3

Systems Engineering for Organic Computing: The Challenge of Shared Design and Control between OC Systems and their Human Engineers

Kirstie L. Bellman[1], Christopher Landauer[1], and Phyllis R. Nelson[2]

[1] Aerospace Integration Science Center, The Aerospace Corporation, P.O. Box 92957, Los Angeles, California 90009-2957, USA.
bellman@aero.org, cal@aero.org

[2] Electrical and Computer Engineering Department, California State Polytechnic University Pomona, 3801 W. Temple Ave., Pomona, California, 91768, USA.
prnelson@csupomona.edu

Summary. The term "emergence" is usually used to mean something surprising (and often unpleasant) in the behavior of a complex system, without further qualification. Designers of OC systems want to manage emergence in complex engineered systems so that it can contribute to, or even perhaps enable, accomplishing the system's performance goals. That is, OC designers aim to construct systems that are more flexible and adaptable in complex environments, to gain some of the advantages in robustness and adaptability that biological systems seem to gain from these phenomena. In this chapter we suggest some principles that we believe underlie the enormous flexibility and opportunistic adaptability of biological systems. We show how these principles might map to systems engineering concepts when they do, and what to do instead when they don't. We then describe five specific challenges for the engineering of OC systems, and how we think they might be addressed. We also discuss the key role played by language and representation in this view of designing and deploying an OC system. Finally, we describe our progress and prospects in addressing these challenges, and thus in implementing systems to demonstrate the capabilities that we have identified as essential for successful OC systems.

Key words: Biologically-inspired system architectures, computational reflection, layers of symbol systems, representational mechanisms, self-modeling systems, systems engineering

3.1 Introduction

Organic Computing (OC) would like to take advantage of one of the key attributes of biological systems; they adapt and change on multiple time scales as they evolve, develop, and grow, and they do so without external direction

R.P. Würtz (ed.), *Organic Computing*. Understanding Complex Systems,
doi: 10.1007/978-3-540-77657-4_3, © Springer-Verlag Berlin Heidelberg 2008

or control. However, such self-design and self-organization is at variance with any of our current engineering methods for designing and controlling complex systems. One of the central challenges to OC systems is that not only do we want somehow to create the foundations for biological-like system properties, but also that we must do so in a manner that allows us to monitor continually, manage and even further develop such systems while they are in operation. Hence, we can learn from biological systems, but in fact OC systems face a unique challenge: OC systems must remain closely linked to us, their designers, builders, and users. This chapter addresses the challenge of how OC systems engineering can be accomplished by providing specific capabilities that enable the system and its human developers and systems engineers to jointly shape system goals and behaviors.

In this chapter, we start with an emphasis on certain characteristics of biological systems and describe how such characteristics – if they were to be imitated in engineered systems – lead to several striking new challenges for the human systems engineer. These difficult tasks include how to share control with a somewhat autonomous system and how to change the traditional role of the systems engineer from attempting to determine and build all system characteristics to a new role of carefully building in key interaction points for evaluating, shaping, guiding, deterring, or preventing certain system behaviors. This new style of interaction between the human engineer and the system implies that there is also a fundamental shift in what we as the engineers believe we can design the system to do, and in how we evaluate what acceptable solutions are. This new methodology changes our notions of sufficiency, optimality and any other evaluation criteria we attempt to apply to the design and the performance of the engineered OC system. Furthermore, since any evaluation criteria will partly develop along with system capabilities, we must design a system that does not have elegant predefined responses, but rather can generate reasonable solutions on-the-fly.

Learning how to effectively share control between humans and partially autonomous systems is already familiar to the research community; it is just made more difficult by the degree of self-modification and self-organization in an OC system. After all, OC is a continuation of automation, except that instead of just responding autonomously, the OC system is able to self-design some of its response capabilities to the world, maybe even including its own sensory as well as "action" capabilities.

To be a systems engineer for a system with the resources to adapt over its operational life requires a redefinition of the concept of "optimal" that has driven traditional design. Specifically, the OC systems we are proposing must contain components and processes that are not optimized for the narrow *a priori* definition of system *specifications* that has traditionally formed the basis for design and validation. For example, new emergent features enable new strategies, and therefore by necessity will fall outside the specifications previously defined for the system. Therefore, we have to evaluate the systems design and performance, which includes developing new metrics, as well as design the system components and processes so that they can be used and evaluated under unexpected – even unintended – circumstances in a monitorable way.

To be a systems engineer for any partly or fully automated system capable of adaptation and reconfiguration of its components and processes requires sharing control with that automated system, and hence a negotiation between the local and even private requirements of the autonomous system and the often more global perspective and requirements imposed by the system developer or user. These differences are not only in viewpoint (for example, how information is locally or globally understood and determined), but in the contexts for requirements and system capabilities. For example, the system developer may need to consider not only the system's operational context, but also legal, political, social and indeed moral contexts for potential uses of the system. Therefore, the human developers and the system may have very different purposes and goals. For example, immediate costs to the autonomous system may bias its reasoning processes and therefore its developmental processes, to the detriment of necessary long-term goals hoped for by the system developers and expected by the system users or owners. These negotiations between human system developer and partly self-determining OC systems lead to a number of distinct challenges for the human system developers and systems engineers.

In the sections that follow, we discuss many of the processes that are central to OC system capabilities. However, before we do so it is worth emphasizing here that there are three classes of processes that we discuss: first are those processes that we believe underlie the distinctive and remarkable properties of biological systems, for which we discuss how we might build analogs appropriate for OC systems; second are those processes that may or may not exist in biological systems, but certainly are critical to OC systems in order to make use of the biological-like processes; third are OC processes that are critical to our ability, as the human engineers, managers, users, and owners, to communicate with the OC system, to manage and to share control with it, and possibly to repurpose it. Because of the importance of system-human communication, we argue that meaningful and context-specific communication between the system and its designers, developers, and users is essential to this endeavor, and that therefore, the creation and use of appropriate and sharable language is fundamental to its success.

In section 3.2 we discuss such biological characteristics as permissive growth and development, how biological systems achieve controlled sources of variation, and the opportunistic nature of biological processes and systems. We close by emphasizing several differences between biological systems and engineered systems that will drive the challenges for systems engineers.

In section 3.3 we examine the systems engineering challenges of developing the above capabilities, focusing on five specific challenges. The first of these challenges is to create generative processes. That is, although traditional design methods include tools for adjusting an operating point within a known parameter space, we will also need to develop processes for our OC systems that can efficiently create new and very different possibilities for the system. Secondly, because OC systems will adapt and change, the instrumentation that provides information about the system's current internal state will also need to rapidly adjust in a number of ways to the system's increasing complexity. This challenge also implies that we will also need to develop tools for creating evaluative processes that express the results of measurements in ways that are useful and understandable to both the system and its engineers, developers and users. The next challenge that we address is how to build the capabilities for reflection and direction that enable an OC system to identify and assess possible responses, and choose, implement, and adjust them as its context and understanding shift. Our fourth challenge is to enable our OC systems to utilize a portion of their resources to "actively experiment", discovering properties, relationships, attributes, and limitations of both their own capabilities and their ability to operate within different environments. The final engineering challenge is to combine the capabilities resulting from the previous four challenges to enable our OC systems to build models of their changing environment, and to use those models to identify unusual features of their situation. That is, we suggest that an OC system must achieve a situational awareness capability that directs its resources toward the aspects of its environment and internal state that present, at the current time, the most important threats or opportunities.

In section 3.4 we discuss processes that enable the OC system to share information and control with its human developers and managers. In order to build the basis for shared control, the system and the human must be able to communicate about system state and control decisions, and also to negotiate plans and goals. Hence we consider the difficult problem of developing shared representations and languages. We also discuss our progress and prospects along these lines.

In section 3.5, we wonder aloud if we could perhaps take advantage of some of the biological principles suggested in this chapter to better organize our own discovery processes as a community of OC researchers and to leverage off of each other's work as we together confront the challenges of achieving the potential of OC systems.

3.2 Key biological principles for an OC system

One of the fundamental goals for OC is to develop systems with key biological-like capabilities to adapt and change on multiple time scales, and to evolve, develop, and grow on their own in response to their current state, their context (including the goals and purposes of their designers, owners, and users), and their history. This goal is motivated by the astonishingly wide variability of responses that are observed in biological systems, as well as their remarkable robustness in responding to sudden large changes in their environment.

In this section we focus our attention on several principles that we believe are essential to providing the foundations of biological-like adaptation and robustness: building processes rather than building components, "permissiveness", generative processes and controlled sources of variation, and "opportunistic" processes.

Hence, in section 3.2.1, we emphasize that one of the most important aspects of biological processing may be, in fact, that biological systems build the processes that create and maintain biological structures, rather than building structures *per se.* Instead of attempting to achieve a particular structure or a particular result, the emphasis is instead on building processes which are analogs of factory floors or assembly lines, an image easily extended to cellular and genetic lines. As we discuss in this section, this approach means that basic elements are constantly rebuilt and renewed, which allows points of entry for all sorts of adaptive possibilities.

In section 3.2.2, we describe how the "assembly lines" of nature do not reproduce the precisely-constrained products that we strive for in engineered systems. Rather, numerous observations of variation of biological components and their further differentiation into new types point to a type of widespread permissiveness. The "permissiveness principle" allows all interactions, relationships, variations, actions and results *unless* any of these are shown to be deleterious or harmful. One could only use such a principle if there are methods for monitoring and discovering the results and effects of such variations. Clearly one can only follow such permissive strategies in the context of populations of elements. The building processes noted above provide both the populations of elements or events and many of the means for changing those elements or events.

The permissiveness principle results in the occurrence of many different kinds of unintended interactions, resulting in turn in side-effects and emergent phenomena (section 3.2.3). These sources of variation and of novelty are critical to enabling the types of changes in a biological system that, if used correctly, can become the basis for adaptive responses over the life of an individual cell or organism or, on a different time scale, over the evolution of a species. However, even though permissive processes provide many novel kinds of variations and occurrences, biological systems have found that the "hit-or-miss" quality of changes and variations stemming from emergence and side-effects is not persistent or consistent enough to meet the requirements for controlled sources of variation required by many adaptive processes. Therefore, biological systems have somehow created active processes that generate variations. In this section, we describe two qualitatively different types of generative processes: ones that create relatively well-defined, persistent, and constrained sources of variation and ones that change the nature of the solution space.

As clearly indicated by the above arguments, the resulting broad range of possible system behaviors could be exploited by adaptive processes. We call such processes "opportunistic" because they are designed to notice and then take advantage of variations and events occurring due to emergence, side-effects or controlled sources of variations. In section 3.2.4, we present some examples of "opportunistic processes" in animal systems and describe some of the capabilities of biological systems that enable these processes.

Lastly, in section 3.2.5, we discuss the differences between OC processes and biological processes that we contend are essential because OC systems are artificial, engineered constructs rather than the result of the evolution of an entire ecology. Even with appropriate analogs of all these biological principles, an OC system requires additional capabilities and processes in order for us, the human developers and users, to monitor, shape, and negotiate with it. This last topic will lead us directly into section 3.3, which presents the challenges for OC systems engineering that we deduce from these biological principles.

3.2.1 Build processes not structures

One of the obvious properties of biological systems is that they grow and they develop. Growth and development are at once adaptive advantages for an organism because it can respond to its changing environment with its growth and development, and a necessary result of life: since there is no external designer and developer of a biological system, it must "bootstrap" itself into existence by this growth and development. This then is the essential reason for the principle that biological systems in fact build the processes that create and maintain biological structures, rather than building a structure directly. Any system that self-organizes and self-designs will require some bootstrapping processes. The interesting question here is whether the bootstrapping processes of our artificial systems will need to use the same biological strategy of building up a family of related elements that can then be differentiated and used by the system.

We begin by considering how biological systems emerged from a less differentiated universe of matter and energy in the first place. Without speculating about this evolution in detail, we draw three important ideas from such imaginings.

First, when persistent biological structures emerge from the dynamics of physical systems, they are indeed persistent and separate, but only in a relative sense. This is because they are created out of the *same materials* — and therefore share in many ways the same fundamental parameters at some basic level — as their surroundings. A cell wall is semi-permeable; a brain region is a recognizable region with sloppy boundaries and extents, and so it goes. One of the implications of this view is that the boundaries of a biological structure are always leaky and somewhat continuous with the world around it.

Another implication, emphasized below, is that because these structures share so much in common with their surroundings they continue to exist only because of active building and maintenance processes. Unlike algorithms or transistors, which one can consider to be "permanent" when viewed in terms of the operational lifetime of the system, there are many examples in biological systems where biological "components" such as cell walls, muscle, or neural pathways go away or change if active maintenance processes change. A good example is what happens to bone and heart muscles in the zero-gravity environment of space [63, 15, 64].

Not only are there many biological examples of components disappearing upon disuse, but there are also many examples where biological systems appear to make use of these active building and maintenance processes to support crucial flexibility in systems. This "flexible modularity" is seen clearly in language and movement. As Bellman and Walter [13] state,

> We have overused the idea of built-in structures by being overly dependent on prewired patterning. This concept places the emphasis on the coherence and the "fixedness" of the assemblages. It largely ignores the means of introducing flexibility and variability into the combinations of elements used in assemblages. Yet the ability to recombine relatively independent elements and hence to decompose the assemblages is an equally important and complementary process to our ability to form those assemblages. Any word or movement can potentially be combined with a very large number of other words or movements to form a large number of sentences or acts. Hence, both language and movement are structurally coherent in the assemblages and are also generative. We use the word generative because it puts the emphasis on producing and originating new forms that conform to a body of rules. We also think of this generative quality as being acted out in an "on-line" fashion. That is, the animal is constantly generating new assemblages as it acts or speaks and as it adjusts for and monitors the context. Many of these assemblages could be temporarily formed for the moment's purpose, which places the emphasis on the processes that combine elements and not on the fixedness of the combinations.

In section 3.2.3.2 we discuss in more detail the types of adaptive behavior supported by such generative processes. Here we simply want to emphasize that structures and behaviors within an OC system will be more like these biological "assemblages". That is, active processes will continually recruit the necessary components, build useful assemblages, and maintain those assemblages, often doing so only for the duration of a specific current context.

The second key idea about the evolution of biological structures is that biological systems continue to use the dynamics among emerged structures to create and maintain new structures, hence building up many layers of structures with complex interactions. One of the results of the above viewpoint is that the emerged structure does not have to be made to fit with its surroundings. Rather, because it has emerged at all, it is *ipso facto* viable within its surroundings. In that sense there is a continual validation – in engineering terms – of the interface (but not necessarily validation of the performance or the functions of that component or set of relationships).

Another result is that the boundaries of the emerged structures may be less distinguishable from the rest of the system and its external environment than we might expect or desire. But since the boundaries are in our terminology more "leaky", they share more properties with the other parts of the system, including many layers of the system at once. Such shared parameters could, through "opportunistic processes", become the means by which the system both integrates across different system elements and adaptively controls parameters that have been found to vary meaningfully with critical differences in system behavior and the accomplishment of different system goals. This property results in complex side effects, but also provides tremendous opportunity for the use of shared characteristics in adaptive and integrative processes. Our design problem, described in section 3.3, is to help develop the types of discovery processes in the OC system so that it finds the ones that are most useful for our purposes.

As an example, an important characteristic often observed in biological systems is that a single control element such as a "master gene" or neurotransmitter can have multiple, diverse, and widely distributed impacts throughout the system's levels and processes. In a system whose structures are continuously maintained, created, and modified by processes that use the same basic raw materials, this distributed effectiveness of a control element allows local adaptations while helping to provide a basis for system-wide integration. In a system with leaky layers, one could imagine how serendipitous combinations of side-effects, if properly captured, could result in such wide-spread effects and help provide the basis for integrating across diverse kinds of elements and layers of elements. Thus, a third key concept we can learn from biology is that as increasingly complex processes and structures are developed some common control elements link them. Often these shared control elements are really families of elements related through the history of their development through common "building processes" and through retained common features. However, there will also be differences among the control elements within a family due to local specialization. That is, because many of these building processes are distributed throughout the entire biological system, their assembly lines can be impacted and specialized to local conditions. Hence the fact that there are building processes is key to both providing the similarities among families of elements and the "entry points" for the adaptations that will occur because of local requirements.

