Visualization and Computer Simulation of Biological Development

A Dissertation Proposal
Presented to the
School of School of Computing Science
Simon Fraser University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

by
Maria Louise Lantin
November 1997

1 Introduction

The ultimate goal of developmental biology is to discover how one cell with limited direct information can develop into a complex organism. Countless experiments have been done and several hypotheses have been put forward to explain the differentiation, migration, growth and division of cells in early plant and embryonic development. Mechanical, chemical and electrical mechanisms (and combinations of these) have been suggested as being responsible for the aforementioned cellular behavior. Through developmental models and computer simulations it is hoped that we can compliment traditional techniques and gain a better understanding of how these mechanisms interact with each other and influence cellular development.

Developmental processes can essentially be divided into three areas: regional specification (pattern formation), differentiation and morphogenesis. All three are equally important to the development of an organism. Regional specification occurs in the earliest stages of development and refers to the process by which cells in different regions choose different pathways of development. In a sense differentiation and morphogenesis can be thought of as consequences of regional specification. Cell differentiation occurs when a cell's behaviour and developmental path changes from that of its ancestors through the synthesis of different proteins. Morphogenesis is the synthesis of multicellular arrangements such as tissues and organs from a seemingly heterogeneous group of cells. It is believed that morphogenesis is brought about by essentially six cellular processes: the direction and number of cell divisions; cell shape changes; cell migration; cell growth; cell death; and changes in the composition of the cell membrane and extra-cellular matrix. These processes are regulated by cell-cell communications which can occur through diffusible substances (hormones, growth factors, morphogens), or direct physical contact [15, 31].

Biological experiments often require a substantial amount of planning and necessitate meticulous care in their execution. While the planning stage would still be an important component of the virtual experiment cycle, because the plan could be adjusted quickly "on-the-fly", an error or unexpected circumstance in the experiment would not mean as significant a time loss. In fact, one

could restart the experiment with a modified plan from a known stable point in the previous execution. A virtual experiment environment could also guarantee repeatability of results, allow the testing of a hypothesis on an isolated cell configuration, and enable easy labeling or marking of significant features of a cell configuration such as chemical concentrations and cell types, sizes, ages, lineage, etc.

2 Thesis Statement

To provide an exploratory software environment that integrates all important aspects of biological development for the simulation of multi-cellular structure formation.

3 Research Description

3.1 General Description

The goal, as stated in the introduction, is to provide a software environment that will be employed to elucidate the mechanisms of plant and embryo development. In recent years, the focus of biological research has been in the area of molecular biology with special attention being given to DNA and its by-products. While such research is necessary to develop an understanding of the processes involved in morphogenesis, it is essentially a bottom-up type of approach. The inherent problem with such an approach is that there are a great number of molecules to observe and not a clear path to the next level of abstraction. We propose a tool to explore, in a top-down fashion, how cellular proteins combined with the physical properties of the cells can lead to some observed developmental phenomena. Neither the top-down nor the bottom-up approach will lead to a complete picture of morphogenesis, but the hope is that, drawing from each other's results, the full picture will emerge more quickly. Lionel Harrison, in the introduction to *Kinetic Theory of Living Pattern* [17], gives a nice overview of the two types of researchers (which he calls "splitters" and "lumpers").

In order for our research tool to be capable of assisting in the quest to speed up the discovery process, it must meet a number of requirements. Firstly,

the simulation concepts within the tool must be closely related to real biological ones. This requirement is important to facilitate the construction of models and the inferences made from them. Secondly, it must be accurate. It is essential that the biologist using the tool trusts the results of the simulation. Thirdly, it must be fast enough to be interactive. It is an unfortunate computing reality that this requirement competes with the second. We will have to make some compromises to achieve reasonable response time, but each one should be assessed carefully as to its impact on accuracy. However, the option should always remain to restore full accuracy (i.e. no optimization) for batch or less interactive experiments. Finally, the environment should be complete in its coverage of the known biological components of morphogenesis. That is, it should cover all three areas of developmental mechanisms and all six cellular processes listed in Section 1.

These are the primary requirement that we will strive to satisfy in this thesis. Another important concern will be the usability of our research tool. We will make every effort to construct a user interface that biologists will find easy to use, although the primary focus will be on the underlying simulation engine.

3.2 Relationship to Other's Work

Many models of cellular development have been put forward in the past (a review can be found in [19]). Our work is most closely related to the Lindenmayer (L) family of models which we will briefly summarize here.

L-systems (short for Lindenmayer systems) were put forward by Aristid Lindenmayer in 1968 [21] as a modelling framework for plants. An L-system is closely related to the concept of a Chomsky grammar in the sense that it successively rewrites a string of symbols using rules specified in a grammar. However, unlike Chomsky grammars, each derivation step in an L-system occurs in parallel for each symbol in the string and there is no distinction between terminal and non-terminal symbols. In an L-system biological model, the symbols represent a component of the organism being modelled such as a cell, a branch or a leaf. The association of a biological concept to a symbol, however, is strictly

in the modeller's mind and the graphical representation of the symbols is specified through LOGO-style turtle directives embedded within the L-system grammar rules [26]. L-systems have been used extensively to model branching structures and non-branching filaments [27].

L-systems evolved into a whole family of models that includes bracketed L-systems [21], map L-systems (cyclic and marker-based forms) [4, 13, 23, 25], cellwork L-systems (3-dimensional map L-systems) [12] and cell systems [2, 9, 20]. Map L-systems made possible the modelling of cell layers by using the graph theoretical concept of maps. Each region, bounded by a group of edges (walls), corresponds to a cell and the rules of the map L-system either operate on walls (marker-based) or regions (cyclic) to bring the map to each subsequent configuration. Cells are brought to their optimal shape through a physically-based cell shape controller which takes into account cell turgor pressure, wall tension and outside forces (other cells, environment, etc.) [14]. Cell systems added a layer of abstraction to map L-systems by having rules operate on cells rather than walls. The physically-based controller introduced with map L-systems has also been successfully applied to cell systems. Appendices B through D contain examples of an L-system, map L-system and cell system respectively.

