

Using Morphogenetic Models to Develop Spatial Structures

Jacob Beal, Jessica Lowell, Annan Mozeika, Kyle Usbeck

October 3, 2011

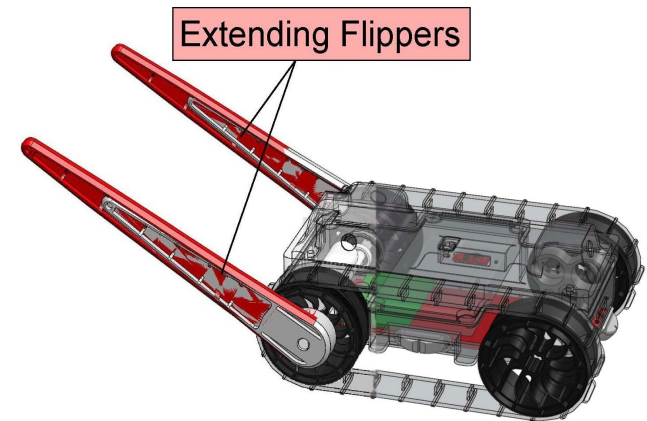
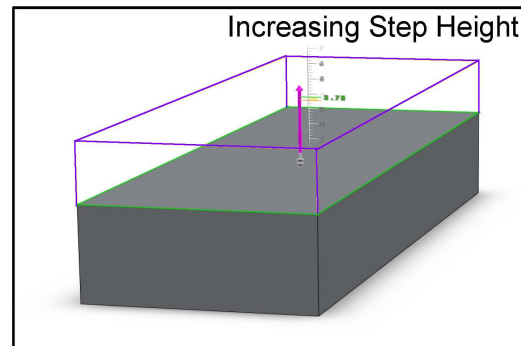
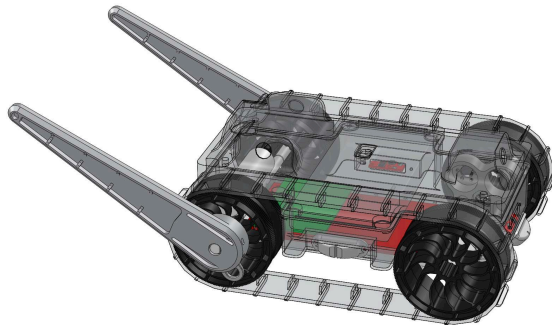
**Spatial Computing workshop at
IEEE SASO 2011**

iRobot[®] **Raytheon**
BBN Technologies

Work sponsored by DARPA; the views and conclusions contained in this document are those of the authors and not DARPA or the U.S. Government.

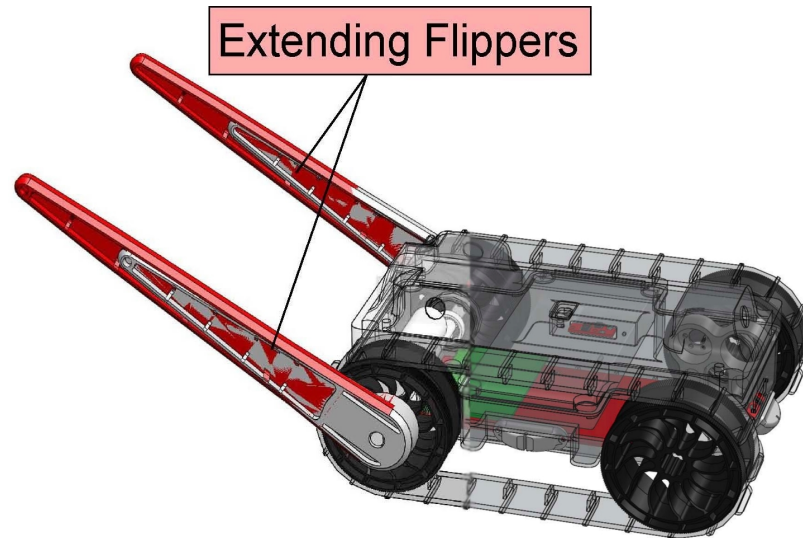
Problem: Form Adaptation

- What needs to change for new conditions?
- How does a change impact other systems?



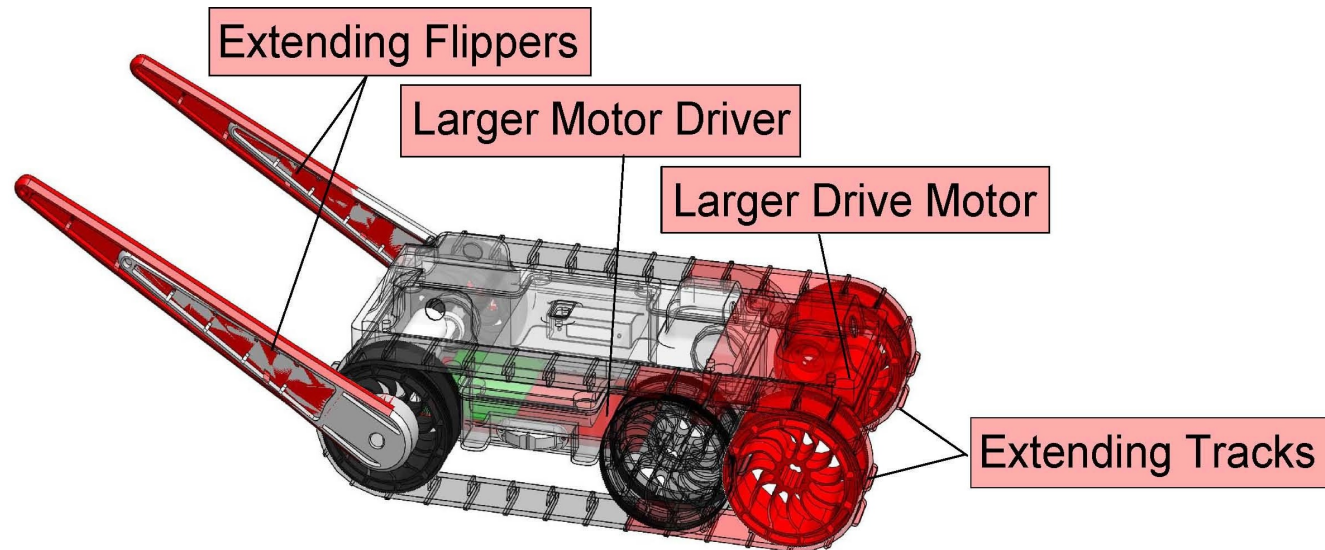
Problem: Form Adaptation

A small change ...



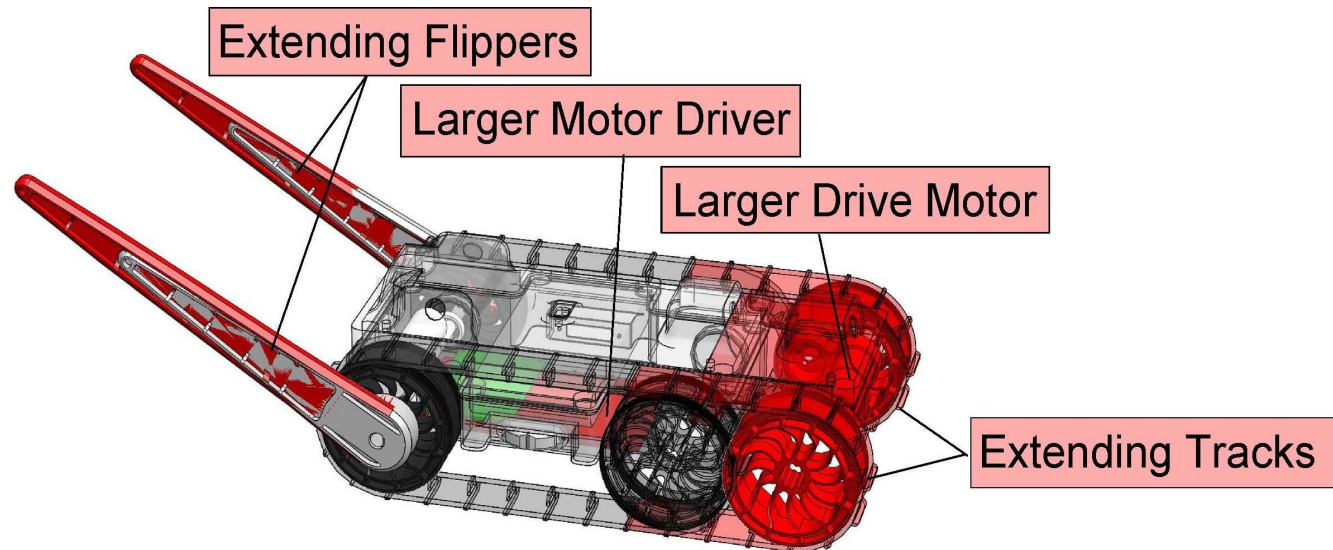
Problem: Form Adaptation

A small change ... has many consequences



Problem: Form Adaptation

A small change ... has many consequences

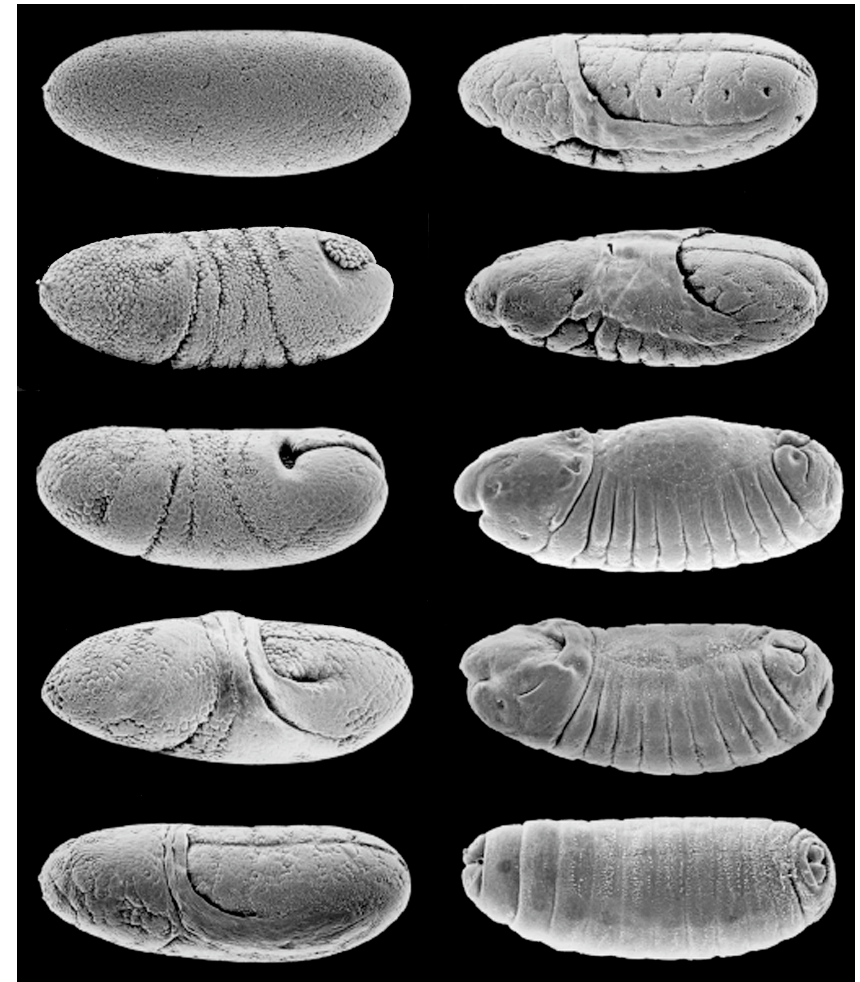


Problems: What to change? *How to change it?*

Approach: Morphogenesis

- Process by which an object's shape develops
- *Developmental Program*
- Bio-inspired design

No biomimicry for the sake of biomimicry!