One of the most important consequences of shifting from building structures to building processes is that a system will have a much broader range of possibilities. The continual renewal of processes and structures gives the system a "safe" region of its "possibility space" within which it is relatively free to adjust, because its existing processes and structures are known to be successful in at least some "nearby" portions of its possibility space. At the same time, because these processes are shared and distributed across many parts of the system's hierarchy, some of the integrated responses can also enable "long leaps". This means that local adaptation at one level can have widespread effects across the hierarchies of emerged structures and components. These long leaps also enable the recruitment of far-ranging and diverse components.

The biological style offers a very rich set of control options that include both controlling for processes and for outcomes. In cybernetic terms, Kreitman's conjecture [36], states that in an environment of arbitrary disturbances, at any particular time one can control for either the process or the outcome, but not both. Biology does both, though clearly at different times. There is a first emphasis on building processes that generate populations of imprecisely replicated and varying elements. In section 3.3.4 we discuss the other necessary processes where, through feedback and selection mechanisms of several sorts, the system refines and modulates these processes to get desired results. When these building processes operate in a permissive biological environment they produce a wealth of interactions and emergent structures that will be utilized by the biological system.

3.2.2 Permissive growth and development

In this section, we explore the implications of the "permissiveness principle", which we consider a fundamental biological principle, one that helps separate biological from engineered behaviors and capabilities. In order to introduce it, we will first start with a brief description of classical systems engineering, and then contrast that with biological "permissiveness".

Systems engineering for traditionally-constructed systems defines and locks in the performance requirements for the system and the interfaces among its components. During the early design and specification stage, often called conceptual design, the foundational mappings of functions onto specific components are identified. These choices become the basis for specifying the rest of the system, so that consideration of alternatives is often frozen out of the ensuing design process. This approach helps to organize and manage the design process, which is focused on the familiar and extremely useful representation of a system as a block diagram that details both the individual subsystems and their allowed interactions. However, the choices of the contents of the boxes (i.e., what hardware and operational capabilities will be grouped together) and the interfaces between them (i.e., what symbols they will exchange and in what directions) can have profound influence on the functionality of the final product. Hence, the concept of a system organizing itself seems not only foreign, but perhaps also a bit dangerous, especially given that unanticipated behaviors of these traditionally-designed systems often result in catastrophic failures.

The challenge for systems engineers in the traditional approach can thus be seen as one of finding the best partition of the system; that is, to define the blocks and their interfaces. However, a focus on the contents of the boxes and on their designed-in interfaces leads us to ignore a wide variety of small interactions with the expectation that they will not contribute to the behavior of the system as a whole. In the context of a system made of a very large number of elements (for example, cutting-edge microprocessor designs have over 700 million transistors in an area less than 0.5 cm^2), there is an increasing potential for “small” interactions to lead to emergent behavior with unintended impacts, some immediately observable as detrimental; some detrimental over much longer time frames.

Let us now consider qualitatively how biological systems differ from this classic style of engineering a system, in which one specifies and designs specific components, engineered to be as uniform as possible, and specific interfaces with other components so that the system controls as much as possible the interactions among components and hence any side-effects. Although there are many critical biological processes for shaping, refining, and controlling the system’s dynamics through monitoring, regulatory and feedback processes of many types, biological systems allow a great deal of variability and imprecision in their components.

Similarly, we already mentioned that there is a great deal of leakiness among biological components and between levels of components. One important aspect of biological leakiness and variability is that many – perhaps even most – of the system’s interactions and structures aren’t controlled for directly. In fact, unlike engineered systems, multiple parallel and overlapping processes and structures exist in biological systems. These parallel and overlapping pathways are another manifestation of the permissiveness principle.

Biological systems appear to be deeply permissive at all levels of organization. That is, anything goes so long as the organism or system hasn’t learned (e.g., over evolution in populations of organisms or by feedback in terms of a single organism) that there is a harmful or deleterious effect. Thus, in a biological system anything that is not physically impossible can become part of its behavioral repertoire or feature set unless it is explicitly disallowed or prevented by processes within the organism. This permissiveness principle will produce a large number of unregulated features and relationships, some of which will be noticed at any time by monitoring and regulatory processes. Regulation could eventually decrease those features and relationships that negatively impact the system’s viability and functioning and by the same mechanisms preserve and even enhance ones that support system’s viability and functioning.

Permissiveness could be one of the essential principles that allows a system to generate emergent phenomena and side-effects and the processes that utilize them. That is, some of these serendipitous combinations of effects relating families of distributed command elements, described before, will later be the basis of the opportunistic processes that are able to make use of such emergent phenomena.

3.2.3 Emergence, side-effects, and controlled sources of variation

In the classic view of adaptive control, a system equipped with its unique combination of sensor, behavioral, and feedback processes assesses its current state and the relevant features of its operational environment, decides what courses of actions are feasible, and then selects, plans and executes the needed control adjustments and behavioral actions. Sensors, feedback, control adjustments and actions all presuppose that the system is able to vary any aspect of itself as a response to its plans or to events in the environment. Those attributes a system is able to change are limited and shaped by its physical and intellectual capabilities. Together these form a space of possible variations, which we call the "possibility space." Any selection of control adjustments, sensor tasking, actions or plans will be a subset of this space. In this section, we suggest that the permissiveness principle creates the proper milieu for emergence and side-effects, and that these, in turn, are critical sources of variation that are captured and made use of by adaptive systems to extend their possibility spaces.

Although permissive processes provide many novel kinds of variations and occurrences, biological systems have found that the "hit-or-miss" quality of changes and variations stemming from emergence and side-effects is not persistent or consistent enough to provide the controlled sources of variation required by many adaptive processes. Therefore, somehow biological systems have created active processes that generate variations. In this section, we describe two qualitatively different types of generative processes: ones that are relatively well-defined, constrained sources of variation that are persistently produced and maintained, and a second type which changes the possibility space and expands the design envelope in unexpected ways.

3.2.3.1 Emergence and side effects

We leave to others the challenge of characterizing emergent phenomena[3] and developing methods for predicting emergence in complex systems. Instead, we accept that side-effects and emergence are known to be possible, and go on to consider how a complex system could develop processes that take advantage of such phenomena when they occur, and make use of them to be more robust and adaptive.

Although emergence and other side-effects continue to plague human-engineered systems, in biological systems it is clear that emergence and other unexpected phenomena are utilized by the system to provide needed sources of change and novelty. We believe that permissive processes promote emergence, since they allow unexpected or unplanned coincidences to reinforce each other.

Cellular automata and other dynamical explorations demonstrate that stable, long-lived structures can emerge from stochastically-generated initial conditions in a "flat" rule space [27]. Although such demonstrations are very

[3] We find the following working definition of emergence by Christian Müller-Schloer to be useful: "Emergent phenomena is where collections of individuals interact, without central control, to produce results that are NOT explicitly 'programmed' and which are perceived as 'orderly.' " [69]

important for developing our understanding of emergence, the conditions in many of them are quite different from the conditions for emergence in biological systems. By examining biological systems, we may learn new strategies for the management and utilization of emergence. Emergence in biological systems 1) does not start with a uniform distribution of identical elements; 2) involves a system with goals, intentions, and learning; 3) must be observed from inside the system by the system. That is, in dynamics demonstrations, the observer is outside the system, but in biology the system itself must be the observer.

In biological systems, emergence occurs in a system that has lots of existing structures, which embody the history of both the individual and its species. These structures and processes can be recruited to support an emerging behavior or structure. Unlike the homogeneous elements of many dynamical models that were used valuably to prove the existence of emergence from very simple and uniformly distributed elements, biological systems have heterogeneous components at many different functional layers. Such existing structures mean that across the biological system there will be regions with quite different ongoing dynamical processes. These existing structures both constrain and shape the dynamics that may lead to emergence and at the same time, become the basic components of a different level of dynamics in the system. For example, the human nervous system has meaningful dynamics occurring at the molecular, cellular, and organ levels, all of which may lead to emergent structures and behaviors with impacts across the different levels.

The second major difference in biological emergence is that biological systems have goals and intentions; they plan and learn on their own and from others. Hence the emergence of new patterns resulting from unexpected dynamics and the side-effects of widely distributed relationships among components both impacts and is impacted by the existence of purposive behavior in biology. In this sense, the central challenge of OC research is also the central challenge of biological systems: how to combine unanticipated patterns and events with intended patterns and events.

Emergence can appear very differently when viewed from inside the system or by an external observer. The whole emergent pattern may never be seen as such from the vantage point of internal observation, but instead look like a set of correlated differences or changes distributed throughout different locally monitored regions. Ian Stewart [81] presents many interesting points about the problems faced by a participant in a dynamical system. These include observing patterns and making decisions based on limited information from within the system and attempting to abstract decision rules from the behavior of the system. Some of these problems can be demonstrated with Langton's ant. These simulations have two simple rules assuming an initially all-white grid. Step into a square; paint the square you came from black if it was white and white if it was black; if the current square is white, turn right and if it is black, turn left.

Stewart describes the patterns generated by simulations of Langton's ant as first "symmetric", then "chaotic", and then the ant "builds a highway forever" in about ten thousand steps. If the simulation starts with a pattern rather than all white squares, apparently it still ends up in a "highway." Stewart's conclusion is that one cannot predict this pattern, but rather only do the simulation: there is no possibility to shortcut the process, which he considers a key characteristic of emergence. Furthermore he considered the calculation of the rules for the behavior as intractable. In his analogs of the ant game, there can be several billion steps before something interesting happens. And then such analogs cannot tell you why or even how such patterns came to be. There are many points in this work that should be considered by those of us building OC systems. However, biological systems may have sidestepped some of these difficulties because they have found ways to utilize emergence without having to perceive the whole pattern or predict the endpoints in some emerging process.

In biological systems, the local-only perceptions and decisions of Langton's ant can have far-ranging effects because of the shared assembly lines and common control elements noted earlier. Furthermore, biology has also developed strategies to combine and coordinate the perception, learning, and decisions of a community of players through diverse memory and communication processes.

Biological systems have developed communication methods that allow them to share their experiences and viewpoints. This is discussed further in section 3.4. The importance here is that, although this does not change the intractability of perceiving the end points in their own evolution, it may change the ability of a biological system to track its own long-term and emergent patterns because other members of its species can observe it from the outside and communicate those observations. For example, if one animal even at a simple level can inform another that it is too close to a dangerous situation, that can motivate that perceived system to discover what changes or clues in its environment or self it ignored and allow it to correct those insufficiencies. Parents, of many species, constantly do that with their offspring.

Recent work in distributed optimization and market-like decision processes also show that if the decision game is set up correctly – including the topology of the relationships among participating decision elements – then local decision-making elements can not only come up with coherent and useful global decisions and effects but, under many conditions, optimal ones [71].

As an example that summarizes all the points we have made here, consider walking. The biological system starts with a number of built-in structures, such as neural pattern generators that contribute to the timing of the gait and successful configurations among muscles, tendons, and bones that contribute to walking for that individual or species. However, the system does not have to predict or lay out the values of all of these muscles and neurons to walk. Rather, it appears to launch the behavior with some rather stereotypic gait patterns and an abstract goal of intended place or direction. It then uses self-monitoring, feedback, reflection and other opportunistic processes to dynamically recruit and organize its components. It constantly adjusts for external conditions, e.g., the slight shifts in ground, and for that system's own performance capabilities. For example, fatigue will impact how high it lifts its limbs or the speed of its gait. Parents help train and shape their offspring in the important movement patterns of their species — and the offspring clearly learns as well through trial and error. The resulting walking pattern details and path could not have been predicted ahead of time and therefore in accordance with Stewart's definition are emergent. However, it was not important to the biological system to know in advance that level of detail in its planning or to predict such details.

Clearly a system's measurement capabilities are critical to its ability to self-monitor and therefore to respond opportunistically to emergent processes. If the appropriate opportunistic processes are available, and if they are coupled with sufficient instrumentation, then neither a system nor its engineers, developers, and users need to distinguish between emergence and other effects. Instead, all of the permissively-attempted paths, together with their results, become available for use in achieving the purposes and goals set for the system.

3.2.3.2 Generative processes

Permissiveness alone may not be enough for a biological system to be able to adapt to changing contexts because the needed amplitudes and types of variations may not be available. Hence, biology has developed generative processes. Some of these produce relatively well-defined and constrained sources of variation that are persistently produced and maintained; these are like the volume knobs or the tuners on a stereo. In some sense, they embody a parameterization of some key characteristics of the system that can be controlled by the system. A second type of generative processes change the nature of the possibility space. They are likely to be much rarer in occurrence and more often fatal to the system. However, a few will survive the test of viability with the rest of the system to provide very new attributes, and can be picked up by the system in surprising ways. Some of these changes could lead to very different ways of doing things and eventually, in higher cognitive processing, to very different choice and decision processes.

There are many examples of processes in biological systems that first generate new combinations of elements and then, after evaluation, incorporate the successful ones into some more persistently available form. Examples of these generative processes occur at all levels of biological systems. At the genetic level, we observe that both meiosis (the normal reshuffling of genes from both parents) and mutations (the "errors" resulting from the imprecise and inefficient genetic replication processes) provide variation in a population that is acted upon by natural selection to create species that are adapted to an environmental niche and also robust enough to handle many unanticipated environmental stresses. Meiosis is a dependable source of variation in the well-defined space of current genes. Mutations are rarer, mostly deleterious or lethal, but once in a while, greatly successful. Meiosis gives rise to variations within a local safe region of the possibility space for the species, while mutation is a source of long leaps to possible new regions.

At the cellular level, neurophysiologists have studied and speculated on the "helpfulness" of variations to allow more resilient pattern generation and to prevent perseveration or over-recruitment among cells. A special case is the ability to generate variations that act like noise generators. As an example, Garfinkel [28] studied the use of such noise generation in the regulation of heart patterns and recovery from abnormalities,

> The extreme sensitivity to initial conditions that chaotic systems display makes them unstable and unpredictable. Yet that same sensitivity also makes them highly susceptible to control, provided that the developing chaos can be analyzed in real time and that analysis is then used to make small control interventions. This strategy has been used here to stabilize cardiac arrhythmias induced by the drug ouabain in rabbit ventricle. By administering electrical stimuli to the heart at irregular times determined by chaos theory, the arrhythmia was converted to periodic beating.

An example of generative processes at the behavioral level is behavioral merging. Behavioral merging is not only an example of generative processes but also an example of the way in which the resulting new combinations and variations in behavioral elements can be used to handle adaptively a common control problem. It demonstrates the resolution of conflict between competing goals, tasks, or requirements, and the ability to map actions to goals in highly flexible ways. As described by Bellman and Walter [13],

> A given instance of behavior can reflect several motivations and work toward several goals at once. Contrary to the usual emphasis in behavioral studies, in which an animal must choose between mutually exclusive acts, an animal in nature is rarely in the situation where it must engage in one behavior to the exclusion of other behaviors. Rather, an animal's movement frequently shows "behavioral merging," in which several motivational goals and action patterns are combined into one coherent pattern. In studying the merging of feeding and aggression behaviors in the lizard, an animal noted for the rigidity of its behavior patterns, Bellman found that when elements of feeding and aggression conflicted, other elements were selected and substituted, so that, overall, both feeding and aggressive patterns were combined into one fluid behavioral sequence.
>
> The behavioral sequence resulting from merging points to a particular type of flexibility in a movement system. A specific movement pattern

> can subserve a number of goals. If this is so, then a specific movement pattern is not necessarily linked to one goal any more than to any other goals (although there may be some kind of weighting, so that a given behavior is most often associated with one particular goal). This implies that a movement is not "released" as a necessary consequence of the occurrence of a particular motivational goal; rather it is "recruited" to serve that goal. Furthermore, from behavioral merging, we see that a whole action pattern need not be recruited but only those elements best fitting the circumstances.