In 1995, we introduced extended context-sensitive cell systems [20] which improved on cell systems by permitting the manipulation of individual cell parameters such as pressure and wall tension. This allows the design of more accurate models by providing finer control of cell shape. Later, we combined the structure of the Cell Programming Language (CPL) [1] with the syntactic features of cell systems to create a more structured syntactic language for cell systems [19]. We named this language "Cell Systems Programming Language" (CSPL). Among its many useful features, the language allows the definition of cell types in a structured way by grouping cell characteristics and behaviour, and provides commands that map semantically to biological processes such as cell division and chemical diffusion. It is worth mentioning however, that as the level of biological abstraction gets higher, the modeller has less control over the emergent behaviour of the model. This is always an issue when

moving from low-level languages to higher-level ones (e.g. consider the move from assembly to C to 4th Generations Languages (4GL)). In programming languages, one loses the ability to explicitly dictate and optimize code but gains freedom in expressing high-level logical constructs. A high-level biological simulation language such as CSPL, denies the modeller individual control over all aspects of a cell, but permits the expression of well-known processes such as chemical diffusion without worrying about the details of such a process. However, just as a programmer must trust that a compiler is correct in its translation of a computer program, so must the modeller believe that correct behaviour can be attained through a combination of high-level process and the manipulation of cell parameters.

The multiple-mechanism developmental model developed by Fleischer [7] is similar to our proposed environment in its intent to provide a comprehensive development environment. However we differ significantly in the cellular representation and medium. The cells in the Fleisher model are disks and spheres which do not vary in shape (only size) and move around freely in a viscous medium. Though the cells can touch and bind through wall binding sites, the shape changes inherent in such interactions are not modelled. Also in contrast to our system, all cell behaviours are specified through partial differential equations and not higher level language constructs as in CSPL.

The power of our proposed environment lies in the syntactic nature of the model specification language (CSPL), in its reference to cells as entities and in its use of the physically-based cell shape controller to model cell layer dynamics. Together these features allow for easier model construction, more realistic simulations, and more direct biological inferences. Appendices E and F show two examples of models constructed with CSPL. One thing to notice here is that even though the CSPL model of *A. Catenula* (Appendix E) uses the same division rules as the corresponding L-system (Appendix B, it produces a more realistic result because of the cell shape controller. Also, using the CSPL model, it would be possible to model the cell division ratio using reaction-diffusion processes inside the cells as suggested in [17] (though it would require the addition of an underlying diffusion grid). It is not clear how one could achieve this with

the L-system model of *A. Catenula.* Although reaction-diffusion theory has been integrated into L-systems by means of interactions between components [27] or between components and their environment (medium) [24, 28], it is not possible to model concentration gradients within a single component of an L-system, simply because the components do not have a shape which can be referenced during the execution of the model.

The models that we have studied (see Table 1) fall into the two categories previously described in [19]: analytical and synthetic. Analytical models strive to prove that a particular developmental feature can be explained by a set of cellular processes. These models are usually limited in the scope of cellular processes they model and tend to be inflexible. Synthetic models are designed without a particular organism or developmental feature in mind. They strive to encompass as many cellular processes as have been described in the literature in order to model a wide range of developmental features. The models described in [1, 5, 6, 8, 21, 23] are in this category but differ in the range of processes they model and the way in which these are implemented. Because cell shape changes can significantly affect the overall structure of the organism - by influencing division patterns for example - it is important to model the shape of a cell with some accuracy. Conversely, the structure of the organism can affect cell shape through mechanical means. For example, the number and sizes of a cell's neighbours will affect the shape that it ultimately takes. Cellular representation plays a large role in determining whether a model will be able to model the cell shape-structure interactions accurately. For example, even though the topological model [5] is able to model all the cellular processes, the cell shape changes are a consequence of cell neighbourhood changes and do not reflect mechanical interactions between cells. That is, even though it models well the impact of shape on structure, it does not do the converse. Both are important in a realistic model of cellular structure and processes. Our choice of polygons and beta-splines to model 2-dimensional cells and the use of a physically-based cell shape controller ensures that the cell shape-structure interactions are modelled accurately.

	Cell Rep.	(L)ayers, (T)issues, (B)ranch **	Division	Diff.	Shape Variable	Growth	Death	Migration	Other
L-systems 1968 [21]	lines, curves, surfaces	B	•	•	•	•	•		
Map-L systems 1979 [4, 13, 23, 25]	polygons, β-splines	L, B	•	•	•*	•			
Cell Systems 1991 [2, 9, 20]	polygons,β-splines	L	•	•	•*	•	•		
Protrusion-contraction 1990 [32]	polygons	L			•*	•		•	•
Free cells 1993 [16]	Free Dirichlet	L			•	•		•	•
Topol. model 1984 [5, 29]	topological	L	•	•	•	•	•	•	•
Cellworks 1984 [12, 22]	polyhedrons, spheres, ellipsoids	L,T	•		•*	•			
FEM 1994 [3]	finite elements	L,T			•*				•
Multi-mechanism 1992 [6]	disks,spheres	L,T	•	•		•	•	•	•
CPL 1995 [1]	points	L	•	•	•	•	•	•	•
Polarization-Elasticity 1995 [30]	points	L		•				•	•

* Uses physically-based shape controller

** A tissue refers to a 3-dimensional group of cells. A layer is a cross-section of a tissue.

Table 1: Summary of developmental models. The column labeled “Other” includes processes such as collision detection and cell adhesion

4 Research Plan

4.1 Approach

We propose to develop a software environment which satisfies all the requirements outlined in Section 3.1. The environment will consist of four main components:

- a language,
- a simulation engine,
- an editor to construct simulations, and
- a interface to control simulation parameters an to display results.

