03 EXPANDING THE FEEDBACK LOOP

Theories of communication between biological and mechanical systems emerged as soon as the foundations of technology appeared. Systems theory was followed by cybernetics, aiming to understand the functions and processes of systems that participate in circular and causal chains—moving from action to sensing to comparison back to action—especially between artificial and biological systems.[1] The investigation between biological and mechanical systems intersects a variety of disciplines, including psychology, ecology, linguistics, anthropology, and visual arts. Similar to the tuning of physical models and simulations, synthetic learning is tuned through historical data and ongoing simulations to define approaches to complex problems. This approach requires two distinct components, clear processes of abstraction and generalization as well as robust interfaces with physical systems. The translation of complex interactions into abstract or generalized models requires an evolving process, where the model is continually evaluated to understand the relationship between fidelity, accuracy, and chance. This interface stems from an abstracted relationship, but must maintain a flexible evolution allowing for new logics to harness the connections that are created. In this manner a relationship develops between programmed logic and physical interaction.

Within the case studies outlined in this text, and much of the work in the field of responsive architecture, technology takes the form of an object communicating with external interactions. The emphasis on the object as mediator, instead of context or environment, reduces the complexity of systems—filtering out fluid connections, varying conditions, and processes that can emerge in a higher order, multiple-loop, or evolving dialogue. Stan Allen formerly advocated this position suggesting that, as a discipline, Architecture should shift focus from objects to "field conditions" to address the form *between* things instead of the form *of* things.[2] It is this liminal zone of landscape (the field) punctuated by architecture that is synthesized by landscape architects. As we design the built environment, addressing the need to control, modify, and interpret ecologies in an effort to enable sustainability and adaptability it is necessary to exploit the intricacies of context as a means to create connections between ecological, cultural, and social systems.

The application of responsive technologies in landscape architecture requires a formal shift in the relationship between user, environment, and computation to focus on context and interrelations. The primary research in computation and interaction, particularly within physical computing have focused on Human Computer Interaction (HCI).[3] The relationship between the user and the machine is an intricate web that alters human perception and mediates between modes of simulation (in software) and forms of articulation (in hardware). This relies heavily on positing that the role of computation centers around human relationships with systems that are both simple and complex. In considering HCI as a device for shaping and mediating physical environments, this anthropogenic approach must be expanded and linked to an interaction model that is contextual. Response to site context is fundamental to landscape architecture, creating metaphorical connections or performative relationships. This form of contextual response becomes embedded within the behavior of responsive systems, creating site specificity through behaviors or outputs rather than solely through metaphor, morphology, or aesthetics.

Feedback denotes a call and response. It is the ability of a system to act, process the actions, and then respond with an updated action. This causal chain of response is the feedback loop and is central to self-regulating or evolving systems. This was discussed briefly in the previous chapter but deserves further exploration to examine the particulars of how this might evolve.

EVOLVING RESPONSE

The feedback loop as a framework denotes an interaction between user and system or device, an updating and cyclical series of events. The common example has been the thermostat, called out early on by the cybernetician Gordon Pask,[4] and described in Usman Haque's Hardspace, Softspace ". . . regulating temperature according to our requirements."[5] Even this example has advanced over time and many current thermostats have evolved to not only control temperature but also to visualize energy consumption and to predict ideal temperature settings. "In the feedback-loop model of interaction, a person is closely coupled with a dynamic system. The nature of the system is unspecified." If this is the case then ". . . the feedback-loop model of interaction raises three questions: What is the nature of the dynamic system? What is the nature of the human? Do different types of dynamic systems enable different types of interaction?"[6] Similarly, from the perspective of landscape architecture, these questions must consider: What is the nature of the non-HCI centric feedback loop? How does feedback change the nature of nature? How do responsive systems function as integral regulators of ecological systems?

As we examine the potentials of these technologies in the environment each of the previous questions is important. Responsive systems that are non-HCI specific ideally expand to incorporate a broader range of constituencies including flora, fauna, and other actors. Broadening the interaction model also broadens the goals that the system attempts to achieve and regulate. The new "natures" that spawn from this approach can be vast, not only evolving in new ways but also performing beyond typical capacities. This form of nature is a product of human designed computational intelligence that evolves in tandem with biological intelligence. These new relationships are inherently unknown but become coupled with the environment. This form of feedback will have the ability to grow in processing capacity, creating a resistant response to ecological systems. It is this resistance that is the primary method of feedback with the potential to create an evolutionary growth between machine and biology. A form of feedback and resistance is central to the design of the environment; it can be seen in a range of contemporary landscape architecture projects. The planting of 25,000 birch trees at Schiphol Airport in Amsterdam by West 8 creates a tension between infrastructural supremacy and the landscape as liminal connector.[7]