This last point reinforces the importance of building processes, which provide the biological system with many points of entry for adaptive responses. These include the ability to drop steps in its processes if existing components of the system or features of the external operational environment permit it.

Lastly, in human language we clearly see generative processes that produce a large variety of phrases and patterns, while remaining consistent with both grammatical and semantic rules. This last example brings to the fore that these generative processes operate not only on physical and behavioral properties of the system, but also on the symbolic and representational capabilities of the system that underlies its cognitive and communicative capabilities.

Generative processes result in many different combinations of processes, structures, and symbols that can accomplish a particular outcome when combined with methods that build up or emphasize some relationships, processes, or symbols while de-emphasizing others. Such a variety of choices enables a system or structure to substitute for a "broken" method or less effective method, an ability that is at the core of the robustness of biological systems. Compare this to the usual engineered system, which carefully specifies and engineers away all sources of variation and interactions among components, and which is brittle in the sense that it usually "breaks" if presented with unanticipated contexts or interactions.

3.2.4 Opportunistic processes

As clearly indicated by the above arguments, the resulting broad range of possible system behaviors could be exploited by adaptive processes. We call such processes "opportunistic" because they are designed to notice and then take advantage of variations and events occurring due to emergence, side-effects, or controlled sources of variations.

One of the classic attributes of adaptive behavior in animals is their ability to take advantage of whatever is in the environment in order to accomplish their goals. Even simple creatures, known for rigid behavioral patterns, usually are able to break their behavioral patterns if something in the environment has made those parts of the behavior unnecessary. [4] Hence if a suitable hole

[4] Konrad Lorenz had some notable exceptions such as the egg rolling behavior of the gray goose or the nut storing behavior of the red squirrel, but such rarer

exists, the organism will not dig it, if a suitable object is nearby the organism will not go further to move another object, and so forth. In more advanced animals this ability to take advantage of aspects of the environment is extended into sophisticated problem-solving capabilities that allow the animal to make use of objects in novel ways. An early example comes from the famous "insight experiments" of the German psychologist Köhler, who demonstrated the ability of apes to use objects in the laboratory in novel ways [35]. One such experiment required the apes to reach a desired banana by discovering that they could link several sections of a pole together. [5]

Jakob von Uexküll, the prolific and observant German ethologist, was one of the first to consider the qualities that go into adaptive and opportunistic biological systems. Especially important was his insight that an animal's perceptions are deeply affected by its effectors (its capabilities to move and do things with some given object or within some given ecosystem). One of his early stories is a description of how a hermit crab's motivational state affects its perception of an empty shell [83, 75]. When the crab was molting and vulnerable, it backed into the shell for protection. When the crab was hungry, it approached the shell displaying hunting behavior. When the crab was mating, it approached the same shell as a potential competitor and showed aggressive displays.

This type of opportunism and adaptiveness appears to occur even in the behavior of single-cell animals. As Jennings [30] observed, unicellular animals are capable of many of the complex and adaptive behaviors of multicellular animals. They respond to the same classes of stimuli to which humans do, they have specialized receptive areas (although not yet specialized for different senses), and they frequently have specialized contractile parts whose action is coordinated. They exhibit spontaneous behavior, early trial-and-error behavior, habituation, and context-dependent responsiveness to stimuli. As Jennings concludes, "We do not find in the nervous system *specific qualities* not found elsewhere in protoplasmic structures. The qualities of the nervous system are the general qualities of protoplasm." [30, page 263]

Jennings provides several fascinating examples of the ability of single-cell animals to interact with the environment and with each other. One example is the predator-prey relationships among *infusoria* such as *Didinia* and *Paramecia* [30, page 186].

---

examples are usually understandable in light of the criticality of that behavior to the animal's survival: one could say that nature in that case has over-engineered the response.

[5] Parenthetically, the apes in this case, claims Köhler, devised a smarter solution than his. Instead of the apes using the extended pole to knock the banana down, they used the extended pole to polevault their way to the banana. The importance of this observation is that the possibility space is determined partly by the animal's unique capabilities. Clearly, Köhler was not agile or light enough to polevault, so he did not recognize this possibility.

His observations demonstrated several startling, adept adaptive capabilities, including the ability for a hunter to cease its hunt of one highly-elusive target and turn to a better target, even though the first target was still visible and potentially available. The hunter subsequently caught the second prey. It did not cease its chase of the first target because it was no longer hungry, or because it was too tired to continue, or because the target was out of the scope of its sensors. Rather, it apparently stopped because it had not been successful enough within a certain amount of time and effort.

These biological examples are behavioral, and hence observable to scientists, but there is no reason to suppose that this opportunistic style of processing is not carried out at all levels of biological systems. In fact we contend that they are. Although we can only touch on a few illustrative examples here, we also believe that it is instructive for OC researchers to study the myriad ways in which different biological systems adapt. Such examples can inform as well as inspire the OC field on the style and power of the opportunistic processes we would like to develop for our artificial systems.

One of the first things that is evident from these biological examples is how critical generalization, differentiation, and learning processes are to the ability of the system to notice, capture, incorporate, and adaptively control the co-occurrences and interactions among a rich set of different types of components. Babies demonstrate wonderful examples of opportunistic developmental processes that are tempered by discrimination and learning. At first, a baby attempts to fit anything and everything into its known behaviors and goals. Everything it can grasp goes into the mouth, is picked up, is dropped, and so forth. Gradually, through refinement, differentiation, and learning processes, the baby learns what is too hot, too acrid, too heavy or just right. Its explorations are carefully constrained by concerned parents.

Opportunistic processing will critically depend on the instrumentation available to the system, which will determine what variations it can perceive, capture and incorporate into its repertoire of repeatable capabilities or reproducible states. Again, at all levels of the system, from genetic, cellular to behavioral and cognitive, we see very different means by which correlations and co-occurrences are retained and eventually culled until the correct aspects of some complex set of events has been retained by the system for future manipulations.

The building processes discussed earlier enable several of these opportunistic strategies. For example, the observation that animals can take advantage of the existence of structures or events in their environment or in their own metabolic pathways happens partly because in the building processes one has developed a set of feedback and monitoring processes that support a sequence of operations by having a large number of steps that are initiated or not, depending upon the occurrence of pre-conditions such as chemical precursors or trigger events. In addition, permissiveness will guarantee a large number of relationships and side-effects, creating the very good chance that the system will often stumble on "shortcuts" in its sequencing of events.

Just as combinatorial processes can be taken advantage of, so can processes that appear to be oppositional. In this case, the manipulation of opposing effects results in good solutions. Examples of this are the ways that combinations of neurotransmitters with opposing effects combine to provide the desired states, as well as a means to modulate such states with very small adjustments. Similar examples occur in the tensions between flection and extension muscle groups for limb movement, as well as the balance between excitatory and inhibitory neural pathways. In all these cases, the benefits of such an arrangement seem to rest on the ease with which small control adjustments of either opposing effect can lead to big changes in the states.

The opportunistic processes feed a number of critical processes that evaluate, reflect on, and utilize the information that has been gathered about correlated states to drive the behaviors of the system. However, opportunistic processes cannot just depend on the co-occurrences resulting from external events to drive their opportunities for correlating, refining, and differentiating the drivers for complex states. Hence they also need to drive their exploration of correlates. In section 3.3 we examine how active experimentation is used by the system to constantly learn more about how its own components interact and affect each other.

3.2.5 Critical distinctions between biological processes and OC processes

It is tempting to concentrate only on those processes that enable the biological-like characteristics that interest us, e.g., those processes that can generate novel behaviors and responses, those that contribute to the discovery and utilization of emergent patterns, etc.. However, in OC systems it is also essential to consider what other capabilities and processes must be added in order to allow human engineers, managers, users, and owners to communicate with and guide the OC system not only during the initial design phase, but also throughout its operational life. Such considerations are especially important for large and costly systems that will experience shifting contexts during their design and life-cycle.

There are two driving differences between the needs of biological systems in general and OC systems. The first is a result of the essentially “alien” nature of the OC system we noted in the beginning of this section. Biological systems are not only somewhat continuous with their environment, but are also part of a complete ecosystem; that is, in collaboration and competition with other systems, all of which are linked to and part of a larger whole. In contrast, the very concept of engineering a system leads it to be disconnected and “alien”; it is developed by us, usually with materials very different from those in its operational environment, doing functions that may have little to do with those of any other system in the operational environment.

In addition, because the biological system is continually renewed and recreated from the raw materials of its surroundings, it may more readily adjust its structure and composition because of small shifts in the available materials or the current conditions. Engineered systems on the other hand have predefined form and substance. Of course, this can have some advantages in an environment that suddenly no longer feeds the building processes of the biological system. However, it will not have the opportunities described above for adaptation through growth and development, and there will be no biological-like renewal of composition and structure.

One result of this is that OC systems require much more work in defining appropriate and deep enough context models of the operational environment, and in ensuring that there are appropriate interfaces for noticing all the needed attributes of the environment in coordination with the system's own sensing behavior and actions. One cannot depend on the "natural" shared parameters based on the physics and natural history of an ecological niche that occur with biological systems.

The second major requirement of OC systems that is different from biological systems is the need of the OC system to always remain tethered to us. They must always be monitored for and shaped by our goals and purposes. This does not mean that the OC system will not be able to act somewhat independently of us; in fact, we argue later that one of the chief modes of interaction with the OC system will be through negotiation and not through the usual fixed control methodologies used in other engineered systems. In so far as an OC system is self-organizing, it will in fact *be* one of the developers and one of the systems engineers of itself, in coordination with the human development team. But this need for entrée into the internal state of the OC system's sensors, effectors, and decision processes leads to the need for sufficient instrumentation, reflection and reasoning, and communication capabilities to work with us. As we discuss in section 3.4, the language of this collaboration and negotiation needs to be co-developed from the experience base of the OC system. We will require these systems to inform us of their state and intentions. Thus, an OC system must not only create appropriate symbols for its own use, but must also be able to explain them to us.

Whether structures, processes, and representations emerge or whether they are supplied at the beginning by the designers, what is important to us as OC systems engineers is to build in mechanisms for recognizing new possibilities, and for guiding the development process to some extent based on our (and the system's) growing understanding of the application problem domain and the processes and structures that have developed within the system in response to it. As much as possible, we also want to keep some of the means to "know" and to "find purpose or meaning" retained by the system, since the design engineers are unlikely to be available for the entire lifetime of the system. Thus, unlike a biological organism, an OC system will always retain a special type of differentiation from its surroundings in that it is deployed into an operational context to achieve purposes other than its own survival.

3.3 Systems engineering challenges

Our biological analogies have led us to suggest that OC system engineers will need to provide certain key capabilities that we believe enable the adaptability, opportunism, and robustness of biological systems. However, doing so presents significant challenges precisely because these capabilities require a radically different approach to engineering design, in which the processes that constantly rebuild, renew and expand the system, rather than the system's components and their interactions, are the central focus. Unlike the processes and resulting structures that we observe in nature, those created or managed by engineers will of necessity be artificial, if only because they are deployed to accomplish externally defined goals and purposes, and hence are in an important sense alien to their environment rather than having evolved as part of a complete ecology. Because we want biological capabilities in systems that have "non-organic" origins, we the developers will have to provide the means for detailed interactions with its operational environment and the starting structures and processes usually provided by evolution.

This means that a number of approaches and designs for such systems cannot work, because some basic assumptions and other design crutches are no longer available, no longer implicit in the approach. Instead, a new set of systems engineering methods and attitudes is needed, and this section begins our exploration of these issues.

The specific engineering challenges we will consider in this chapter are:

- creating appropriate generative processes (section 3.3.1);
- inventing appropriate instrumentation and evaluation processes (section 3.3.2);
- providing the capability for the system to analyze and reflect on the information it has gathered (section 3.3.3);
- enabling the system to actively experiment so that it improves its (and our) representations and models (section 3.3.4); and
- providing methods for the system to refine the symbols, representations, and models of both the system and its context, to create a kind of "situational awareness" (section 3.3.5).

We now discuss each of these challenges, as well as their implications, in more detail.

3.3.1 Creating generative processes

By generative processes we mean something much more interesting and much more challenging to construct than search processes, because the latter search a fixed and given space using predetermined parameters, whereas the former actively create both the parameterizations and the search space as they proceed. The additional capabilities of these generative processes as compared to traditional techniques are expected to yield some surprising results, desired and undesired, that the system can discover, evaluate, and choose to exploit or suppress, as well as new possibilities that offer increased effectiveness in controlling for processes and qualities that support the system's purposes and goals. These generative processes thus need to be able to efficiently create new and very different possibilities for the system, as well as to discover and describe linkages that couple processes and symbols, enabling coordinated responses among multiple processes and functional units at lower levels of the system's hierarchy.

Designing generative processes thus means developing capabilities to recognize and coordinate coherent activity among subsets of the processes that build functional units, as well as methods for describing this coherence by creating new symbols or variables. Such an approach is fundamentally different from the traditional conception of searching a predefined space using predefined variables and predefined criteria for success. For example, in computing algorithms we usually assume a predetermined basis set of characteristic and well-behaved variables, write a generic search through a large space chosen because it is easy to describe in terms of those variables, and then apply the success constraints to eliminate large portions of the space. An alternative approach, which seems to be more like what humans do, is to create the search space and new characteristic descriptions of it as we go along. We start with the construction of the search space from the viewpoint of our purposes or goals, which lets us incorporate many of the constraints into the construction itself, so that we automatically avoid consideration of large but uninteresting portions of a generic search space, and concentrate on those regions that seem interesting or useful, even if the resulting possibility space is made up of disconnected and oddly-shaped portions of what would have been the generic search space, and even though the resulting space would have been difficult to describe in terms of a set of predetermined variables. We also organize our understanding of this search space by making up new interpretable descriptions of the various options we discover, a form of reparameterization that allows us to more easily use these new possibilities in other contexts.

The previous example suggests that purposes and goals are central to the efficient construction of generative processes, since they provide a ranking of high-level criteria for success. We saw many examples of this continuous mapping between the goals and the generative processes in the biological examples described in section 3.2.3.2.

When we humans solve problems we actually seem to use two basic strategies: we start with familiar possibilities and search for relevant or useful combinations, but we also use long leaps to radically different possibilities. This example of human problem-solving illustrates the two types of capabilities that we include in the overall description of "generative processes". One is more local and continuous, resembling refinement or "pushing the envelope", while the other involves disconnected leaps. In section 3.2.1, we speculated on how the assembly lines and building processes support these long leaps by producing families of distributed control elements that share features and whose domains of action include simultaneously different levels of modules or components in the system. Generative processes as we conceive of them must thus include some methods that are able to efficiently "leap to" and evaluate regions of the system's possibility space that are "far" from those it has already experienced, as opposed to only searching for a new combination of established parameters. Although these exploratory leaps could be accomplished by random variation of basic units, such a random combinatorial search is not sufficiently efficient in terms of resources and time even at fairly modest levels of complexity. In fact, the size-, context-, and time-dependent structure of these possibility spaces makes it impossible even in principle for random searching to explore the space in any useful amount of time.

We envision the following characteristics for "leaps". They will produce relatively large changes from the current state, frequently "land" in "useful" final states, utilize previous discoveries of successful strategies, be closely coupled to evaluative, reflective and directive processes that result in the depreciation or elimination of unsuccessful results and the reinforcement of interesting ones, and be strongly context-dependent.

