An Endosymbiotic Model for Modular Acquisition in Stochastic Developmental Systems

John Rieffel[1] and Jordan Pollack[1]
[1] DEMO Lab, Computer Science Dept, Brandeis University, Waltham MA
jrieffel@cs.brandeis.edu

Abstract

Hierarchical modular composition is often cited as a requisite for scalable complexity. The most popular reference in this regard is Herbert Simon's allegory of the watchmakers Tempus and Hora. And yet, while numerous studies have used this story as a jumping-off point to explain the emergence of hierarchical modular composition in evolutionary systems, relatively few emphasize the role of *noise* in the parable. Developmental representations, which model "biological assembly", are a suitable lens through which to explore this latter aspect, since noise and error during ontogeny can have a significant negative impact on the progress of evolution. In this work we present an endosymbiotic model for modular acquisition in a developmental representation, and demonstrate its ability to hierarchically assemble large objects in the presence of developmental noise.

Introduction

In his landmark essay "The Architecture of Complexity", Herbert Simon (1962) makes the case for the evolutionary necessity of hierarchical modularity through his parable of the watchmakers Tempus and Hora. Tempus builds his watches incrementally, piece by piece, and when interrupted, puts down the watch he is working on, which then falls apart into its constituent pieces. Hora, on the other hand, first combines pieces into separate small modules, and then combines those modules together into a final watch. Consequently, he only loses the particular module that he is working on, and so is significantly more likely to build a complete watch than Tempus is. Simon uses this story to claim that increasingly complex forms are nearly impossible to evolve without such hierarchical composition of *stable* modules.

Many researchers have used this parable as inspiration for exploring adaptive representations and the relationship between modularity and evolvability. In this context the stability of a form is interpreted as evolutionary stability - that is, insulation from potentially deleterious effects of mutation and crossover. As such, the evolvability of adaptive representations comes from their ability to dynamically generate modules in the process of search (Wagner and Altenberg, 1996).

Our interest here, however, is in a different perspective on the story - that of the relationship between modularity and noise in developmental representations,which take their cue from the biological processes of ontogeny and growth. Since they seek to model "biological assembly" (albeit of systems quite distinct from watches), these systems lend themselves to a rather straightforward interpretation of Simon's parable.

Like watchmakers, developmental systems seek to assemble complex forms from primitive constituent parts, and perform their tasks under the risk of interruption. While for the watchmakers interruption during assembly took the form of a telephone call, in developmental systems the role is performed by error and noise during ontogeny. Like watchmakers, in order to generate complex objects, adaptive developmental representations require modularity and, much like Hora's subcomponents, in order to be useful, these developmental modules must be stable and reliably producible in the presence of noisy assembly.

Unfortunately, conventional models of modular acquisition, which form representational modules by compartmentalizing favorable genetic sequences, can be stymied by developmental representations in which the phenotypic consequence of a genetic sequence is highly contingent upon its context. The challenge of modular acquisition in such *context dependent* developmental representations therefore lies in figuring out a process by which the *meaning*, not just the syntax, of a favorable genetic sequence can be preserved across contexts.

In this work we provide an *endosymbiotic* model of modular acquisition which overcomes the context sensitivity of developmental representations by encapsulating complete phenotypes, rather than genetic sequences, into the set of primitives available to the evolutionary developmental system. Using this model we describe an adaptive representation which exhibits the emergence of the hierarchical assembly of *stable* modules in the presence of developmental noise which otherwise severely retards evolutionary progress.

Modularity and Hierarchy in Representation

The importance of representational modularity lies in its ability to couple functionally related portions of the genotype while simultaneously decoupling them from functionally unrelated portions. Changes to a representational module have few side effects in the remaining genome, and changes outside a module have few effects upon the module. Wagner and Altenberg (1996) persuasively argue that the evolvability of a system is highly contingent upon its representation's ability to adapt by discovering and incorporating evolutionary modules.

Several models of representational modularity in evolutionary computation exist. Many systems, such as Hornby's L-systems (Hornby, 2005) and Bongard's Gene Regulatory Networks (Bongard, 2002) feature representations which are implicitly modular.