Drosophila Embryo Morphogenesis

Photo Credit: Parkhurst Labs

Spatial Computing

- Morphogenesis as a method of developing spatial structures
- Cellular-level execution of developmental programs

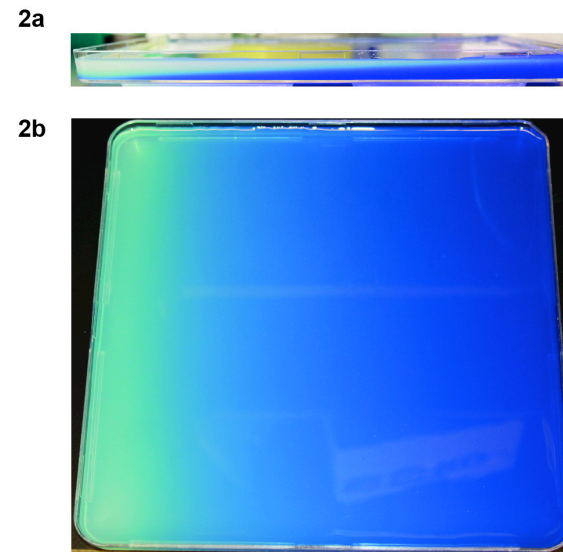
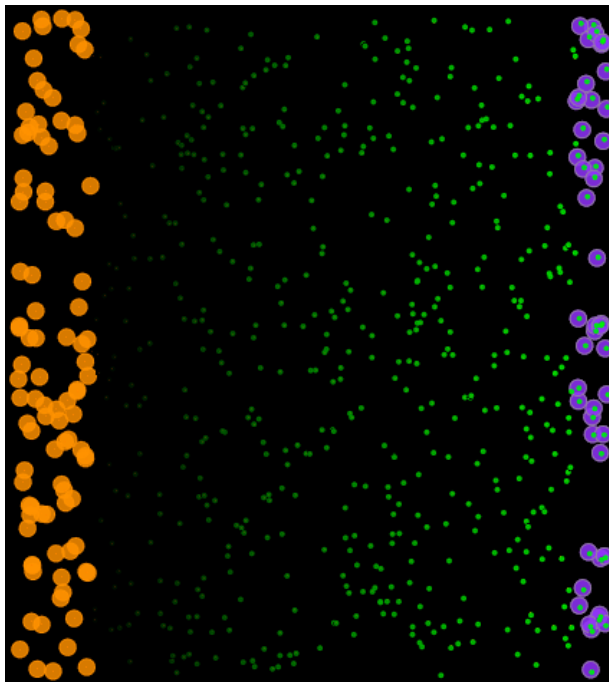


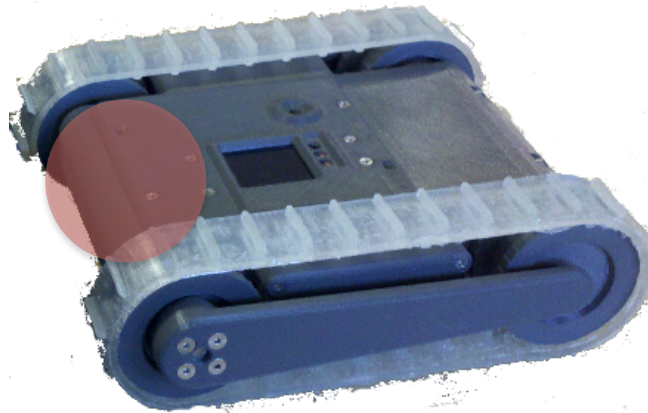
Photo credit: Center for Brain Science, Harvard Uni.

Outline

- Goals for Developmental Programs
 - Developmental epochs
- Structure of Developmental Programs
 - Developmental primitives
 - Developmental rules
- Benefits of Developmental Programs
 - Reduce parameter dimensionality
 - Adaptable
 - Implicitly create a reference architecture for the engineered system
- Contributions and Future Work

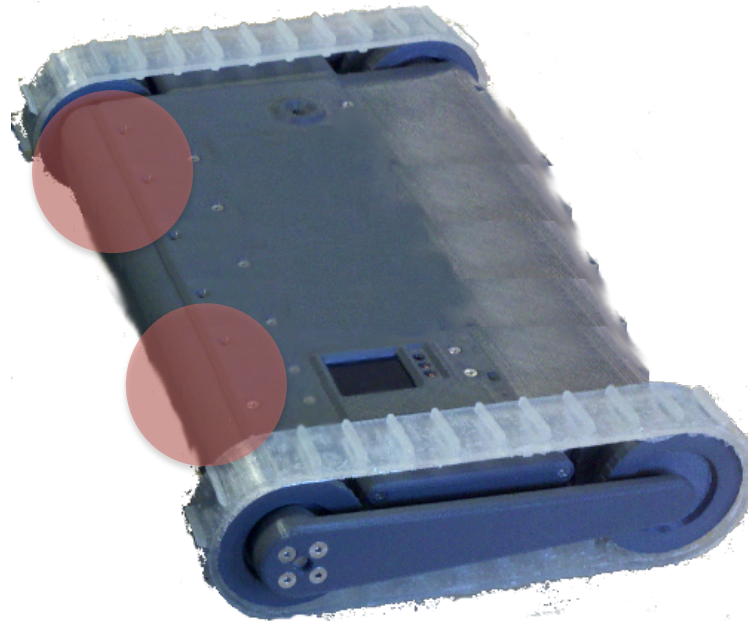
A Complex Transformation...

- Both flippers are driven by one servo...



A Complex Transformation...

- Both flippers are driven by one servo...
... what if the robot widens to need two?



How can we support this **radical** change?

Developmental Epochs

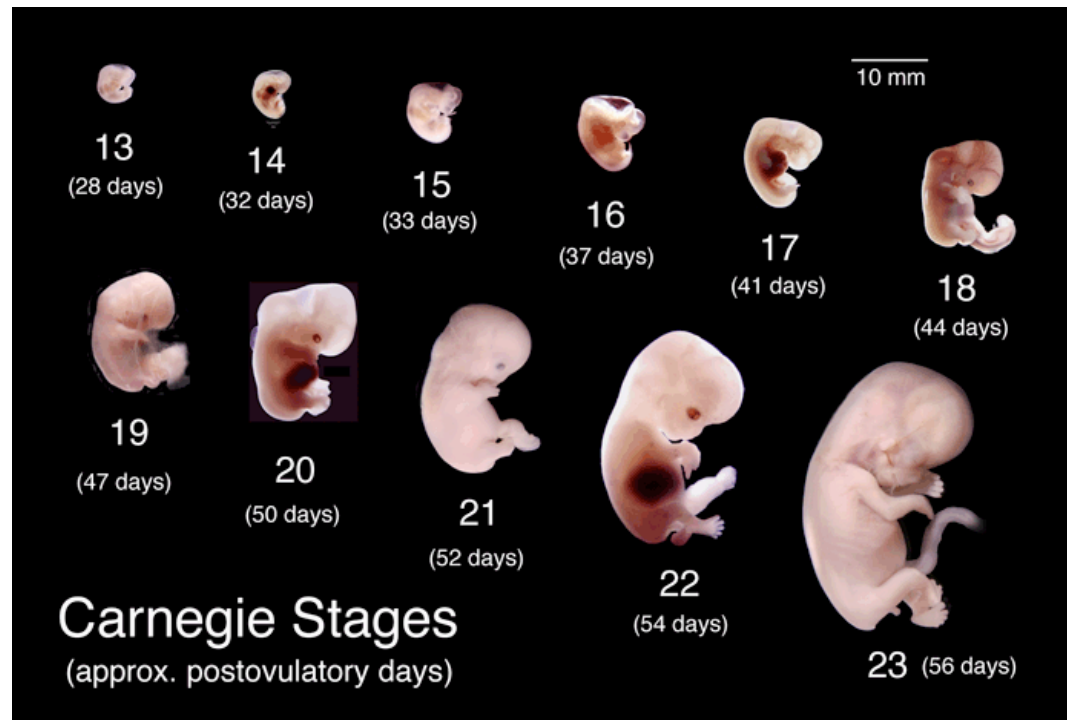
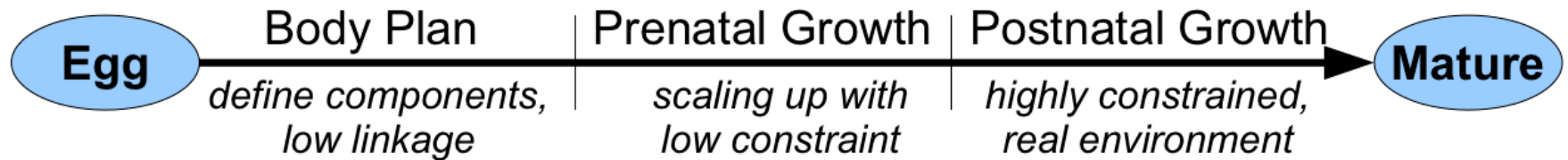
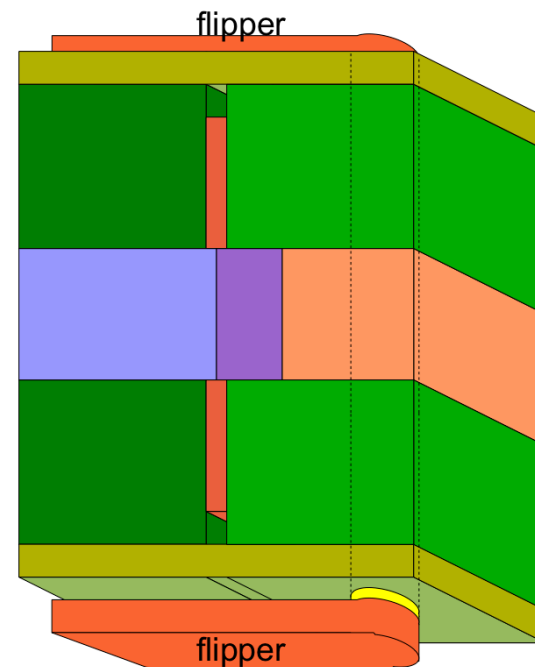
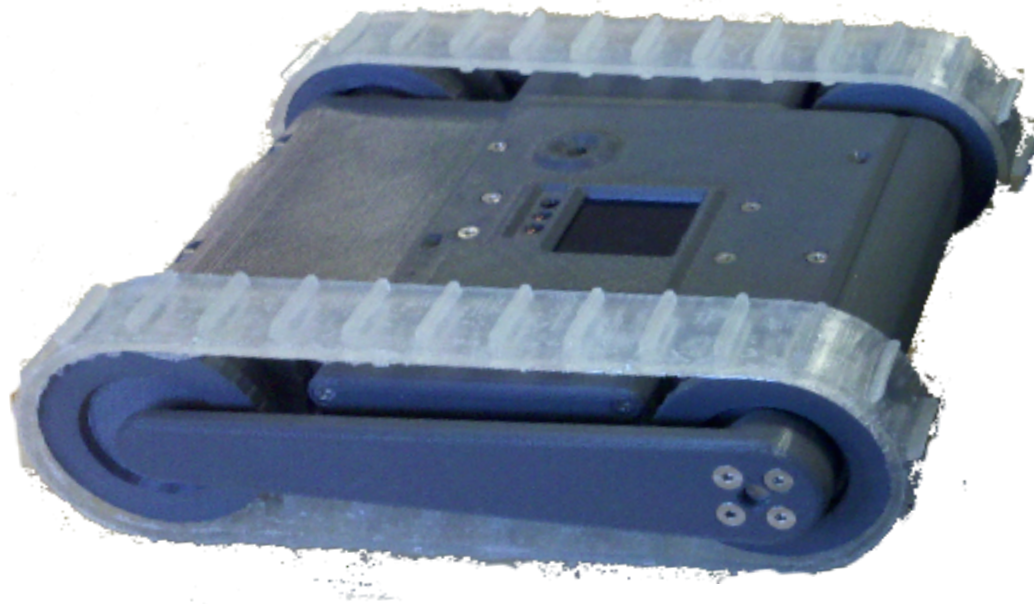


Photo credit:
U. Michigan

Goals for Developmental Programs

- Layout primary body plan components
- Establish relationships between components
- Easy to modify