We suggest that generative processes in OC systems are likely to be based on the "modularity" of the higher-level processes and structures that the system knows so far, and perhaps even more particularly to be based on the representations of those processes and structures. Biology seems to take advantage of modularity to achieve many of those characteristics that we desire in our artificial systems. Deem [23] has described biological modularity, as well as the hierarchy of complexity that accompanies it, as follows.

> A modular structure to the molecules of life allows biological information to be stored in pieces. The existence of this modularity means that evolution need not proceed just by changes of one base of the genetic code or movement of one atom or amino acid at a time; rather, functional units can be exchanged among living organisms. For example, [...] proteins often comprise almost independent modules, and the genetic information that codes for those modules may be transmitted through evolution. The modular structure of proteins is hierarchical, with identifiable elements at the levels of atoms, amino acids, secondary structures, and domains. Hierarchical elements continue through the levels of proteins, multiprotein complexes, pathways, cells, organs, individuals, and species.

Thus, one way to produce both meiosis- and mutation-like generative processes is to mix previously-created and relatively high-level functional units in new ways (perhaps by rearrangement or reuse in a new internal context), because existing functional units embody known successful strategies. The results of such operations with higher-level units of the system hierarchy not only provide methods for expansion of the system's possibility space, but do so in a way that ties to methods of building new descriptive terms based on existing ones. Another way to understand the potential of modularity is to recognize that the building processes that continually rebuild and renew the system's components and structures are also in a sense modular, with various entry points depending on context. This means that these same processes can be recombined in new ways using a context-dependent combination of sequences of entry points.

One potentially valuable way to model these effects may be the class of small world effect models currently studied in network science [84, 2]. These models are used to study the impacts of having local neighborhoods of links interspersed with a few long links.

Modularity-based approaches are likely to be much more efficient than those that modify or combine low-level processes and structures. If given sufficient resources, searches based on low-level components will certainly eventually discover not only the same "reshufflings" and "leaps" as modular schemes, but potentially much other useful information as well. However, we contend that approaches based on low levels of a system's hierarchy will generate many more unsuccessful proposals because they largely ignore the knowledge of the possibility space and trajectories through it that are represented by previous successful discoveries embodied in the modules.

Since search processes need a space to search in, part of the problem here is to construct that search space in a sufficiently flexible way to also allow sufficiently fast searching. We specifically do not use the term "efficient" here, since efficiency is the opposite of the robustness we want the system to exhibit. The modules themselves are like safe regions; that is, safe configurations with local allowable variation. The combination of modules is a way of combining disconnected safe regions in the possibility space for constructing a search space.

We have been discussing the advantages of modularity; this is one of the chief legacies that biological systems get from their evolution. That is, there will be many side-effects that are at best neutral or, as in the case of genetic mutation, largely deleterious. It takes a special narrow combination of constraints and coincidental events to show the benefits of an effect. One of the ways that a biological system benefits from being a member of a family of biological systems is the leverage of many failed possibilities being constrained away before it comes into existence. Many of the partial structures and properties of the biological system are in fact the embodied memory of these constraints now physically imposed by genetics, the structure of its physical and cognitive components. We as the human systems engineer working with the developing OC system will play the role of evolution through simulation and our experience across families of like systems. Providing some of this experience base and the constraints is in fact one of the chief jobs of the systems engineer for OC systems.

The ability of an OC system to create its own languages and representations is an important key to achieving the efficiency of modularity-based generative processes in part because these internally-produced symbols can be exchanged with other processes and structures both within a single system and across multiple realizations of a system. Such exchanges of symbols in effect disseminate the knowledge of processes and structures that have been proven to succeed in at least some part of the system's possibility space, a point to which we will return in section 3.3.4.

All types of generative processes that we advocate are powerful tools for building adaptive systems, but they are not themselves sufficient to produce adaptability, robustness, opportunism, and other biological-like characteristics that we want in OC systems. The results of these generative "experiments", both those operating within a search space and those violating its boundaries, must be measured and then evaluated in terms of the symbols and languages known to the system (section 3.3.2).

The system and its engineers must also be able to reflect on the results of measurements, comparing them to models of internal operation and external context. These comparisons involve the critical (and difficult) ability to compare models and results on the fly, adjusting the models to fit new, and especially novel, results. In addition, the capability for reflection (which takes place at many levels of the system hierarchy) is needed to identify possible responses and project their consequences into the future. At any given time, the system must choose how it will respond. We call this "direction."

Coupling generative processes to instrumentation, evaluation, reflection, and direction in a strongly permissive milieu gives the system an ability to actively experiment (section 3.3.4) when the resources to do so are available. We contend that generative processes play an essential role in enabling effective active experimentation, and conversely, devising and evaluating active experiments is essential for the creation of effective generative processes. Therefore, because of these mutually enabled roles, we believe that research on strategies for designing these processes is essential to progress in engineering OC systems.

A final challenge related to developing generative processes is that, in addition to supplying the original generative processes, the raw materials which they manipulate must be built as part of the engineering process. We make no assumptions here about where these beginnings come from, but we expect that a lot of it will be imposed from the outside, by the designers, who will be making decisions concerning the raw materials, initial conditions and processes, basic symbols, evaluation and validation criteria, and other starting points.

3.3.2 Instrumentation and evaluation

The traditional block diagram view of a control system as shown in figure 3.1 implicitly assumes the ability to measure values of the reference input, actuating signal, manipulated variable, controlled variable, and feedback signal. The instrumentation for these measurements is implicit in the diagram, since each block uses one or more of these variables as its input. In this section we consider the challenge of implementing appropriate instrumentation in OC

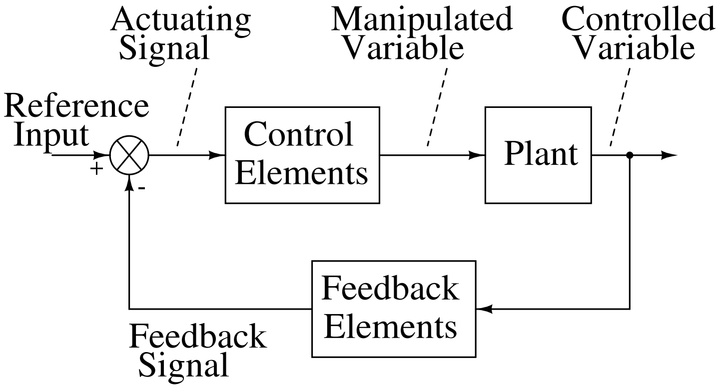

Fig. 3.1. Feedback control system block diagram

systems, where the system will need to adjust and even perhaps create its instrumentation in response to its changing processes, structures, symbols, representations, languages, models, and context. We also consider the closely related challenge of evaluation, which is the capability to use measurements obtained from the system's instrumentation, together with models of its state and trajectory, to achieve its purposes and goals.

Although the approach represented in figure 3.1 has supported an immense body of successful engineering analysis and design, the sense of simplicity and comfort we perceive from these successes is tempered by the knowledge that some engineered systems demonstrate complex and challenging phenomena such as bifurcation, chaos, and emergence. One of the goals for OC systems is to enable our engineered systems to recognize and, where possible, make use of these phenomena to achieve their purposes and goals. We are thus led to the questions of what to measure and how to measure in order to give our systems the information that enables this capability.

These instrumentation questions are not unique to OC systems. In fact, they resemble both theoretical and experimental challenges in physics having to do with modeling and measurement of systems that are discrete at the lowest scale but also have average effects at higher scales such as pressure and temperature that are *usually* sufficient to describe the collective behavior of a large number of these discrete elements in a particular region. Pippard [74] has captured one version of the physicist's view as follows:

> Can it be that the systematic reduction of complex processes to their basic constituents, obeying laws of marvelous simplicity, has left us with a body of knowledge whose usefulness is rather problematical? It has been the habitual claim of physicists that they could make predictions whose verification underpinned the laws and conferred on science a validity that no other branch of learning could aspire to.

> Was this a delusion? Of course not, but the claim may have been overoptimistically expressed.
> It was always recognized that complexity might preclude detailed prediction — no one ever hoped to follow the motion of each molecule in a gas. But long before statistical mechanics provided a theoretical foundation, it was clear that the average properties of pressure, velocity, temperature, etc., obeyed quite straightforward laws of thermodynamics and hydrodynamics. Straightforward though they were, the equations expressing them were still capable of yielding highly irregular solutions, and this time there is no molecular complexity to explain turbulence away — it is intrinsic to the equations. This should have been enough to alert us to the potential in almost any non-linear differential equation to surprise us by the diversity of its solutions.

The difficulties of predicting physical phenomena such as phase transitions and the turbulence transition in fluid flow, and especially the challenge of recognizing such changes in global behavior in real time from a set of local measurements made inside the medium, give us a concrete analogy for considering instrumentation and measurement in the context of complex systems, and particularly in OC systems.

Basically, we need to answer two questions: "How much instrumentation or information is good enough?" and "What kind of instrumentation or information is good enough?" In the examples that phase transitions and fluid dynamics demonstrate, instrumenting every process and structure down to the equivalent of the discrete molecules in a fluid does not necessarily ensure that our instrumentation or the information it produces is "good enough" in the sense that we can link it to useful descriptions, parameterizations and models, and, ultimately, to purposes and goals, even when the "molecules" have fixed properties.

Instrumenting everything is also not a useful approach for other reasons. For example, too much instrumentation may fundamentally alter what we instrument, or the instrumentation and measurement may consume so much of the available resources that it overwhelms the capability to process it, meaning that it is impossible to make use of the resulting data. For example, in wireless networks, as the number of mobile nodes increases the messages tracking their location can overwhelm the capacity of the network, precluding its ability to accomplish its goal of transmitting content messages.

Also, if we measure all the details then we need models at the same level of complexity. (Otherwise we might as well focus our measurement efforts on determining averages at appropriate scales.) Our notion of "particularity" is useful here. The enormous amount of detail that can be relevant in a complex dynamic environment for any system means that the system needs a lot of help in observing and describing it. Any proposed modeling method will suffer from the complete inability to reach a "critical mass" of information until there is enough descriptive detail, since the important interactions may include what would otherwise seem to be minor effects, but which can sometimes combine to be dominant. However, the more detailed the model the more information it may require before the remaining uncertainties are small enough that the results are useful. What this means to us is that the system that is attempting to survive in and interact with its environment needs multiple methods in addition to multiple resolutions for describing it, and that verification and validation are essential (i.e., every hypothesis is tested, and every conclusion is provisional).

In fact, because there are very good higher-level continuum models that are almost always adequate for describing the overall state and trends of the system's operation, it seems far more useful to implement almost all instrumentation at levels of the system's hierarchy that inform these "continuum" models. This leaves the question of how to recognize those cases where these models are inadequate, which we address by implementing and exploiting the use of reflection and situational awareness, a topic to which we return in section 3.3.3 and 3.3.5. Notice how this instrumentation and these models will correspond to modules.

Another way to approach the question of designing instrumentation is to ask if there are characteristic scales that are especially significant for recognizing emergent phenomena. Again, to take the physicist's view,

> We know now that the invisible hand that creates divergences in some theories is actually the existence in these theories of a no man's land in the energy (or length) scales for which cooperative phenomena can take place, more precisely, for which fluctuations can add up coherently. In some cases, they can destabilize the physical picture we were relying on and this manifests itself as divergences. [24]

This conception that there may be some appropriate scale at which emergent phenomena first become significant gives hope that it could be possible to build instrumentation that identifies at least some emergent phenomena. Additionally, the modularity-based approaches we have suggested as a basis for generative processes may be helpful in addressing this challenge. As generative processes develop new possibilities for the system they in effect "parameterize" their descriptions of those possibilities to give new higher-level descriptions that are exactly correlated with useful regions of the system's possibility space, and include to some degree this sense of scale. From this view, the challenge of instrumentation can be restated as how to describe the disjoint regions of possibility space that have been found to be interesting, and how to represent the possibilities within each of those regions. Just as generative processes offer enormous advantages for simplifying the process of finding "good enough" strategies quickly, we suggest that instrumentation that is linked to the structure of the system's evolving set of possibilities will leverage the same simplifications.

Linking instrumentation to the modularity of processes and structures is particularly significant in the kinds of OC systems we are advocating because their processes and structures are not fixed. Unlike the clearly delineated blocks, interface protocols, and variables of the system in figure 3.1, we expect that OC systems will have multiple and interacting processes using and affecting the same measured variables, precisely because these variables are linked to the structure of the system's possibility space. Thus, in OC systems as in biological ones, the same variables may be used in multiple models at various scales of resolution (levels of the system's hierarchy). Since we expect that OC systems will be used in changing contexts, they will also face measurement challenges related to resolution and dynamic range, as the required granularity of analysis depends on the operating context. It is therefore necessary to consider very carefully the problem of how the system gains its knowledge of internal and external events and processes, which leads us directly to instrumentation issues.

Instrumenting an OC system is in our view especially challenging because it is critically interdependent with the capability to continuously evaluate the measurements produced by that instrumentation. However, we see opportunity for addressing these very significant questions in both this interdependence and in the biological paradigm of introducing processes that create structures. The layered hierarchy and leaky, overlapping processes and structures of biological systems, together with the foundational concept that the system's structures are built from processes rather than being fixed for life as in figure 3.1 means that the system has the capability to adjust its instrumentation, or even to create new instrumentation, in response to changing needs. That is, such an approach satisfies the need for multiple descriptions coupled to verification and validation.

Such self-modification of instrumentation requires the capability to evaluate the available data to determine the system's position and trajectory within its possibility space. More specifically, we must invent new types of models that are able to continuously accept measurement data, identify and rank the importance of sources of uncertainty, and propose changes to the instrumentation that can reduce those uncertainties. This concept of dynamically integrating instrumentation and evaluation has been considered in the NSF Dynamic Data Driven Applications Systems (DDDAS) program.

> DDDAS is a paradigm whereby application/simulations and measurements become a symbiotic feedback control system. DDDAS entails the ability to dynamically incorporate additional data into an executing application, and in reverse, the ability of an application to dynamically steer the measurement process. Such capabilities promise more accurate analysis and prediction, more precise controls, and more reliable outcomes. The ability of an application/simulation to control and guide the measurement process, and determine when, where and how it is best to gather additional data, has itself the potential of enabling more effective measurement methodologies. Furthermore, the incorporation of dynamic inputs into an executing application invokes

> new system modalities and helps create application software systems that can more accurately describe real-world complex systems. This enables the development of applications that adapt intelligently to evolving conditions, and that infer new knowledge in ways that are not predetermined by startup parameters. [22]

The phrase "adapt intelligently" is significant here because it ties the changes in the system to purposes and goals. Thus, a strong guiding principle for instrumentation is that it supplies information in terms of representations of the state of the system that can be related to purposes and goals. It is this relationship of instrumentation, representations and models to purposes and goals that allows the system to evaluate their effectiveness and make choices of alternatives and refinements.

To summarize, we believe that we need to develop new types of models that support evaluative methods and processes for the OC system and the OC system engineer to shape, control, divert, and correct the generative processes through reflection on the information produced by feedback and instrumentation. We expect that multiple measurement and evaluative processes will be in operation at all times at various levels of the hierarchy of an OC system. Then reflection is used to analyze the effectiveness of the system's own processes in context, by continuously comparing these evaluations with models to estimate possible trajectories in the system's possibility space, ultimately resulting in the system choosing some combination of actions or directions that are expected to support achieving its goals and purposes.