Of those which provide for the *explicit* encapsulation of modular components, the most common fall under the rubric of Hierarchical Genetic Programming (HGP), where encapsulated modules become new primitives in the language. Koza (1992) developed Automatically Defined Functions (ADFs), in which sub-functions are allowed to evolve their own function and terminal sets. Angeline expanded upon ADFs with *module acquisition* (MA), which co-evolve a representational genetic "library" of encapsulated primitives which are universally available to evolving programs(Angeline and Pollack, 1994). Subsequently, Rosca and Ballard introduced Adaptive Representation through Learning (ARL), which replaced the randomness inherent in modular acquisition in ADFs and MA with a "usefulness" heuristic based upon fitness contribution and activation frequency of subtrees (Rosca and Ballard, 1996).

More recently, de Jong co-evolved a representation and its corresponding population of genotypes (2003). Candidate modules were chosen by finding the most frequent pair of alleles in the current population and, drawing from Watson's work on symbiotic composition (2002), were added to the language as primitives only if they their fitness contribution was at least as good as the fitness contribution of all other possible pairs in a randomly chosen context.

While they vary in their details, each of these models of modular encapsulation involve incorporating *genotypic* sequences, thereby protecting them from the deleterious effects of mutation and crossover, and then adding them to the language of representation. As such, encapsulated modules are simply shorthand for the genetic sequence they represent - one can be substituted for the other without consequence. Below, we will motivate a system of encapsulation which, by contrast, involves incorporating *phenotypic* results into the language or representation.

Why Endosymbiosis?

Because of their prescriptive nature, developmental representations display a measure of context dependency: the same sequence of operations can have vastly different results depending on where in the process it occurs. Consider a developmental representation as a *recipe*. The set of instructions which produce egg whites in a souffle recipe (separate yolks and whites, place whites in a bowl, whip into soft peaks) would produce a mess (if anything at all) if they occurred later in the recipe or, for that matter, in an omelet recipe.

To make matters worse, in developmental systems such as the one we describe below, a contiguous portion of the phenotype which we might want to modularize may have been produced by disjoint portions of the genotype. Similarly, a contiguous portion of the genotype may produce disjoint phenotypic results.

These factors can therefore stymie adaptive models such as HGP, which generate modules by extracting and compressing favorable genetic sequences. A genetic sequence which produces a favorable trait in one context will not necessarily preserve that result when transferred to another context. Furthermore, favorable phenotypic traits may not be attributable to modularizable portions of the genotype, and modularizable portions of the genotype may not produce useful phenotypic modules.

The challenge of modular acquisition in developmental representations, then, lies in preserving not the syntax, but rather the *meaning* of a desired phenotypic result. *Endosymbiosis*, the encapsulation of an entire organism by its host, is the model which we propose for this.

In our model of endosymbiotic encapsulation (See Figure 1), complete organisms, not just specific portions of their phenotype, are used to form modules. As such, endosymbionts become *precompiled* phenotypes, and join the set of primitives available to the representation. To continue the metaphor of the recipe, endosymbiosis is analogous to the parallel creation of a *sous chef*, who specializes in producing that one particular higher-order ingredient, such as stiffened egg whites.

When referenced during the course of development, it is the phenotypic result – the complete endosymbiont – rather than the genetic sequence responsible for creating the endosymbiont, that is used by the developmental process. In this manner, the meaning, rather than merely the syntax, of a module is preserved, and can be applied consistently across contexts.

The Role of Noise in Development

Stochastic effects which lead to error and noise during development can significantly complicate the task of evolution. When subjected to noise during development, a genotype is capable of developing into an entire range of phenotypes, each with a corresponding fitness (Figure 2) Determining which, if any, is the phenotype most representative of the originating genotype is a difficult, and in some cases, entirely misleading task (Viswanathan and Pollack, 2005a).

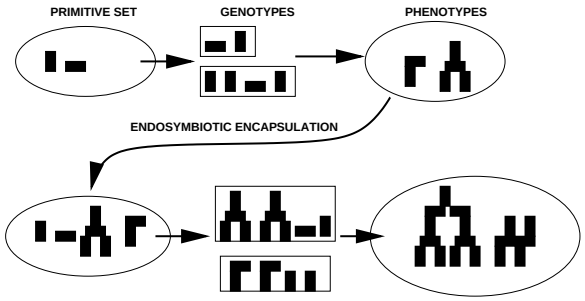

Figure 1: In the endosymbiotic model of module acquisition, only complete phenotypes, rather than genetic samples, are added to the set of primitives.