Outline

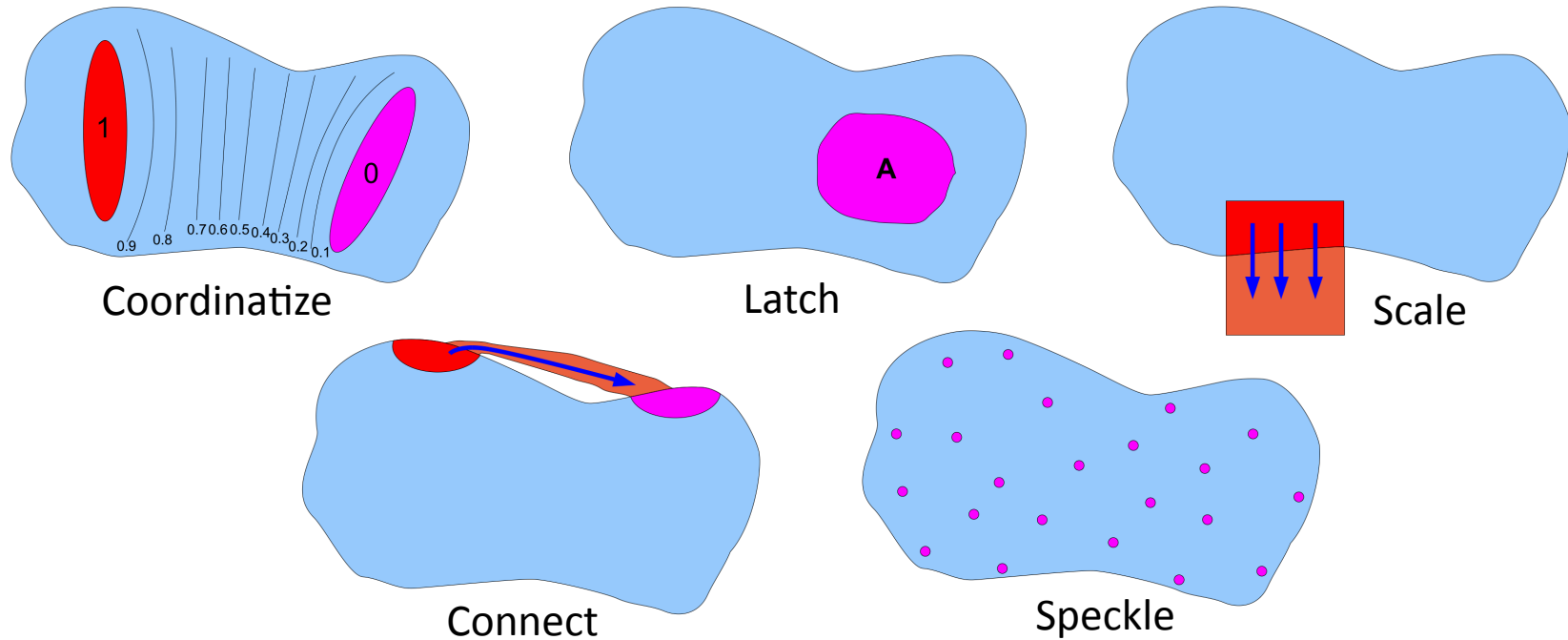
- Goals for Developmental Programs
 - Developmental epochs
- Structure of Developmental Programs
 - Developmental primitives
 - Developmental rules
- Benefits of Developmental Programs
 - Reduce parameter dimensionality
 - Adaptable
 - Implicitly create a reference architecture for the engineered system
- Contributions and Future Work

Structure of Developmental Programs

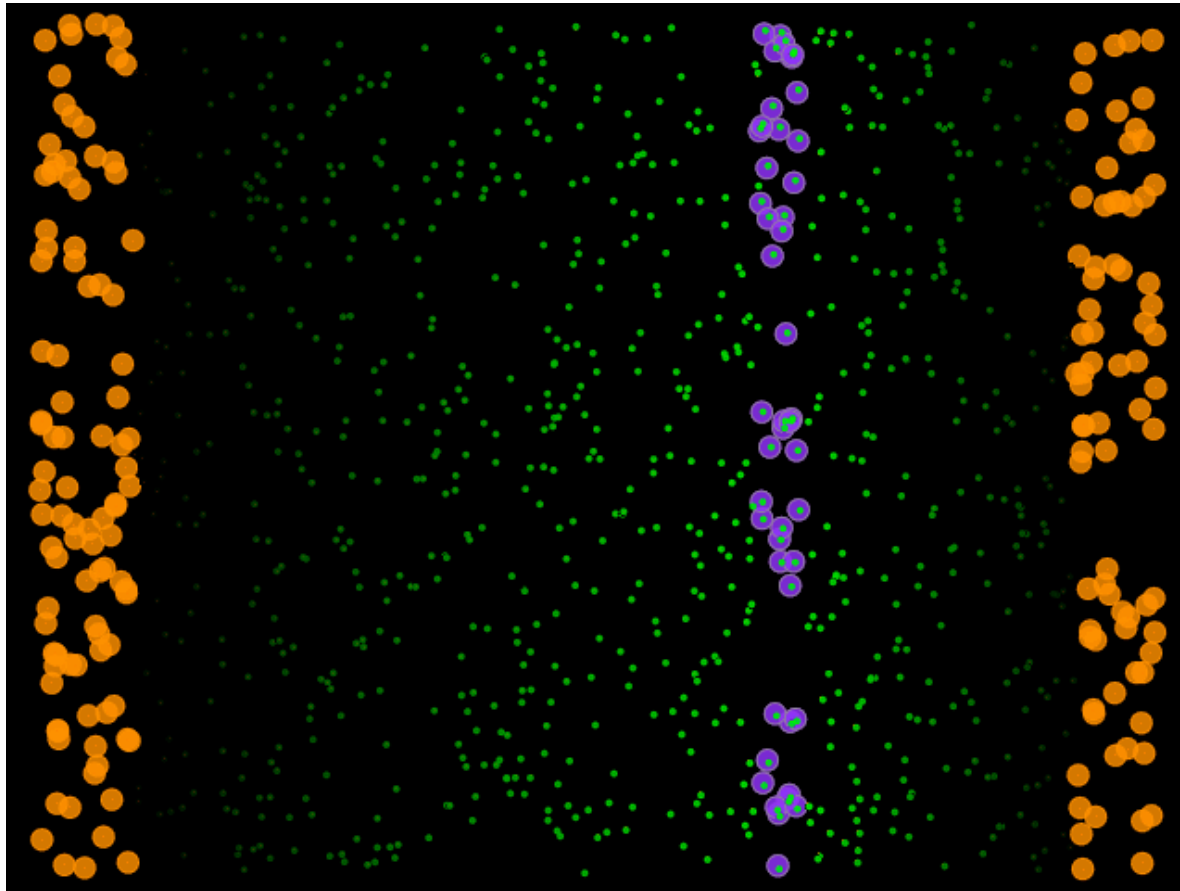
- Set of *developmental rules* specifying preconditions and effects
- Effects are compositions of *developmental primitives*
- Why developmental rules?
 - Parallel application,
 - Implicit relations,
 - Easy to modify/insert
- How rules work
 - Continuous manifold evolution,
 - Conflict resolution by actuator blending

Developmental Primitives

We begin with 5 biology-inspired manifold operations:

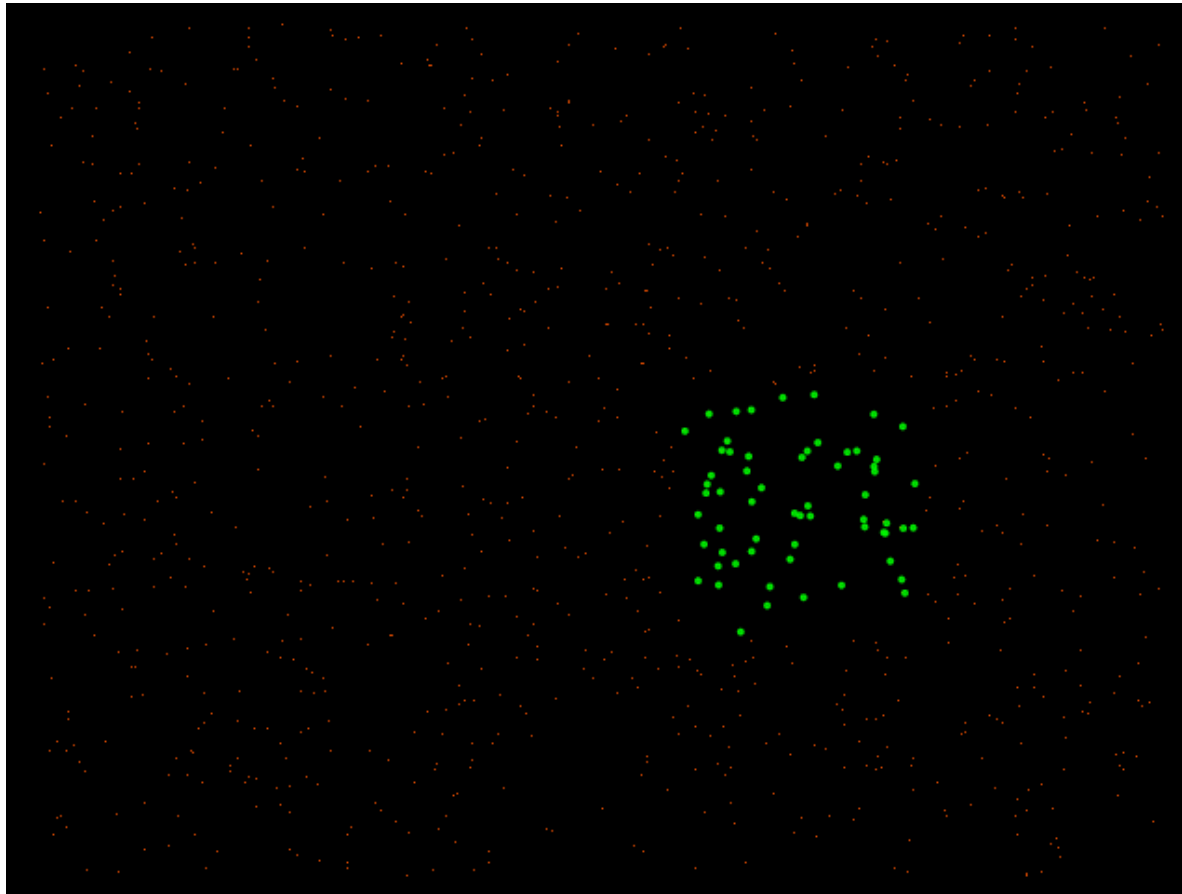


Coordinatize Primitive



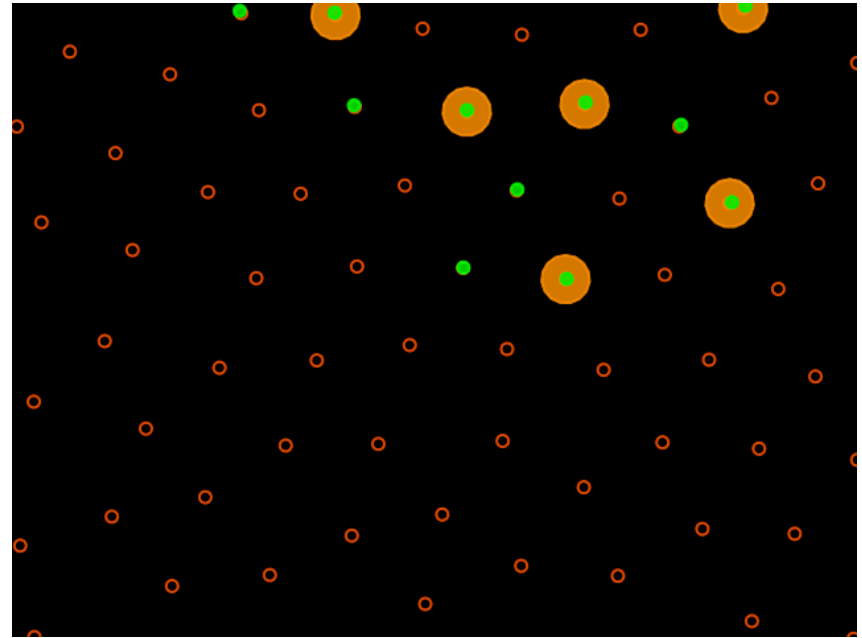
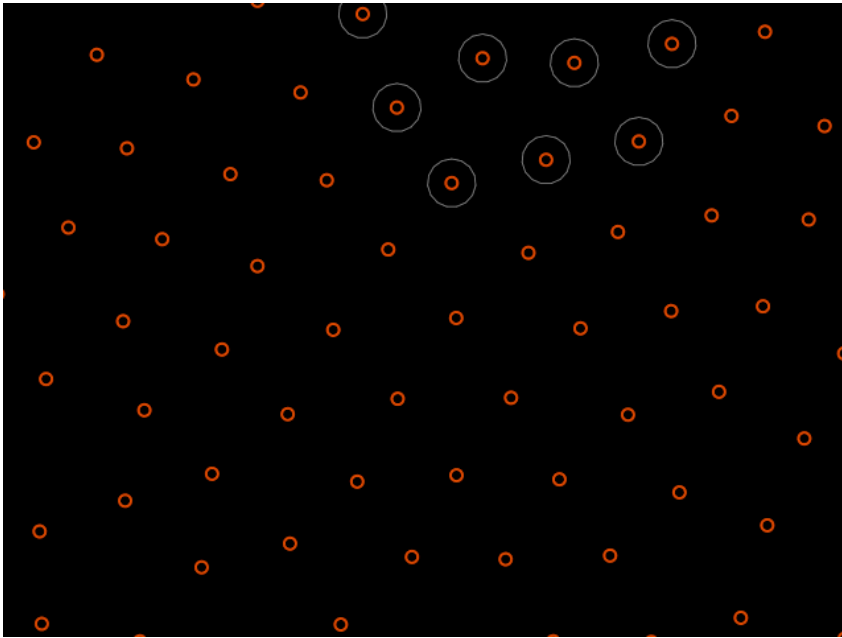
coordinatize (0-region, 1-region)