3.3.3 Reflection and direction

Like others, we [49, 50] have argued that self-perception and self-monitoring are critical features for goal-oriented autonomous systems in order for them to move around their environments. In other words, one can imagine designing an organism or a robot with bumper-car feedback that hits a wall and stops or turns. In many ways, that can suffice for certain types of simple activities in very constrained environments like a room with four rectangular walls and a hard floor. But even in elementary creatures, such as crabs, lizards and crayfish, we see much more sophisticated adaptive mechanisms [13, 83]. Animals are very competent at knowing how high they can leap, how fast they can run, and what hiding places they can enter. This knowledge is only partly about their own capabilities, but more importantly about how their capabilities map into their environment. As Churchland said [18, page 74], "self-consciousness on this view is just a species of perception [...] self-consciousness is thus no more (and no less) mysterious than perception generally." He goes on to emphasize the considerable variety of "self-monitoring" [p. 185] that occurs at different levels. Recognition and perception of what is "oneself" and what is not oneself are difficult processes, but we readily can identify their occurrence in a number of biological systems, from single cells in immune systems [73] to

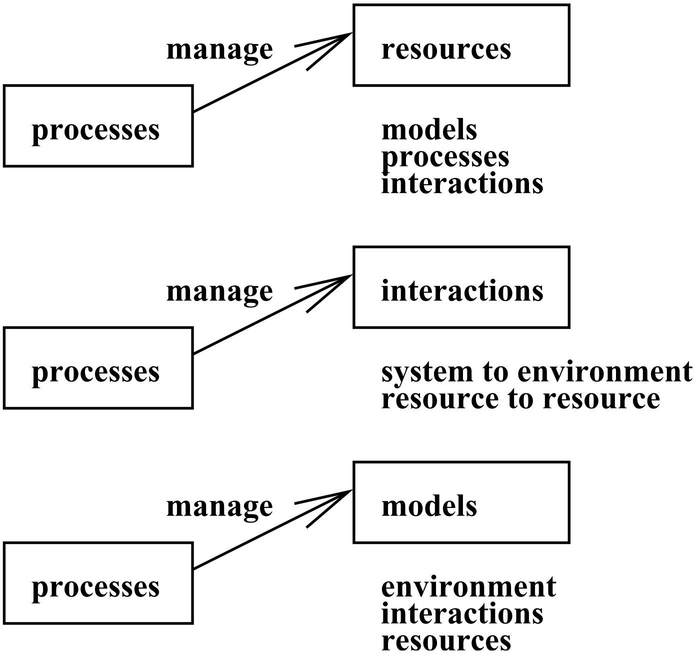

Fig. 3.2. Reflective software system processes

mammals. It is easy to imagine mechanisms that could make those perceptions available to higher level and more cognitive systems.

Self-monitoring capabilities are not the same as "self-reflection". For example, the means to monitor internal state and respond to that internal state are available to a thermostat. Ironically, although we have indisputable evidence for self-reflection in humans, our most concrete definitions of self-reflection capabilities come from the world of computer programs. Patti Maes [65] defines reflection as "the process of reasoning about and/or acting upon oneself." Practically speaking, in computers, computational reflection means having machine-interpretable descriptions of the machine's resources. We have found in our approach [42] that it is extremely useful to have not only state information available but also general meta-knowledge about the limitations and required context information for all the system's resources. There are then processes that can act on this explicit knowledge about capabilities and state in order to better control the system in its performance and maintenance [12, 42, 45]. It is clear from Damasio's discussions [19] that he is thinking of his "third type of image" as being available for both self-monitoring and self-reflection in a sense compatible with the ideas described here.

The engineering challenges we have considered so far will give us systems that discover new options (generative processes), and also measure and model their current state (instrumentation and evaluation). The next challenge that we consider is how to implement processes that generate, identify, and evaluate possible responses based on the system's reflective capabilities. These reflective capabilities will explicitly reason about what relevant information and what relevant resources the system currently has, the effectiveness of the current strategies and approaches to reaching goals, and the current shortfalls in both capabilities and information. Then other processes can, DDDAS-like, task sensors to collect additional information or creatively use combinations of existing resources, or replan current approaches.

As we have discussed in the previous section, our proposed OC systems implement instrumentation and evaluation at multiple resolutions in possibility space and in time and at multiple levels of their hierarchy, and reflection and direction must be implemented over these diverse scales as well. To use a biological analogy, reflexes supply rapid responses in situations that are likely to be critical to an organism's survival. Our systems will undoubtedly need this same "rapid response" capability, as well as a means to identify the conditions that should trigger these responses. At the same time, just as with biological systems, OC systems will need to adjust their original responses as they obtain and evaluate more information. This continual reflection at different time scales and levels results in multiple responses to a particular situation. In addition, these responses may have different levels of complexity and sophistication, as well as being operative over different time scales.

A fully reflective system has processes that collectively and cooperatively manage all of the resources, including the models, processes, and their interactions; it has processes that manage all of the interactions, including system to environment and resource to resource, at multiple time, space, and conceptual scales; and it has processes that manage all of the models, including those of the environment, the resources, and their interactions [61]. These processes are illustrated in one way in figure 3.2, and again in figure 3.3 on page 57.

The reflective capabilities provide a great deal of the information and processes that will be needed by the human developers for their monitoring and evaluation of the OC system, including relatively detailed models of what the system perceives and knows about its goals, state, environment, and options.

3.3.4 Active experimentation

Traditional engineered systems are designed to respond to changes in conditions, but biological systems exhibit a much more active style. For example, engineered systems often take advantage of relatively stable conditions by shutting off many of their subsystems to conserve energy. Biological systems, on the other hand, present a much more complex approach to the utilization of their resources: they devote some of their capacity to actively experiment with their environment, their capabilities, and their limitations. That is, biological systems perform a much more complex overall optimization strategy in choosing how to respond that recognizes the potential of self-modifying systems to find new or alternative strategies that expand its possibilities so

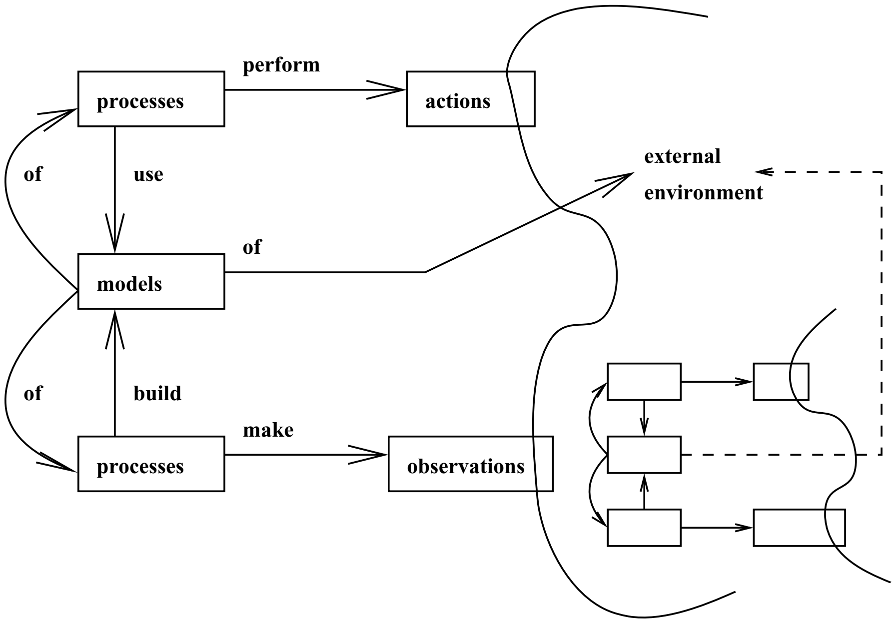

Fig. 3.3. Reflective system in environment

that it will have more options in the next "emergency". Also, if the system is already active, it may have some operating processes that can be adjusted to provide a very rapid response while other capabilities are recruited and the overall response is adjusted on a longer time scale. That is, if the system is already active, it doesn't have to start its response from scratch.

Active experimentation can be seen in part as a calibration effort in the sense that the system chooses to repeat previous actions in what it hypothesizes are similar conditions, or to perform similar actions in what it hypothesizes are the same conditions, evaluating the results in order to search for correlations and differences. Such active experimentation also provides opportunities to test the limitations of capabilities in safe situations (e.g., animal play) so that emergency responses can whenever possible be accomplished within those limitations.

Another use of active exploration is to improve models of the local conditions. Examples of this type of active exploration are well documented in the ethological literature, where an animal essentially moves randomly about its environment, discovering and building a cognitive map of features of that environment that are relevant to its size, abilities, etc.. These features can later become vital for rapid responsiveness. For example, we may observe that a lizard darts into a hole at our approach. The speed of this response is possible because of the lizard's prior knowledge of the location of the hole and that it is of appropriate size and shape for a refuge.

In another application of active experimentation, which we have called continual contemplation, the same approach is applied to the mapping and organization of the internal states and capabilities of a system. We have previously described continual contemplation as "continual exploratory data analysis not only on external or domain knowledge introduced to the system through sensors or data sources, but also continual exploratory data analysis on the system's own state and its own use of its resources as it attempts to support user requests and to solve problems." [9]

Finally, active experimentation can be applied to all levels of system processes, up to and including its language and reasoning levels. These experiments make use not only of its own capabilities, but artifacts that it may have built in its environment. For example, human beings are using computer systems to expand our possibility space by leveraging the particular capabilities of computers to perform certain kind of operations faster, more uniformly, or more often than we do.

Each of the previous engineering challenges yields system capabilities that are utilized in active experimentation: the permissive application of generative processes, the ability to measure and evaluate in new ways, reflection that gives the system access to its own structure, language processes that can express and interpret new models, variables, and control strategies, and the ability to implement multiple responses at various scales and hierarchical levels. Biological systems seem to demonstrate that using these capabilities even when they are not immediately needed to respond to current conditions has long-term advantages. That is, the potential future efficiencies gained from discovering alternative strategies, testing correlations and limits, mapping and synthesizing information on local conditions and system capabilities, and shaping operating processes rather than starting new processes in response to an emergency all seem to give biological systems an overall benefit that outweighs the additional expenditure of resources.

3.3.5 Situational awareness and context modeling

Because reactive planning and response is not always fast enough, the system can gain a great advantage by advance planning. It must be remembered that almost all advance planning is not used, although it is clearly not useless because planned and rehearsed responses are much faster than new reactions. This means that a lot of advance planning is needed, much more than reactive planning, with the corresponding advantage in viability. However, in order to do advance planning, it is necessary to construct rich enough models of the "niche" for the system, or in our case, the operational context of the performing OC system. Because the OC system is not really built from the dynamics of the world, but through our view of it, it is not sufficient only to give it models and model-building capabilities based on our current view of the environment and the composition of the system. Rather, the system will critically depend on the instrumentation we discussed earlier, as well as reflection, evaluation, and active experimentation capabilities that test the adequacy of its models within the current environment. One of the challenges, clearly exemplified by von Uexküll's story of the crab discussed earlier, is that under different modes of operation and with different goals, the same environment may in fact mean very different things.

Thus the context models for the operational environment are not completely predetermined but must rather be constructed with more attention paid to those parts of the environment needed for a given activity and with the appropriate "hooks" for monitoring the essential parameters determinable from that particular environment and relevant for the system at that time. In other words, as part of an action plan in its operational environment, a system will be dynamically recruiting not just the components to do things, but also the sensors that can provide the feedback, the analysis processes that can assess the feedback within its operational context, and the processes that reason about the feedback for its implications on the world and on the goals or performance of the system.

3.4 Representation and language

The ability of a system to use representations or even systems of representations, such as language, provides enormous advantages to the system in its capabilities. As we will show these advantages are so profound that one sees the use of representations occurring very very early in the development of living organisms. In this part of the chapter, we will first describe this early development of representation in biological systems, and then relate the use of representations and language to several critical capabilities within an OC system, e.g., representing goals and negotiation with other reasoning systems. In the last part of this section, we will describe how an OC system might start to build up its own set of representations and meaningful terminology, drawing on early work in Artificial Intelligence and describe why these capabilities are so critical to allowing us, the developers and users of an OC system, to continually monitor and shape the behavior, goals, and results of an OC system.

3.4.1 Representation in biological systems

It has not been widely appreciated in the computing community until recently just how complex biological systems are [4, 13]. In this subsection, we replay a compelling (at least for us) argument [10] that representations are the key to biological flexibility, and that they occur starting with the very smallest of animals.

Earlier, we presented evidence that unicellular animals have a rich repertoire of behaviors resulting from the coordination of body parts and internal structures. This internal coordination requires communication among the internal structures. Tomkins's [82] model of biological regulation assumes the use of internal symbols in unicellular and multicellular animals. If we ignore for a moment the biochemical details, his argument on the evolution of biological regulation is elegantly straightforward: even "ancient molecular assemblages" possessed cellular properties capable of self-replication. Nucleic acid and protein synthesis are endergonic reactions; hence primordial cells were required to capture energy from the environment. However, changes in the environment that diminished the supply of monomeric units necessary to polymer synthesis or altered the formation of adenosine tri-phosphate (ATP, an essential component of energy management and metabolism) were probably lethal. Therefore, survival would require regulatory mechanisms that maintain a relatively constant intracellular environment.

Tomkins divides this biological regulation into two modes. In simple regulation there is a direct chemical relationship between the "regulatory effector molecules" and their effects. As examples, he cites enzyme induction, feedback inhibition of enzyme activity, and the repression of enzyme biosynthesis. The critical point here is that in simple regulation, the control of the internal environment is tenuous at best, since the regulatory molecules are themselves important metabolic intermediaries. Therefore the animal's internal environment is still closely tied to the availability of essential nutrients. In complex regulation, there are metabolic "symbols" and "domains". To quote Tomkins [82, page 761], "The term *symbol* refers to a specific intracellular effector molecule which accumulates when a cell is exposed to a particular environment." As two examples, he cites adenosine 3'5'- cyclic monophosphate (cAMP), which in most microorganisms is a symbol of carbon source depletion, and guanosine 5'-diphosphate 3'- diphosphate (ppGpp), which is a symbol of nitrogen or amino acid deficiency. Importantly, "metabolic symbols need bear no structural relationship to the molecules that promote their accumulation in a nutritional or metabolic crisis ... cyclic AMP is not a chemical analog of glucose." [82, page 761] Tomkins also points out that metabolic lability is another attribute of intracellular symbols that allows their concentrations to fluctuate quickly in response to environmental changes. However, note that this lability is different from the troublesome lability of the simple regulation mechanisms. In the case of simple regulation, since the regulatory molecules are themselves metabolic intermediaries, they (and hence the internal environment) will fluctuate in a direct manner according to the supply of external nutrients and conditions. However, in the case of complex regulation, the symbols will respond rapidly to the external environment, leaving protected for some time the metabolic processes they control. This protected time is exactly the time in which the organism has the chance to make some adaptive response to the environment (e.g., swim away from the carbon-depleted region), and this, it turns out, is exactly what bacteria do. For example, carbon-starved *Escherichia coli* develop flagella, which allow the bacteria to be motile; cAMP is critical to the development of flagella. By incorporating a symbol "level" the animal gains time in which it can protect its metabolic processes from external conditions. In Tomkins's examples of the necessity of cAMP to the development of flagella in *E.coli*, we see that the effects controlled by the symbol are not all metabolic but also include adaptive behavioral responses that will protect the metabolic processes. He also points out that many symbols may share in the control of a given process.

Later he extends his argument from single cells to multicellular animals and uses the slime mold, *Dictyostelium discodium*, as a model of transition of intracellular symbols to intercellular symbol use. In the slime mold, the cells exist as independent myxamoebas until starved. At this point, cAMP accumulates in the cells, similarly to *E.coli*, as a symbol of carbon depletion, but unlike *E.coli*, it is also released from the cells into the external medium where it acts as the attractant that causes myxamoebas to aggregate into one multicellular slime mold. As Tomkins states [82, page 762], "Cyclic AMP thus acts in these organisms both as an intracellular symbol of starvation and as a hormone which carries this metabolic information from one cell to another." But, as noted earlier, cAMP is labile and therefore, Tomkins argues, is not suitable for the long distance required for intercellular communication in large metazoa. He proposes that hormones, more stable chemical compounds, took over the role. As he emphasizes, the process in intercellular communication always begins and ends in the internal primary codes of individual cells.

