Modular Interdependency in Complex Dynamical Systems

Richard A. Watson*
Computer Science
University of Southampton
Highfield, Southampton, SO17 1BJ
UK
raw@ecs.soton.ac.uk

Jordan B. Pollack
Computer Science
Brandeis University
Waltham, MA 02254
pollack@cs.brandeis.edu

Abstract Herbert A. Simon's characterization of modularity in dynamical systems describes subsystems as having dynamics that are approximately independent of those of other subsystems (in the short term). This fits with the general intuition that modules must, by definition, be approximately independent. In the evolution of complex systems, such modularity may enable subsystems to be modified and adapted independently of other subsystems, whereas in a nonmodular system, modifications to one part of the system may result in deleterious side effects elsewhere in the system. But this notion of modularity and its effect on evolvability is not well quantified and is rather simplistic. In particular, modularity need not imply that intermodule dependences are weak or unimportant. In dynamical systems this is acknowledged by Simon's suggestion that, in the long term, the dynamical behaviors of subsystems do interact with one another, albeit in an "aggregate" manner—but this kind of intermodule interaction is omitted in models of modularity for evolvability. In this brief discussion we seek to unify notions of modularity in dynamical systems with notions of how modularity affects evolvability. This leads to a quantifiable measure of modularity and a different understanding of its effect on evolvability.

Keywords
Dynamical systems, modularity, nearly decomposable systems, evolvability

1 Evolvability and Modularity in Dynamical Systems

Modularity is a familiar characteristic of a large class of natural systems. It is commonly interpreted as the condition that interactions or dependences between subsystems are weak or sparse relative to interactions within subsystems. Although modules in natural systems are, of course, not completely independent, there is a general intuition that modules must be, in some sense, approximately independent *by definition*. But this concept of modularity is easily oversimplified. In particular, intermodule interactions are often assumed to be unimportant and are sometimes omitted altogether. Such a simplification does not match common observations.

For example, if a cell is a meaningful module for a multicellular organism, it may well be the case that the cell encapsulates a complex network of intracellular interactions, but it is obviously not the case that the proper functioning of the cell is independent of the context provided by other cells. Similarly, a gene may be a meaningful module of the genome in that the detail of the nucleotide sequence within the gene can be abstracted into a relatively low-dimensional set of features such as the three-dimensional shape of the protein that the gene codes for, but it is obviously not the case that the functional behavior of the gene is independent of other genes. This coincides with our

* Corresponding author.

understanding of engineered modularity also—the engine of a vehicle is a subsystem with significant internal complexity, but the remaining interface with the rest of the vehicle is still integral to its proper functioning and operation. In short, modularity does not logically require that intermodule dependences be weak or unimportant.

This oversimplification may arise from a tendency to conflate structural modularity with the functional behavior of a system. For example, it is often useful to describe a complex system using a graph that represents the interactions between components, and modularity is taken to mean that intermodule edges are sparser than intramodule edges (e.g., [9, 4]). However, this interpretation of modularity is merely a structural description of the graph of interactions, and for complex dynamical systems a structural description of interconnectedness is not sufficient to determine the behavioral independence of one module with respect to another. In general, the dynamical dependence or independence of a module with respect to another module is related to how changes in the state of one module affect changes in the state of the other. The structural interconnectedness of modules tells us something about the likelihood or strength of immediate effects between one module and another, but it is not necessarily indicative of the extent of consequent state changes over time. In principle, one module may be strongly and nonlinearly sensitive to small state changes in another module despite being sparsely connected. Other important dynamical properties such as the number, location, and stability of attractors may also be affected despite sparse connections. The exact consequence of interactions between modules is dependent on the exact nature of the systems involved—but in general, it is not correct to assume that sparsely connected dynamical systems have only small effects on one another's dynamical properties.

A more general concept of modularity derives from the property of encapsulating internal complexity: This encapsulation may leave a relatively low-dimensional interface with the remainder of the system, but this does not imply that resolving dependencies through this interface with other parts of the system is unimportant or insignificant. Moreover, this conception of modularity is in fact necessary for making sense of *hierarchical* modularity. The extent to which the grouping of dependencies can be important is limited by the extent to which all dependencies are assumed to be important: In the limit, if we assume that modules at one hierarchical level are independent, then modules at the next hierarchical level become meaningless.

In the evolution of complex systems, modularity is often believed to increase *evolvability*—the ability of an evolutionary process to optimize or adapt a system. For example, modularity may enable subsystems to be modified and adapted independently of other subsystems, whereas in a nonmodular system, modifications to one part of the system may result in side effects in other parts of the system as dependencies among different parts of the system are broken or become maladapted. But with respect to evolvability, it seems difficult to retain the idea that intermodule dependencies may be important. If they have a significant effect on fitness, then broken dependencies with other subsystems may prevent the independent adaptation of subsystems, making the system effectively nonmodular. From this point of view, it seems that in the context of evolvability, significant intermodule dependencies are directly opposed to modularity—as though there is a direct tradeoff between how important intermodule dependencies are and how modular the system is. For example, Kauffman [5] provides a means to describe arbitrary systems of interdependent variables, and the *NKC landscape* describes a system where subsets of variables have more intraconnections than interconnections. But here the addition of significant intermodule dependencies is directly opposed to the definition of the modules themselves. In models of how modularity affects evolvability, can the notion of modularity as the encapsulation of internal complexity be retained without introducing the assumption of unimportant intermodule interactions?