Latch Primitive



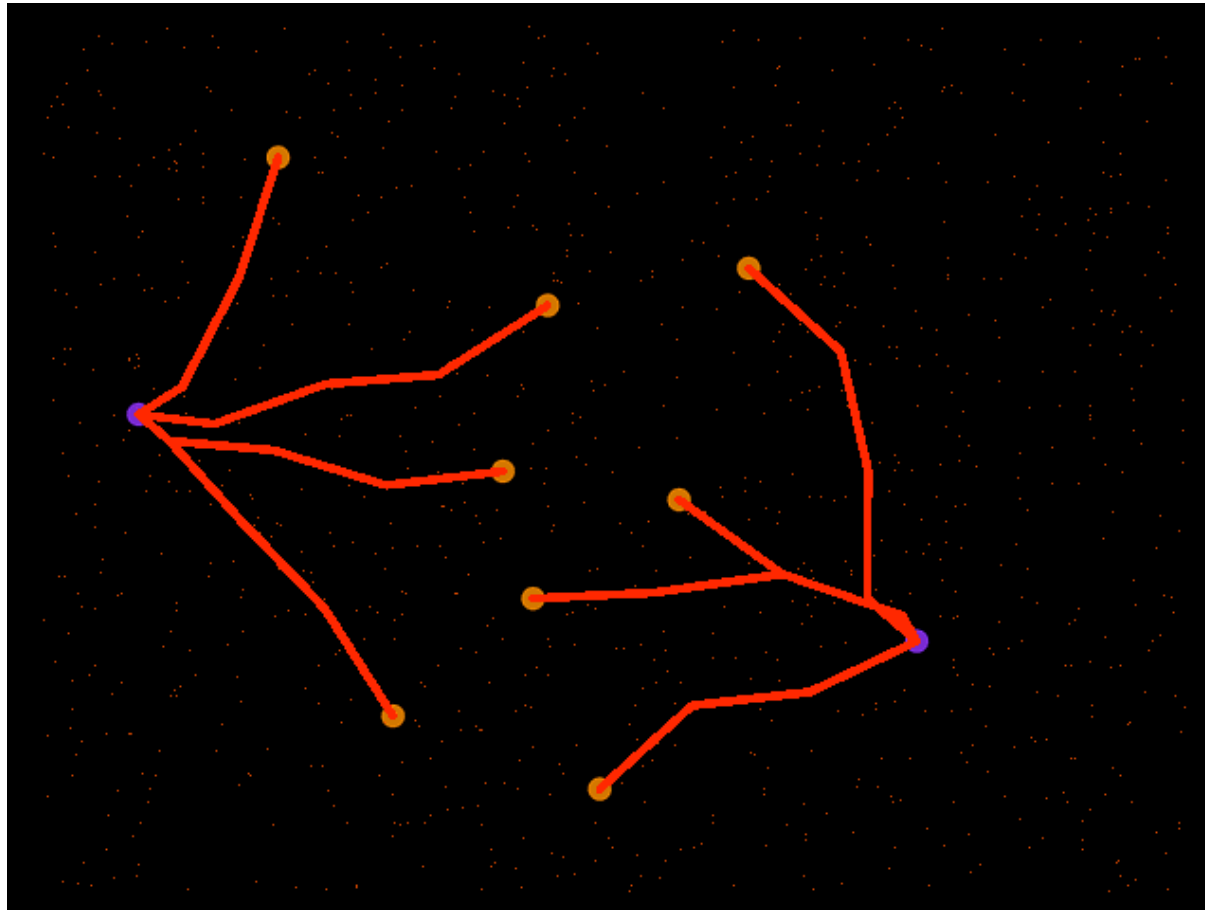
latch (region, type)

Scale Primitive



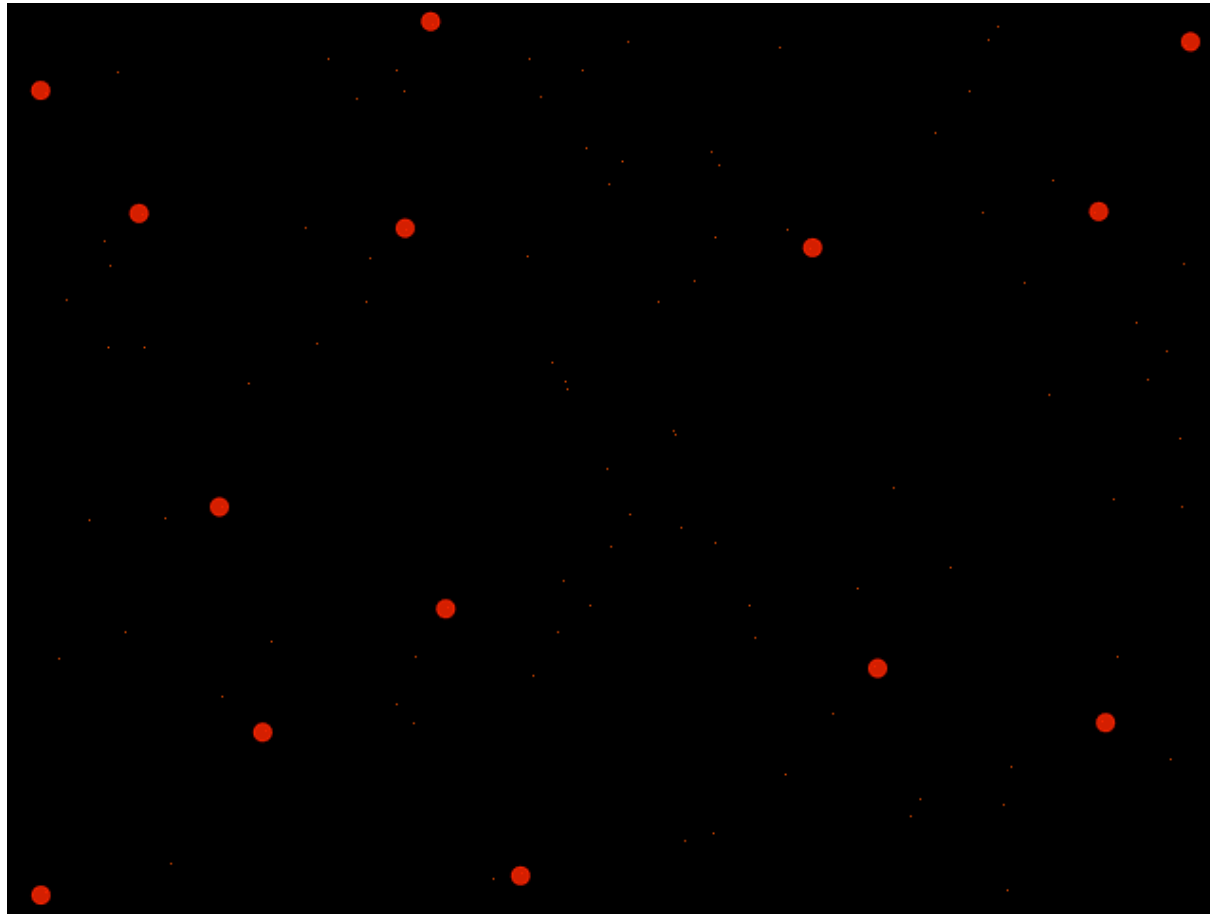
scale (**region**, scale-factor)

Connect Primitive



connect (**source-region**, **destination-region**)

Speckle Primitive



speckle (region, expected-separation)

Developmental Rules

Precondition: Tissue = egg
Anteroposterior > 0.5
Mediolateral > 0.33
Effects: Latch(limb-bud)

Precondition: Tissue = egg
Anteroposterior < 0.5
Mediolateral > 0.33
Effects: Latch(limb-bud)

Precondition: Tissue = limb-bud
Effects: Coordinatize(Proximodistal)

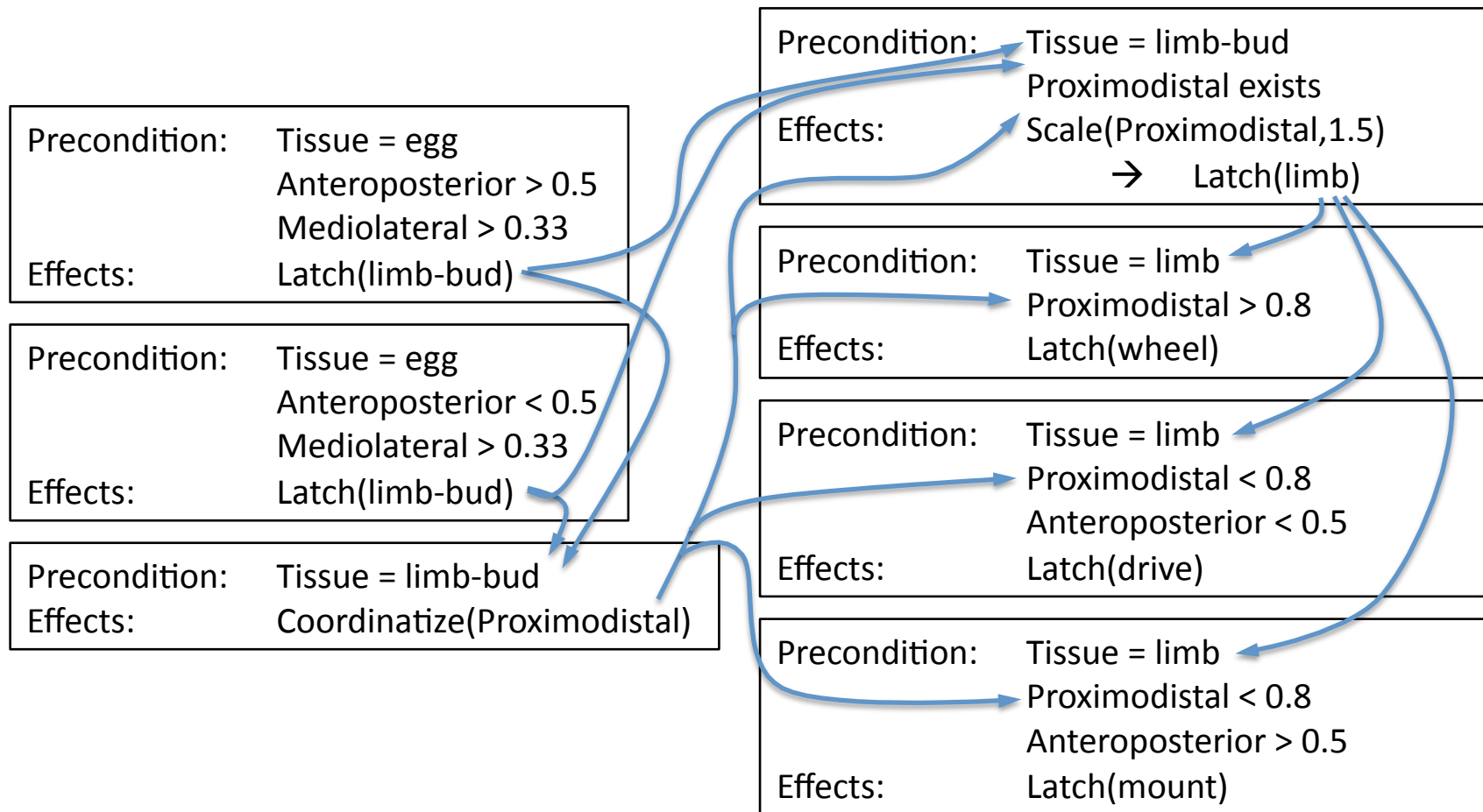
Precondition: Tissue = limb-bud
Proximodistal exists
Effects: Scale(Proximodistal,1.5)
→ Latch(limb)

Precondition: Tissue = limb
Proximodistal > 0.8
Effects: Latch(wheel)