Just as in the case of internal communication processes of unicellular animals, the intercellular communication processes of multicellular organisms are symbol-based. Note how these internal communication processes, in both unicellular and multicellular animals, make possible behavior or coordinated goal-directed movements. Movement is fundamentally a cooperative phenomenon, requiring communication among the organism's parts. As we saw in the example of motility in *E.coli*, even the most primitive movement is controlled and mediated by the use of symbols.

In other words, in the unicellular animal, we have a collection of symbols, like cAMP, which together with the way they affect the processes under their control and the way this collection of symbols affect each other, constitutes a primitive brain without nerves. This primitive brain without nerves is elaborated in multicellular animals in two ways. (1) The labile symbols of the unicellular animal are replaced by hormones, which are more stable chemical symbols, and by nerves, which provide more specific routes of information than chemical diffusion. (2) Layers of symbols develop to the point that "domain" (in Tomkin's sense) becomes not a set of body processes but rather a set of brain processes.

As stated in [10, page 918],

> An increased use of symbols disassociates the intracellular processes of unicellular organisms from the environment. This means that an event

> which occurs in the receptive space of the organism does not produce an immediate response. What it produces is an internal reaction that symbolizes the event in the environment. These symbols of external events then become part of the internal processes of the organism. The more an organism has the ability to symbolize external events and the greater its capacity to manipulate those symbols internally, the more it is freed from non-adaptive, direct responses to fluxes in the energy and matter surrounding it. It begins to have the capability to organize delayed actions, which give it the freedom to plan, simulate, and act when its own internal processes deem it appropriate; such actions can take place at greater and greater distances in time and space from the initial external event.

This "disassociation" is critical to representation, but also to the reflective processes described earlier. The separation enabled by the disassociation of symbols allows the time and the freedom from external "realities" within which the system can essentially "simulate" or do what-ifs and other types of reasoning about possibilities, e.g., different behaviors, different contexts, different results, before committing itself to the actual energy to perform actions.

Clearly, there is a careful and context-dependent trade-off between the *timeliness* of rapid responsiveness and the *timelessness* permitted by representation for reasoning and reflection. Biological systems typically begin to address this trade-off by taking advantage of the multiple layers of their responsiveness; while some layers of a system are doing immediate actions, other layers are doing longer term reasoning processes. This is clearly seen in the crayfish emergency "tail flip", which is one of the most rapid responses known: it is just 10 ms from the stimulation of the flight response to an undirected flip. Meanwhile, a slower system within the crayfish is carefully figuring out a trajectory for swimming away from the possible threat that led to the emergency response.

This trade-off between timeliness and timelessness is also seen in the generative processes, discussed in section 3.2.3.2, where reasoning and reflection are part of the system's ability to assess and alter its rapid construction and maintenance of current configurations or assemblages of components. That is, the generation of assemblages does not have to be perfect, but rather because of adaptive processes can be generated and then fine-tuned rapidly, depending on the changing circumstances and changing needs of the system. These adaptive processes clearly require reflection and reasoning. This point is especially relevant to "substitutability", which is the ability of a system to rapidly adapt by changing the processes and structures used to accomplish a given purpose or goal.

This brings us to the key topic of representing purpose, goals and meaning in OC systems. These terms are not intended to imply anything about consciousness or even awareness on the part of the OC system. Rather, through explicit or implicit means the system must represent goals in order to evaluate the results of actions.

3.4.2 Purpose, goals and meaning

Part of the adaptive behavior of biological systems is seen in the sophisticated capabilities that animals have for developing and encoding "meaningful" representations about their environment and their own states, and for developing processes using these representations to plan actions that achieve desirable states. These "desirable states" will always be situated. In other words, they will always include a combination of features in the world in relationship to features exhibited within the biological system. For example, in Tomkins' examples above the feature of carbon availability in the external world was being represented in single-cell animals, as well as their own state of energy availability. These desirable states are the lowest level of "purpose" and "goals", and as pointed out in the section above, like other representations can become increasingly separated in time and space from external events. That is they can become increasingly "abstract", and with abstraction, impact more diverse parts of the total system.

Animals show sophisticated abilities to represent and process the actions needed to support diverse goals. One such ability is to satisfy multiple goals with a single course of action. This is called merging and was studied by one of the authors because of its possible use as a source of variation to help explain the amazing flexibility of responsiveness in biological systems. But it also has profound implications for the underlying adaptive decision processes available to animal systems and the way that goals are represented. One of the consequences of merging is that there can be multiple goals for any action and multiple actions for any goal. A given instance of behavior can reflect several motivations and work toward several goals at once.

Contrary to the usual emphasis in behavioral studies, in which an animal must choose between mutually exclusive acts, an animal in nature is rarely in the situation where it must engage in one behavior to the exclusion of other behaviors. Rather, an animal's movement frequently shows "behavioral merging", in which several motivational goals and action patterns are combined into one coherent pattern. In studying the merging of feeding and aggression behaviors in the lizard [4], an animal noted for the rigidity of its behavioral patterns, Bellman found that when elements of feeding and aggression conflicted, other elements were selected and substituted, so that, overall, both feeding and aggressive patterns were combined into one fluid behavioral sequence. The behavioral sequence resulting from merging points to a particular type of flexibility in a movement system. A specific movement pattern can subserve a number of goals. If this is so, then a specific movement pattern is not necessarily linked to one goal any more than to any other goal (although there may be some kind of weighting, so that a given behavior is most often associated with one particular goal). This implies that a movement is not "released" as a necessary consequence of the occurrence of a particular motivational goal; rather it is "recruited" to serve that goal. Furthermore, from behavioral merging, we see that a whole action pattern need not be recruited but only those elements best fitting the circumstances.

Clearly, these abilities have advantages both in terms of expended effort and in terms of rapidity of response. However, as in many other qualities of biological systems, it is efficiency of a peculiar type. It is highly efficient in allowing future adaptiveness and in robustness, but not efficient or optimal for any single given set of actions. Part of the reason for this style of efficiency may have much to do with the type of complex multi-criterion optimization within a rapidly changing environment required by biological systems. That is, in conventional man-made systems, which are engineered to optimize their performance within carefully specified environments, one can develop planning processes that optimize the performance given a fairly fixed set of criteria. The emphasis can thus be on the efficiency of the fixed course of action. In biological systems, and in the systems we are trying to invent in OC, the complexity of the system's interactions and requirements and the changing environment require an emphasis on the ability to rapidly adapt and hence to change course or replan. This adaptiveness requires all sorts of properties that in single-purpose systems are redundant or excessive.

We have already noted in this paper many capabilities which support the ability of a system to change course and replan, including generative and opportunistic processes, reflection, and active experimentation. However, in order to develop appropriate evaluation methods for choosing the best solutions among combinatorial possibilities and to deal with the synthesis of the information we will have in the necessarily explicit models of an OC system, we will also need many new ideas on what we mean by optimization or even satisficing in these systems. As difficult as it will be to represent goals, it is even more difficult to state the evaluation criteria that will determine "goodness" and "fitness" for the OC system.

One shift will be away from any long-term or overall optimization or satisficing and toward strategies of local and short-term optimization with methods designed to rapidly capture and summarize the wide-spread impact of decisions. Having layers of relationships will help because small continuous change at one layer can have much wider and diverse impacts that can be monitored for from the standpoint of different levels of recruited modules.

It should be noted here that adaptation is not like optimization, and is not usefully implemented by optimization. It is always just satisficing, not optimizing, and usually the time constraints on decision making mean that even formal satisficing is not possible either, so some combination of experience and guessing is needed.

3.4.3 Negotiation with other reasoning systems

So far we have been emphasizing the importance of representations in the ability of animal systems to regulate themselves and then as a continuation of that self-regulation, to represent and reason about their goals and to adaptively plan actions in a dynamic environment. As is hinted at in Tomkins's examples of the slime mold, the same set of internal self-regulatory symbols that are used intercellularly to coordinate the actions of populations of cells can also be, by many mechanisms, made visible to other organisms in order to coordinate their behavior at the population level. Hence the symbols excreted, secreted, vocalized, enacted, etc., allow animals to coordinate their mating, hunting, fighting for territory, learning, and many many other types of needs for communication and coordination. That the animal kingdom displays such diversity in the types of symbols, the reasons for symbols, and the mechanisms for conveying symbols speaks to the enormous importance of shared representation in adaptive complex systems. Interestingly, in the animals with increasingly complex reasoning capabilities there is a correspondingly increasing complexity of communication capabilities, including those displaying the nuances of emotional state (which contains a wealth of information about the motivational state and likely goals and intentions) and social needs for negotiation and coordination. It is our belief that OC systems, because of their complexity and because of our need to monitor and shape their behaviors, require similarly sophisticated negotiation and coordination capabilities.

As we describe the needs of representation and communication in OC systems, it is important to note that in a complex system:

- there is a time delay, both for making up or choosing the information to convey, and for getting the information out; hence the need for some autonomy, and
- there is often a difficulty in characterizing complex states; think for a moment of how difficult it is for you, as a human, to describe your unobservable symptoms to a medical doctor.

These delays are one source of emergence: if there are arbitrary time delays in a communication process, or other feedback process, then that process can exhibit instabilities and emergences. However, communication is so important to both the individual and the group that there are multiple, overlapping, and even redundant symbols, representations, and modalities in order to ensure that critical information is conveyed.

One of the advantages we have in biological systems is the situatedness of our systems in the physics of the world and the common evolution among members of a species and even among all mammals. We understand because we are. Much of our ability to reason from the outside about another human's state and meanings has a lot to do with our commonalities as humans. For example, consider the deep problems that autistics have in communicating as an example of how even minor human variations can have vast impact on our abilities to negotiate and communicate. On the other hand, there are some vocalizations that, like reflexes, act as immediate emergency responses: it is a curious feature, especially in mammals, that distress calls and other broadcasts are occasionally so important that they can be recognizable across species.

In OC systems we will be building an "alien" system with no shared evolution, with little shared constraining world (the dynamics, etc., will be different), and, potentially, little overlap in our operating environments. As we develop and use language in an OC system, we might want to consider how to develop co-evolutionary strategies, recognizing that both we and our systems will be changed by that co-evolution.

The important question here is how we will create livable systems. We certainly need to understand how an OC system negotiates with its human engineers and builds up a common set of symbols, etc., so that its communication may be understood and its effects will be appropriate. Here there are two meanings of "livable": for the system to be able to live in its environment and for us to be able to live with it. We mean both.

3.4.4 Use of language

An OC system needs to model its surroundings and its own behavior for self-assessment and self-improvement. Since the designers cannot know everything about the system's environment and development, the system will have to create new models or modify existing ones. It will therefore have to have ways to assess the efficacy of its models, and change them as it deems necessary.

More fundamentally, the languages in which the models are written may not be adequate for all development paths, so the system will also have to create new kinds of models and new languages in which to define them, and sometimes re-express its older models and processes in the new language.

We write "languages" in the plural because we do not believe that any one modeling language or paradigm can be sufficient, even in principle, to model all relevant or important aspects of a complex environment [14]. We therefore advocate the use of a collection of "little languages", instead of trying to fit everything into one big one. Of course, the collection of "little languages" has multiple underlying assumptions, and this multiplicity requires some integration process, but we have developed an integration mechanism that is well-suited to complex system integration, called "Wrappings", described in section 3.4.6 below.

We have argued here that the creation and use of language internal to the system is fundamental to the success of OC systems in complex environments. The study of this symbolic aspect of systems design and operation is called "Computational Semiotics", and it lies at the intersection of the edges of mathematics, linguistics, philosophy, logic, and computation [43, 44]. It is about the creation and use of symbol systems by constructed complex systems, but we are trying to push it much farther in the direction of interaction with human language, since the system has to tell us about itself.

A nice introduction to the general topic can be found in [25] (if you can ignore the intrusions of justifiably annoyed comments about transformational grammar), with some comparisons to the early writings in language [76] and philosophy [85, 86]. There are also other approaches based in logic [79] and computation [70]. Another description from a different viewpoint can be found in [17].

Our attention to the use of language includes whatever programming or specification notations are to be used, since they are almost always too precise for what the designer knows about a system, so they require the designers to make too many decisions before it is possible to know enough to make those decisions properly [77].

We intend that the system will help the designers create the language, by operating for a while, so that the system can know enough to make some good choices, and that it can present enough information to the designers so that they can make other good choices.

In the most general terms, we can describe the operation of such a system as follows:

- system observes external and internal behavior
  - developers must provide initial languages
  - system use languages to record these observations
  - system assesses the adequacy of its own languages
  - system changes the languages or invents new ones as necessary
  - the process cycles back to the system's use of languages
- system creates models
  - developers must provide initial notations
  - system uses notations to record these models
  - system assesses the adequacy of the notations
  - system changes the notations or invents new ones as necessary
  - the process cycles back to the system's use of notations
- system inherits or creates goals
  - developers must provide initial goals
  - system reasons about the models in pursuit of its goals
  - system assesses the adequacy and consistency of the goals
  - system changes or replaces the goals as necessary, according to the results of negotiations with developers
  - the process cycles back to the system's use of goals

To describe these processes in more detail, and to explain how we expect these systems to work, we start with our emphasis on context, then proceed to symbol systems, language, and models. We end with a discussion of our progress and prospects for OC systems.

3.4.4.1 Communication and cooperation

A significant aspect of how we will cooperate with our complex computing systems is our ability to communicate with them and their ability to communicate with us. To that end, and since we are no longer expecting to design every aspect of these systems, they will have to be able to use symbols of their own devising, created to represent some meaning significant to them, which must also be conveyed to us. There are examples of early artificial intelligence systems that generate symbols to represent things that they have learned about the world and their own internal states [31, 72].

These systems will have to make new models of both their environment (context) and their internal processes and states, evaluate the effectiveness of those models, and revise them or build new models again as necessary. These systems will also have to communicate their internal models to us, so we can monitor their actions and predict their expected activity to ensure it is in line with our intentions for the system.

This is why we emphasize the semiotics of these systems; if we can understand enough about the processes of language formation and use, we can design systems that will be able to explain themselves to us. To do that, we need to understand the symbol systems, what they are used for, how they are defined (whether by the designer or internally), how a system can evaluate them in the context in which they are being used, and how a system can change its symbols appropriately and tell us what it did.

3.4.5 Symbol systems and representational mechanisms

We start with a description of our approach to representational systems, and show how engineered systems can be expected to create and use them.

It may seem as though we are starting from "too far back" in the design process, namely before the domain is well understood, but in our opinion we must start there, because the different philosophy on adaptation that we have developed above requires fundamental changes in the nature of our computing systems and devices, and the development processes that lead to them.

Besides, it is our opinion that every complex system design process actually starts (and usually finishes) before the domain is well understood, often long before, even though that fact is not generally known in advance (though we claim that it could and should be expected).

For us, a representational mechanism is the same as a modeling mechanism. That is, any computational scheme that derives a computationally accessible object (or process) to represent some phenomenon of interest (either external to the system or internal) is a representational mechanism.

The modeling scheme is better if the model is better. The better models capture more properties of the phenomenon (or at least more of the properties important to the modeler), and the representational mechanism is better when it can represent more phenomena of interest to the system.

There must be processes that identify the phenomena and create the models, processes that can transport the resulting models in time (memory) or space (communication), and processes that analyze the models for making decisions or convert them to other forms for further analysis (translation).

