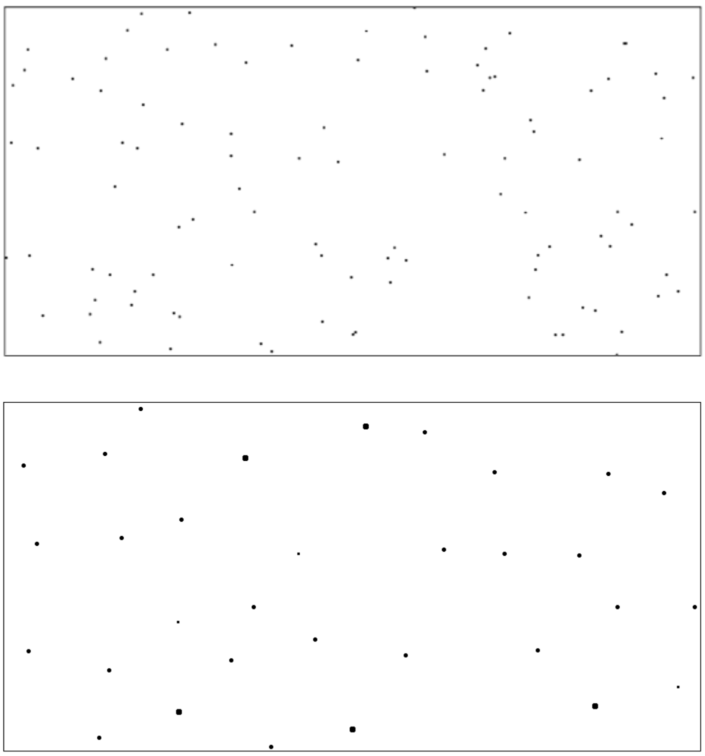

Fig. 3. Result of the growth grammar `irreg` (Table 1) after 2 steps (pattern of seedlings; upper part) and after 3 steps (pattern of mature trees, lower part). View from above

Table 1. The sensitive growth grammar `irreg`

```
/* Stand with random coordinates, close neighbourship
   excluded */

\const x_extens 2000,   /* Extension of stand */
\const y_extens 1000,
\const nbseedl 100,     /* initial number of seedlings */
\var rvar uniform 0 1,  /* uniformly-distributed random
                           variable */
\var juv_h uniform 5 20,     /* heigth of seedlings */
\var height normal 250 1000, /* heigth of trees (Gaussian
                                distribution)*/
\var dist function 2 1, /* distance function with min length
                           argument */
\const inhib_r 150,     /* distance which inhibits growth */
\angle 90,
\var hg length,

/* Generative rules: */
* → stand(x_extens, y_extens, nbseedl),
stand(x, y, n) → border(x, y) P2
 &(n) < [ + f(rvar*x) RL-90 f(rvar*y) - seedling(juv_h) ] >,
(dist(hg) > inhib_r) seedling(jh) → tree(height),
seedling(jh) → ,    /* mortality of outcompeted seedlings */

/* Interpretive rules: */
border(x, y) ⇒ [ P14 + F(x) RL-90 F(y) RL-90 F(x)
                  RL-90 F(y) ],
seedling(jh) ⇒ D6 P11 F(jh),
tree(h)      ⇒ D(0.05*h) F(h)
```

In this example, the height of the mature tree was assumed to be independent from that of the seedling, and is supposed to follow a normal distribution with given mean and given variance. However, it would be easy to include in the rule system some arithmetical expression relating the heigth of the seedling and the heigth of the mature tree, thus reflecting the well-known phenomenon of rank preservation [24]. E.g., the third rule in Table 1 could be replaced by

```
(dist(hg) > inhib_r) seedling(jh) →
                          tree(factor*jh + epsilon),
```

where "*factor*" is an empirical constant and "*epsilon*" an error term which can be declared as a random variable with normal distribution.