The simulation language will be based on Cell Systems [2, 9] and CPL (Cell Programming Language) [1], and will incorporate features to allow cell division, cell growth, cell branching, chemical diffusion between cells and their environment, and environmental parameters such as gravity, temperature and lighting level / direction. Work on this language which we have called CSPL is already underway; the current specification of the language can be found in Appendix A. Specifically, we want to add a branching command, a migration command, an underlying diffusible medium, collision detection, and programmable environmental parameters such as light, gravity and temperature. By programmable, we mean that the modeller will be able to control these parameters just as they would a cell (i.e. the environmental parameters will have behaviour).

The simulation engine is the means by which we perform the commands included in the language. It encompasses the cellular representation, the language parser, the command interpretor, the cell shape controller, and the simulation director. Some of the issues regarding the simulation engine have already been decided, such as our choice of the cellular representation and of a physically-based cell shape controller. However a lot of work still needs to be done in optimizing the cell shape controller and extending it to include environmental factors and collision detection between cells. The optimization of the

cell shape controller is especially important to the interactive nature of the software. Currently, as the number of cells increase (and cell multiplication is often exponential), the time in reaching structural equilibrium increases significantly because we must iteratively calculate forces acting on every vertex within the structure. We are hoping to decrease the number of steps to equilibrium by limiting the sphere of influence of a cell. For example, if a cell changes shape on the one side of the structure, it should have little impact on the opposite side. We envisage that this improvement will result in a performance gain especially when dealing with structures exhibiting apical growth (growth mainly from the tip (apical cells) of the structure) which is common in plants. We are also investigating a top-down approach to structure dynamics by recursively calculating dynamics on groups of cells. This may decrease the number of iterations that we need to perform over the entire structure to reach equilibrium.

The simulation editor will be a graphical user interface to help in defining cell types and their respective behaviour, specifying the starting cell configuration and setting default parameters for the simulation (see Figure 1 for screen shot).

The simulation viewer is the main window which displays the starting cell configuration and the animation of the developing structure. A number of control panels which are part of the viewer, allow dynamic manipulation of simulation parameters such as individual cell pressure and wall tension, parameters of the physically-based cell shape controller, simulation step size, etc. (see Figure 2 for screen shot).

4.2 Artifacts to be Produced

Two artifacts will be produced:

- the software environment, and
- a complete model of *Physcomitrella Patens*

As a demonstration of the features of the software environment, a realistic model of the development of the moss *Physcomitrella Patens* will be constructed using our software and the help of a biologist, Dr. Neil Ashton from

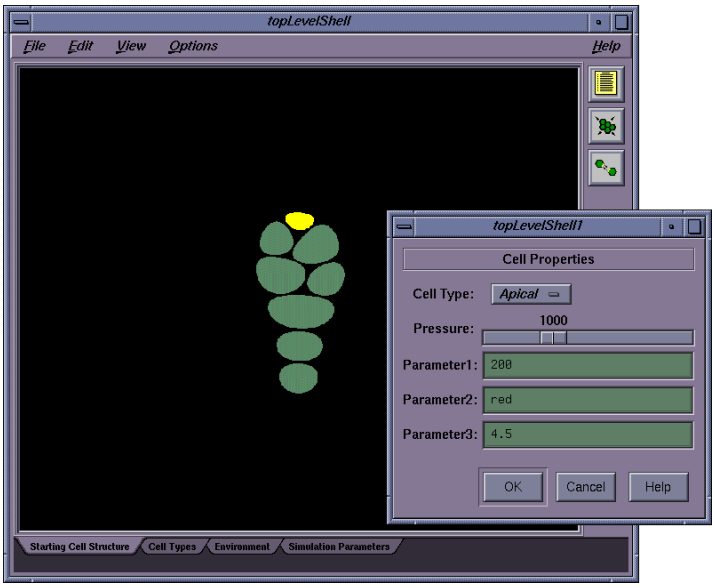

Figure 1: Screen shot of simulation editor

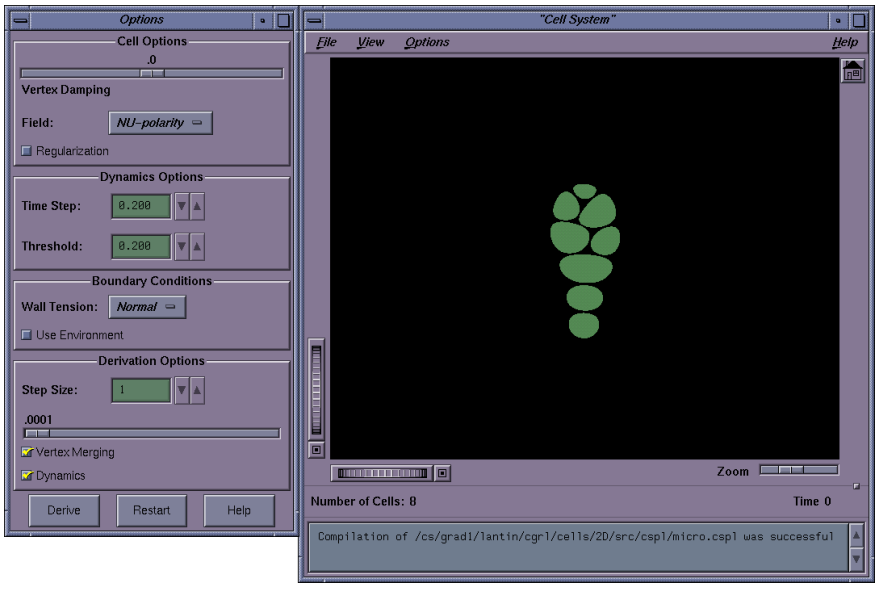

Figure 2: Screen shot of simulation viewer

the University of Regina. A good description of the development of *P. Patens* appeared in [11]:

> Generally speaking, the development of *P. Patens* under defined standard culture conditions begins with a spore which germinates to produce filaments of green chloronemal cells which grow by apical divisions approximately 20 hours. Subapical cells produce branches which form new apical cells. After roughly six days some chloronemal apical cells divide to produce caulonemal apical cells. These caulonemal apical cells divide much faster (approx. every five hours) giving rise to reddish caulonemal filaments, most of which exhibit clockwise curvature. Most subapical caulonemal cells divide to form side branch initials. Between 85 to 95% of these branches give rise to secondary chloronemata, 5 to 6% become secondary caulonemata, and the remaining either do not develop further or produce a bud. These buds develop into gametophores (leafy shoots). Much of the development remains relatively planar, with the exception of the leafy shoots which grow upwards.