Symbol systems are one kind of representational mechanism, chosen for simplicity and ease of computation. A symbol system consists of a finite set of basic symbols and a finite set of combination methods. The analogue is a phrase-structure grammar with constraints [26].

There may be different types of symbols and structures, and different kinds of combination methods, but it is important that they be finitary, which means that each combination method can only combine a fixed number of structures at each use (each combination method can have various kinds of restrictions on what structures it can combine, which we take to have the power of context-sensitive grammars). All structures in a symbol system can be pictured as finite trees, and the combination methods are ways to combine trees into larger ones.

The “get-stuck” theorems tell us that the systems need to be able to evaluate and adjust not only their models, but their basic symbol systems and modeling mechanisms [53, 47]. Basically, for any given symbol system, we can compute the maximum number of expressions of each length, and then argue that adequate modeling of more complex environments eventually leads to expressions that are too large to process quickly enough: the system “gets stuck”, and the only way out is to change the symbol system.

We also want to make it very clear that symbol systems are just one kind of representational mechanism, one that we have chosen because they are easy to use and analyze, not because they are necessarily the only or even the best choice for all modeling problems.

3.4.5.1 Language formation

As the system operates in a complex environment, it gathers information about what choices it has made and what activities it has seen in the environment and in its own internal operation. It is our intention to have these systems create private descriptive language for their own use, and explain it to their human users. To that end, the systems have to have many more empirical modeling capabilities than usual. As a first step, one can do very simple, straightforward syntactic analyses of language and language use [38]. With such methods, we have shown how language formation might occur, with identifying common or replicated patterns in the structures, the processes, or in the relationships [48].

Here the system can do some empirical invention. It can accumulate commonalities and replications of structure and process, in context. It can accumulate commonalities and replications of descriptions and relationships, and it can assign symbols to those clusters and recognize them when they occur again, gradually describing more over time.

Digital computers can only do three things: move and copy data, compare data, and interpret limited range models of arithmetic (computers do not do arithmetic). All of these operations are entirely syntactic. In a sense, we are trying to make these devices compute with semantics, which means defining syntactic representations of semantics relationships, and computing up the meaning hierarchy from data through information to knowledge [39].

For example, there is a kind of abstraction that is part of "continual contemplation" in reflective systems [46]. Any process or any structure can be decomposed, the parts abstracted with an attached context of their use in the combination, and then reassembled and reintegrated computationally. The parts then become process or structure components that can be put together in other ways, with other components, according to their Wrappings. This kind of abstraction occurs very often in mathematics, as proof steps become methods and sometimes subjects in their own right.

3.4.5.2 Model evaluation

After a system has built models, then it needs methods to assess them, and, if necessary, replace them. In particular, a system needs to be able to determine that a model is inadequate.

This process is called model-deficiency analysis, and it proceeds from two sources of information: intentional goals and observed behaviors, and most particularly, from places in which the behaviors do not match the goals. These will be described in terminology that is internal to the system, which also means that the language used must be adequate to describe them. This is another force towards development by the system of multiple little languages and also of better languages.

The notion of allowable variation, when applied to language, means to us that the system should use several different sets of foundations simultaneously, that is that the internal languages occur at different resolutions, with different local contexts and different interactions, so that their efficacy for particular problems can be compared. This choice is already known to be important to simulation systems [20, 68], at least at the level of temporal and spatial resolution. We assert that it is equally important in other domains.

Computational reflection offers important advantages in the evaluation of models. However, because reflection will explicitly represent system processes and structures, we encounter an interesting dualism. As soon as processes are made explicit, they become descriptive structures, and as soon as structures have interpreters, they become processes. This dual view allows both kinds of things, that is, both descriptive structures and processors, to be processed in different ways for different purposes.

3.4.6 Progress and prospects

In this subsection, we describe what we can do now in terms of realizing our (admittedly extremely ambitious) concepts for representations, symbol systems, languages, and model construction and evaluation.

Once we find the right mechanisms, we can implement these systems using our Wrapping approach to integration infrastructure. Wrappings also provide several useful notions for implementing and combining the little languages noted above. The *Problem Posing* programming paradigm [51] separates the *information service requests* in a program from the *information service providers*. It is always clear which is intended, and the distinction is known to the compilers and interpreters of the notation. We have shown that it applies to programming notations from all of the major programming paradigms: imperative, declarative, relational (constraint), functional (applicative), object (message), and others.

For example, function definitions are information service providers and function calls are information service requests. We normally associate the two by using the same name, but the names are in completely different name spaces. The Problem Posing interpretation allows us to break the direct association and reconnect them in much more interesting and flexible ways. We define *problems* as information service requests, and *resources* as information service providers, so that we can treat them separately. In particular, we can then study the notion of a *problem space* as an explicit representation of the goals and purposes of various processes in a particular application domain, without needing to specify *a priori* how those problems are to be addressed [14, 54].

One of the more interesting ways to connect problems to resources is with *Knowledge-Based Polymorphism*, that is, with a knowledge base that maps problems into resource uses in context. Our Wrapping approach to integration infrastructure takes this mapping as fundamental.

We start with the widely observed notion that declarative knowledge has the advantage of being analyzable. But declarative knowledge does not *do* anything; it needs an interpreter, and we need to make those interpreters explicit for study. The Wrapping approach is based on these two fundamental aspects of computation in constructed complex systems, the descriptions and the interpreters:

- *Wrapping Knowledge Bases* (WKBs) describe the uses of all resources, not just how, but also whether, when, and why to use a particular resource in a particular context.
- *Problem Managers* (PMs) interpret the WKBs to select and apply resources. PMs are also resources, are also Wrapped, and therefore also selectable.

Such a system has no privileged resources at all. Any part of the system can be replaced (actually superseded) by a corresponding provided part. This flexibility allows all of the integration processes to be studied in the same system.

We mentioned above in the language summary that we expect the OC system to use a multiplicity of languages, and hence the Wrappings will have to integrate a multiplicity of languages. What is needed is to interpret the models or other situations in which language fragments are used as problems requiring certain semantic information, and the language fragments as resources providing some semantic information, and the Wrappings as a knowledge-based connection from needed to provided semantics.

We have described a Wrapping-based architecture for systems that have models of themselves [52, 61], which they can use to examine and change their own behavior. These systems have descriptive models of every process in them, and interpreters to produce that process behavior from the descriptions. The interpreters are also processes, and also have descriptions, so the system is completely self-describing.

We have built systems using Wrappings for small but difficult integrations, as well as for larger systems. One example was a system with 48 resources for evaluating the effects of new technology insertion into a situation management and rapid response system. The general point being made here is that even fairly complicated systems can be put together rapidly with Wrappings, provided that they and their expected behavior are well-understood. We have also built systems with models of (some of) their own behavior, and systems that create their own symbol systems (a word identification program using grammatical inference [3]), but not (yet) reflective ones. We have built systems that re-express parts of themselves, but so far only the descriptions, not (yet) the processes.

These applications lead us to believe that we can implement some of the appropriate computational resources for OC systems, including some version of the following capabilities, which will allow systems to

- manage their own computational processes. Wrappings identify the resources they have and the classes of problems they address.
- manage their own modeling processes [5, 41].
- are partially self-modeling and self-modifying, as shown in the previous section.
- manage their own symbol systems and invent new ones.

The hardest part of this approach is the symbol creation necessary for this kind of constructive semantics, that is, how a system can represent the connections between purely syntactic data (which is all that computers can contain) and semantic meanings (connections to the phenomena).

That boundary is the fundamental phenomenon in the use of symbols, often called the symbol grounding problem [29], which has generated a large amount of discussion, including claims that it is already solved [80]. In our view, for OC systems, the designers make the first choices of symbols, and that reduces the importance of this problem.

In any case, this is where all symbolic processing starts: feedback loops and other stable structures and correlations lead to symbols, e.g., a system moves and it sees a motion, or it speaks and it hears a sound, or any of a number of connections from simple outputs to multiple inputs.

In the absence of a good solution for this (hard) problem, all is not lost. We can still proceed under the assumption that our system has a fairly limited set of different kinds of actions that it can perform, and a limited set of different kinds of events or activities that it can recognize.

We will not expect the system to create new kinds of external interfaces, though it may be able to create new instances of many kinds of interface. We will expect the developers to provide a rich set of initial interfaces, so that the system has enough information to study its environment and enough capability to affect it appropriately.

The key to building these linguistically capable systems is to provide the right set of language producing and modifying mechanisms. Empirical statistics has discovered a large set of notations and methods that are useful in this regard, for representing and understanding large sets of correlated time series, and we expect each application domain to have its own special methods also. More methods and more different kinds of methods are needed, though, and we expect that developing the new methods needed for this detailed level of language use within the system is still hard. Integrating these methods is a challenge well addressed by Wrappings, and model-deficiency analysis holds great promise for future developments.

3.5 Conclusions

In this chapter we considered three major and mutually reinforcing types of developments in a successful adaptive system: The first is creating the "possibility space". This possibility space includes much more than the history and development of an individual or even a population of individuals; in fact, for biological systems the possibilities start in the physics of the environment which will become the system's ecological niche. The second set of developments could be thought of as creating processes that both enlarge and constrain the shape of this possibility space. The last set of developments is the more traditional concern of adaptive systems research: the control processes that enable the system to navigate through the possibility space. That is, given its goals and its current state within the possibility space, what exactly can the system perceive (of its possibilities), what can the system control, and what can it do.

In the case of each of these major sets of developments, we first described what we consider to be biological versions of these processes, and then what that might imply in terms of engineering OC analogs of such processes. But in addition to creating analogies to existing biological processes, we also discussed some unique challenges for OC systems; the greatest of which is that these systems must always be accessible, monitorable and coordinated with our goals and intentions for the systems. This implied to us that OC systems require sophisticated instrumentation, self-monitoring and reflection capabilities as well as the ability to represent their states to us, communicate and negotiate with us, and hence share the development of its control and organization with us.

Aside from the small progress we have made, particularly on reflective processes described in the previous section, we feel that OC in order to make progress critically needs to develop several capabilities as a community of OC researchers. That is, in the systems engineering challenges we discussed the issues involved in exploring the possibility space in OC systems, and some of the strategies that a biological system uses to do its explorations. Two of the foremost strategies of biological systems are 1) to use a population of individuals to explore formidably large possibility spaces and do so with active experimentation, and 2) provide the means for communication and coordination among the entire group so that the perceptions and experience of individuals can be combined. So can we not in fact treat the community as a population of organisms determined to explore the possibility space of OC systems? And could we not, with coordinated efforts, act like an active experimentation process, carefully correlating across our different experiences? Of course in order to do this we need to deal with the reproducibility of our results within OC systems and we will need to work very hard on building up much better models of the goals and operational contexts for our individual demonstrations. Lastly, we would all, as a community of researchers, benefit from whatever methods are developed to give us overviews in many different ways of these complex systems.

References

1. Harold Abelson, Gerald Sussman, with Julie Sussman, *The Structure and Interpretation of Computer Programs*, Bradford Books, now MIT Press (1985)
2. Lada Adamic, "The Small-World Web", on the Web at URL "http:// www. hpl. hp. com/ research/ idl/ papers/ smallworld/ smallworldpaper. html" (availability last checked 15 July 2007)
3. Dana Angluin, Carl H. Smith, "Inductive Inference: Theory and Methods", *Computing Surveys*, Volume 15, Number 3, pp. 237-269 (September 1983)
4. Kirstie L. Bellman, *The Conflict Behavior of the Lizard, Sceloporus Occidentalis, and Its Implications for the Organization of Motor Behavior* (PhD Dissertation), 225 pages, University of California, San Diego (1979)
5. Kirstie L. Bellman, "An Approach to Integrating and Creating Flexible Software Environments Supporting the Design of Complex Systems", pp. 1101-1105 in *Proceedings of WSC'91: The 1991 Winter Simulation Conference*, 8-11 December 1991, Phoenix, Arizona (1991)
6. Kirstie L. Bellman, "Developing a Concept of Self for Constructed Autonomous Systems", pp. 693-698, Volume 2 in *Proceedings of EMCSR'2000: The 15th*

European Meeting on Cybernetics and Systems Research, *Symposium on Autonomy Control: Lessons from the Emotional*, 25-28 April 2000, Vienna (April 2000)
7. Kirstie L. Bellman, "Emotions: Meaningful Mappings Between the Individual and Its World", (to appear) in R. Trappl, P. Petta (eds.), *Emotions in Humans and Artifacts*, MIT Press (2001, expected)
8. Kirstie L. Bellman, "The Challenge for a New Type of Computational Semiotics: The Roles and Limitations of Diverse Representations in Virtual Worlds", *Proceedings of VWsim'01: The 2001 Virtual Worlds and Simulation Conference*, *WMC'2001: The 2001 SCS Western MultiConference*, 7-11 January 2001, Phoenix, SCS (2001)
9. Kirstie L. Bellman, "Peacemaker 2020: a System for Global Conflict Analysis and Resolution, a Work of Fiction and a Research Challenge" in Robert Trappl, ed., *Programming for Peace: Computer Aided Methods for International Conflict Resolution and Prevention*, Springer Dordrecht the Netherlands 2006.
10. Kirstie L. Bellman and Lou Goldberg, "Common Origin of Linguistic and Movement Abilities", *American Journal of Physiology*, Volume 246, pp. R915-R921 (1984)
11. Kirstie L. Bellman, Christopher Landauer, "Integration Science is More Than Putting Pieces Together", in *Proceedings of the 2000 IEEE Aerospace Conference (CD)*, 18-25 March 2000, Big Sky, Montana (2000)
12. Kirstie L. Bellman, Christopher Landauer, "Towards an Integration Science: The Influence of Richard Bellman on our Research", *Journal of Mathematical Analysis and Applications*, Volume 249, Number 1, pp. 3-31 (2000)
13. K. L. Bellman and D. O. Walter, "Biological Processing", *American Journal of Physiology*, Volume 246, pp. R860-R867 (1984)
14. Richard Bellman, P. Brock, "On the concepts of a problem and problem-solving", *American Mathematical Monthly*, Volume 67, pp. 119-134 (1960)
15. Patrick L. Barry, "Good Vibrations", NASA News 02 November 2001, on the web at URL "http:// science. nasa. gov/ headlines/ y2001/ ast02nov_ 1. htm" (availability last checked 02 June 2007)
16. Braitenberg, V., *On the Texture of Brains: an Introduction to Neuroanatomy for the Cybernetically Minded*, New York, Springer-Verlag, 1977, p 121.
17. Daniel Chandler, "Semiotics for Beginners", on the Web at URL "http:// www. aber. ac. uk /media /Documents /S4B" (availability last checked 02 June 2007)
18. Paul Churchland,. *Matter and Consciousness*, Cambridge, Mass.: The MIT Press, New York: Avon Books (1984)
19. Antonio Damasio, *Descartes' Error: Emotion, Reason, and the Human Brain*, New York: Avon Books (1994)
20. Paul K. Davis, "An Introduction to Variable-Resolution Modeling and Cross-Resolution Model Connection", pages 1-43 in [21]; revised version in RAND report R-4252-DARPA, RAND Corp. (1993)
21. Paul K. Davis, Richard Hillestad (eds.), *Proceedings of DARPA Variable-Resolution Modeling Conference*, 5-6 May 1992, Herndon, Virginia, Conference Proceedings CF-103-DARPA, RAND Corp. (March 1993)
22. "DDDAS: Dynamic Data Driven Applications Systems", on the web at URL *http://www.nsf.gov/funding/pgm_summ.jsp?pims_id=13511* (availability last checked 05 July 2007)
23. Michael W. Deem, "Mathematical adventures in biology", *Physics Today*, January 2007, pp. 42-47.