To address this question we draw together notions of modularity in evolvability with notions of modularity in dynamical systems. Herb Simon [10] discusses a broad-ranging set of systems—from business organizations to biological systems—that exhibit what he calls the property of being *nearly decomposable*. He describes the characteristics of nearly decomposable systems dynamically: "the short-run behaviour of each of the component subsystems is approximately independent of the short-run behaviour of the other components," and "in the long run the behaviour of any one of the components depends in only an aggregate way on the behaviour of the other components.'' The second part of this description explicitly acknowledges the presence of intermodule interactions, without implying that they are insignificant or unimportant, by using the idea that the details of subsystem dynamics are aggregated in some way. Simon's book spends some considerable effort on discussing the implications of modularity for the evolvability of systems, but ultimately his examples of how modularity affects evolvability do not fit properly with this description of modularity in dynamical systems, as we will discuss.

In the remainder of this article we first detail some of Simon's examples to introduce some of the useful concepts and conceptual difficulties that they illustrate. We then describe a simple dynamical system that exhibits the properties of a nearly decomposable system. Following common intuitions, in its structural description this system has strong/numerous intramodule dependencies and relatively sparse/weak intermodule ones. However, we see that this system also has significant intermodule dependencies with respect to simple dynamical properties. For example, examining the dependence and independence of dynamic stability of subsystem configurations, we see that the most stable subsystem configuration is strongly dependent on the state of the remainder of the system. Nonetheless, there is still a meaningful sense in which this property is semi-independent of the rest of the system. We provide a quantifiable measure of this form of *modular interdependency.*

By relating fitness to a modular property of the system (such as system stability in this example), we produce a model that illustrates how modularity of this form can influence evolvability. We briefly discuss some prior results [11, 14], using a fitness landscape based on this construction [12]. These show that this form of modularity has a different effect on the evolvability of systems than the naive form of modularity. In particular, they show that it is difficult to resolve intermodule dependencies in systems with modular interdependency using linear incremental improvement, but evolutionary mechanisms such as sexual recombination, which can exchange adapted subsystems among individuals, can evolve this sort of system easily under certain circumstances.

2 Simon's Example Systems

In this section we examine the concept of nearly decomposable systems introduced by Simon and discuss some of his well-known examples in some detail. To understand how modularity affects evolvability (for our purposes, how likely it is that an evolutionary process can make a system well adapted), we assess the combinatorics of the evolutionary search process. That is, we assess the number of configurations that an evolutionary search process is likely to sample before finding high-fitness configurations of the system: if this is low, then the system is more evolvable than if it is high. Simon provides some examples that discuss the effect of modularity on this kind of combinatorics, and some examples that discuss modularity in dynamical systems. However, no one example describes both aspects adequately, illustrating the conceptual difficulties in bringing these ideas together. However, the three well-known examples detailed below do provide important conceptual components for the model that follows.

Example 1 (Combination lock). In discussing the expected time for a complex system to evolve, Simon uses an analogy based on finding the correct combination for a combination lock on a security safe. Specifically, suppose a lock has 10 dials, each with 100 positions—by blind trial and error we would expect to need on average half of 100^{10} guesses to open the safe. Simon contrasts this with a *defective* lock where a click can be heard when the dial is at the correct setting. In this case, we would expect to require on average about half of 100×10 guesses to open the lock, because opening the lock is merely 10 independent repetitions of finding the correct setting for each dial. This example is easy to interpret as two evolutionary scenarios differing only in the structure of the fitness contributions: In the first case the fitness landscape would give no feedback to evolutionary search until the setting for *all* dials was correct, whereas the defective lock would give a fitness increment for each dial that was correctly set. Note that in the defective lock the solution to each module is entirely independent of other modules; the correct setting for a dial, and the feedback for setting it correctly, are not in any way sensitive to the context of the other dials.

As suggested above, an important consequence of assuming independent modules is that modules of this kind cannot form a meaningful hierarchy. For example, let us take a look at what a hierarchical lock might be. Let us suppose that a *hierarchical superlock* has 10 groups of 10 dials each, and in the defective version of this lock, a group of dials makes a big click when all 10 dials it contains are set correctly. If the individual dials are defective as before, what is the expected number of guesses to open the lock? We can see that it is merely half of 100 × 10 × 10. Note that this is exactly the same as would be expected for a defective *flat superlock* that has 100 defective dials. In other words, when each dial is independent, subsequent grouping of dials is redundant and does not change the combinatorics involved. As an example of a completely decomposable system, the defective-lock scenario is adequate—it makes clear quite simply the intuition about the effect of modularity on combinatorial search. But from Simon's descriptions, it is not at all clear how to modify the lock analogy to add any intermodule dependencies and make the dials nearly decomposable.