A model of this organism has already been developed using L-systems [10] with good results but the investigation of some aspects of the moss development such as the space-filling of chloronemal cells and the effects of gravity, light and chemicals requires a more powerful cell-based simulation tool.

4.3 Limitations of the Dissertation

Because our primary aim is to develop a biologically complete simulation environment, we will not be producing a full-blown user interface study of our software. However, we believe that, through working with a biologist in the development of the *Physcomitrella Patens* model, we will gain valuable insights into the usability of the tool. Undoubtedly, some changes will be made to the interface through this inter-disciplinary cooperation and even though such a study is not inclusive, it will be very beneficial in making our software more usable.

Our model uses 2-dimensional cells, and while this is not a limitation when dealing with low-level plant development and cell layers, it would be useful to extend the model to use 3-dimensional cells to enable the study of higher plants and animals. In particular this would be necessary for the modelling of the leafy shoots of *P. Patens*. However, 3-D cells would require a significantly different mechanism for the specification of the cell division plane and 3-D structures would need much more computational power to reach equilibrium.

5 Research Papers

- A complete software environment for the investigation of developmental processes
- A computer simulation of the development of *Physcomitrella Patens*
- The Cell System Programming Language (CSPL) - a simulation language suited to biological investigation

6 Contributions

6.1 Computing Science

At my depth exam, one of the examiners astutely asked the question: “Why is this computing science?”. To answer this question we must first answer: “What is computing science?”. Computing science like any other field has two components: theoretical and applied. The theoretical component is closely related to mathematics and deals with issues of pure computation and optimization. The applied component in its most traditional sense deals with the development of general algorithms for the computer such as those used in communication protocols, operating systems and database management. As computers spread rapidly into other fields however, the applied component broadened to include algorithms and methods developed for specific applications (e.g. web protocols and visualization methods). This branch of computing science uses the most recent advances in the field to push the boundaries of the computer as

a research and productivity tool. This is where we feel our contribution is most significant.

On a more traditional level, we have also contributed a tool that can be used to study the behaviour of irregular cellular automatas. The use of a language to easily define state changes under variable and programmable conditions will be a powerful addition to the investigation of processes occurring on irregular grids such as pattern formation.

6.2 Biology

Our contribution to biology is in the production of a tool for the study of biological development processes which can also be used as a basis for the development of more advanced modelling tools. It is our hope that this tool will help in furthering the understanding of morphogenesis in plants and embryos.

We will also contribute a complete model of *P. Patens* which we hope will lead some new insights into the development of lower plants.

7 Dissertation Schedule

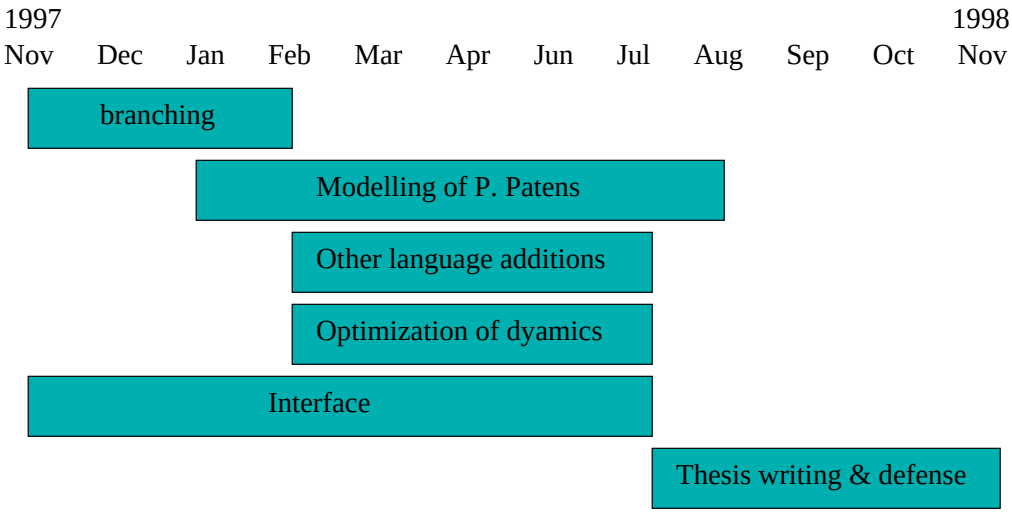

Figure 3: Completion timeline

Acknowledgments

Thanks to Dave Fracchia for helpful comments and suggestions. Also, much appreciation goes to Gabor Melli for his rigorous approaches to my questions (logic meets hand-waving...). Finally thanks to Kopka and Daly for their fabulous time-saving LaTeX 2_ε book [18]!