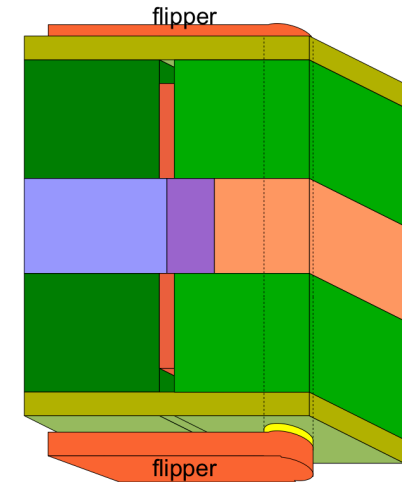
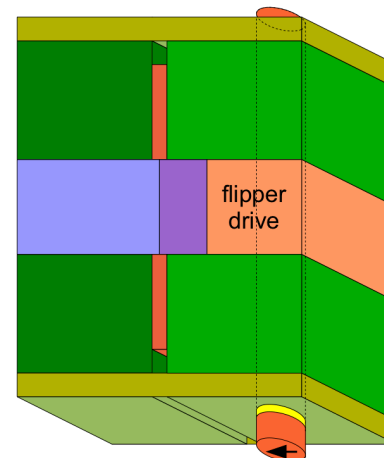
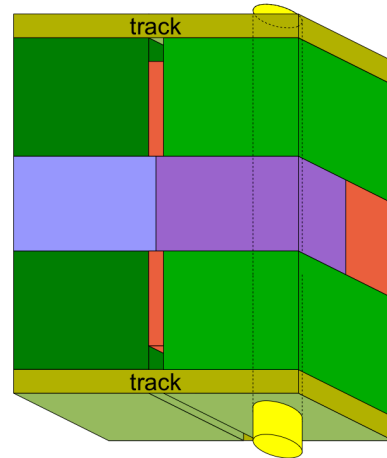
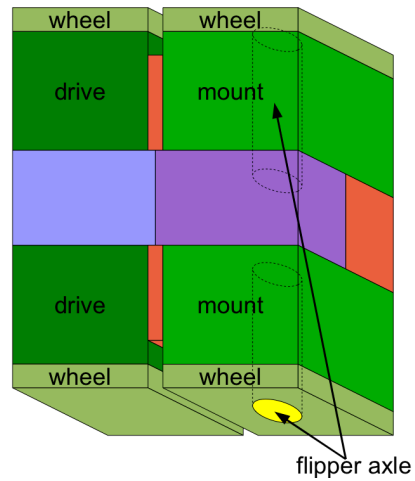
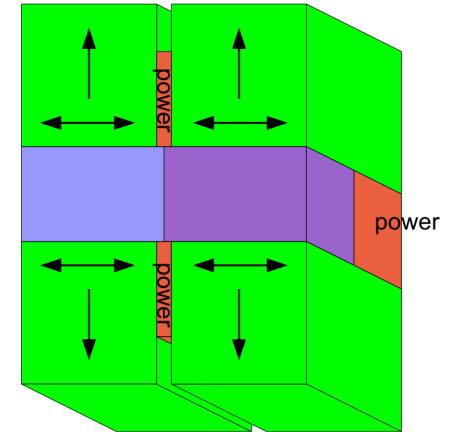
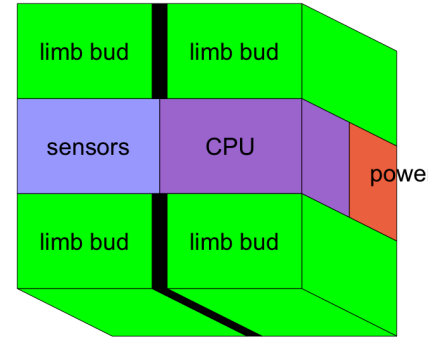
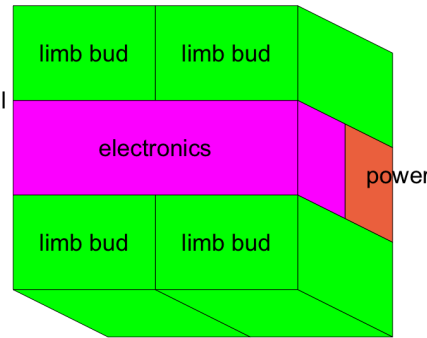
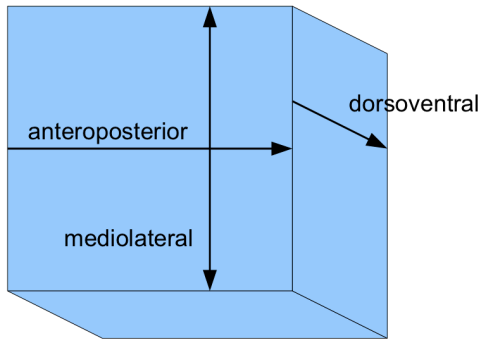
Precondition: Tissue = limb
Proximodistal < 0.8
Anteroposterior < 0.5
Effects: Latch(drive)

Precondition: Tissue = limb
Proximodistal < 0.8
Anteroposterior > 0.5
Effects: Latch(mount)

Developmental Rules



Developmental Program for Body Plan



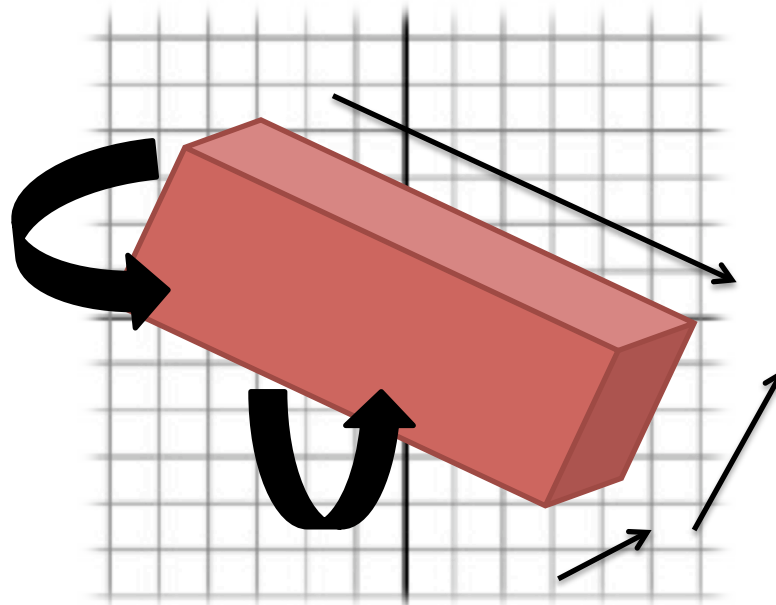
Next: prenatal growth

Outline

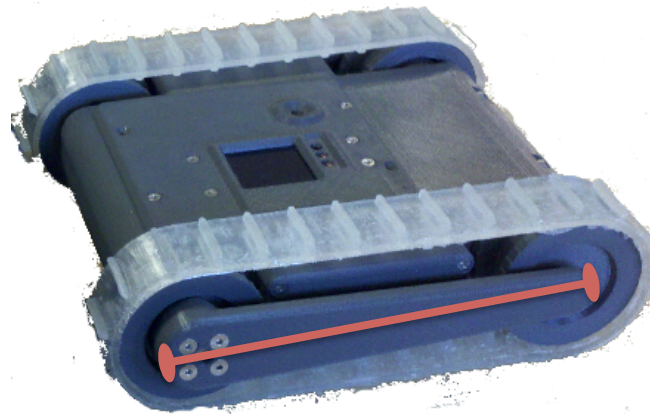
- Goals for Developmental Programs
 - Developmental epochs
- Structure of Developmental Programs
 - Developmental primitives
 - Developmental rules
- Benefits of Developmental Programs
 - Reduce parameter dimensionality
 - Adaptable
 - Implicitly create a reference architecture for the engineered system
- Contributions and Future Work

Benefits of Developmental Programs

- **Reduce parameter dimensionality**
- Adaptable
- Implicitly create a reference architecture for the engineered system

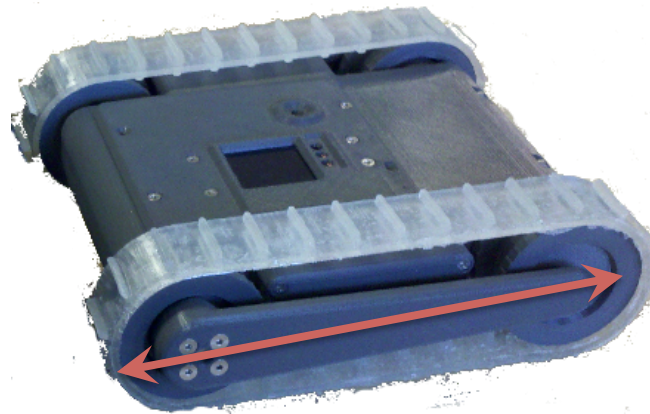


How Hard is it to Change Flipper Length?



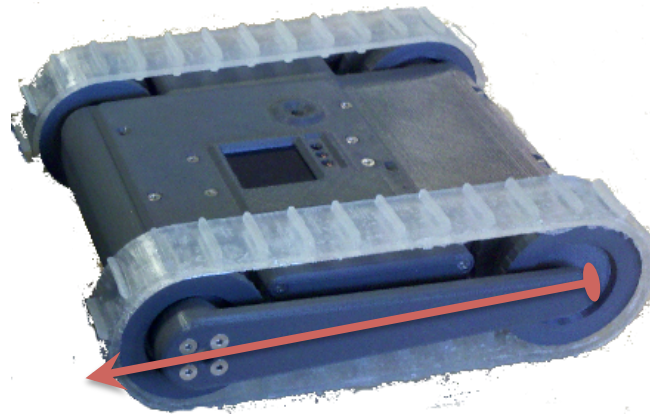
How Hard is it to Change Flipper Length?

Does it grow from the center?



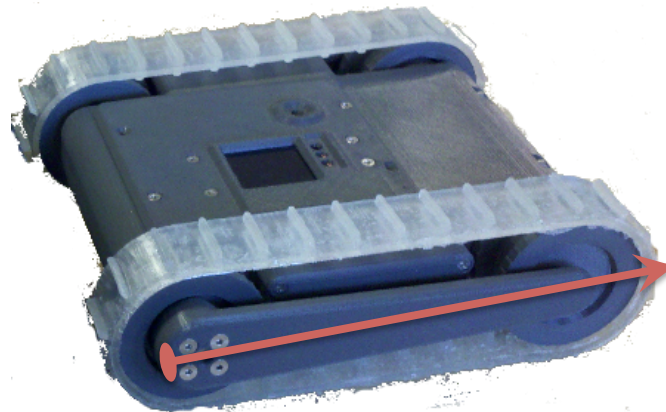
How Hard is it to Change Flipper Length?

Does it grow from the center? the front?



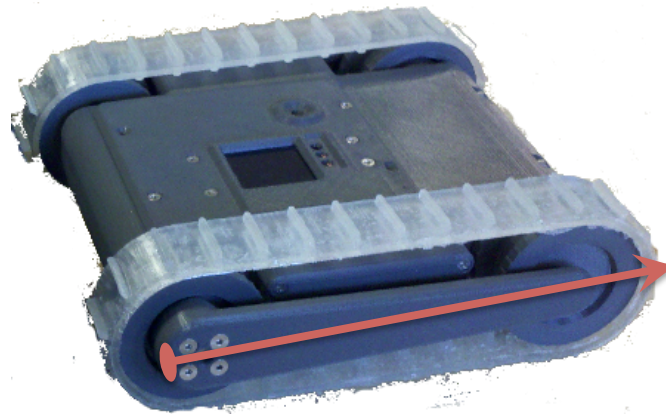
How Hard is it to Change Flipper Length?

Does it grow from the center? the front? the back?



How Hard is it to Change Flipper Length?

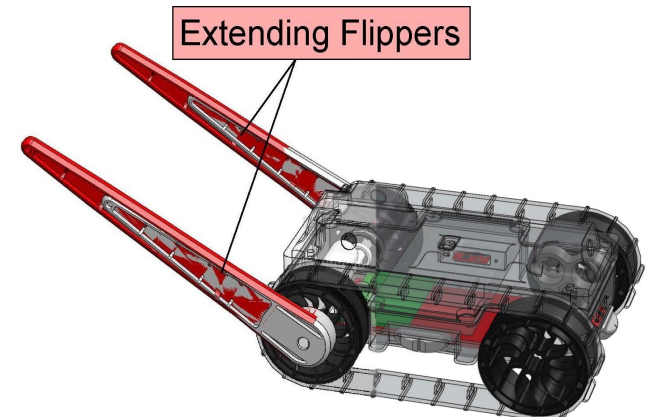
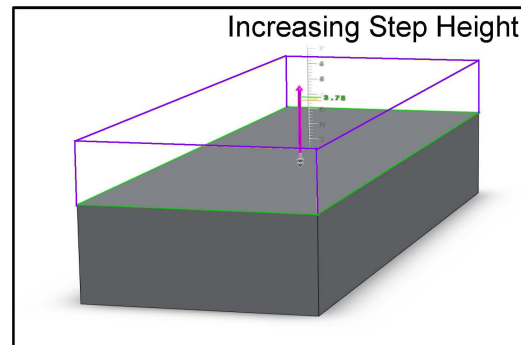
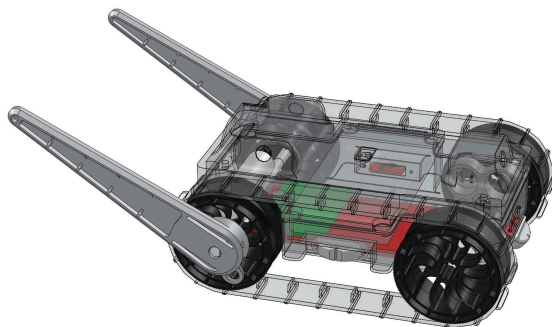
- Only a few parameters are *key*, while most are constrained by their relationships to key parameters.