Before going deeper into feedback and response in the environment it is important to discuss some underlying concepts regarding interaction and feedback. Responsive technologies often rely on self-regulating systems that operate on singular goals or outcomes. The goal defines a relationship between the system and its environment, which the system seeks to attain and maintain. A simple self-regulating system, one with only a single feedback loop, cannot adjust its own goal; its goal can be adjusted only by something outside the system. Such single-loop systems are called "first order."[8] Many typical forms of feedback engage this first order and trend towards regulation based on a pre-determined goal. In computer programming, feedback refers to basic logics: the testing of a condition and the execution of code in response. Testing statements such as if: then and if: else create responses as the code parses information. This fundamental binary language is the underlying root of responsive technologies and is inherent to the creation of software. The hurdle for designers in software comes from developing greater nuance and ranges of feedback. To develop computational logics that evolve, learn, and expand with biological systems, there is a requirement that these forms of binary logic advance to understand the role of the device and the language that connects the device to its context.

Describing the environment as a series of actors, devices, or machines connotes a formal schema that translates the seemingly inanimate into active objects. The term machine, as defined by Levi Bryant, packages physical forms, political philosophies, and biological processes—among all other things—as agents that possess their own properties and behave through a series of internal and external flows.[9] This is an object-oriented model, where the expression of the machine takes on the characteristics of its relationships. Responsive technologies have this form of agency, creating an ontological relationship through sensing, feedback, and response. This form of design hinges on fields of influence and contextual awareness, described as "intelligent ambience"—an attempt to move beyond the capabilities of the machine and to instead focus on connections and relationships. Lucy Bullivant describes this as a way to emphasize "intelligence and its distribution through an environment" highlighting an "understanding of people's character, behavior, and context."[10]

Understanding the communication between systems as a form of language, either co-ordinated or translated, is a useful conceptual envelope to discuss the transfer of data between designed and evolved objects. This metaphorical language exists in a liminal realm between mechanical or digital devices, and biological or ecological systems. Language, and a process of abstraction, addresses simultaneous modes of complexity through common communication protocols. This comes in multiple forms, from a library for design typologies to a discrete interface between entities. The process of abstraction becomes extremely important, taking infinitely complex organisms or systems and describing them through abstract representations that elicit their fundamental performance.

Data is collected, filtered, reduced, and abstracted in order to become manageable. When data is diminished in this way, valuable pieces of context can be lost and connections are severed. The process of abstraction translates essential elements as a way to analyze or classify the properties while maintaining connections to higher fidelities. By separating characteristics, or hiding details of one or more properties, disparate systems can be arranged and compared, allowing one to concentrate on different concepts in isolation from others.[11] In computer architecture, there is a series of abstraction layers that operate between the physical hardware and the software that runs the hardware. These layers hide the implementation details and differences between hardware so that they can function within the same system. Libraries translate commands provided by the programs into the specific device commands needed by each individual piece of hardware. While unessential details are hidden, they are not lost and can be retrieved in the event of later necessity. Abstraction makes it possible to change the quantity of information represented, allowing a reformulation in a simpler formalism to become possible while preserving the quantity of information involved.[12]

As a metaphor, this form of translation from ecological systems, biology, and software is an intriguing way to imagine a new form of the abstraction layer. The operating system or integrated development environment can be useful metaphors to imagine the meta-space that would enable this form of connection or transaction,[13] essentially an evolving translator or compiler that interprets between hardware (wetware) and software (virtualizations). This ecological abstraction layer is the liminal space that serves as a linkage, a form of integrated development environment for the creation of new forms of biological engineering and ecological management. Systems that find common modes of abstraction begin to create opportunities for larger inter-relationships between disparate ecological, biological, social, and cultural modes. This common layer of abstraction creates an over-arching opportunity to develop, analyze, or accentuate patterns of linkages rather than design self-contained devices or architectures. Each element expresses itself through physical data with underlying streams of complexity that stem from local inputs but are linked and modified within larger systems. The larger system has no aim of control but instead trends toward local stabilities, adaptation, and modification across subsets of systems.