24. Bertrand Delamotte, "A hint of renormalization", *Am. J. Phys.* vol. 72 (2004) 170-184
25. John M. Ellis, *Language, Thought, and Logic (Rethinking Theory)*, Northwestern University Press (1993)
26. J. M. Foster, *Automatic Syntactic Analysis*, American Elsevier (1970)
27. Martin Gardner, "The fantastic combinations of John Conway's new solitaire game 'life' ", *Scientific American*, Vol. 223, pp. 120-123 (October 1970)
28. A. Garfinkel, M.L. Spano, W.L. Ditto, J.N. Weiss. "Controlling cardiac chaos", Science 257 (5074), pp. 1230-1235 (Aug 28 1992)
29. Steven Harnad, "The Symbol Grounding Problem", *Physica D*, Vol. 42, pp. 335-346 (1990) on the web at URL "http:// www. ecs. soton. ac. uk/ harnad/ Papers/ Harnad/ harnad90. sgproblem. html" (availability last checked 02 June 2007)
30. Jennings, H. S., *Behavior of the Lower Organisms*, Bloomington, In., Indiana Univ Press 1906 (Republished with new foreword by C. King, 1976)
31. L. Kaelbling, M. Littman, and A. Cassandra, "Planning and Acting in Partially Observable Stochastic Domains", *Artificial Intelligence* 101(1-2): 99-134, 1988.
32. Catriona M. Kennedy, "A Conceptual Foundation for Autonomous Learning in Unforeseen Situations", Tech. Rpt. WV-98-01, Dresden Univ. Technology (1998); also on the Web at URL "http:// citeseer .ist .psu .edu /kennedy98conceptual .html" with abstract at URL "http:// citeseer .ist .psu .edu /kennedy98conceptual .html" (availability last checked 02 June 2007)
33. Catriona M. Kennedy, "Distributed Reflective Architectures for Adjustable Autonomy", in David Kortenkamp, Gregory Dorais, Karen L. Myers (eds.), *Proceedings of IJCAI-99 Workshop on Adjustable Autonomy Systems*, 1 August 1999, Stockholm, Sweden (1999); a similar note is on the web at URL "http:// citeseer .ist .psu .edu /kennedy99distributed .pdf" with abstract at URL "http:// citeseer .ist .psu .edu /251251 .html" (availability last checked 02 June 2007)
34. Gregor Kiczales, Jim des Rivières, Daniel G. Bobrow, *The Art of the Meta-Object Protocol*, MIT Press (1991)
35. Wolfgang Köhler, *Intelligenzenprüfungen an Anthropoiden*, (1917), translation *The Mentality of Apes* (1925)
36. Kevin B. Kreitman, "Kreitman's Conjecture and the Law of Requisite Heterarchy", *Symposium on Cybernetic Factors for Socio-Economic and Management Framework*, 14th International Congress on Cybernetics, 21-25 August 1995, Namur, Belgium (1995)
37. George Lakoff, *Women, Fire, and Dangerous Things*, U. Chicago Press (1987)
38. Christopher Landauer, "Syntax is More Than You Think", pp. 801-806 in *Proceedings of ISAS'98: The 1998 International Multidisciplinary Conference on Intelligent Systems and Semiotics*, 14-17 September 1998, NIST, Gaithersburg, Maryland (1998)
39. Christopher Landauer, "Data, Information, Knowledge, Understanding: Computing Up the Meaning Hierarchy", pp. 2255-2260 in *Proceedings of SMC'98: The 1998 IEEE International Conference on Systems, Man, and Cybernetics*, 11-14 October 1998, San Diego, California (1998)
40. Christopher Landauer, "Some Measurable Characteristics of Intelligence", Paper WP 1.7.5, *Proceedings of SMC'2000: The 2000 IEEE International Conference on Systems, Man, and Cybernetics (CD)*, 8-11 October 2000, Nashville Tennessee (2000)

41. Christopher Landauer, Kirstie L. Bellman, "Knowledge-Based Integration Infrastructure for Complex Systems", *International Journal of Intelligent Control and Systems*, Volume 1, Number 1, pp. 133-153 (1996)
42. Christopher Landauer, Kirstie L. Bellman, "Integration Systems and Interaction Spaces", pp. 161-178 in *Proceedings of FroCoS'96: The First International Workshop on Frontiers of Combining Systems*, 26-29 March 1996, Munich (March 1996)
43. Christopher Landauer, Kirstie L. Bellman, "Semiotics of Constructed Complex Systems", pp. 35-40 in *Intelligent Systems: A Semiotic Perspective*, *Proceedings of the 1996 International Multidisciplinary Conference*, *Volume I: Theoretical Semiotics*, *Workshop on Intelligence in Constructed Complex Systems*, 20-23 October 1996, Gaithersburg, Maryland (1996)
44. Christopher Landauer, Kirstie L. Bellman, "Mathematics and Linguistics", pp. 153-158 in *Intelligent Systems: A Semiotic Perspective*, *Proceedings of the 1996 International Multidisciplinary Conference*, *Volume I: Theoretical Semiotics*, *Workshop on New Mathematical Foundations for Computer Science*, 20-23 October 1996, Gaithersburg, Maryland (1996)
45. Christopher Landauer, Kirstie L. Bellman, "Model-Based Simulation Design with Wrappings", pp. 169-174 in *Proceedings of OOS'97: The 1997 Object Oriented Simulation Conference*, *WMC'97: The 1997 SCS Western MultiConference*, 12-15 January 1997, Phoenix, SCS International (1997)
46. Christopher Landauer, Kirstie L. Bellman, "Computational Embodiment: Constructing Autonomous Software Systems", pp. 131-168 in *Cybernetics and Systems: An International Journal*, Volume 30, Number 2 (1999)
47. Christopher Landauer, Kirstie L. Bellman, "Situation Assessment via Computational Semiotics", pp. 712-717 in *Proceedings ISAS'98: the 1998 International Multidisciplinary Conference on Intelligent Systems and Semiotics*, 14-17 September 1998, NIST, Gaithersburg, Maryland (1998)
48. Christopher Landauer, Kirstie L. Bellman, "Language Formation in Virtual Worlds", pp. 1365-1370 in *Proceedings of SMC'98: The 1998 IEEE International Conference on Systems, Man, and Cybernetics*, 11-14 October 1998, San Diego, California (1998)
49. Christopher Landauer, Kirstie L. Bellman, "Generic Programming, Partial Evaluation, and a New Programming Paradigm", Paper etspi02 in *Proceedings of HICSS'99: The 32nd Hawaii Conference on System Sciences (CD)*, *Track III: Emerging Technologies*, *Software Process Improvement Mini-Track*, 5-8 January 1999, Maui, Hawaii (1999); revised and extended version in [51]
50. Christopher Landauer, Kirstie L. Bellman, "Computational Embodiment: Agents as Constructed Complex Systems", Chapter 11, pp. 301-322 in Kerstin Dautenhahn (ed.), *Human Cognition and Social Agent Technology*, Benjamins (2000)
51. Christopher Landauer, Kirstie L. Bellman, "Generic Programming, Partial Evaluation, and a New Programming Paradigm", Chapter 8, pp. 108-154 in Gene McGuire (ed.), *Software Process Improvement*, Idea Group Publishing (1999)
52. Christopher Landauer, Kirstie L. Bellman, "New Architectures for Constructed Complex Systems", in *The 7th Bellman Continuum*, *International Workshop on Computation, Optimization and Control*, 24-25 May 1999, Santa Fe, NM (1999); in *Applied Mathematics and Computation*, Volume 120, pp. 149-163 (May 2001)

53. Christopher Landauer, Kirstie L. Bellman, "Symbol Systems in Constructed Complex Systems", pp. 191-197 in *Proceedings of ISIC/ISAS'99: The 1999 IEEE International Symposium on Intelligent Control*, 15-17 September 1999, Cambridge, Massachusetts (1999)
54. Christopher Landauer, Kirstie L. Bellman, "Lessons Learned with Wrapping Systems", pp. 132-142 in *Proceedings of ICECCS'99: the 5th International Conference on Engineering Complex Computing Systems*, 18-22 October 1999, Las Vegas, Nevada (1999)
55. "Virtual Simulation Environments", pp. 191-196 in *Proceedings of AISP'00: The 2000 International Conference on Artificial Intelligence, Simulation, and Planning*, 6-8 March, 2000, Tucson (2000)
56. Christopher Landauer, Kirstie L. Bellman, "Reflective Infrastructure for Autonomous Systems", pp. 671-676, Volume 2 in *Proceedings of EMCSR'2000: The 15th European Meeting on Cybernetics and Systems Research*, *Symposium on Autonomy Control: Lessons from the Emotional*, 25-28 April 2000, Vienna (April 2000)
57. Christopher Landauer, Kirstie L. Bellman, "Some Measurable Characteristics of Intelligence", Paper WP 1.7.5, *Proceedings of SMC'2000: The 2000 IEEE International Conference on Systems, Man, and Cybernetics (CD)*, 8-11 October 2000, Nashville Tennessee (2000)
58. Christopher Landauer, Kirstie L. Bellman, "Symbol Systems and Meanings in Virtual Worlds", *Proceedings of VWsim'01: The 2001 Virtual Worlds and Simulation Conference*, *WMC'2001: The 2001 SCS Western MultiConference*, 7-11 January 2001, Phoenix, SCS (2001)
59. Christopher Landauer, Kirstie L. Bellman, "Computational Infrastructure for Experiments in Cognitive Leverage", in *Proceedings of CT'2001: The Fourth International Conference on Cognitive Technology: Instruments of Mind*, 6-9 August 2001, Warwick, U.K. (2001)
60. Christopher Landauer, Kirstie L. Bellman, "Wrappings for One-of-a-Kind System Development", Paper STSSV04 in *Proceedings of HICSS'02: The 35th Hawaii International Conference on System Sciences (CD)*, *Track IX: Software Technology*, *Advances in Software Specification and Verification Mini-Track*, 7-10 January 2002, Waikoloa, Hawaii (Big Island) (2002)
61. Christopher Landauer, Kirstie L. Bellman, "Semiotic Processing in Constructed Complex Systems", *Proceedings of CSIS2002: The 4th International Workshop on Computational Semiotics for Intelligent Systems*, *JCIS2002: The 6th Joint Conference on Information Sciences*, 08-13 March 2002, Research Triangle Park, North Carolina (2002)
62. Christopher Landauer, Kirstie L. Bellman, "Refactored Characteristics of Intelligent Computing Systems", *Proceedings of PERMIS'2002: Measuring of Performance and Intelligence of Intelligent Systems*, 13-15 August, NIST, Gaithersburg, Maryland (2002)
63. Michael E. Long, "Surviving in Space", National Geographic, January 2001, also on the web at URL "http:// www. nationalgeographic. com/ ngm/ 0101/ feature1/ index. html" (availability last checked 02 June 2007), with many useful links to more information
64. Barbara F. Lujan, Ronald J. White, "Human Physiology in Space", National Space Biomedical Research Institute, Houston, Texas, on the web at URL "http:// www. nsbri. org/ HumanPhysSpace/ index. html" (availability last checked 02 June 2007),

65. Patti Maes, "Computational Reflection", Technical Report 87-2, MIT AI Laboratory (1987)
66. P. Maes, D. Nardi (eds.), *Meta-Level Architectures and Reflection*, *Proceedings of the Workshop on Meta-Level Architectures and Reflection*, 27-30 October 1986, North-Holland (1988)
67. Colin McGinn, *The Mysterious Flame: Conscious Minds in a Material World*, New York: Basic Books (1999)
68. Alex Meystel, "Multiresolutional Architectures for Autonomous Systems with Incomplete and Inadequate Knowledge Representations", Chapter 7, pp. 159-223 in S. G. Tzafestas, H. B. Verbruggen (eds.), *Artificial Intelligence in Industrial Decision Making, Control and Automation*, Kluwer (1995)
69. Christian Müller-Schloer, "Quantitative Emergence." Talk presented at Dagstuhl Seminar on Organic Computing, January 2006, p. 8.
70. Alex Meystel, *Semiotic Modeling and Situation Analysis: An Introduction*, AdRem, Inc. (1995)
71. David C. Parks and Moshe Tennenholtz (eds.), *Games and Economic Behavior*, Special Issue Dedicated to the ACM Conference on Electronic Commerce (EC'07), Elsevier, 2008 (to appear).
72. H. Pasula, L. Zettelmoyer, and L. Pack Kaelbling, "Learning Symbolic Models of Stochastic Domains", *Journal of Artificial Intelligence Research* 29(2007).
73. Candace Pert et al., "Neuro-peptides and their receptors: a psychosomatic network", J. Immunology 135(2), pp. 820-826 (1985)
74. A. B. Pippard, *Response and Stability: An Introduction to the Physical Theory*, Cambridge University Press, Cambridge, 1985, p. 127.
75. Torsten Rüting, "History and significance of Jakob von Uexküll and of his institute in Hamburg", *Sign System Studies*, Vol. 32, Nos. 1/2, pp. 35-72 (2004)
76. Ferdinand de Saussure, *Cours de linguistique générale*, Payot, Paris (1916), translated by W. Baskin as *A Course in General Linguistics*, Fontana/Collins, Glasgow (1977)
77. Mary Shaw, William A. Wulf, "Tyrannical Languages *still* Preempt System Design", pp. 200-211 in *Proceedings of ICCL'92: The 1992 International Conference on Computer Languages*, 20-23 April 1992, Oakland, California (1992); includes and comments on Mary Shaw, William A. Wulf, "Toward Relaxing Assumptions in Languages and their Implementations", *ACM SIGPLAN Notices*, Volume 15, No. 3, pp. 45-51 (March 1980)
78. Brian Cantwell Smith, "Varieties of Self-Reference", In J.Y. Halpern (Editor), *Reasoning about Knowledge*, *Proceedings of TARK 1986*, AAAI Publication, pp. 19-43 (1986)
79. John Sowa, *Knowledge Representation*, Morgan Kaufmann (1999)
80. Luc Steels, "The symbol grounding problem is solved, so what's next?", in M. De Vega, G. Glennberg and G. Graesser (eds.), *Symbols, embodiment and meaning*, Academic Press, New Haven (2007)
81. Ian Stewart and Jack Cohen, *Figments of Reality: The Evolution of the Curious Mind*, Cambridge University Press (1999)
82. G. M. Tomkins, "The Metabolic Code", *Science* 189: 760-763, 1975.
83. Jakob von Uexküll, "Streifzüge durch die Umwelten von Tieren und Menschen", (1934); translated and edited by Claire H. Schiller as "A stroll through the worlds of animals and men", pp. 5-80 in Claire H. Schiller (eds.), *Instinctive Behavior: The Development of a Modern Concept*, International Universities

Press, New York (1957); also reprinted in *Semiotica*, Vol. 89, No. 4, pp. 319-391 (1992)
84. Duncan J. Watts and Steven H Strogatz, "Collective dynamics of 'small-world' networks", *Nature* 393 (1998)
85. Ludwig Wittgenstein, *Logisch-philosophische Abhandlung* Annalen der Naturphilosophie (1921), translated into English by C. K. Ogden as *Tractatus logico-philosophicus*, Routledge and Kegan Paul (1922); also on the Web at URL "http:// www. kfs. org/ jonathan/ witt/ tlph.html" (availability last checked 31 May 2007); also in Project Gutenberg at URL "http:// www. gutenberg. org/ etext/ 5740" (availability last checked 31 May 2007);
86. Ludwig Wittgenstein, *Philosophische Untersuchungen*, translated into English by G. E. M. Anscombe as *Philosophical Investigations*, Prentice Hall (1999), Blackwell (1953/2001) (which includes the original German text also)
87. Carl Zimmer, "From Fins to Wings", National Geographic, November 2006, also on the web at URL "http:// www. nationalgeographic. com/ ngm/ 0611/ feature4/ index. html" (availability last checked 02 June 2007), with many useful links to more information