Another refinement of the model can be made by including a tendency of clustering. Instead of spreading the seedlings directly at random, we can simulate a two-phase stochastic process by first spreading clusters of seedlings:

```
stand →
  &(n) < [ move_to_random_position cluster(rcl) ] >,
```

and then by expanding each cluster into a circular area where seedlings are distributed with random polar coordinates:

```
\var rr uniform 0 360, /* random rotation angle */
\var rvar uniform 0 1,
             /* random variable between 0 and 1 */
cluster(r) →
    &(sd_per_cl) < [ RH(rr) + f(rvar*r) –
                     seedling(juv_h) ] >.
```

Here, "*sd_per_cl*" is the number of seedlings per cluster, which again can be declared as random variable, e.g. following a binomial distribution with given parameters *p* and *n*:

```
\var sd_per_cl binomial 0.5 25.
```

"`RH(rr)`" denotes a random rotation, and "`f(rvar*r)`" a random movement in the interior of the circle with radius *r*, where the start point is the centre of the cluster. An additional sensitive rule can be specified which deletes seedlings which fall by chance outside the borders of the stand. Figure 4 shows the resulting distribution of clusters (second step; upper left part of the Figure), seedlings (third step; lower left part) and mature trees (fourth step; upper right part), all seen from above, and a view on the mature trees, represented as cylinders, from an oblique angle (lower right part). Trees from different clusters have different grey tones.

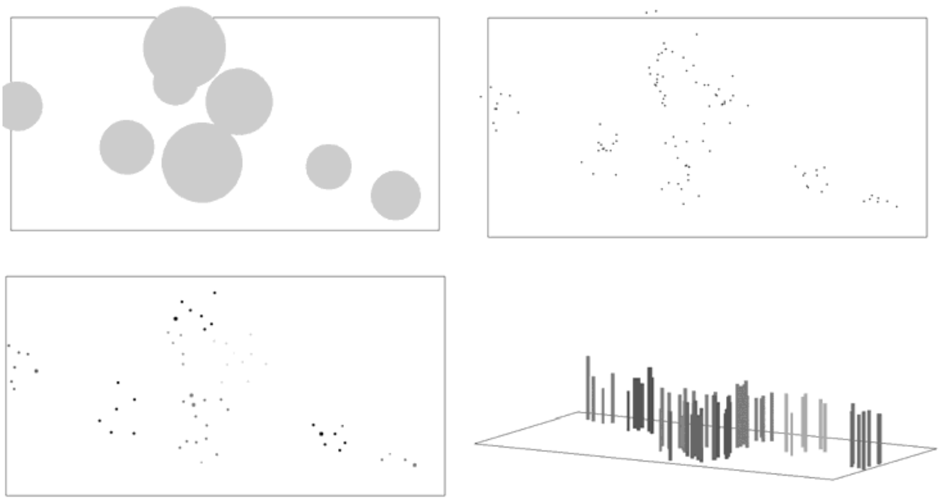

Fig. 4. Result of the growth grammar `clusters` (Table 2) after 2, 3 and 4 steps, seen from above, and after 4 steps from oblique perspective

Like in the previous example, the trees respect a minimum distance, which is smaller than the cluster radius. The complete grammar is shown in Table 2.

Table 2. The sensitive growth grammar `clusters`

```
/* Stand with randomly distributed clusters of trees,
   close neighbourship excluded */

\const x_extens 2000,       /* Extension of stand */
\const y_extens 1000,
\const nbclust 8,           /* number of clusters */
\var rcl uniform 100 200,   /* radius of cluster (random) */
\var sd_per_cl binomial 0.5 25,/*seedlings per cl.(random)*/
\var juv_h uniform 5 20,    /* heigth of seedlings */
\var height normal 250 1000,/* heigth of trees */
\const inhib_r 35,      /* distance which inhibits growth */
\var dist function 2 1, /* distance function
                           with min length argument */
\var i index,
\var rvar uniform 0 1,   /* random factor between 0 and 1 */
\var rr uniform 0 360,   /* random rotation */
\var xx xcoordinate,
\var yy ycoordinate,
\var hg length,
\angle 90,

/* Generative rules: */
* → stand(x_extens, y_extens, nbclust),
stand(x, y, n) → border(x, y)
    &(n) < [ + f(rvar*x) RL-90 f(rvar*y) - P(i+2)
             cluster(rcl) ] >,
cluster(r) → &(sd_per_cl) < [ RH(rr) + f(rvar*r) -
                               seedling(juv_h) ] >,
(xx < 0 || yy < 0 || xx > x_extens || yy > y_extens)
    seedling(jh) → ,
    /* delete seedlings fallen out of stand limits */
(dist(hg) > inhib_r) seedling(jh) → tree(height),
seedling(jh) → ,     /* delete outcompeted seedlings */

/* Interpretive rules: */
border(x, y) ⇒
            [ P14 + F(x) RL-90 F(y) RL-90 F(x) RL-90 F(y) ],
cluster(r)   ⇒ D(2*r) P9 F10,
seedling(jh) ⇒ D6 P11 F(jh),
tree(h)      ⇒ D(0.05*h) F(h)
```