A CSPL Language Specifications

simulation : [*sim_attributes*] *celltypes environment observer cells*

sim_attributes : *sim_attribute* [*sim_attributes*]

sim_attribute : *field_t*
| *regular*
| *time*
| *threshold*
| *tension*
| *step*
| *merging*
| *dynam*

field_t : **field_type =** *field_type*

field_type : one of
NU_POLARITY U_POLARITY Vector

regular : **regularization =** *on_off*

on_off : one of
On Off

time : **time_step =** *real*

threshold : **dynamics_threshold =** *real*

tension : **outside_wall_tension =** *tension_opt*

tension_opt : one of
NORMAL HALF

step : **step_size =** *integer*

merging : **vertex_merging =** *on_off*

dynam : **dynamics =** *on_off*

celltypes : *celltype* [*celltypes*]

celltype : **type** *typename* [**: like** *typename*] {
[**colour =** *colour*]
[*declarations*]
behaviour
}

type_name : one of
environment observer *identifier*

colour : *identifier* (X colour name)
| (*integer*, *integer*, *integer*) (RGB colour)

declarations : *declaration* [*declarations*]

declaration : *datatype identifier*
| **biochemical** *identifier* = [*expression*, *colour*] [*expression*, *colour*]

datatype : one of
real int bool

behaviour : *states*
| *commands*

states : *state* [*states*]

state : **`state`** *identifier* [**`virtual`**] {
commands
}

environment : **`environment`** {
var_inits
}

observer : **`observer`** {
var_inits
}

var_inits : *var_init* [*var_inits*]

var_init : *identifier* = *expression*

cell : **`cell`** {
`type` *identifier*;
cell_inits
`location` = { *coordinates* }
}

cell_inits : *lcell_variable* = *expression* [*cell_inits*]

coordinates : *coordinate coordinate coordinate*
| *coordinate* [*coordinates*]

coordinate : '[' *expression*, *expression*, *expression* ']' [**=** *identifier*]

commands : *command* [*commands*]

command : *divide*
| *assignment*
| *diffuse*
| *if_then*
| *for_loop*
| *differentiate*
| *goto*
| *write*
| *echo*
| **`die;`**
| **`image;`**

divide : **`divide`** [*expression*`%`] **`at`** (*expression*, *expression*) **`into`**
daughtercell daughtercell

daughtercell : *identifier* {
[*cellvar_assignments*]
}

cellvar_assignments : *cellvar_assignment* [*cellvar_assignments*]

cellvar_assignment : *lcell_variable assign_operator expression*

lcell_variable : **`pressure`**
| *identifier*
| **`neighbour`**(*expression*).*lcell_variable*
| **`wall`**(*expression*).*wall_variable*

wall_variable : one of
`tension permeable`

assign_operator : one of
= += -= *= /= ++ --

assignment : *cellvar_assignment*
| *simvar_assignment*

simvar_assignment : *lsim_variable assign_operatrs expression*

lsim_variable : one of
time_interval steps

diffuse : **diffuse** *identifier* {
rate = *expression*
production = *expression*
}

if_then : **if** *expression 1orbcommands* [**else** *1orbcommands*]

1orbcommands : *command*
| { *commands* }

for_loop : **for** *identifier expression* **to** *expression 1orbcommands*

differentiate : **differentiate** *daughter_def*

go_to : **goto** *identifier*

write : **write** *expression*

echo : **echo** *quoted_string*

expression : *integer*
| *real*
| **`True`**
| **`False`**
| *variable*
| (*expression*)
| **`-`** *expression*
| **`!`** *expression*
| *expression operator expression*
| **`atan2`** (*expression*, *expression*)
| *unaryfunc* (*expression*)

operator : one of
`+ - * / ^ == != <= >= < > && ||`

unaryfunc : one of
`sin cos tan ceil floor asin acos`
`atan exp log log10 sqrt`

variable : *cell_variable*
| *sim_variable*

cell_variable : *identifier*
| **`age`**
| **`area`**
| **`num_neighbours`**
| **`pressure`**
| **`neighbour`**(*expression*).*cell_variable*
| **`wall`**(*expression*).*wall_variable*

sim_variable : one of
`time time_interval steps`

B Parametric context-sensitive L-system of *A. Catenula*

```
#define CH 900 /* high concentration */
#define CT 0.4 /* concentration threshold */
#define ST 3.9 /* segment size threshold */
#ignore f ~ H

ω :  -(90)F(0,0,CH)F(4,1,CH)F(0,0,CH)
p1 : F(s,t,c) :  t=1 & s >= 6 ⟶ F(s/3*2,2,c)f(1)F(s/3,1,c)
p2 : F(s,t,c) :  t=2 & s >= 6 ⟶ F(s/3,2,c)f(1)F(s/3*2,1,c)
p3 : F(h,i,k) < F(s,t,c) > F(o,p,r) :  s>ST | c>CT ⟶
                                (s+.1,t,c+0.25*(k+r-3*c))
p4 : F(h,i,k) < F(s,t,c) > F(o,p,r) :  !(s>ST|c>CT) ⟶
                                F(0,0,CH)   H(1)
p5 : H(s) :  s < 3 ⟶ H(s*1.1)
```

**F cells are drawn as rectangles and H cells as circles.

Figure 4: Simulation of the development of *Anabaena Catenula*

C Map L-system of *Microsorium Linguaeforme*

$$
\begin{array}{llll}
r_1: & \vec{A} \longrightarrow \overleftarrow{a}\ [-\ \overleftarrow{B}]\ \vec{I} & r_2: & \vec{B} \longrightarrow \vec{E}\ [+\ \vec{b}]x[-\ \vec{H}]\ \vec{D} \\
r_3: & \vec{D} \longrightarrow [-\ \vec{M}]\ \vec{F} & r_4: & \vec{F} \longrightarrow \vec{G}\ [+\ \overleftarrow{H}]x[-\ \vec{H}]\ \vec{D} \\
r_5: & \vec{H} \longrightarrow x[+\ \vec{F}]x & r_6: & \vec{I} \longrightarrow \vec{C} \\
r_7: & \vec{C} \longrightarrow \vec{I}\ [-\ \overleftarrow{F}]\ \vec{I} & r_8: & \vec{E} \longrightarrow x[-x]x \\
r_9: & \vec{G} \longrightarrow x[+x]x[-x]x & r_{10}: & \vec{J} \longrightarrow \vec{L} \\
r_{11}: & \vec{K} \longrightarrow \vec{N} & r_{12}: & \vec{L} \longrightarrow x[+\ \overleftarrow{M}]x \\
r_{13}: & \vec{M} \longrightarrow x[+\ \vec{L}]x & r_{14}: & \vec{N} \longrightarrow \vec{O} \\
r_{15}: & \vec{O} \longrightarrow x[+\ \overleftarrow{L}]\ \vec{N} & &
\end{array}
$$