Example 2 (Watchmakers). Next we look at Simon's famed watchmaker parable. The first watchmaker assembles watches made of 1000 components. The second watchmaker assembles watches made of 10 modules, each of which is composed of 10 sub-modules, each of which is composed of 10 components. The difference in the success of the two watchmakers arises in their robustness to interruptions. When either watchmaker is interrupted, the current assembly or subassembly being worked on disassembles. For the first watchmaker an interruption nearly always occurs before completing the watch, and all the work performed on an incomplete watch is wasted. The second watchmaker is often able to complete a stable subassembly before an interruption occurs (and less assembly steps are wasted per interruption), so much of the work is usefully retained. The advantage for the modular watch assembly procedure in terms of the expected time to complete a working watch is easily quantified for a given interruption probability. This, Simon claims, explains why modular complex systems, through their inherent stability, are more likely to be evolved than nonmodular ones.

This fable describes the advantages of stable intermediates in a stochastic assembly procedure, and here, unlike the lock example, there is a benefit to additional layers of hierarchical modularity. However, the assembly procedure is difficult to interpret as an evolutionary search process. In the lock example, we search a space of dial-position combinations, but the watchmaker fable has no explicit search space—neither watchmaker is required to search for configurations that are stable or fit; each watchmaker knows the correct assembly for a watch or a module without search. In the lock example it is easy to interpret the search of dial configurations as adaptive evolutionary search, where clicks in the defective lock are fitness rewards that inform evolutionary search. But in the watchmaker fable such an interpretation is problematic. Specifically, if we imagine that evolutionary adaptation is searching a larger space of possible watch-part assemblies and is rewarded by increased fitness for modules that are stable, then the utility of hierarchical modularity is lost. That is, additional fitness increments for assemblies of submodules into modules, or modules into whole watches, are redundant in the same way as additional clicks are redundant in the hierarchical superlock—guiding evolutionary search using fitness increments at the lowest level only would be sufficient. In trying to make a model of how an evolutionary process would be better able to evolve the modular watch than the nonmodular watch, it is not clear how to map interruptions into fitness contributions of system configurations without making all but the first level of modularity redundant. Despite some effort here, it is difficult to see how to convert the benefit of stable intermediates in an assembly procedure, illustrated in this parable, into an example of how modularity affects the ability of evolutionary search to find fit configurations of a system.

Example 3 (Heat Exchange). In a third example, Simon describes modularity in a dynamical system. This is a system of heat exchange in a building that is subdivided into rooms and cubicles. Heat exchange between cubicles within a room is more rapid than heat exchange between cubicles in different rooms. This system exhibits the properties that Simon lists for nearly decomposable systems: the short-run dynamics of heat exchange in a room is approximately independent of the heat exchange dynamics in other rooms, and the long-run dynamics of heat exchange in a room depends (approximately) only on the mean temperature of cubicles in other rooms. Note that the ultimate temperature of a room depends on the temperature of other rooms, despite the fact that the short-term dynamics of temperature among cubicles within a room are largely unaffected by the temperatures of cubicles in other rooms.

However, Simon does not describe how this kind of modularity affects the evolvability of a system. How is the modular system easier to evolve than the nonmodular one? What is the analogue for fitness, or for evolutionary search through a configuration space? For example, it does not make sense to make an analogy between evolvability and the speed at which the system reaches equilibrium, because modularity in the heat exchange parameters slows the approach to equilibrium. If this is an analogy for a system that might be evolved, we do not know how changes in state relate to combinatorics that reflect evolvability.

Summarizing Simon's three examples: The lock example provides a quantitative combinatorial argument for modularity, but the modules are entirely separable, and this kind of modularity cannot be hierarchical. The watchmaker example has meaningful hierarchy, but describes the advantage of robustness in a stochastic assembly procedure, not an advantage of modularity in a combinatorial evolutionary search process. The heat exchange example illustrates a hierarchical dynamical system with meaningful modularity in the dynamics of the system at several scales, but it is not clear how to map this example to an evolutionary scenario where we can examine its effect on evolvability.

Our purpose in discussing these example systems from Simon in detail is not to suggest that Simon is unusually misguided. On the contrary, these examples provide some very valuable concepts of modularity that we can build upon. But they are also indicative of a widespread difficulty in reconciling the basic idea that evolvability can be enhanced when subsolutions can be identified independently, with the basic idea that modularity should *not* mean that subsystems are *fully* independent. The combination-lock example is a good illustration of the former, and the heat exchange example is a good illustration of the latter. But we are missing an illustration of both together. Thus we are also missing an example of modularity in a combinatorial search model that is hierarchical. The heat exchange model is hierarchically modular (we can see that the inter-room dynamics of the system operate at a different temporal scale but are otherwise just like those of the intercubicle dynamics), but this model does not provide a combinatorial argument. The combination lock example provides a combinatorial argument, but here hierarchy is not possible (we cannot make a sensible hierarchical lock, because there are no inter-dial dependencies). Although at first glance the watchmaker parable seems to provide a model of hierarchical modularity, in fact, as noted, it does not describe a combinatorial search process.