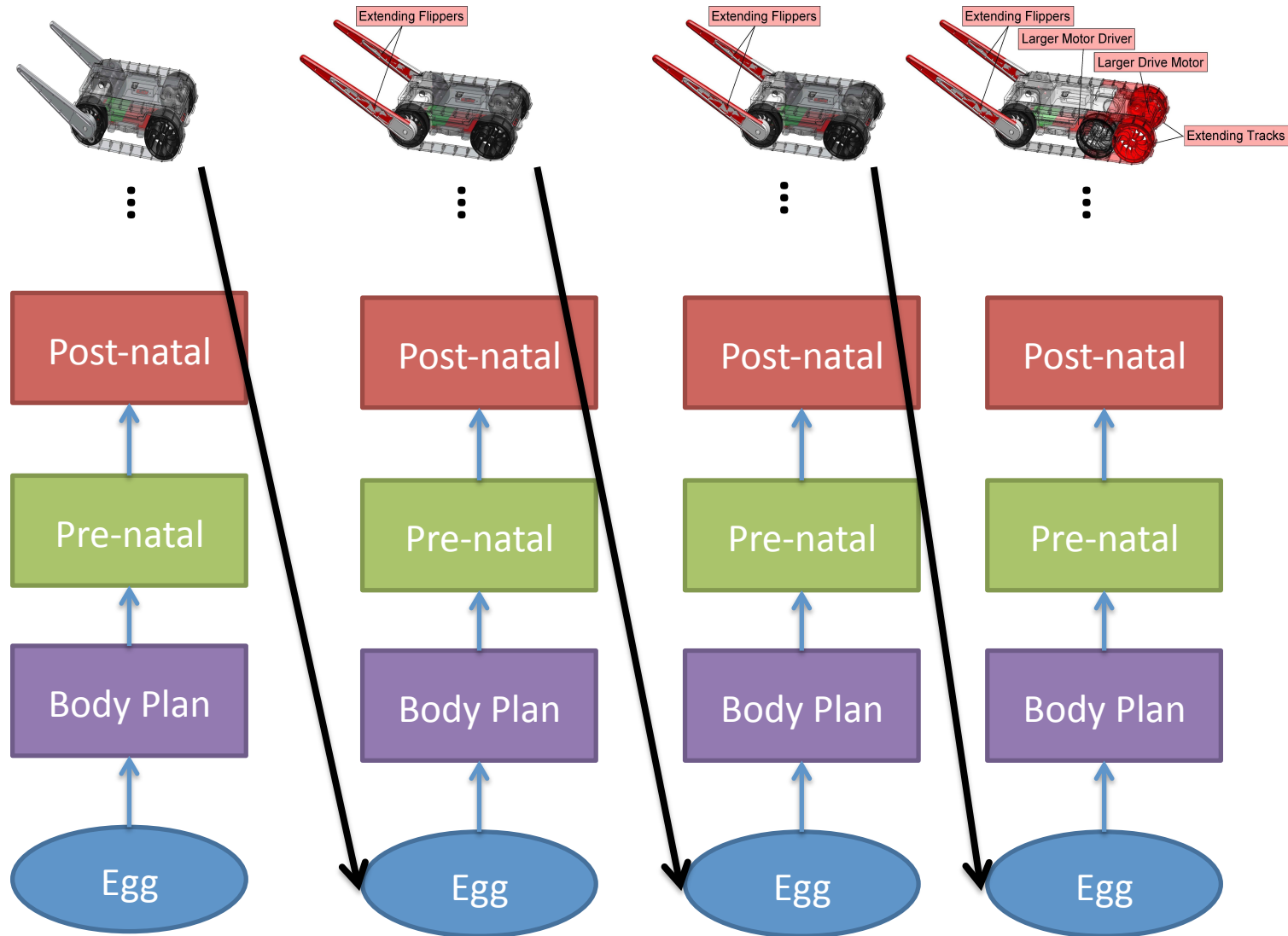
Developmental programs encode the spatial relationships between components, effectively reducing the number of design parameters.

Benefits of Developmental Programs

- Reduce parameter dimensionality
- **Adaptable**
- Implicitly create a reference architecture for the engineered system



Embryonic Adaptation

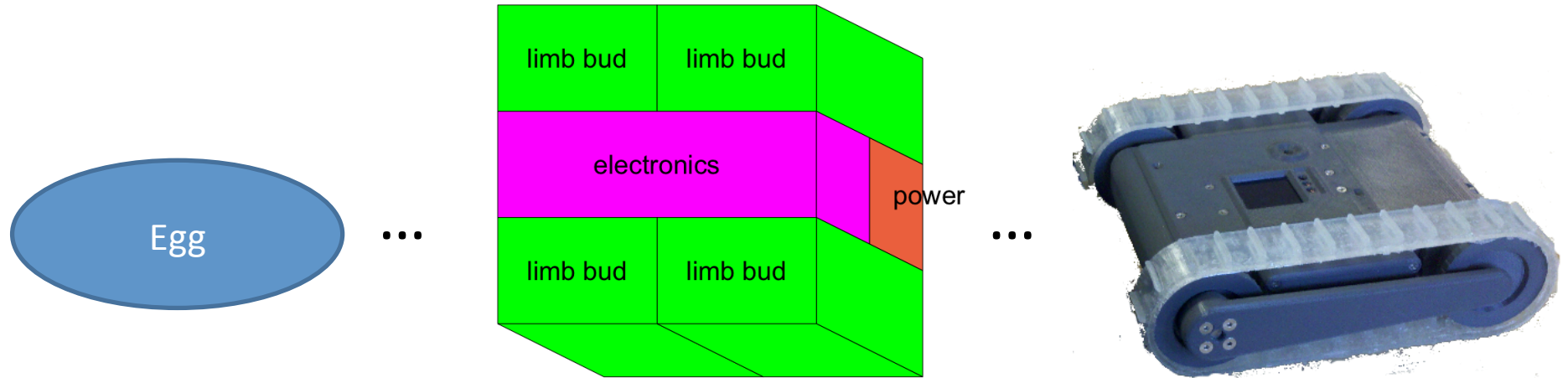


Benefits of Developmental Programs

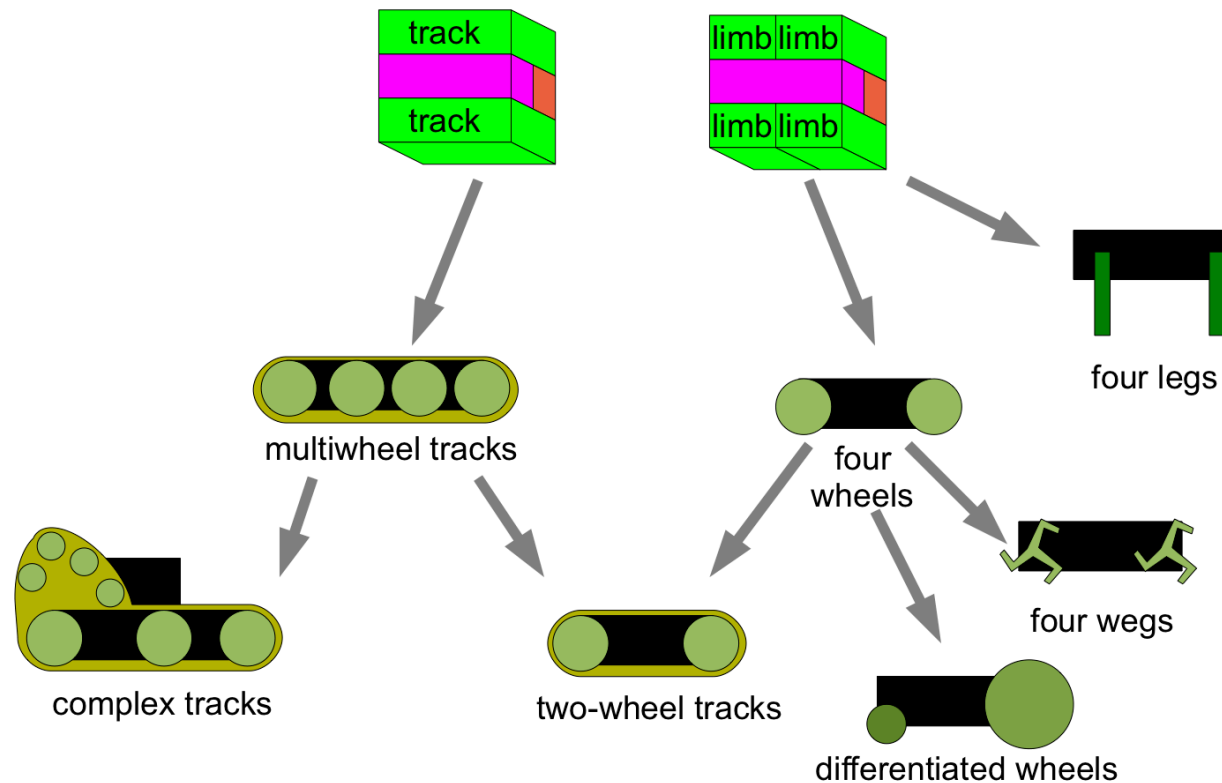
- Reduce parameter dimensionality
- Adaptable
- **Implicitly create a reference architecture for the engineered system**

A reference architecture captures the “best practices” for an architectural solution in a particular domain.