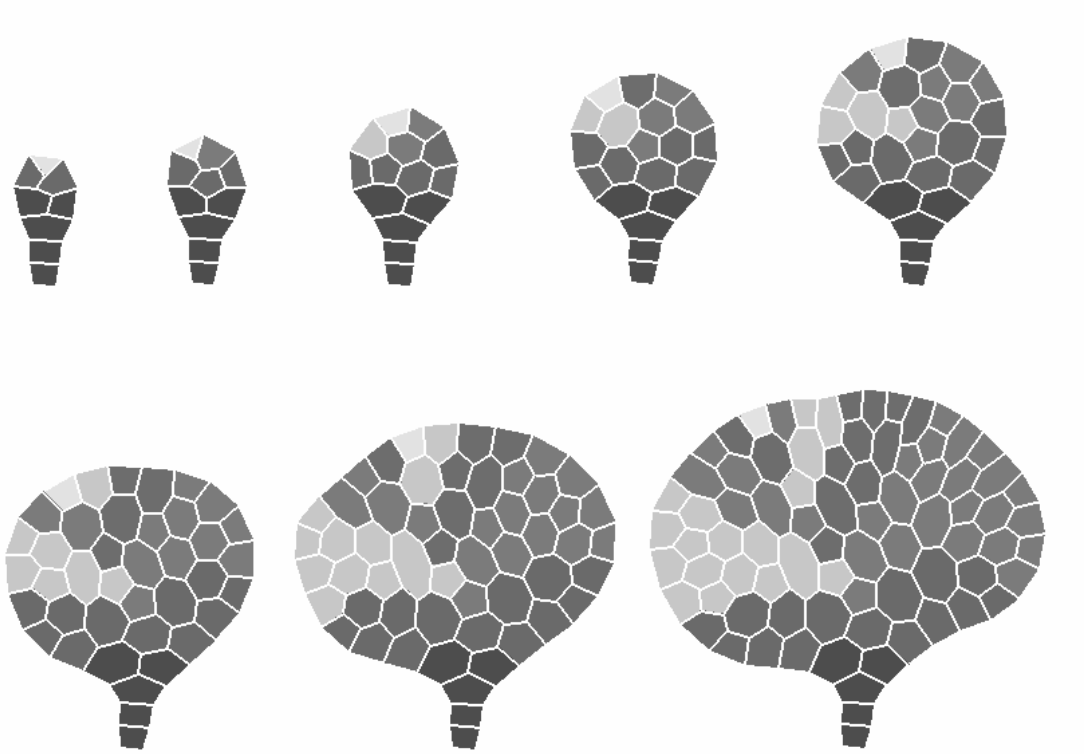

Figure 5: Simulated development of *Microsorium Linguaeforme*

D Cell System Model of *Microsorium Linguaeforme*

```
#types
ONE()
TWO()
THREE()
C()
D()
E()
F()
G()
L()
R()
X()

#axioms
F()@[300,100,0],[200,0,0],[200,-70,0],[200,-100,0],[300,-100,0]
L()@[100,100,0],[200,0,0],[300,100,0]
C()@[100,-70,0],[200,-70,0],[200,0,0],[100,100,0]
X()@[200,-70,0],[100,-70,0],[100,-200,0],[200,-200,0],[200,-100,0]
X()@[300,-100,0],[200,-100,0],[200,-200,0],[300,-200,0]
X()@[100,-200,0],[100,-300,0],[300,-300,0],[300,-200,0],[200,-200,0]
X()@[100,-300,0],[100,-400,0],[300,-400,0],[300,-300,0]
X()@[100,-400,0],[100,-500,0],[300,-500,0],[300,-400,0]

#rules

L-->R()^(-45,.3)ONE()
R-->ONE()^(45,.7)L()
ONE-->TWO()^(-98,.5)THREE()
TWO-->ONE()^(90,.5)ONE()
THREE-->X()^(0,.5)X()
C-->D()
```

```
D-->E()^(-90,.5)X()
E-->X()^(90,.5)D()
F-->G()^(-90,.5)X()
G-->F()^(90,.5)X()
X-->X()
#end
```

E CSPL model of *A. Catenula*

```
type environment {
  biochemical nitrogen = [0, yellow] [10, RED]
}
type H  {
  state growing {
    wall(2).tension *= .7;
    wall(4).tension *= .7;
    if (age == 2)
        goto static;
  }
  state static {}
}
type anacell {
  real size;
  state growing {
    size += .1;
    diffuse nitrogen  {
      rate = .25;
      production = -.75;
    }
    wall(1).tension -= .5/20;
    wall(3).tension -= .5/20;
    if (size >= 6) goto mature;
    if (nitrogen < .7 && size < 4) goto starved;
  }

  state mature : virtual {}

  state starved {
    differentiate_to H { nitrogen = 900; }
   }
```

```
}
type L : like anacell {
  state mature {
    diffuse nitrogen  {
      rate = .25;
      production = -.75;
    }
    divide at (0, .67) into {
      R {
        size *= 2.0/3;
        wall(1).tension = 1;
        wall(3).tension = 1;
      }
      L {
        wall(1).tension = 2;
        wall(3).tension = 2;
        size *= 1.0/3;
       }
     }
     goto growing;
   }
}
type R : like anacell {
  state mature {
    diffuse nitrogen  {
      rate = .25;
      production = -.75;
    }
    divide at (0, .67) into {
      L {
        size *= 2.0/3;
        wall(1).tension = 1;
```

```
        wall(3).tension = 1;
      }
      R {
        size *= 1.0/3;
        wall(1).tension = 2;
        wall(3).tension = 2;
       }
    }
    goto growing;
  }
}

environment {
  nitrogen = 900;
}
cell {
  type R;
  size = 4;
  nitrogen = 3;
  wall(2).permeable = True;
  wall(1).tension = 1;
  wall(3).tension = 1;
  wall(4).permeable = True;
  location = {
    [-50, 50, 0][-50, -50, 0][50, -50, 0][50, 50, 0]
  }
}
```