So, working from the lock example, how can the correct position for a dial be identifiable so as to reduce the combinatorial search required to open the lock without making the dials *fully* independent? And, working from the heat exchange example, how can we interpret the intramodule and intermodule dynamics as an evolutionary search process? In the following example system we draw together the dynamical system properties that Simon describes with a combinatorial search interpretation of evolvability. This utilizes a hierarchically modular dynamical system with a mapping from the dynamical stability of state configurations to fitness contributions. Importantly, the same hierarchical modularity that Simon provides in the heat exchange example is thus manifest in the fitness interdependencies of modules and submodules in a combinatorial search problem.

3 Modular Interdependence

In this section we use a simple example dynamical system to unify ideas of modularity in dynamical systems with ideas of modularity in evolvability. We show that structural modularity in this system does not imply independence, or near-independence, of the dynamical properties of modules. Thus there is a meaningful sense in which a system may be modular in some respects and yet have strong interdependencies between modules in other respects. We define how the modularity of a system may be quantified with respect to some property of interest—for example, the stability of the dynamical system. If we map a modular property such as that to fitness, then we can equivalently quantify the modularity of the system with respect to evolvability. We use this to contrast and clarify some of the concepts introduced by Simon above.

3.1 An Example Modular Dynamical System

Consider a gene regulation network as an example dynamical system. We suppose, for simplicity, that each gene may be in one of two states: *high*, meaning highly expressed, or otherwise *low*. Let the future state of each gene be determined by some function of the states of the genes that regulate its expression. (In general, this describes a *random Boolean network* [5].) We can use a graph to represent the connectivity of regulation activity, where nodes represent genes, and edges represent regulation interactions. Here we will assume that all genes regulate every other gene (i.e., a fully connected interaction graph), but we will modify the strength of interactions between certain genes. We also assume that all nodes have a self-recurrent connection so that a node's future state is a function of the nodes it is connected to and its own state at the previous time step.

To define a modular network, we modify a (fully connected) network so that the interactions within particular subsets of genes will be stronger than the interactions between genes in different subsets. To represent this we will simply use multiple edges between genes (since we do not really wish to distinguish between numerous interactions and strong interactions). Figure 1 thus shows an example system that exhibits a clear two-module structure.

This graph describes the structural modularity of our system—to this we must add a description of its dynamics. For illustration we use a simple voting-style discrete-time update rule. Specifically, the probability of a gene taking a given state in the next time step will be equal to the proportion of regulating connections from genes that are themselves in that state at the current time step. Let "high" = 1 and "low" = 0; then

$$
\begin{aligned}
&P(A_{t+1} = 1) = \frac{1}{k_A} \sum_{i=1}^{k_A} B_{it}, \\
&P(A_{t+1} = 0) = 1 - P(A_{t+1} = 1), \qquad (1)
\end{aligned}
$$

where A_t is the state of gene A at time t, k_A is the number of edges connecting to A, and B_{it} is the state of the ith regulating gene connected to A at time t. This update rule is popular in models of

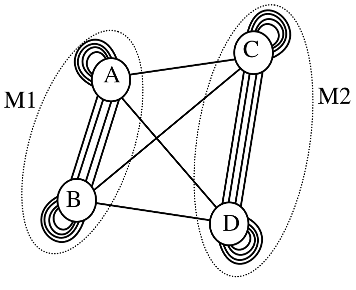

Figure 1. An example system of four genes, *A*, *B*, *C*, and *D*, arranged in two modules, M1 and M2, where intramodule (and self-recurrent) interactions are of strength 4, and intermodule interactions are of strength 1.

physical dynamical systems such as Ising models. These dynamics define a system that meets Simon's criteria for a nearly decomposable system. The short-run behavior can be understood by equating a low stability value with rapid changes; and the proportion of ones is a sufficient aggregate effect to determine the intermodule effect on stability.

The functional (rather than structural) modularity of a system may be determined by dynamical properties such as the stability and location of attractors. The long-term (rather than immediate) effects of changes in one module on state changes in another module can be characterized by changes in the attractors of a system. In general, if the attractors of a subsystem or module are unaffected by state changes in another module, then this is an important form of dynamical independence. At the opposite extreme, if we find that the attractors of a subsystem are completely different, this indicates an important form of intermodule dependency.

By way of example, let us analyze our example system by examining which configurations of M1 are most stable, and how these configurations differ with state changes in M2. For these purposes it is sufficient to use a simple measure of stability: specifically, let us define the *stability* of a network, or subnetwork, as the probability that no gene in the network changes state. (This definition of stability closely parallels the *free-energy* measure of a configuration in an Ising model.) The stability of a network configuration is therefore the product of the stabilities of all genes in the network. For example, in a system of two genes, A and B, where the probability of A remaining in its current state is $S(A)$, and the probability of B remaining in its current state is $S(B)$, the stability of the system AB is simply $S(A)S(B)$.