By defining a tree object in a sub-grammar (not shown), using the feature of "object instancing" provided by the software system GROGRA (see [13]), it is easy to improve the visual design of the resulting stand (Figure 5). Of course, further improvement of this sort of output, using techniques from computer graphics provided in established software systems, is possible, leading to photo-realistic visualisations of forests and landscapes [8].

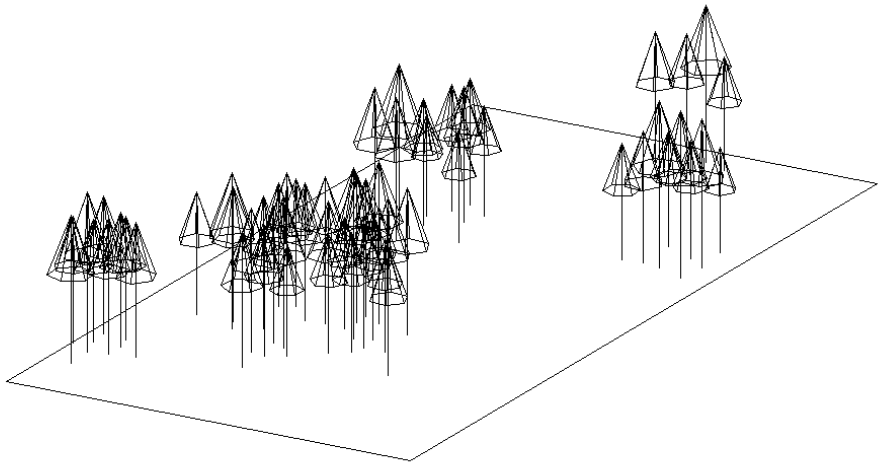

Fig. 5. A stand with clustered structure and visual standard tree objects, defined in a sub-grammar of the stand model

4 A Model of Crown Radius Dynamics Under Competition

The previous models contained only a very coarse representation of interactions between neighbours. If more detailed information about crown dimensions is available, a sensitive function can be used to simulate the reaction of crown radius to the presence of competitors. In an ad-hoc model, i.e., without using a specification in terms of a growth grammar or other higher-level language, this approach was used by Pretzsch [19, 20]. He described the dynamics of horizontal crown expansion, using 8 predefined directions, in dependence upon distance to neighbouring trees. To formalize this approach, we use a sensitive function `sf` which returns the distance to the next geometrical element inside a 45° sector along the direction of the considered crown radius. If this distance is large enough, i.e., above a given threshold *ds*, then the considered crown radius continues to grow (parameter *c*=1) and increases its length to *r*+1:

```
(sf(ang) > ds) s(c, r) → s(1, r+1),
```

otherwise, a status of "shrinking" (*c*=2) is assumed and the radius is shortened by 0.3 length units:

```
s(c, r) → s(2, r−0.3).
```

(Notice that the second rule is applied in each case when the first one is not applicable.) We can further make the assumption that the crown radii in the status of "shrinking" are counted for each tree, and if this number exceeds 5, the tree is removed because of deadly suppression by its competitors. In the grammar, this mechanism is realized by a local variable *x* counting the shrinking crown radii (i.e., with crown radius status *c*=2), and by a "cut operator" ("`%`") which switches off the turtle interpretation of subsequent symbols. Table 3 shows the complete grammar, and Figure 6 shows its application to a rectangular, regularly-spaced stand of 180 trees. Each tree is represented by its 8 crown radii.

