A Self-Constructing Artificial Organism

Roland UNTERBERGER

June 27, 2008
Overview

1. Pattern formation in biology
2. Ideas from Fabian Roth
3. Software
4. Biologically inspired implementations
All animals look different. One would expect thousands of different development mechanisms.

Basic machinery of development is the same, only few evolutionarily related genes make difference.

50 percent of genes were already present in the common ancestor of worms, flies and humans.

Other 50 percent of genes make the difference in development.
Different looks with instruction plan

- Like a child’s construction kid, elements are the same but the assembling makes the difference
- In cell development it is the DNA code, with dozens of separate regulatory elements
Sister Cells can be born different

- This happens by asymmetric cell division
- After symmetric cell division -> change into different cell by signals
- When should a cell divide into a different cell?
Possible processes to get different cells

- Signals from the neighboring cells.
- A signal from cells outside the group drives group into a different development pathway.
- Long-range signals by morphogens
Morphogens are long-range inducers that exert graded effects

- Signaling molecule diffuses out from a localized source
- Create a signal concentration gradient
- Different distance -> different behavior
Idea

- Construct a multicellular organism
- Start with a single blast cell provided with intracellular factories
- Blast cells versus structure and functional cells
Cell has basically a constructor, competences and reactors
The environment

- A cell is placed into the internal environment
- The internal environment itself is embedded in the world environment
The structure is described in a lineage-tree.

After setting the parameters for the competences and reactors, we are able to build a structure.

These previous points are like the DNA of an animal.
<gene name="B0" type="O">
    <regulator>
        <condition chem="B0" thresh="eta" type="ON"/>
        <condition chem="b0" thresh="b0_head" type="OFF"/>
        <thresholds name="eta" value="1.0"/>
        <thresholds name="b0_head" value="0.8"/>
    </regulator>
    <action>
        <factory name="DivideCompetence"/>
        <factory name="MigrateCompetence">
            <argument name="b0" value="1.0"/>
        </factory>
        <factory name="SourceReactor">
            <argument name="b0" value="0.17"/>
        </factory>
        <factory name="ConstReactor">
            <formula result="B0" arguments="alpha,b0_head,B0"/>
            <argument name="b0_head" value="2.1"/>
            <argument name="alpha" value="0.1"/>
        </factory>
    </action>
</gene>
This thesis gave us the idea of building our own simulation system.
Our software

- In principle the approach of Fabians thesis but
- Different cell types, to handle other tasks
- A GUI (with Repast Simphony)
- A better neuronal system
Additional requirements

- Fast simulation with large number of cells
- Extendable for future implementations
- Solutions:
  - Stand alone without graphical output
  - Combination of Prototype Factories and Abstract Classes
  - Standard implementation of Competences and Reactors.
  - Sample classes: DivideCompetence, MigrateCompetence, SourceReactor, AxisReactor, ...
Chemical division

Two implementation approaches

- Mother cell and daughter cell get the same internal chemicals (Fabian’s approach)
- Split-up the chemicals with a controlable probability (more biological approach)
- First approach easier to handle, and gives good results.
- Second approach more likely but the parameter setting is difficult.
Chemical diffusion

- Diffusion in the internal space and between cell and internal space
- Different diffusion coefficients for each chemical in the two diffusions
- Different diffusion coefficients of each chemical

\[
c_r^i \leftarrow c_r^i + \tau \cdot D_i \cdot \sum_{r \sim r'} (c_r^i - c_r^i),
\]
Is very hard to find good working parameters.

Two parameters make a big change in the behavior

Many parameters have to be set.
...a little movie...
Unlimited cell growth

- Step = 700
- Source reactor $b_0 = 1.6$
- Integration constant = 0.18
Cell stable

- Step = 700
- Source reactor b0 = 1.7
- Integration constant = 0.17
AxisReactor equation

- Create opposing gradients of morphogens with Gierer-Meinhardt model.
- \( s_1 \) and \( s_2 \) are the long range cross-facilitators and have a higher diffusion coefficient.
- \( g_1 \) and \( g_2 \) are the short range cross-facilitators.
- \( g_1 \) produces \( s_1 \) and in turn cross-facilitate \( g_2 \) over a longer distance.

\[
\begin{align*}
\dot{g}_1 &= \frac{cs_2}{a + g_1^3} - \alpha g_1 \\
\dot{g}_2 &= \frac{cs_1}{a + g_2^3} - \alpha g_2 \\
\dot{s}_1 &= \gamma g_1 - \gamma s_1 \\
\dot{s}_2 &= \gamma g_2 - \gamma s_2.
\end{align*}
\]
Most results

- The gradient building is not as expected.
- Mostly random and not two section
Pattern formation in biology
Ideas from Fabian Roth
Software
Biologically inspired implementations

\begin{align*}
g_{1} & \sim 1.400 \\
g_{2} & \\
g_{1} - g_{2} & \\
\int g_{1} & \\
\int g_{2} & \\
\int g_{1} - g_{2} & \\
\end{align*}
Pattern formation in biology
Ideas from Fabian Roth
Software
Biologically inspired implementations

<table>
<thead>
<tr>
<th>g1 1950</th>
<th>g2</th>
<th>g1 - g2</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="g1_1950.png" alt="Image" /></td>
<td><img src="g2.png" alt="Image" /></td>
<td><img src="g1-g2.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>g1</th>
<th>g2</th>
<th>g1 - g2</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="g1.png" alt="Image" /></td>
<td><img src="g2.png" alt="Image" /></td>
<td><img src="g1-g2.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int g1</th>
<th>int g2</th>
<th>int g1 - int g2</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="int_g1.png" alt="Image" /></td>
<td><img src="int_g2.png" alt="Image" /></td>
<td><img src="int_g1-int_g2.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Biologic Axon growth

- Axons move through their environment via the growth cone
- Growth cone explores the surrounding environment.
- Target cells diffuse chemicals in the environment which act as short range attractors
AxonCompetence

- Growth cone is attracted or repulsed by specific morphogens
- Growth cone splits if the attracted morphogens increases
- The program has the possibility of different connection mechanisms
- Standard connection probability depends on amount of connections (implemented now)
Construction of neural level

- Lightweight version of a simulator like CSIM
- General input cells which receive input from the external world
- Output cells which send output signals to the external world
- Cells with neural competences which communicate with other cells
Neural simulation

- Have to be much faster than the cell structure process.
- After connection with another neural cell, we create a new Synapse object.
- Every cell has an output delay queue which depends on the distance between the connected cells.
- Two different simulation approaches: analog and spiking neurons.
Thank you!
References

- Alberts at al., *Molecular biology of the cell*, 2002
- Carroll at al., *From DNA to Diversity*, 2005