Writing Developmental Programs

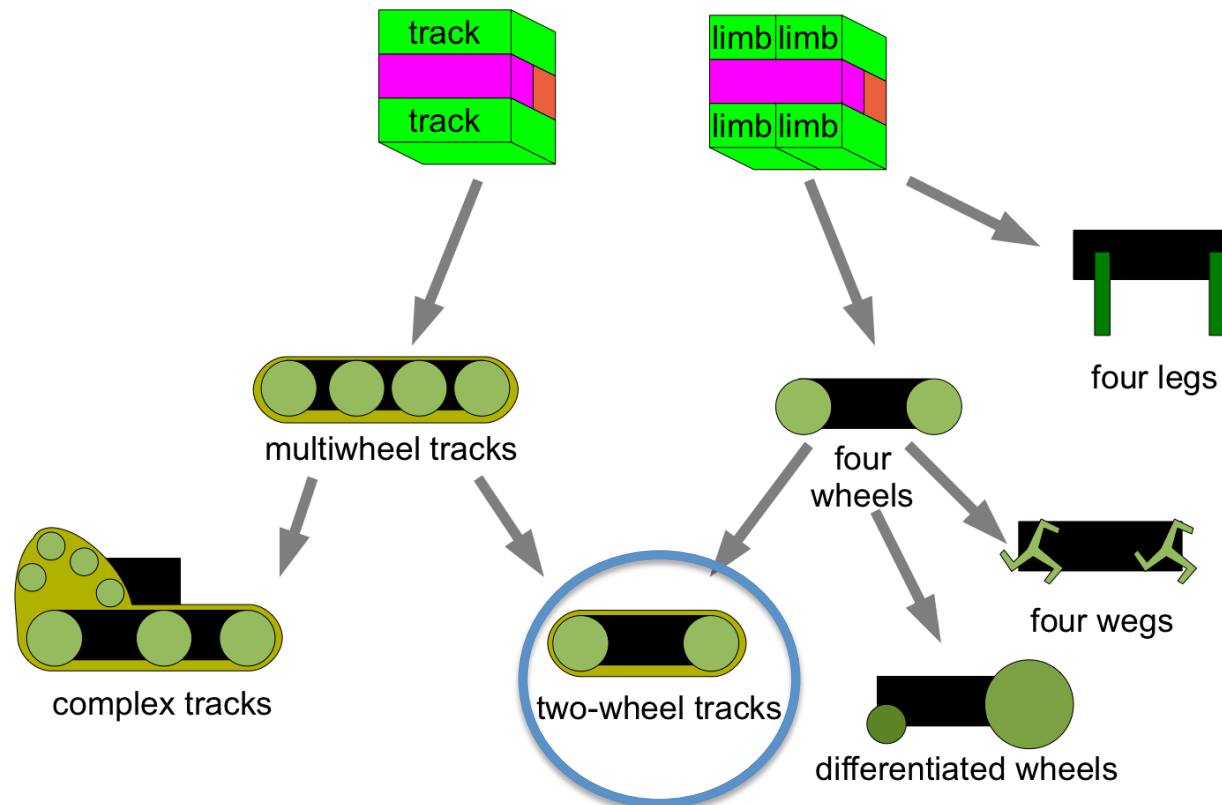


Creating a Reference Architecture



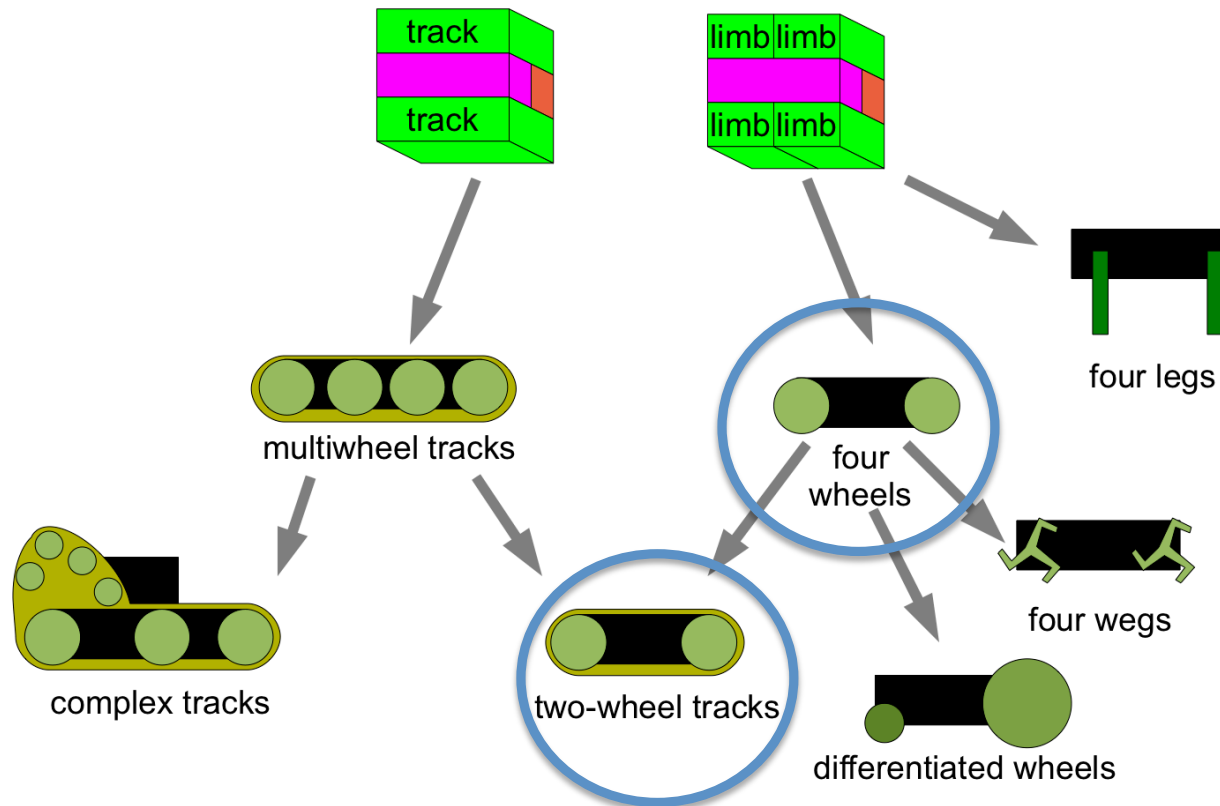
- The sequence of development for a body plan implies a *prioritization* of major design features, selecting a family of more accessible variants

Creating a Reference Architecture



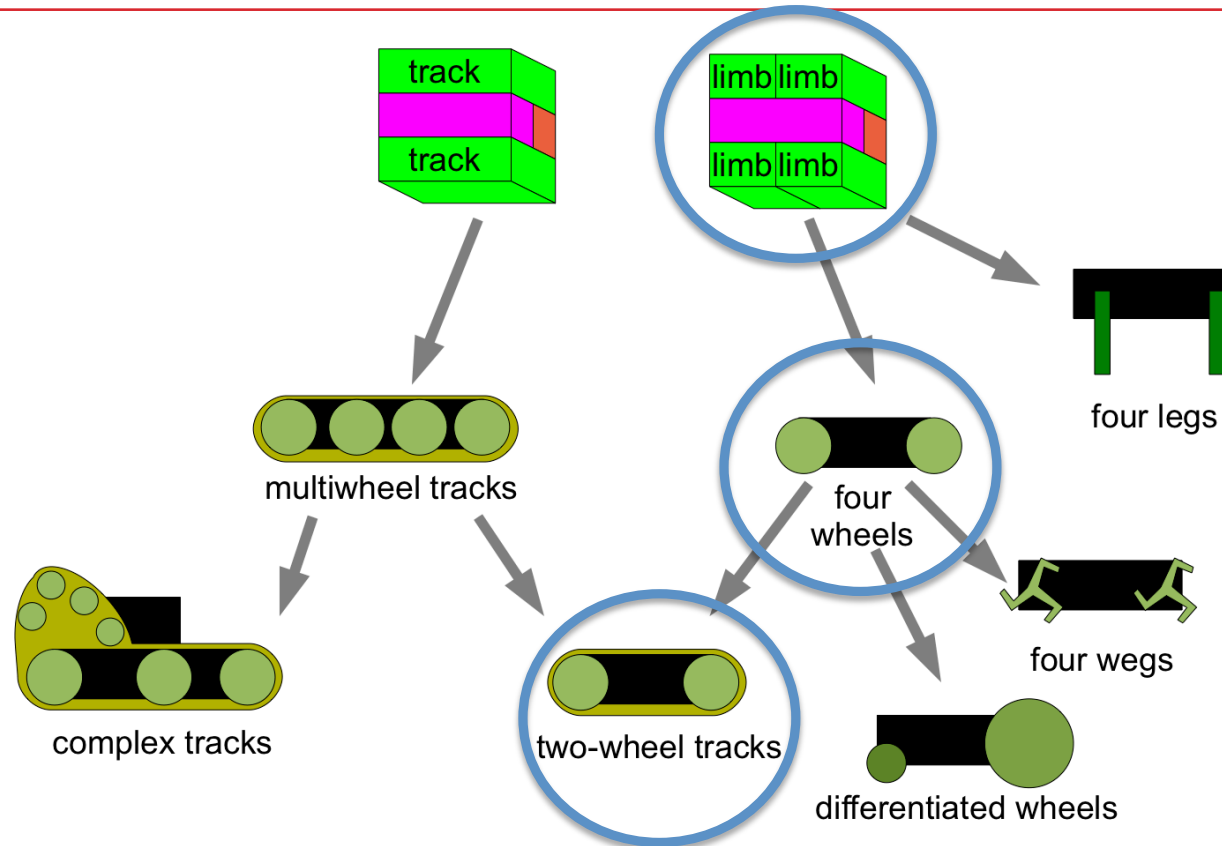
- The sequence of development for a body plan implies a *prioritization* of major design features, selecting a family of more accessible variants

Creating a Reference Architecture



- The sequence of development for a body plan implies a *prioritization* of major design features, selecting a family of more accessible variants

Creating a Reference Architecture



- The sequence of development for a body plan implies a *prioritization* of major design features, selecting a family of more accessible variants

Contributions & Next Steps

- Proposed morphogenesis as a method of developing spatial structures
- Explained how developmental programs encode relationships between design decisions
- Described our preliminary morphogenetic engineering framework
- ***Next:*** A full implementation of tissue-level execution of developmental programs
- ***Goal:*** Cellular execution of developmental programs

Project Team:

Raytheon
BBN Technologies

Jacob Beal (PI)

Aaron Adler (co-PI)

Susan Katz (PM)

Brett Benyo

Jeff Cleveland

Jessica Lowell

Hala Mostafa

Kyle Usbeck

Fusun Yaman

Interns:

- Katie McGuire
- Taylor Campbell

iRobot

Annan Mozeika

Intern:

- Gretchen Markiewicz

Team website:

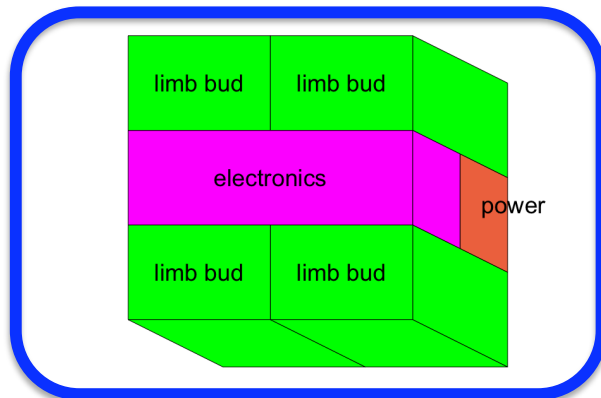
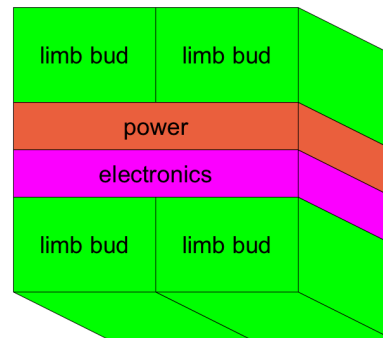
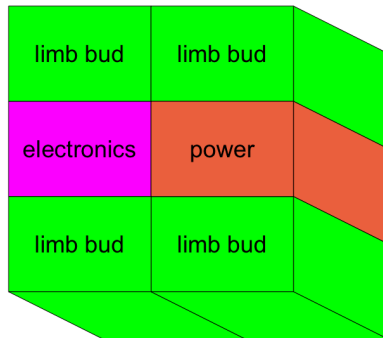
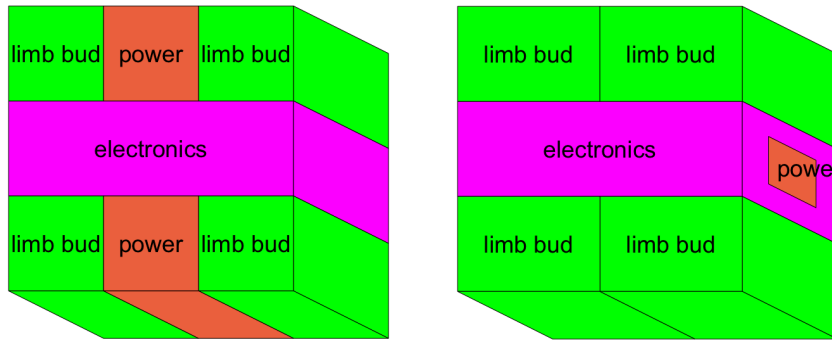
<http://madv.bbn.com>

Sponsored by:



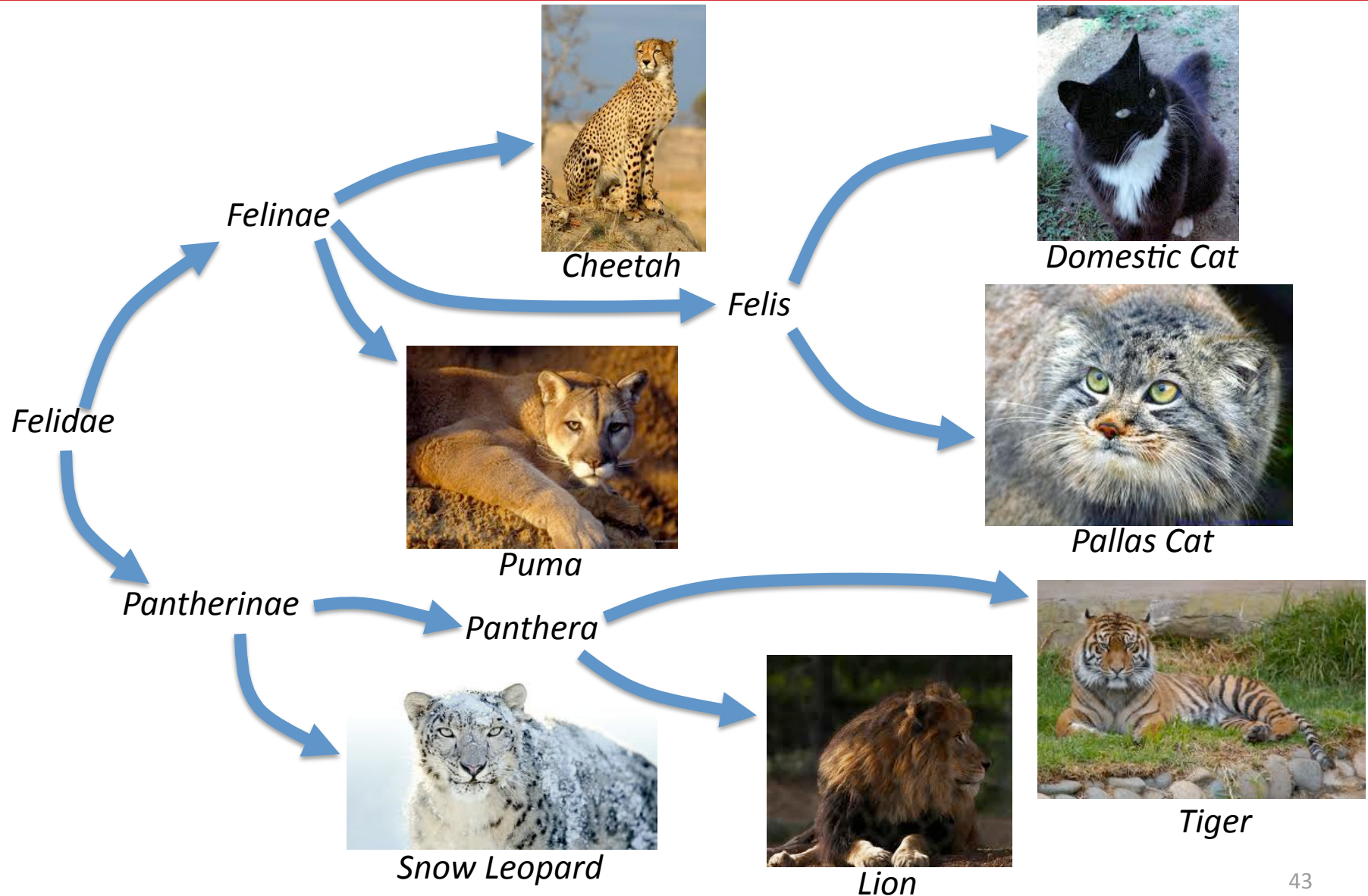
Backup

Considering Alternative Body Plans



- Don't split electrical or power components
- Nested body adds complexity
- Component weight differences affect stability