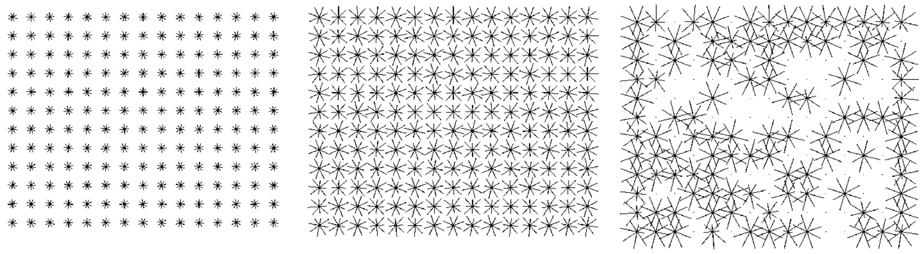

Fig. 6. Results of the sensitive growth grammar `radii` (Table 3) after 4, 8 and 12 steps. In the rightmost part, several trees have died because of too many shrinking radii due to competition, and other trees begin to invade the resulting gaps with their crowns. From [10]

Table 3. The sensitive growth grammar `radii`

```
\const ds 3,   /* threshold distance for competition */
\const dp 12,  /* distance between the planting positions */
\const ang 22.5,        /* opening angle of sensitive cone */
\var rr uniform 0 360, /* random rotation */
\var i index,
\var sf function 9 1,  /* competition function */
\var x local 0,        /* count of shrinking crown radii
                          of each tree */
\var col color,
* → &(15) < [ &(12) < [ F(0) K(x) tree ] RU90 f(dp)
                         RU-90 > ] RL-90 f(dp) RL90 >,
tree → A(x, sum(col=2, 1)) cut RU90 RL(rr)
        &(8) < [ RL(i*45) s(1, 1) ] >,
(sf(ang) > ds) s(c, r) → s(1, r+1),
s(c, r) →  s(2, r-0.3),
(x >= 6) cut → %,
s(c, r) ⇒ Pl(c) F(r)
```

Note that the only stochastic component in this model is the initial orientation of the "star" of 8 crown radii representing a tree. This is sufficient to generate random gaps in the aged stand.

5 A Model of Plant-Herbivore Interaction

The following example was inspired by a model constructed by Breckling [4] (who did not use grammars), and is explained in further detail in [11]. Going further beyond the previous examples, we now include the natural reproduction of plants by spreading of seeds. A plant is represented by the symbol `p` and has two parameters, age *t* and size *r*. Size is assumed to be proportional to carbon content or energy content of the plant. Geometrically, plant `p(t, r)` is represented by a circle with radius *r*. We use some heuristic rules for mortality (this example does not intend to represent a refined model of carbon metabolism):

```
(t > pmaxage) p(t, r) →
```

This rule is applied when the plant has reached a given maximal age, *pmaxage*. The right side of the rule is empty, i.e., the plant disappears (mineralisation and nutrient cycle are not represented in this model).

```
(r < 0) p(t, r) →
```

This means that the plant dies because of negative carbon budget.

```
(sh > 0) p(t, r) →
```

Here, we use a sensitive function `sh` ("shadow"), which returns a positive number when the plant is situated inside the crown radius of a larger plant. We assume in this case, similar to the example `irreg` above, that the plant is outcompeted. This model of competition for light is, of course, much simpler than many approaches which are described in the literature (e.g., [18]). See [14] for grammar representations of more detailed light competition models.

If none of the rules for mortality is applicable, the plant grows, and its age is increased by 1:

```
p(t, r) → p(t+1, rad+pgrow).
```

The amount of growth, *pgrow*, is a constant. The new radius, *rad*, is hold as a local variable which is accessible by objects described in other rules (e.g., representing herbivores). Therefore, we do not simply use *r*, but instead *rad* has to be initialized by *r* in an interpretive rule which specifies at the same time the graphical representation of the plant as a circle (not shown). Finally, there is a rule for reproduction. It is activated if one of two fixed (arbitrary) age stages is reached, and if in the same time *r* is above a given threshold. This condition is expressed using the logical operators || (or) and && (and), taken from the language C [7]:

```
(((t = pgenage1) || (t = pgenage2))
                        && (r >= pminrad))
     p(t, r) → &(pgenfac*rad)
                < [ RH(rr) + f(dist) – p(0, 0) ] >
                p(t+1, rad)
```

The repetition operator & says how many seeds are dispersed, their number being proportional to the radius of the plant. Spreading of seeds (`p(0, 0)`) is done in a similar way than in the example clusters above; *ro* and *dist* are random variables. The mother plant ages, but does not grow (`p(t+1, rad)`). Already with this simple grammar, consisting of 5 generative and one interpretive rule, a richness of spatial patterns emerges. We obtain clusters of smaller plants and larger gaps which are later again invaded by plants. Figure 7 shows, as an example, a stand which has evolved since 54 time steps from a single plant which was located near the centre of the picture.