There are two types of abstraction in computation: control and data. Control, process or procedural, abstraction separates the way a procedure or action is used from how it is implemented and focuses on actions. The external representation is the presentation or interaction that allows programmers to intuitively communicate with hardware. Without a form of control abstraction the process of communicating directly with hardware lacks flexibility, is not efficient, and lacks the intuition of abstract arithmetic operations. Data abstraction separates the elements of behavior that are not critical to the procedure from those that are, and allows programmers to hide data representation details behind a simple set of operations.[14] This creates a contract on behavior between data and code. Anything that is poorly defined or not defined when implemented can change, thus breaking the system. Both types of abstraction can be used discretely or jointly, depending upon the requirements of the system. It is important to understand these two forms of abstraction and to develop a layer of complexity that is exposed to constructed systems.

This layer sets an evolving framework and limits the interaction by controlling inputs and outputs.

A system that focuses on defined methods of abstraction must provide methods for separating qualities of each component and classifying those that are similar providing for moments of overlap and possibilities for communication. The merging of disparate systems such as ecology, technology, and culture requires abstraction methods that link each system, preserving desirable properties while allowing complex disparities to be hidden. Abstraction must provide a method that expands the boundaries of programmed systems, which are currently built for specific interactions and lack the opportunity to create evolving organic interactions. Creating a system that expresses intricate complexities is typically an issue of maintaining access to modes of high data fidelity through time.

> Abstraction may be discovered or produced, may be material or immaterial, but abstraction is what every hack produces and affirms. To abstract is to construct a plan upon which otherwise different and unrelated matters may be brought into many possible relations. To abstract is to express the virtuality of nature, to make known some instance of its possibilities, to actualize a relation out of infinite relationality, to manifest the manifold.[15]

Abstraction affects the interrelation between systems through a reformulation of the subject. Each mode has unique properties but a similarity must be considered in order to maintain data fidelity and flexibility when abstracted. Each of these aforementioned systems can be categorized as input, processing, or output creating a component of a larger inter/reactive system. Inputs are typically sensing systems gathering data from the environment but can also come from storage or database systems depending on the implementation. Processing systems are analytical or transformative in nature, causing data to take on new forms. Output systems are varied but are typically either physical/mechanical, or sensorial visualization systems. What does it mean to abstract any of these systems? Can we envision a robust methodology of abstraction that intends to maintain accessibility while promoting an underlying complexity? How does the composite of the systems form a responsive framework that allows for free evolution while maintaining stability?

Dialogue and conversation occur when these systems exchange information in a continuous process, acknowledging and/or influencing each other's responses, creating a progression of feedback loops. These loops can be either single (closed) or multiple (open).

Figure 03.02 Ecolibrium, *Kim Nguyen, Devin Boutte, Martin Moser, and Joshua Brooks, Responsive Systems Studio, 2011*

In a single-loop conversation, outputs are determined by filtering, and the system feeds information back into itself. In a multiple-loop conversation, new information from each system influences future dialogue and depends upon cycles of response.[16] In a true multiple-loop scenario, the output is not pre-determined but is predicated on evolving contextual input. The multiple-loop scenario can be extrapolated to an under-specified or evolving system ordering a new set of possibilities. Instruction is embedded in each device, machine, object, or infrastructure creating rich interactions with the environment and adjacent systems. Systems coagulate and disperse looking for contextual stability; the framework is in flux with shifting priorities and goals.

INTELLIGENCE

Landscapes are inherently intelligent, the biologies that comprise landscapes have their own individual behaviors, logics, and reasoning that allow these systems to evolve through connectivity and response. Speaking about intelligence in landscape speaks to resistance, counterpoints, and individual behavior, it is not what we think of when we use terms such as smart. Intelligence promotes free will and sentience where "smartness is intelligence that is cost-efficient, planner-responsible, user-friendly, and unerringly obedient to its programmer's designs."[17] Intelligence comes from methods of learning that are difficult to obtain through computation and typically has very little in common with methods of efficiency and obedience.

While metaphorical, in this context the concept of intelligence in landscape refers to a feedback loop between the virtualization of a system (landscape) and the physical, real-time manipulations of an environment. Virtualization refers to the removal of landscape data through sensing and monitoring and recreating landscape as an artifact of processing or reaction. This is a loop of monitoring, processing, and actuation that require methods that create independent intelligences within the physical landscape. The establishment of an intelligent system begins with monitoring. To understand the relationship between biological intelligence and computation it is important to outline some of the methods that have led to advances in artificial intelligence and machine learning.