Morphogenesis enables natural variation



A taxonomy of engineered systems?



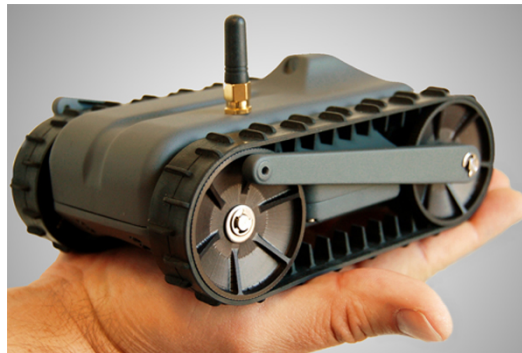
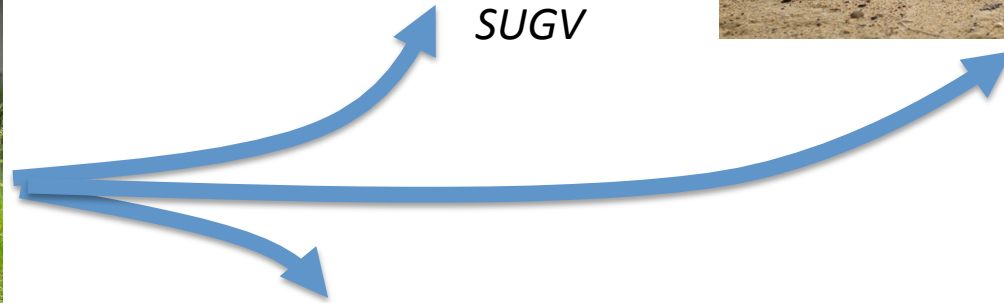
PackBot



SUGV



Warrior



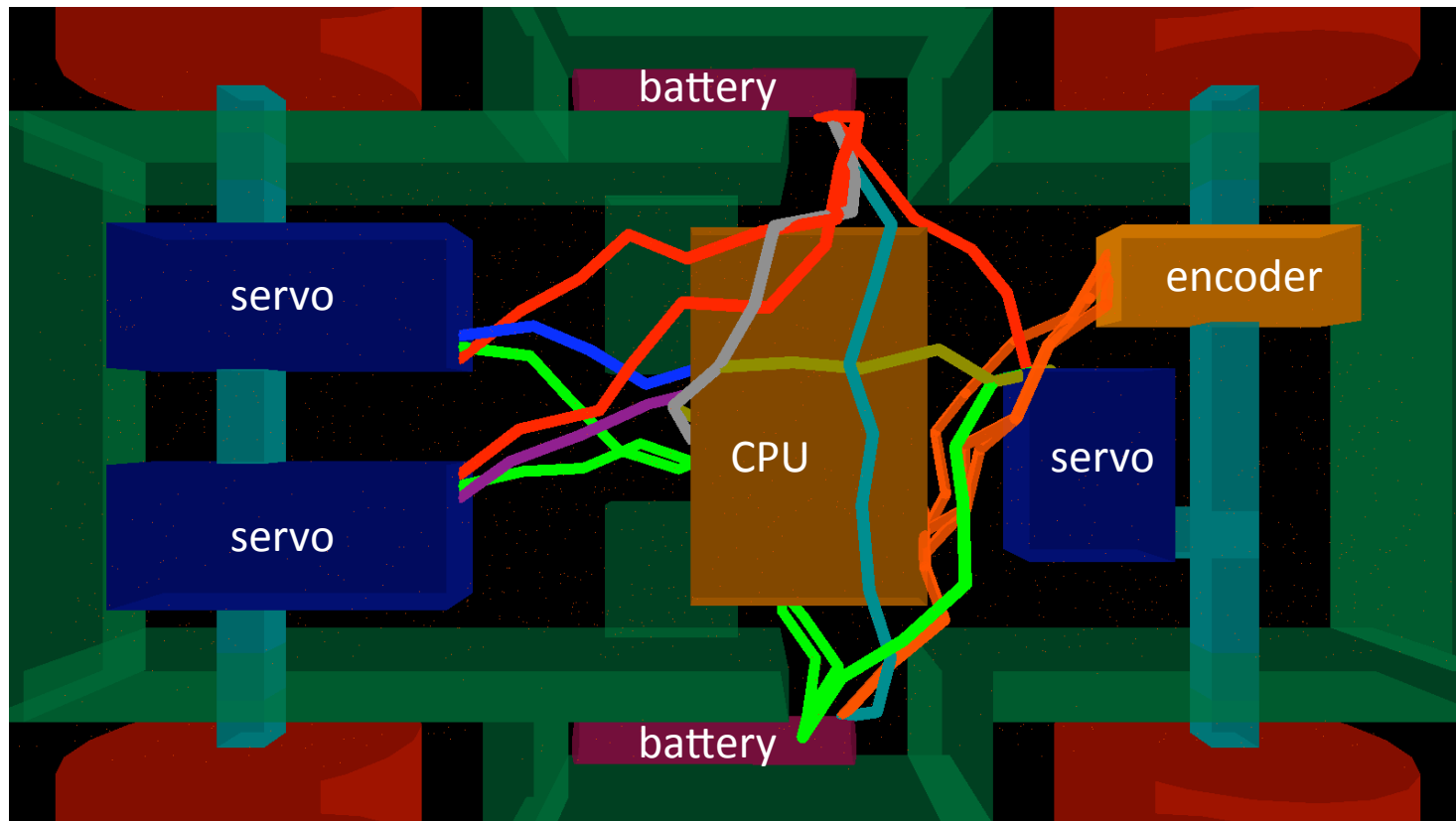
LANdroid



miniDroid

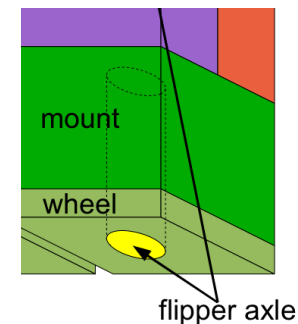
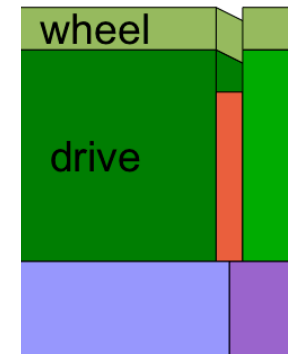
Details: Wiring

- Chemotactic model:

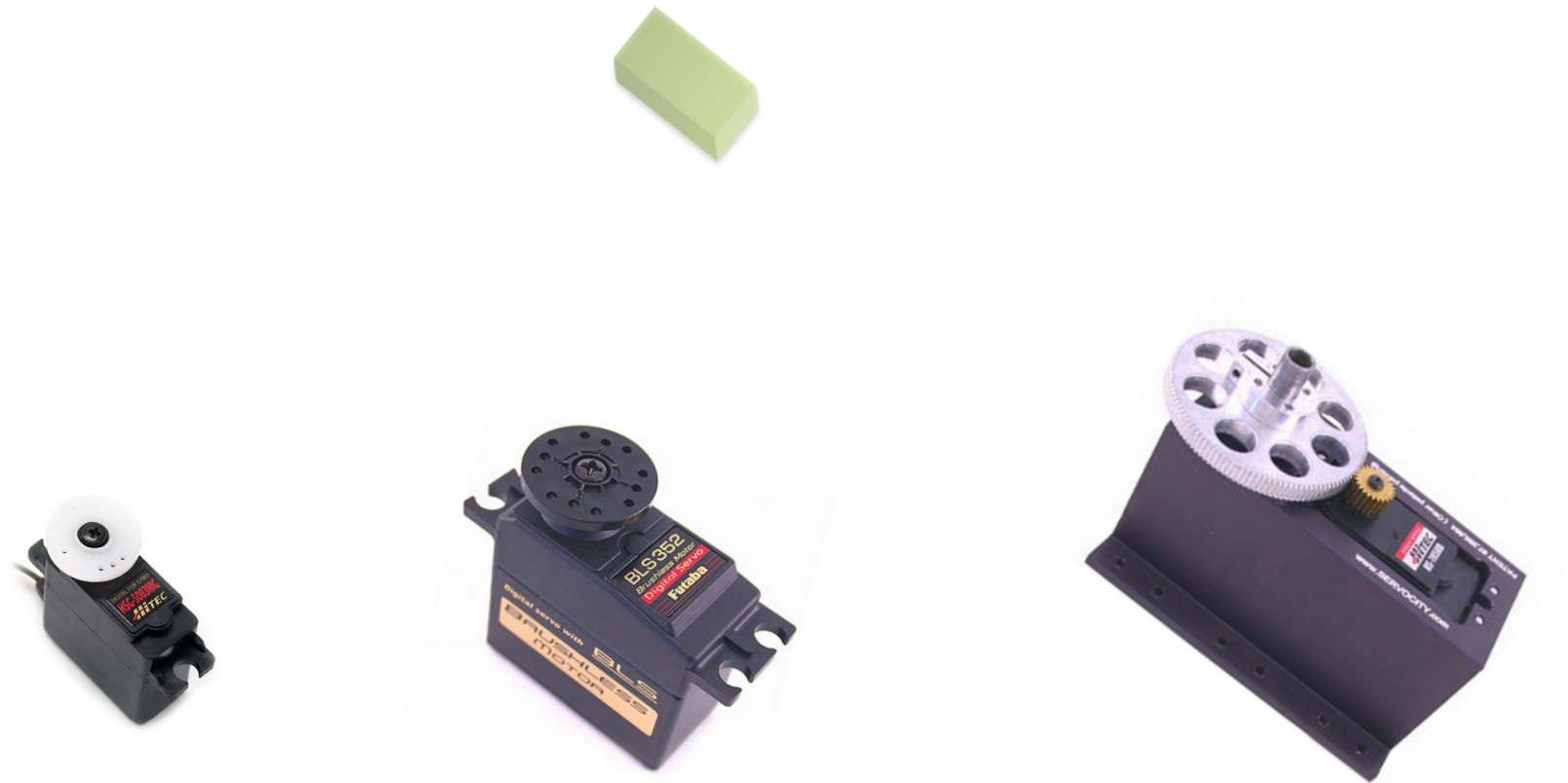


Other Key Developmental Decisions

- Wheel attachments: packaged component with a “base limb” included within wheel
- Flippers: based on wheel axles ensures flipper/wheel structure integrity, but makes them hard to separate



Details: Packaged Components



Approach: interpolate across parts from a
“Component Model Library”

Cellular Scale Operator

- The cells in the tissue seek uniform density.
- We select the cells we want to scale.
- The selected cells latch and turn green.
- The green cells duplicate themselves (producing more green-type cells).
- The cells continue seeking uniform density.

```
proto -m -l -L simple-life-cycle "(all (green (rep once 0 (if (and (= 0 once) (sense 1)) (all (clone (sense 1)) 1) once))) (mov (disperse)))"
```