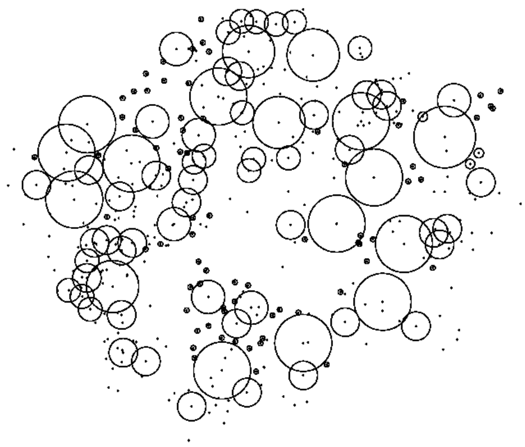

Fig. 7. Stand without herbivores after 54 steps of development, resulting from the plant grammar described in the text. Interaction between plants happens by the "shadow" rule. Points without surrounding circle represent seeds which will sprout in the next time step

We can add further rules to represent animals which take their energy for living from the plants. An animal is symbolized in the grammar by `a(t, e)`, where *t* is age and *e* the reserve of carbon or energy. The animals are represented graphically by small circles. To have a simple reproduction rule, we assume that the animals reproduce by division. In the real world, we can think of some microorganisms (bacteria, fungi) behaving this way. This time, there is only one mortality rule:

```
(e <= 0) a(t, e) →
```

(again, the right side is empty; organic matter from the dead animal is not fed back into the system.) In contrast to the plants, the animals are mobile; they perform a random walk which is influenced only by the presence of plants. If an animal is not in contact with a plant, it is in a "search" status and makes long steps (distance "*long*"), causing a loss of energy ("*respi*"):

```
a(t, e) → RH(rr) + f(long) – a(t+1, e–respi)
```

Preceding this rule, we specify a rule which is applied when the animal has come into contact with a plant. This condition is checked by a sensitive function *f*:

```
(f > 0) a(t, e) →
          RH(rr) + f(short) – a(t+1, e+eat–respi)
          Ar+(rad, –eat)
```

Here, the step of movement is shorter than in the case of search for food.
The energy budget is diminished by "*respi*" and increased by an amount "*eat*" which is taken from the plant (assignment command `Ar+` with "*–eat*" as argument for the plant). Here, the grammar specifies a sort of communication between two objects. Reproduction takes place when an animal is large enough:

```
(e > thr) a(t, e) →
      [ RH(rr) + f(short) – a(0, e/2 – respi) ]
        RH(rr) + f(short) – a(0, e/2 - respi)
```

Both children move away in random directions and get *e*/2, half of the energy content of the parent, diminished by respiration.

Only some start rules, an interpretive rule for the graphical appearance of the animals and some declarations of parameters (not shown) have to be added to the given rules to complete the sensitive growth grammar `phytophag`. Figure 8 shows two possible results after 160 steps, obtained with different parameterizations. Both simulation runs started with one plant and one animal. We see that complex spatial patterns can emerge. The dynamics in time does also depend on the choice of parameters. Figure 9 shows two examples: In the simulation run depicted on the left side, the system collapses, i.e., the plants (and later, inevitably, also the animals) die out because of too much grazing. In the run depicted on the right side, the plant population recovers after a while.