Clearly, since there are intermodule edges, the stability of a given configuration is sensitive to the state of genes in the other module, and the configuration that is most stable may also differ. Let us write the states of the four genes in our system as A, B, C, and D, and write the stability of a subset of variable states x, given the state of a subset of variables y, as $S(x, y)$; for example, the stability of a configuration of M1, given the state of M2, is written as $S(AB,CD)$. Given the intramodule symmetries in our example system, the effects of 01 and 10 are the same, so we only list combinations of 00, 01, and 11 for each module. Table 1 shows the stability of these configurations for our example system, using Equation 1.

We immediately notice that the configuration of M1 that is most stable is strongly sensitive to the configuration of states in M2. Specifically, the most stable configuration of M1 when CD = 00 is

Table 1. Stability of configurations.

A B C D	*S(A,BCD)*	*S(B,ACD)*	*S(AB,CD)* = *S(A,BCD)S(B,ACD)*
0 0 0 0	10/10	10/10	1
0 1 0 0	6/10	4/10	0.24
1 1 0 0	8/10	8/10	0.64
0 0 0 1	9/10	9/10	0.81
0 1 0 1	5/10	5/10	0.25
1 1 0 1	9/10	9/10	0.81
0 0 1 1	8/10	8/10	0.64
0 1 1 1	4/10	6/10	0.24
1 1 1 1	10/10	10/10	1

AB = 00, and the most stable configuration of M1 when CD = 11 is AB = 11. The systems' symmetries provide corresponding stabilities for M2.

This example system is therefore sufficient to illustrate the following: Although we have a system that is clearly modular in one respect, it also has strong intermodule dependencies in another respect: Specifically, it is *structurally* modular in the sense that it has stronger (more numerous) intramodule dependencies than intermodule dependencies, but it has strong *functional* intermodule dependencies in the sense that a simple dynamical property—the most stable configuration for a module—is strongly dependent on the state of other modules.

From this example, one might conclude that (despite the apparent structural modularity) this system is in fact not modular with respect to the property of "most stable configuration." However, note that although the configuration of a module that is most stable is strongly dependent on the state of the other module, there is nonetheless some degree of *independence* in this property also. Specifically, we notice that although the most stable configuration of states may be either 00 or 11, it is always one of these two possibilities, and never 01 or 10, regardless of the state of the other module. This means that there is something we know about the property of interest, the most stable configuration, that is *independent* of intermodule interactions.

From these observations we arrive at a means to quantify the independence of a module, given a property of interest. Specifically, given some property that identifies particular configurations, like the configuration that is most stable, we can assess what we know about the identity of such states that is independent of the states of other modules (the rest of the system). In this example, there are four possible configurations for a module: $C = 4$. Accordingly, a meaningful definition of nondecomposability, given some property that identifies particular configurations of the system, is that without knowing the state of the rest of the system we know nothing about which of these four possible configurations satisfies this property. For example, suppose that for every configuration of M1, there is some configuration of M2 (or the remainder of the system) that would make that configuration of M1 the most stable—if this were the case, then without knowing the state of M2, we would know nothing about which configuration of M1 satisfies the property of interest, and we might reasonably assert that the system is nonmodular with respect to this property. In our example system that is not the case. In fact, there are only two configurations of each module that could be maximally stable. Specifically, regardless of the state of M2, the configurations of M1 that may satisfy this property will be either 00 or 11. We will refer to the total number of configurations that satisfy the property of interest (taken over the set of all possible contexts) as C'. As stated, if $C' = C$, we will call this a nondecomposable system. But in our example, the number of different configurations of M1 that are maximally stable for some configuration of M2 is $C' = 2$.

In general, we will say that when $C' < C$ the system is *decomposable*. But note that there is a special case of decomposability where $C' = 1$, which means that the configuration of interest is always the same regardless of the state of the remainder of the system. In this case, the module is fully independent in the property of interest, a case we call *separable*. In the general case where a system is decomposable but not separable (i.e., $1 < C' < C$, as in our example), we say the system exhibits *modular interdependence* [11].

Note that in the above combination lock example, described by Simon, we have $C' = 1$, that is, the correct setting for a dial is separable from the setting of other dials. For illustration, we can describe a small modification to the lock example that introduces strong intermodule dependencies. Suppose that there are n ($1 < n < 100$) positions of each dial that produce a click, and one of these is the correct position for opening the lock. In this case, we must search combinations of the n dial positions of each dial to open the lock, but the total number of combinations is reduced from 100^{10} to n^{10}. Thus $C' = n$ may be much smaller than 100 but greater than 1 (as used in Simon's description)—so the modified system will exhibit modular interdependency. This example shows how modularity may, in principle, reduce the combinatorial search required without assuming that modules are independent.