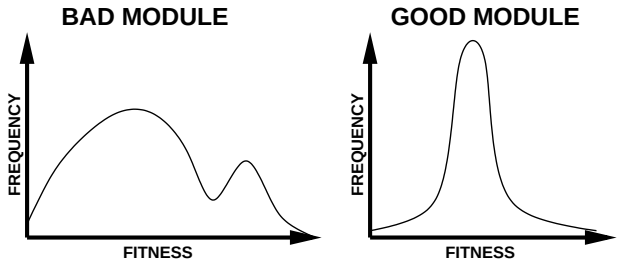

Figure 2: Under the presence of developmental noise, each genotype develops into an entire **range** of phenotypes, each with an associated fitness. A bad module is one which exhibits high variance or multimodality across multiple builds; a good module will be unimodal with very low variance.

Although the effect of noise and error during development has been extensively studied in biological systems, the matter has only lately begun to attract attention in developmentally-inspired artificial systems. Yilmaz and Wu, for instance, recently explored the relation between genetic redundancy and developmental noise (Yilmaz and Wu, 2005).

Viswanathan (Viswanathan and Pollack, 2005b) has studied the impact of stochastic development on assembly, and has demonstrated the ability of adaptive processes which measure the state and progress of the system to achieve higher reliability than purely ballistic processes.

In earlier work of ours (Rieffel and Pollack, 2004), we used a two-dimensional development environment, with a"tetris"-like physics to evolve assembly plans capable of building a predefined goal arch. Evolved assembly plans were able to reliably build the goal arch despite a stochastic noise model, which could knock over structural elements. The key to the assembly plans' robustness was the use of *ontogenic scaffolding* - the placement of intermediate structural elements that were removed once the structure was completed.

Modularity in Noisy Development

If stochastic development imposes one-to-many relation on the genotype-phenotype map, such that each genotype can grow into an entire distribution of phenotypes, then another criterion for modular acquisition arises: that of developmental *stability*.

An important aspect of the utility of modules is their reusability. If, therefore, a module is to be generated and reused multiple times, then it stands to reason that multiple copies of the module should exhibit low variance. Consider Figure 2. A genotype which, under noise, develops into a wide or multi-modal distribution of phenotypes, such as the one of the left hand side of the figure, is not ontogenically stable, and so would not make a very reliable module. A more ontogenically stable genotype, on the other hand, typically develops into a tight, unimodal distribution with low variance, such as the one on the right hand side of the figure, meaning that it will generate near-identical copies, and so makes for a more suitable module.

Experiments

The first purpose of these experiments is to demonstrate the deleterious effects of noisy ontogeny on the evolution of developmental representations. The second is to demonstrate that an adaptive representation with endosymbiotic modular acquisition is capable of overcoming these effects.

Our Artificial Ontogeny takes the form of *Evolutionary Fabrication*: the prescriptive genotype is a linear set of instructions to an external assembly mechanism. In this case the mechanism is a LOGO-like turtle, capable of movement in the X-Z plane, and of depositing objects in the environment. When strung into a sequence, commands to the turtle (move, rotate, put object, take object) form an *assembly plan*. Commands which would cause the turtle to move outside the target area, or place a brick where a brick already exists, are ignored.

The "put" command takes as an argument a unique identifier corresponding to an object in the library of encapsulated modules. Initially, the only objects available to the "put" command are primitive $2 \times 1 \times 1$ bricks. As new modules are encapsulated, they are inserted into the library as new objects and can be referenced by the "put" command (for instance put(brick) or put(module15).) As the modular library grows, the mutation operator selects from any of the modules currently available.

Assembly occurs within a realistic physics environment based upon the Open Dynamics Engine (ODE) [1] the widely used open-source physics engine, which provides high-performance simulations of 3D rigid body dynamics.

Assembly falls into three stages. In the first, the turtle interprets the assembly plan, moving and placing bricks as directed. In this stage, each brick is a separate component in

[1] `www.ode.org`

Figure 3: Illustration of the Fitness Function. The structure itself is black, and the gray region beneath is considered "shaded".

the environment, subject to gravity and interactions (such as collisions) with other objects.

It is this aspect, of construction being situated in a realistic physics environment, that gives the assembly plans particularly high *context dependency*: where a brick is placed and where it ultimately lands is contingent upon the exact turtle location, as well as the presence of supporting and surrounding objects.