Machine learning and artificial intelligence are important computational paradigms that provide practical methods to create systems that have the ability to learn, evolve, and respond to complex systems. The term "artificial intelligence" is a larger concept that has evolved through advancements in search, machine learning, and statistical analysis. Although the term "artificial intelligence" was initially coined in 1956 by John McCarthy when he hosted a conference that explored

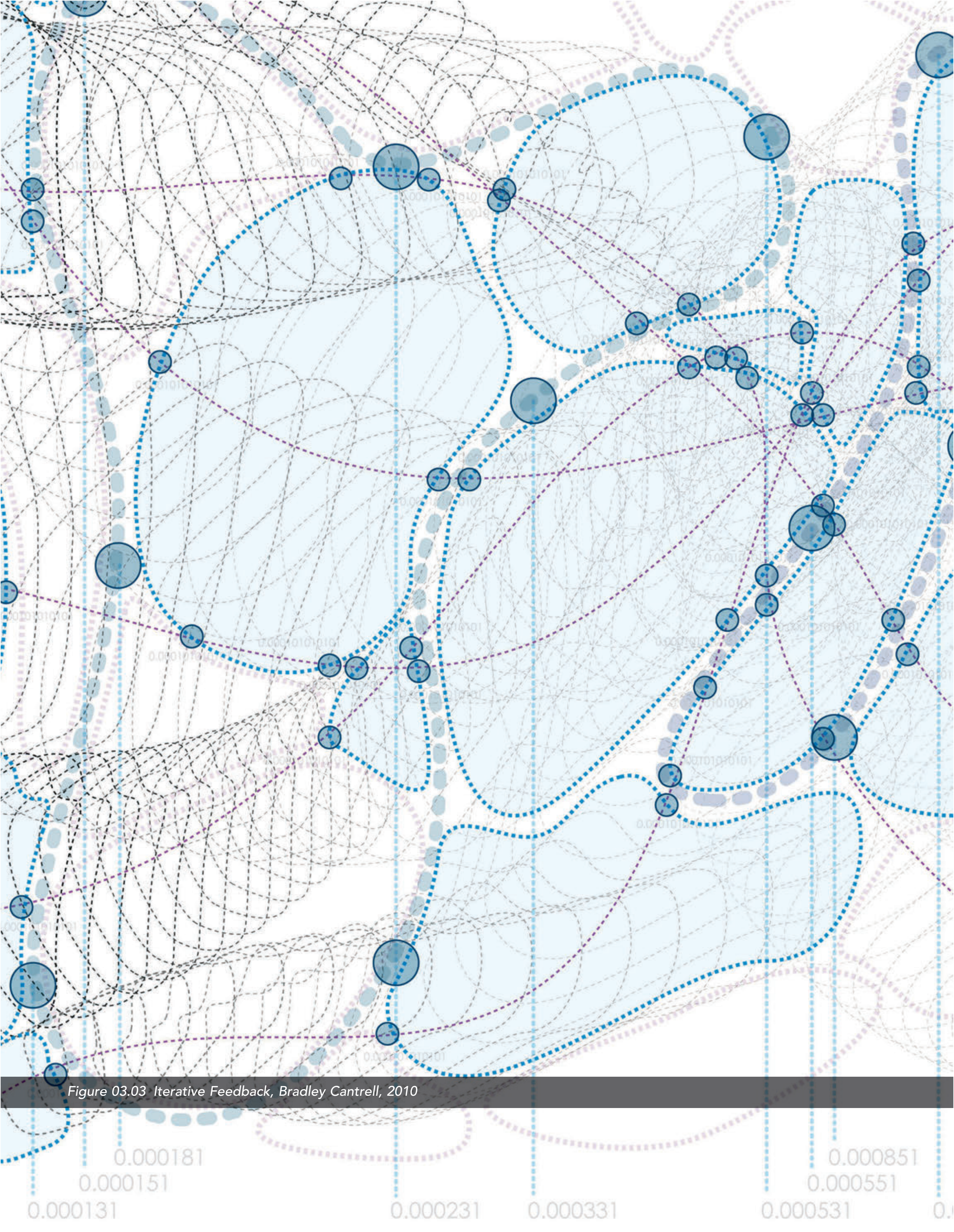

Figure 03.03 Iterative Feedback, Bradley Cantrell, 2010


0.000651
0.000631
0.000731
0.000831
0.000851
0.000931
0.001051
0.001031
0.00113

the topic, the idea was proposed by Vannevar Bush's memex, a device that extends human memory and cognition. Alan Turing, in 1950, set forth the question, "Can machines think?" in a paper entitled "Computing Machinery and Intelligence."[18] It is in this paper that Turing introduced the concept of the "Turing test," which was referred to as "the imitation game." The Turing test sets an important, if not pragmatic, benchmark for artificial intelligence that is indistinguishable from human intelligence. Although this form of intelligence has not yet been reached through computation it remains as a continuing goal.[19] Is there a form of the Turing test that is useful for ecological systems? In some ways, this could be said to have been achieved through modeling of simple life forms such as the modeling of C. elegans in the OpenWorm Project.[20] The obsession with the Turing test in the field of artificial intelligence frames human intelligence as the predominant form of sentience rather than imagining new intelligences that may be better suited in a range of contexts.

What constitutes artificial intelligence is continually evolving and the benchmark is always extended. Where many in the 1950s would have declared that a computer that could play chess on par with a human opponent was a form of artificial intelligence, this is no longer the case. Generally, forms of artificial intelligence are subtle, parsing logistics data to determine new efficiencies or giving recommendations for search terms, rather than performing as sentient beings that take on the form and interaction of human proxies. While the estimation of what is to come often outpaces the current state of artificial intelligence, the effect of intelligent systems serves as an extension or mediator of human intelligence. Human memory and processing are offloaded into computational systems creating space for human intelligence to take on other tasks. This intelligence as an extension of humans is a form of feedback that alters our evolution through a form of transhumanism, this becomes apparent as certain technologies become robust and last for multiple generations.[21] A similar form of extension also pushes environmental systems to develop novel ecologies, interactions between digital computation and analog computation. It is this extension of intelligence that is important for ecological systems, it is less important to create computational models that are indistinguishable from ecological systems and more important to extend or respond through new forms of response and computation.