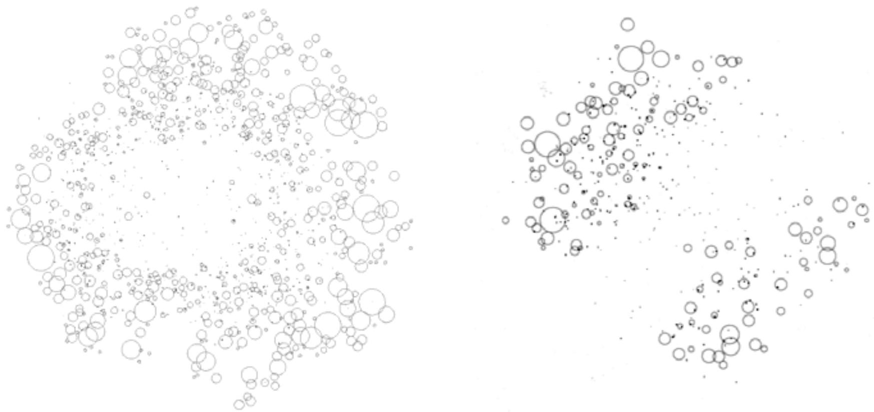

Fig. 8. Results of the grammar `phytophag` (described in the text) after 160 time steps. Both simulation runs started with one plant and one animal and differ in the parameterization. Grey circles: plants, black points: animals

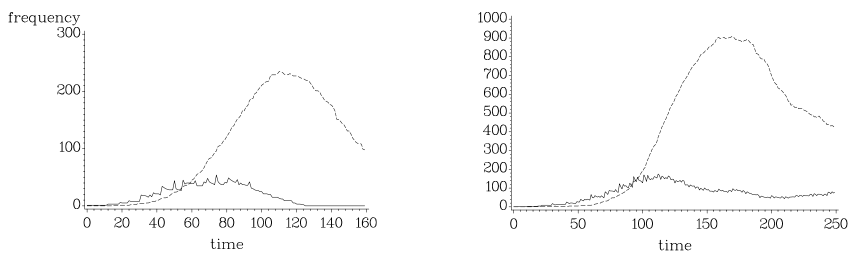

Fig. 9. Development of the numbers of individuals (broken line: animals, smooth line: plants) in two simulation runs with the growth grammar `phytophag`, differring in their parameterization

6 Discussion

Further studies would be necessary to explore the parameter spaces of the presented models systematically. This was not the aim of this overview. Instead, it was intended to demonstrate the descriptive power of the formalism of sensitive growth grammars for the specification of various types of stand models – from simple descriptions of spatial patterns (using point processes and similar models as the mathematical basis) to population dynamics. Several advantages of the approach are obvious:

- All results were obtained with one and the same software tool (GROGRA 3.3, see [9, 10]) which had not to be recompiled for the different grammars. The grammars, which specify the essential features of the models including parameterization, are easy to manipulate.
- To a certain degree, the rules are intuitively comprehensible and describe directly the behaviour of plants and animals (growth, reproduction, seed dispersal).
- Not only the global behaviour of the simulated stand, but also "local histories" of certain trees or regions in the model plane can be investigated. Thus, comparisons with intensively-monitored research plots in real forests are possible. Furthermore, individual treatment of highly-valued trees can be simulated.
- The appearance of singularities (catastrophes), e.g. breakdown of a stand from herbivore attack, can be studied in detail (cf. [4]).
- The universality of the approach is made plausible by the successful re-implementation of plant models from the literature which were originally not specified in terms of grammars and which can now all be studied in the form of growth grammars using the same software shell [12].

Some possible extensions are:

- the inclusion of several trophic levels (predators),
- more refined rules for foraging and reproduction of herbivores,
- more detailed growth and competition models for the trees, using approaches from the literature which are until now not implemented as growth grammars,
- transmission of (simple forms of) genetic information in the reproduction rules.

A weakness of the growth grammar formalism in its current form is the treatment of sensitive functions, which enable the inclusion of external influences and of competition and communication. These functions cannot yet be completely specified in the grammar itself; a link to procedurally-defined functions (as part of the GROGRA software) is still necessary. Thus, a part of the flexibility and independence guaranteed by the grammar approach is lost again. It is a topic of current research to extend the grammar formalism by possibilities to specify such functions independently from the source code of the software, using high-level language constructions adapted to phenomena of competition and communication in living systems.

Acknowledgements

The authors whish to thank Helge Dzierzon, Broder Breckling and Hans Pretzsch for fruitful discussions and inspiration. Funding was obtained from BMBF and DFG. All support is gratefully acknowledged.