Once assembly is complete and the structure is stable, the scaffolding is removed and adjacent bricks are glued together (but not to the floor). Finally, once the scaffolding is gone, the final structure is allowed to come to a rest before being evaluated.

Noise Model

Noise is injected into the system using a "shaky hand" model. When instructed to place object, the turtle puts anywhere within some range $\triangle x$ around its current position, with uniform probability. Noise settings were given as percentages of a brick's width.

Evaluation

Our design task was to create a structure which maximizes the total open volume beneath it, thereby rewarding structures which both maximize height and maximize the number of empty spaces beneath them (Figure 3).

We used a simple Evolutionary Multi Objective Optimization (EMOO) Algorithm over a set of objectives described below.

- Length Of Assembly Plan (minimizing)
- Mass of Objects in Environment (minimizing)
- Shaded Area or Sum of Heights (maximizing)

To measure the effect of developmental noise on evaluated genotypes, each assembly plan was interpreted 10 times, and average values over each objective were used for selection.

Module Selection

Candidate endosymbiotic modules were selected from the phenotypes of the evolving population every 10 generations. The criteria for selection are as follows:

- Fitness - must be a member of the current pareto front.
- Single Piece - must only consist of a single piece. Obviously if an assembly plan results in a structure with two distinct pieces, then it cannot be usefully encapsulated as a context-independent building block.
- Reliability - must have sufficiently low variance in fitness (see Figure 2.)

If a phenotype matches the criteria then it is added, as a whole, to the set of objects available to the "put" instruction.

Module Rejection

The *evolutionary viability* of the modules is determined by their reference count in the population of evolving assembly plans. Whenever a module's reference count in the evolving population drops to zero it is deemed irrelevant, and removed from the object library.

Algorithmic Details

Initial population size was 30 individuals, each with a randomized length of between 8 and 40 instructions. After each generation was evaluated, the N non-dominated individuals (i.e., pareto front) were selected as parents, and N new individuals generated using two-point crossover (60%) and mutation (2% per locus). In order to limit population sizes, duplicate genotypes were rejected, and duplicate objective values were limited using crowding with a limit of 3 individuals per bin.

Results

Three sets of experiments were run for 1000 generations each, with noise set to 0.1%, 1.0% and then 2.5% of a brick width, For comparison, parallel setups without modular acquisition were run at noise levels of 0, 0.1 and 1.0%.

Figure 4 demonstrates the deleterious effects of noise on non-modular development. In the absence of noise, evolution proceeds fairly well. But, even with relatively modest noise, at 0.1%, average fitnesses are half of what they are in the noiseless case. As noise is increased to 1.0%, performance drops even further.

Figure 5 shows a comparison in performance for modular assembly across a range of noise values. Not surprisingly, the modular setups reach near-optimal fitness rather quickly, and outperform the non-modular ones in Figure 4. Furthermore, there is very little difference in performance for modular assembly across the range of noises. Interestingly, the modular noisy evolution shown in Figure 5, across the entire range of noise values even outperforms non-noisy non-modular evolution in Figure 4. We discuss this further below.

Tables 1 and 2 contain representative assembly graphs for some of the evolved objects. These graphs were created by observing each evolved assembly process, and capturing frames of each module which was used in a subsequent structure. The graphs provide some insight into the

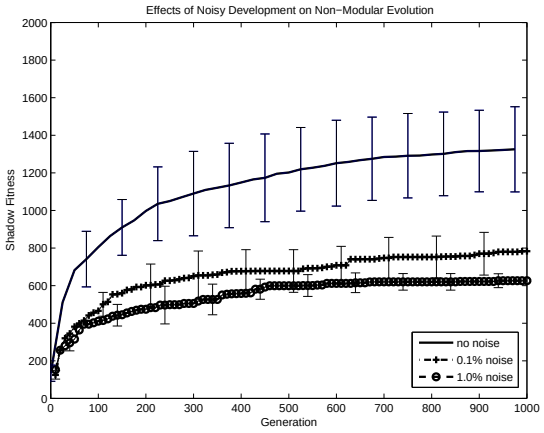

Figure 4: A demonstration of the effects of noisy development. Without noise, non-modular evolution proceeds well. Even with relatively low noise (0.1% of brick width), however, runs do significantly worse. As noise increases, performance decreases. Averaged across 22 runs (noiseless) and 10 runs (noisy), with error bars.