The understanding of modularity given above clearly allows meaningful decomposability when $C' < C$, and yet still allows the possibility of important interactions between modules when $C' > 1$. Our example system shows that structural modularity defined by the strength or number of interactions in a dependency graph may indicate useful modularity without implying that modules are approximately independent or that intermodule interactions are weak or unimportant. In this example, a simple dynamical property—which configuration is most stable—is strongly dependent on the state of other modules in the system. By defining a concept of modularity that accommodates strong and significant intermodule dependencies, we both allow a more general notion of modularity, and also permit the construction of hierarchically modular systems where modules are important at all levels, as shown below. In contrast, when modules are separable, additional hierarchical levels are redundant. This approach quantifies the modularity of a dynamical system, like those described by Simon's notion of nearly decomposable systems, with respect to a given property of interest.

4 Evolvability of Systems with Modular Interdependency

Thus we see that, in principle, it is possible for a system to be modular in a quantifiable sense and yet have strong intermodule dependencies. In this section we discuss the impact of strong intermodule dependencies on the evolvability of complex systems. In order to facilitate this we must describe some mapping from states of the system to fitnesses. If these fitnesses exhibit modular interdependency, then this will mean that the evolution of subsystems will be partially but not fully independent of other subsystems. For example, we may suppose that an evolutionary process is adapting a complex system that is modular in the manner of our example dynamical system, and the fitness of a state configuration is proportional to the dynamical stability, or robustness to perturbation, of that configuration. Such a mapping provides an appropriate connection between notions of modularity in dynamical systems and notions of modularity for evolvability. That is, we can analyze how this kind of modularity affects the likelihood of an evolutionary search process discovering high-fitness configurations of the system.

As we mentioned earlier, it is a familiar idea that if a system can be decomposed into modules that are basically independent, then an evolutionary process can optimize such a system easily. This intuition aligns directly with the reasoning that Simon provides in the combination lock example. However, it is also common wisdom that when module interdependencies are strong, having a significant influence on fitness and requiring large changes in subsystem configurations to resolve intermodule dependencies, the system is not in fact decomposable and therefore not evolvable. Our example system shows that this view is perhaps too simplistic, because there may still be meaningful modularity in a system even though intermodule dependencies are important. The question addressed in this section is what effect modular interdependence (i.e., $1 < C' < C$) has on evolvability. In particular, can an evolutionary process exploit this kind of decomposable but not separable system, or is evolvability enhanced only when subsystems are separable?

For these studies we utilized a system with modular interdependency based on our example system described above. To describe a fitness function exhibiting modular interdependency, we may, for example, simply equate fitness with the stability of a state configuration. With $C' > 1$, additional hierarchical levels of decomposition are significant, and here we use a system with several hierarchical levels as defined below. This function, known as *hierarchical equality*, is a generalization of *hierarchical if-and-only-if* [13]. To maintain alignment with previous work, the fitness function below sums the sizes of all modules that have internally equal states (rather than taking a product of module stabilities), but the functions' modularity is structurally the same as that described in our example modular system above. Specifically, it has the same local optima, and the configuration that is most stable for a module in the earlier system is the configuration that is fittest in this function. The function $g(S_1,\ldots,S_N)$ is a recursive function that gives the fitness of a system with variables $S_1,\ldots,S_N$:

$$g(S_1,\ldots,S_N) = \begin{cases} 1 & \text{if } N = 1, \\ N\,f(S_1,\ldots,S_N) + \sum\limits_{i=1}^{k} g(S^i) & \text{otherwise,} \end{cases} \tag{2}$$

where S_i is the ith variable of the configuration; S^i is the ith disjoint subpartition of the variables [e.g., for equal-size submodules, $S^i=(S_{1+k(i-1)},\ldots,S_{ki})$]; $N = k^H$, where $H \in Z^+$ is the number of hierarchical levels in the system or subsystem; k is the number of submodules per module; and f is the fitness contribution function, defined below

$$f(p_1,\ldots,p_k) = \begin{cases} 1 & \text{if } (\forall ij\colon p_i = p_j), \\ 0 & \text{otherwise.} \end{cases}$$

where $p \in \{0, 1\}$. This function can be used to describe modular systems with any number of submodules per module, and any alphabet of symbols rather than just a binary one. However, it is sufficient to use two equal-size submodules per module, and binary states. For illustration, Table 2 lists all four-bit strings with their fitnesses.

Examination of these fitness values shows that 0011 and 1100 are local optima separated from both 1111 and 0000 (which are the global optima) by Hamming distance 2. Thus for the left subset of genes, the fittest configuration is either 00 or 11, but which of these maximizes fitness is dependent on the state of the other two variables, that is, $1 < C' < C$. As this system is scaled up through successive hierarchical levels to 8 variables, 16 variables, and so on, the number of local optima increases and the distance of each local optimum to the closest point of higher fitness also scales up. However, as with our example system, the fittest configuration is always one of a small number of possibilities ($C' = 2$), but which of these is fittest is dependent on the configuration of the rest of the system. These properties, as before, mean that the system is decomposable but the modules are not separable. This modularity can be quantified using exactly the same analysis used for the example dynamical system above, but by using the property of interest "which configuration is fittest" rather than "which configuration is stablest." Note that this analysis applies to any fitness