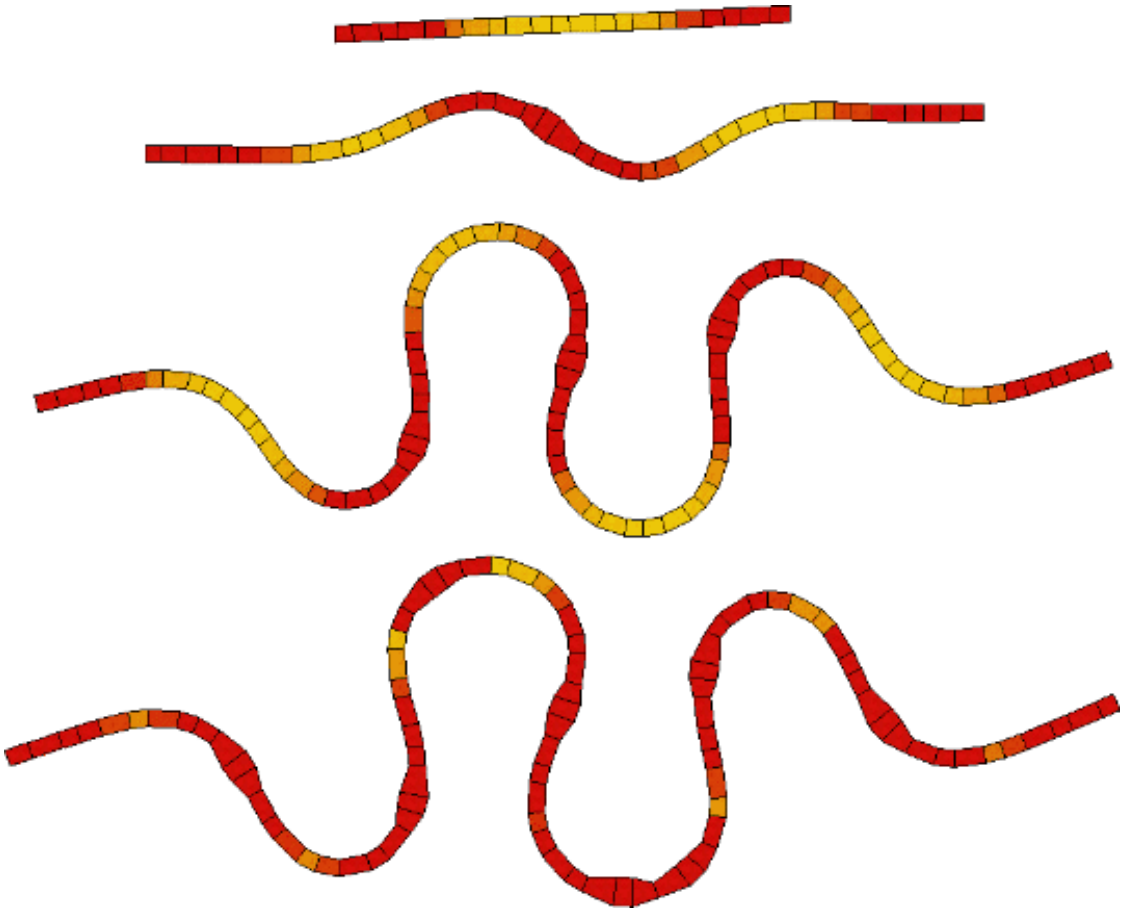

Figure 6: Simulated development of *A. Catenula*

F CSPL model of *Microsorium Linguaeforme*

```
field_type = NU_POLARITY;

type A {
    colour = pink;
    int side;
    divide at (side*45, .3) into {
        A { side = -side; }
        S1{}
    }
}
type S1 {
    int side;
    divide at (side*98, .5) into {
        S2{ side = -side; }
        S3{}
    }
}
type S2 {
    int side;
    divide at (side*90, .5) into {
        S1{ side = -side; }
        S1{}
    }
}
type S3 {
    pressure *= 1.5;
    divide at (0, .5) into {
        T{}
        T{}
    }
}
```

```
type B {
    int side;
    int delay;

    if (delay == 0) {
        divide at (side*90, .5) into {
            B { side = -side; }
            T {}
        }
    }
    else
        delay--;
}
type T {
  state vegetate{}
}

cell {
   type B;
   delay = 0;
   side = 1;
   location = {
       [200,-100,0][300,-100,0][300,100,0][200,0,0][200,-70,0]
   }
}
cell {
    type A;
    side = -1;
    location = {
       [100,100,0][200,0,0][300,100,0]
    }
}
```

```
cell {
    type B;
    delay = 1;
    side = -1;
    location = {
        [100,-70,0][200,-70,0][200,0,0][100,100,0]
    }
}
cell {
    type T;
    location = {
        [200,-70,0][100,-70,0][100,-200,0][200,-200,0][200,-100,0]
    }
}
cell {
    type T;
    location = {
        [300,-100,0][200,-100,0][200,-200,0][300,-200,0]
    }
}
cell {
    type T;
    location = {
        [100,-200,0][100,-300,0][300,-300,0][300,-200,0][200,-200,0]
    }
}
cell {
    type T;
    location = {
        [100,-300,0][100,-400,0][300,-400,0][300,-300,0]
    }
}
```

```
cell {
    type T;
    location = {
        [100,-400,0][100,-500,0][300,-500,0][300,-400,0]
    }
}
```

References

[1] P. Agarwal. The cell programming language. *Artifical Life*, 2(1):37–77, 1994.

[2] M. J. M. De Boer, F. D. Fracchia, and P. Prunsinkiewicz. A model for cellular development in morphogenic fields. In G. Rozenberg and A. Salomaa, editors, *Lindenmayer Systems*, pages 351–370. Springer-Verlag, Berlin, 1992.