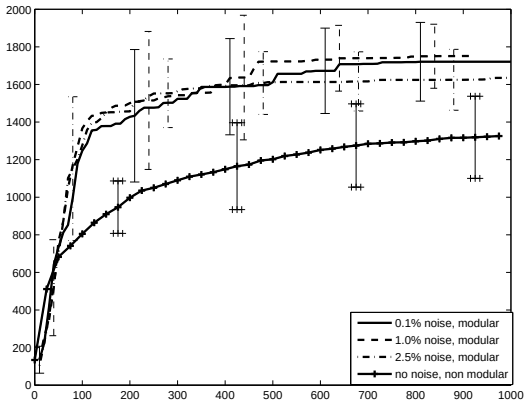

Figure 5: Across a range of noise levels, not only does modular evolution outperform non-modular noisy evolution, but it also outperforms non-modular, non-noise evolution. Averaged across 10 runs of each setup, with error bars.

processes by which evolved assembly plans were able to hierarchically assemble the objects. Full color animations of the assembly processes can be found at the author's web page[2].

These graphs only show modules which contribute structurally to a higher-order module, but not those which are used by the assembly process and subsequently removed, nor those which are referenced by assembly plans but which would cause a collision with an existing object and are therefore ignored by the assembly process.

As a consequence the graphs shown potentially underestimate the number of modules used in each assembly. In particular, it is likely that some modules serve a valuable *temporary* function, much like the ontogenic scaffolding we discovered in (Rieffel and Pollack, 2004). We are currently implementing methods of automatically parsing assembly trees to generate these graphs.

Discussion

It is interesting to observe that modular assembly outperforms non-modular assembly, even in the absence of noise. Consider that non-modular assembly must place structures brick by brick, and so is in general limited to incremental improvements in fitness over the course of evolution. The strength of modular assembly, on the other hand, lies in its ability to add larger sub-assemblies to its vocabulary, and then place them as a single unit, thus enabling faster progress.

This aspect affects not only the speed of evolution, but also the type of structure that is evolved. As we first noted in (Rieffel and Pollack, 2005), non-modular assembly plans in a non-stochastic environment tend to generate arches, even though tree-shapes are a more optimal solution. Our conjecture at the time was that this was due to the difficulty in building balanced trees brick by brick: both matching branches of the tree must be discovered in parallel, and most mutations to a balanced tree would unbalance it. Arches, on the other hand, are more evolutionarily stable, because they are supported on two legs, and can be discovered by a process which first creates a filled arch and then slowly learns to empty out the middle portion.

It is interesting to observe, therefore, that the majority of structures evolved in this noisy environment with modular assembly plans are trees. This is because the adaptive representation is able to generate larger, multi-brick modules, and then place them as a single, balanced unit atop a column. As can be seen, in every case it is a single module which forms both branches of the tree.

Several of the assembly trees shown in the tables also exhibit what might be considered a form of *exaptation* in the evolutionary process, of which the tree in the second row of Table 1 is a particularly recursive example. Because modules are selected for, among other things, their presence on the pareto front, they often have a measurable fitness when encapsulated as modules. When they are used in hierarchical assembly, however, instead of being placed in a manner which takes advantage of this inherent fitness (for instance, by placing them in parallel to form an arch), they are rotated and placed sideways atop a newly formed trunk. As such they serve a new function - for instance as a branch instead of a trunk, and in that role they are able to contribute more fitness than they do alone. This phenomenon is worthy of further exploration.

[2] www.cs.brandeis.edu/~jrieffel/modularity/

Conclusion

We have introduced a form of modular encapsulation based upon phenotypic symbiogenesis which addresses the context dependency of developmental representations. Structures which exhibit *high reliability* in the face of noisy development are chosen as candidate modules, and added as a whole, to the set of primitives available to the assembly process. Not only does this method of modular assembly overcome the deleterious effects of noisy development in an artificial ontogeny across a range of noise levels, but even at the highest noise level tested it also outperforms non-modular methods evolved without noise. The strength of this method lies in a form of *developmental bootstrapping* - small subcomponents are composed into larger *stable* modules available to the representation, and in that manner multi-tier hierarchically composed assembly methods emerge. As the size of the sub-assemblies increases, evolution is able to make incrementally larger structures with relatively fewer instructions.