Table 2. Fitness of configurations.

g(0000) = 12	g(1000) = 6
g(0001) = 6	g(1001) = 4
g(0010) = 6	g(1010) = 4
g(0011) = 8	g(1011) = 6
g(0100) = 6	g(1100) = 8
g(0101) = 4	g(1101) = 6
g(0110) = 4	g(1110) = 6
g(0111) = 6	g(1111) = 12

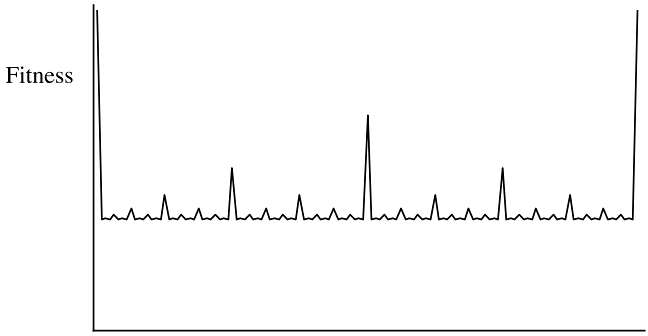

Figure 2. A cross section through the fitness landscape (defined by Equation 2) corresponding to the stability function or energy surface of the modular dynamical system (Figure 1).

function and does not specifically require that there be a direct mapping to the stability of a corresponding dynamical system. That is, if the configuration of a fittest module is dependent on context to some extent (i.e., $C' > 1$) but not entirely ($C' < C$), then the system exhibits modular interdependency with respect to fitness.

In the following sections, we first discuss the evolvability of systems with modular interdependency under normal gradualist mechanisms of evolution: that is, adaptive mechanisms that linearly accumulate random changes in genetic material. Then we discuss the evolvability of systems with modular interdependency under mechanisms we call *compositional*: that is, adaptive mechanisms that involve the exchange of preadapted sets or subsets of genetic material between different lineages [11]. We will find that the evolvability of systems under these two classes of mechanisms is quite different.

4.1 Evolvability under Normal Gradual Evolution

We can get some understanding of the evolvability of this kind of system by examining a particular cross section through the fitness landscape shown in Figure 2. This curve runs through points from one global optimum to the other, making single point-mutation changes; see [11]. The separation of peaks on this curve accurately reflects the Hamming distance from a point of this fitness to the nearest point of higher fitness.

This curve shows that the landscape is highly rugged and has many local optima, creating broad fitness saddles. It also includes *irreducibly complex* adaptations that cannot be reached by a succession of gradually changing proto-systems [11]. All these characteristics are usually associated with evolutionary difficulty, and accordingly, it should not be surprising that simple optimization processes are unable to find high-fitness points in this landscape. In fact, no mutation-based methods can be guaranteed to succeed in time less than exponential in N (the number of genes), regardless of the mutation rate used. So neither a mutation hill climber, an evolutionary algorithm using only mutation, nor simulated annealing can resolve the interdependencies between modules. In short, accretive variation mechanisms that cannot manipulate modules as wholes cannot resolve the interdependencies between modules to escape from suboptimal configurations of the system. An evolutionary process using sexual recombination and fitness proportionate selection is also unable to resolve intermodule dependencies, because the population quickly converges around the best configuration found thus far, and this prevents useful variation from crossover.

In general, the time to resolve intermodule dependences is exponential in the size of the system, N, for gradual mechanisms [11]. This seems to confirm the commonplace intuition that if modules have strong interdependences, then they are not evolvable and may as well be understood as being nondecomposable. However, in the next subsection we argue otherwise.

4.2 Evolvability under Compositional Evolution

In recent work [11], we introduced the term *compositional evolution* to describe evolutionary processes using mechanisms that combine systems or subsystems of genetic material that have been semi-independently preadapted in parallel in different lineages. Examples in nature include:

- normal mechanisms of sexual recombination (under particular conditions of population diversity and genetic linkage);
- mechanisms of interspecific genetic integration such as horizontal gene transfer, [3], or "symbiotic encapsulation," including endosymbiosis [6] and other mechanisms that encapsulate a group of simple entities into a complex entity at a higher level of organization, as exhibited in several of the major transitions in evolution [7, 8].

Compositional evolution stands in contrast to the normal gradualist framework of evolutionary processes involving the linear accumulation of random variations—that is, where the new genetic material introduced by variation has not been preadapted elsewhere as a set. Assuming that large random variations are less likely to be fitness-positive than small random variations, this *accretive* view of evolution supports the familiar assumption of gradual evolutionary change, that is, "successive slight modifications" [1].