[3] G. W. Brodland. Finite element methods for developmental biology. *International Review of Cytology*, 150, 1994.

[4] M. J. M. de Boer and A. Lindenmayer. Map OL-systems with edge label control: comparison of marker and cyclic systems. In H. Ehrig, M. Nagl, A. Rosenfeld, and G. Rozenberg, editors, *Graph Grammars and their application to computer science*, number 291 in Lecture notes in Computer Science, pages 378–392. Springer-Verlag, 1987.

[5] S. Duvdevani-Bar and L. Segel. On topological simulations in developmental biology. *Journal of Theoretical Biology*, 166:33–50, 1994.

[6] K. Fleischer and A. H. Barr. A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis. In Langton C. G., editor, *Artifical Life*, volume XVII, pages 389–415. Santa Fe Institute, Addison-Wesley, June 1992.

[7] K. W. Fleischer. *A Multi-Mechanism Development Model for Defining Self-Organizing Geometric Structures*. PhD thesis, California Institute of Technology, May 1995.

[8] F. D. Fracchia. *Visualization of the Development of Multicellular Structures*. PhD thesis, Univeristy of Regina, September 1991.

[9] F. D. Fracchia. Integrating lineage and interaction for the visualization of cellular structures. In *Graph Grammars and their application to computer science;Fifth International Workshop*, pages 241–246, 1994.

[10] F. D. Fracchia and N. W. Ashton. Case study: A visualization tool for studying the development of the moss *Physcomitrella patens*. In *Proceedings of IEEE Visualization '95*, pages 364–367. IEEE Computer Society Press, 1995.

[11] F. D. Fracchia and Neil W. Ashton. Case study: A visualization tool for studying the development of the moss *Physcomitrella patens*. In *Proceeding of IEEE Visualization '95*, pages 364–367. IEEE Computer Society Press, 1995.

[12] F. D. Fracchia and P. Prunsinkiewicz. Physically-based graphical interpretation of marker cellwork L-systems. In *Graph Grammars and their application to computer science;Fourth International Workshop*, volume 532 of *Lecture Notes in Computing Science*, pages 363–377, Berlin, 1991. Springer-Verlag.

[13] F.D. Fracchia, P. Prunsinkiewicz, and M.J.M. De Boer. Visualization of the development of multicellular structures. In *Proceedings of Graphics Interface*, pages 267–277, 1990.

[14] F.D. Fracchia, P. Prunsinkiewicz, and M.J.M. DeBoer. Animation of the development of multicellular structures. In *Computer Animation*, pages 3–18. Springer-Verlag, 1990.

[15] S. F. Gilbert. *Developmental Biology*. Sinauer Associates, 4th edition, 1994.

[16] F. Graner and Y. Sawada. Can surface adhesion drive cell-rearrangement? part II: A geometrical model. *Journal of Theoretical Biology*, 164:477–506, 1993.

[17] L. G. Harrison. *Kinetic Theory of Living Pattern*. Developmental and cell biology series. Cambridge University Press, 1993.

[18] Helmut Kopka and Patrick W. Daly. *A Guide to LaTeX 2ε*. Addison-Wesley, second edition, 1995.

[19] M. L. Lantin and F. D. Fracchia. Computer simulations of developmental processes. Depth Paper. Submitted to *Computer Applications in the Biosciences (CABIOS)*, May 1996.

[20] M. L. Lantin and F.D. Fracchia. Generalized context-sensitive cell systems. Presented at the *First International Workshop in Information Processing in Cells And Tissues*, September 1995.

[21] A. Lindenmayer. Mathematical models for cellular interactions in development, parts i and ii. *Journal of Theoretical Biology*, 18:280–299, 300–315, 1968.

[22] A. Lindenmayer. Models for plant tissue development with cell division orientation regulated by preprophase band of microtubules. *Differentiation*, 26:1–10, 1984.

[23] A. Lindenmayer and G. Rozenberg. Parallel generation of maps: Developmental systems for cell layers. In V. Claus, H. Erigh, and G. Rozenberg, editors, *Graph Grammars and their application to computer science;First International Workshop*, pages 301–316, Berlin, 1979. Springer-Verlag.

[24] R. Mech and P. Prusinkiewicz. Visual models of plants interacting with their environment. In *SIGGRAPH '96 Proceedings'*, volume 30 of *Computer Graphics*, pages 397–410. ACM, 1996.

[25] A. Nakamura, A. Lindenmayer, and K. Aizawa. Some systems for map generation. In G. Rozenberg and A. Salomaa, editors, *The Book of L*, pages 323–332. Springer-Verlag, Berlin, 1986.

[26] P. Prunsinkiewicz. Graphical applications of L-systems. In *Proceedings of Graphics Interface*, pages 247–253, 1986.

[27] P. Prunsinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, 1990.

[28] P. Prusinkiewicz, M. James, and R. Mech. Synthetic topiary. In *SIGGRAPH '94 proceedings*, volume 28 of *Computer Graphics*, pages 351–358. ACM, 1994.

[29] R. Ransom and R. J. Matela. Computer modelling of cell division during development using a topological approach. *Journal of Embryology and Experimental Morphology*, 83 suppl.:233–259, 1984.

[30] D. Savić. Model of pattern formation in animal coatings. *Journal of Theoretical Biology*, 172:299–303, 1995.

[31] J. M. W. Slack. *From Egg to Embryo.* Cambridge University Press, 2 edition, 1991.

[32] M. Weliky and G. Oster. The mechanical basis of cell rearrangement I. epithelial moprhogenesis during *fundulus* epiboly. *Development*, 109:373–386, 1990.

This dissertation proposal by Maria Louise Lantin is accepted in its present form by the School of School of Computing Science of Simon Fraser University as satisfying the dissertation proposal requirement for the degree of Doctor of Philosophy.

David Fracchia, Committee Chair

David Baillie, Committee Member

John Dill, Committee Member

Stella Atkins, Graduate Chair