While the structures evolved in this study were constrained to a relatively small bounding box, this method opens the door to the increasing scales and complexity provided by more open ended problems. We are particularly interested in the effects of noise level on the topology of evolved hierarchies of assembly. As Simon's parable would lead us to believe, increasing noise reduces the size of intermediate stable forms, which means that tree spans would decrease and tree depths would increase.

References

Angeline, P. J. and Pollack, J. B.: 1994, Coevolving high-level representations, in C. G. Langton (ed.), *Artificial Life III*, Vol. XVII, pp 55–71, Addison-Wesley, Reading, MA

Bongard, J. C.: 2002, Evolving modular genetic regulatory networks, in *Proceedings of the 2002 IEEE Conference on Evolutionary Computation (CEC2002)*, pp 1872–1877, IEEE Press, Piscataway, NJ

de Jong, E. D.: 2003, Representation development from pareto-coevolution., in *Proceedings of the 2003 Genetic and Evolutionary Computation Conference (GECCO-2003)*, pp 265–276, Springer-Verlag, Heidelberg

Hornby, G. S.: 2005, Measuring, enabling and comparing modularity, regularity and hierarchy in evolutionary design, in *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation (GECCO-2005)*, pp 1729–1736, ACM Press, New York, NY, USA

Koza, J. R.: 1992, *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, MIT Press: Cambridge, MA

Rieffel, J. and Pollack, J.: 2004, The Emergence of Ontogenic Scaffolding in a Stochastic Development Environment, in K. D. et al. (ed.), *Proceedings of the 2004 Genetic and Evolutionary Computation Con-*