From an adaptationist perspective, the important characteristic of compositional mechanisms is that they allow for complex entities to be assembled from a number of simpler entities evolved in parallel. For example, the organelles of the eukaryote cell (and accordingly all plants and animals) originated from the union of more than one prokaryote cell [6]. The variation space of possible assemblies of extant entities is clearly different from that of random modifications in genetic material, whether small or large. This compositional variation can provide better evolvability than linear accumulation of random variation when preadapted genetic material relocated from one lineage to another has a better chance of producing a fitness-positive change than does random genetic material. Several factors influence the likelihood of this: not least, the availability of a variation mechanism that manipulates appropriate (non-arbitrary) subsets of genetic material—for sexual recombination this places requirements on the ordering of genes on the chromosome [11].

Another requirement, of interest to us here, is the contextual (in)dependence of the fitness benefit of a set or subset of genetic material—in other words, the modularity of the genetic interactions. This latter condition is met by systems that are strongly decomposable even if they are not separable, and accordingly, there are conditions under which compositional mechanisms can resolve the interdependences between modules [11]. In compositional processes, different lineages can maintain different high-fitness configurations of a module, and then, by exchanging subsets of genetic material between lineages, search combinations of different module configurations to resolve interdependencies between them. This process is able to exploit the combinatorics that are described for the modified defective-lock example; and accordingly, the expected time for sexual recombination, or for an abstract symbiogenesis model, to resolve module interdependencies is, under certain assumptions, polynomial in the size of the system [11]. This kind of combinatorial advantage provided by assembling modules over successive hierarchical levels is developed and formalized in [2].

5 Conclusions

In this article we have discussed different types of modularity and different examples of modular systems. We showed that structural modularity does not necessarily imply independence or near-independence of subsystems with respect to simple dynamical properties of the system. This allows important functional dependencies to exist between modules as per our observations of common modular systems. We used Simon's examples to show that it is easy to assume that modular systems have no important dependencies between modules, but we are also able to show that this need not be the case. We also argued that modular systems without significant module interactions result in degenerate forms of hierarchies where successive levels of modularity have little or no effect. However, modular systems that are decomposable but not separable, which we call systems with modular interdependency, can form hierarchical systems where all levels of organization are significant. Our example system relates clearly the concept of hierarchical modularity in dynamical systems (like that described by Simon) to quantifiable measures of decomposability in corresponding combinatorial optimization problems—a relationship that is absent in Simon's hierarchical examples.

In the later sections of this article we discussed the evolvability of systems with significant intermodule dependencies. As might be expected, such systems can be problematic for evolution. However, these systems are evolvable under certain evolutionary scenarios, which we call compositional evolution. These observations and results cause us to think differently about the adaptive potential of compositional evolutionary mechanisms—especially if biological systems are hierarchically modular [9]. More generally, these observations assist us in clarifying concepts of modularity and accompanying assumptions, especially with respect to hierarchical dynamical systems and their evolvability.

Acknowledgments

Thanks to Edwin de Jong, Anthony Bucci, Alex Platt, Daniel Weinreich, and the reviewers of [12] for helpful discussion and comments.

References

1. Darwin, C. (1859). *On the origin of species by means of natural selection*. London: John Murray.
2. De Jong, E. D., Thierens, D., & Watson, R. A. (2004). Hierarchical genetic algorithms. *In Proceedings of the 8th International Conference on Parallel Problem Solving from Nature PPSN-04* (pp. 232–241).
3. Doolittle, W. F. (2000). Uprooting the tree of life. *Scientific American*, *282*, 90–95.
4. Hartwell, L., Hopfield, J., Leibler, S., & Murray, A. (1999). From molecular to modular cell biology. *Nature*, *402*(supp.), C47–C52.
5. Kauffman, S. A. (1993). *The origins of order*. Oxford, UK: Oxford University Press.
6. Margulis, L. (1970). *Origin of eukaryotic cells*. New Haven, CT: Yale University Press.
7. Maynard Smith, J. M., & Szathmary, E. (1995). *The major transitions in evolution*. New York: W. H. Freeman.
8. Michod, R. E. (1999). *Darwinian dynamics, evolutionary transitions in fitness and individuality*. Princeton, NJ: Princeton University Press.
9. Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., & Barabasi, A.-L. (2002). Hierarchical organization of modularity in metabolic networks. *Science*, *297*, 1551–1555.
10. Simon, H. A. (1969). *The sciences of the artificial*. Cambridge, MA: MIT Press.
11. Watson, R. A. (2002). *Compositional evolution: Interdisciplinary investigations in evolvability, modularity, and symbiosis*. PhD Dissertation, Computer Science Dept., Brandeis University, Waltham, MA.
12. Watson, R. A. (2002). Modular interdependency in complex dynamical systems. In E. Bilotta et al. (Eds.), *Workshop Proceedings of the 8th International Conference on the Simulation and Synthesis of Living Systems*. UNSW Australia.
13. Watson, R. A., Hornby, G. S., & Pollack, J. B. (1998). Modeling building block interdependency. In Eiben, Back, Schoenauer, & Schweffel (Eds.), *Proceedings of Parallel Problem Solving from Nature V* (pp. 97–106). Springer-Verlag.
14. Watson, R. A., & Pollack, J. B. (2003). A Computational model of symbiotic composition in evolutionary transitions. *BioSystems*, *69*(2–3), 187–209.