References

1. Abelson, H., diSessa, A.A.: Turtle Geometry. MIT Press, Cambridge (1982).
2. Acock, B., Reynolds, J.F.: Introduction: Modularity in Plant Models. Ecological Modelling **94** (1997) 1–6.
3. Bormann, F.H., Likens, G.E.: Pattern and Process in Forested Ecosystems. Springer, New York (1979).
4. Breckling, B.: Singularität und Reproduzierbarkeit in der Modellierung ökologischer Systeme. Dissertation, University of Bremen, FB2 (Biology/Chemistry) (1990).
5. Breckling, B.: An Individual Based Model for the Study of Pattern and Process in Plant Ecology: An Application of Object Oriented Programming. EcoSys **4** (1996) 241–254.
6. Ermentrout, G.B., Edelstein-Keshet, L.: Cellular Automata Approaches to Biological Modelling. Journal of Theoretical Biology 160 (1993) 97–133.
7. Kernighan, B.W., Ritchie, D.M.: Programmieren in C. 2nd edn. Hanser, London (1990).
8. Knauft, F.J.: Entwicklung von Methoden zur GIS-gestützten Visualisierung von Waldentwicklungsszenarien. Dissertation, University of Göttingen, Fakulty of Forest Sciences and Forest Ecology (2000). webdoc.sub.gwdg.de/diss/2000/knauft/inhalt.htm.
9. Kurth, W.: Growth Grammar Interpreter GROGRA 2.4. Forschungszentrum Waldökosysteme der Universität Göttingen, Ser. B **38** (1994).
10. Kurth, W.: Die Simulation der Baumarchitektur mit Wachstumsgrammatiken. Wissenschaftlicher Verlag Berlin, Berlin (1999).
11. Kurth, W.: Spezifikation räumlicher Bestandes- und Populationsmodelle mit sensitiven Grammatiken. In: Saborowski, J., Sloboda, B. (eds.): Dt. Vb. Forstl. Forsch.anst., Sekt. Forstl. Biometrie u. Inf., 12. Tagung 1999. Biotechn. Fak. Ljubljana (2000) 259–278.
12. Kurth, W.: Towards Universality of Growth Grammars: Models of Bell, Pagès, and Takenaka Revisited. Ann. For. Sci. **57** (2000) 543–554.
13. Kurth, W., Sloboda, B.: Tree and Stand Architecture Described by Formal Grammars. I. Non-sensitive Trees. J. Forest Sci. **45** (1999) 16–30.
14. Kurth, W., Sloboda, B.: Tree and Stand Architecture Described by Formal Grammars. II. Sensitive Trees and Competition. J. Forest Sci. **45** (1999) 53–63.
15. Lewandowski, A., von Gadow, K.: Ein heuristischer Ansatz zur Reproduktion von Waldbeständen. Allg. Forst- u. Jagdzeitg. **168** (1997) 170–174.
16. Lindenmayer, A.: Mathematical Models for Cellular Interactions in Development. Journal of Theoretical Biology **18** (1968) 280–299, 300–315.
17. Penttinen, A., Stoyan, D., Henttonen, H.M.: Marked Point Processes in Forest Statistics. For. Sci. **38** (1994) 806–824.
18. Pfreundt, J., Sloboda, B.: The Relation of Local Stand Structure to Photosynthetic Capacity in a Spruce Stand: A Model Calculation. Lesnictví / Forestry **42** (1996) 149–160.
19. Pretzsch, H.: Kronenformen und ihre Veränderung unter Konkurrenz. In: Jahrestagung 1992 d. Sekt. Ertragskunde des Dt. Vb. Forstl. Forschungsanstalten, Grillenburg 40–61.
20. Pretzsch, H.: Modellierung der Kronenkonkurrenz von Fichte und Buche in Rein- und Mischbeständen. Allg. Forst- u. Jagdzeitg. **163** (1992) 203–213.
21. Pretzsch, H.: Models for Pure and Mixed Forests. In: Mies, E. (ed.): Natural and Socioeconomic Analysis and Modelling of Forest and Agroforestry Systems in Southeast Asia. DSE/ZEL, Feldafing (1998) 1–25.
22. Prusinkiewicz, P., Lindenmayer, A.: The Algorithmic Beauty of Plants. Springer, New York (1990).
23. Prusinkiewicz, P., Hammel, M., Hanan, J., Mech, R.: Visual Models of Plant Development. In: Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages. Vol. 3: Beyond Words. Springer, Berlin (1997) 535–597.
24. Sloboda, B.: Möglichkeiten der mathematischen Vorhersage der Holzproduktion im Wirtschaftswald. Forstarchiv **54** (1983) 134–142.