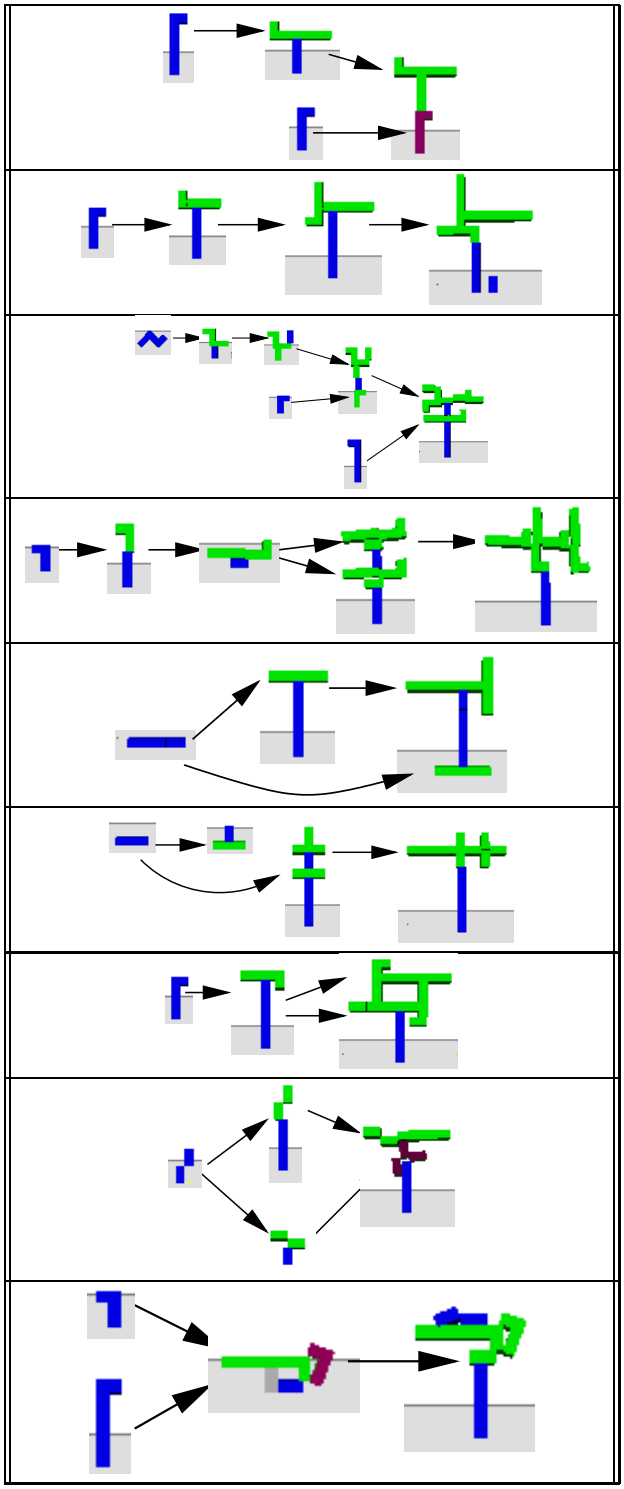

Table 1: A graph based visualization of the hierarchical assembly of modules at 0.1% noise. Each node is a module discovered by the endosymbiotic algorithm, with arrows indicating incorporation into a higher order module. The graphs only show modules which contribute structurally to a higher order module, and so may under-represent the total number of modules referenced by the assembly plan. The final structure is the rightmost.

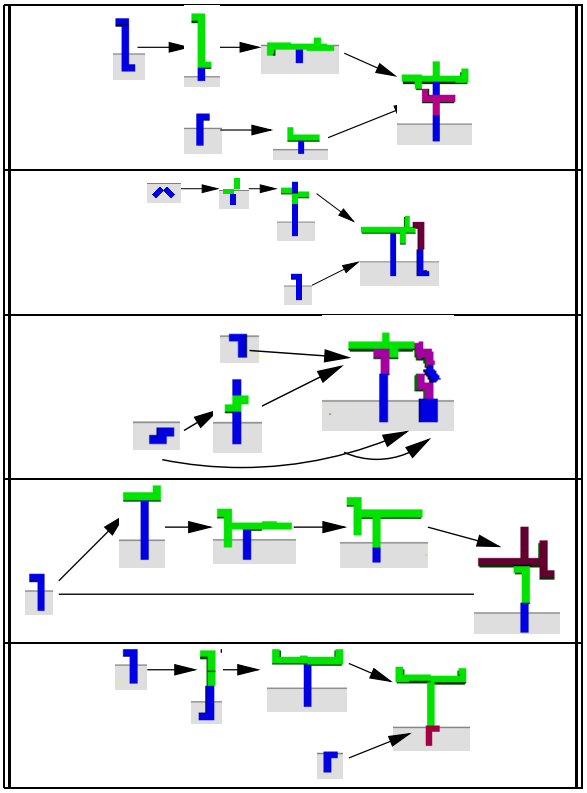

Table 2: Hierarchical Assembly of Modules at 1.0% noise

ference (GECCO-2004), pp 804–815, Springer-Verlag, Lecture Notes in Computer Science Vol. 3102, Seattle, Washington, USA

Rieffel, J. and Pollack, J. B.: 2005, Automated assembly as situated development: Using artificial ontogenies to evolve buildable 3-d objects, in *Proceedings of the 2005 Genetic and Evolutionary Computation Conference (GECCO-2005)*, pp 99–106, ACM Press, New York

Rosca, J. P. and Ballard, D. H.: 1996, Discovery of subroutines in genetic programming, in P. J. Angeline and K. E. Kinnear, Jr. (eds.), *Advances in Genetic Programming 2*, pp 177–202, MIT Press, Cambridge, MA, USA

Simon, H. A.: 1962, The architecture of complexity, *Proceedings of the American Philosophical Society* 106(6), 467–482

Viswanathan, S. and Pollack, J.: 2005a, How artificial ontogenies can retard evolution, in *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation (GECCO-2005)*, pp 273–280, ACM Press

Viswanathan, S. and Pollack, J.: 2005b, On the robustness achievable with stochastic development processes, in J. Lohn, D. Gwaltney, G. Hornby, R. Zebulum, D. Keymeulen, and A. Stoica (eds.), *Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware*, pp 34–39, IEEE Press, Los Alamitos, CA

Wagner, G. P. and Altenberg, L.: 1996, Complex adaptations and the evolution of evolvability, *Evolution* pp 967–976

Watson, R.: 2002, *Compositional Evolution: Interdisciplinary Investigations in Evolvability, Modularity, and Symbiosis.*, Ph.D. thesis, Brandeis University, Dept. of Computer Science, Boston, MA, USA

Yilmaz, A. S. and Wu, A. S.: 2005, Preservation of genetic redundancy in the existence of developmental error and fitness assignment error., in *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation (GECCO-2005)*, pp 1317–1324, ACM Press, New York, NY