Machine learning is a merger between statistics and artificial intelligence that encompasses a range of methods that allow computers to "learn." In some sense machine learning is a way for computers to automatically write programs rather than being explicitly programmed. This is accomplished through inputs that serve as examples that the computer uses to learn from, these examples then drive the creation of a logic. As a framework, this type of learning creates a way to process a range of data and to create a logic that is dependent upon these inputs, therefore responding and evolving as the dataset or input evolves. One of the first examples of machine learning is from Arthur Samuels who, in 1959, developed software that analyzed the game of checkers. The program Samuels developed looked at which moves tended to win and which tended to lose, and created a strategy based on probabilities of the moves available at each turn. The computer then played tens of thousands of games against itself to learn; this created a form of adaptation and allowed the software to become very good at checkers and to continue evolving based on its opponents. It is this form of adaptation in computation that creates potential for our interface with ecology and environment, an integrated method of analysis and response.[22]

The implementation of machine learning requires the positing of well-formed questions. Forms of learning can be structured such that they focus on experience related to an operation. To evaluate the learning, performance must be measured as it relates to the operation; if this improves or degrades the experience then an inference can be made. The creation of a "well-posed learning problem" was posited by Tom M. Mitchell and Avrim Blum, and varies based on implementation.[23] What becomes readily apparent is that this form of learning can be extrapolated to encompass relationships between infrastructure, urbanism, and ecology. A form of learning that derives new forms relationships that are the product of evolving biotic and abiotic intelligence.

SYNTHESIS

Intelligence is a powerful concept when contemplating landscape systems, particularly when considering feedback and interaction. The project of artificial intelligence gives insight to the larger question of how designed landscapes respond with higher levels of complexity and have a form of agency, evolution, and resistance. The relationship between designed intelligence and evolved intelligence presents an additional tool in landscape architecture's repertoire of media. The complexity of ecological systems exceeds human capacity to understand, manage, and control. As designers and engineers our methods have predominantly approached landscape as a medium that requires control and manipulation, a need to fully articulate the future of place. Another view of this control can frame it as a form of abstraction, simplifying the complexity of the landscape system to provide a method that designers can act upon. This is not a fault of the profession but instead comes from a multitude of cultural and legal traditions that go beyond the scope of this conversation.

As forms of intelligence, simulation, and language are more deeply integrated within landscape a form of sentience develops, an agency that is embedded within objects. A system or infrastructure based on mutual abstraction would be contextually aware through local interactions such as responsive buildings, traffic monitoring, and/or environmental sensors. The system would have pre-determined goals of safety, but evolving goals of management linked across time, location, and content. Creating a contextual narrative relayed through abstraction allows for a multitude of designers to develop robust local systems that are inherently interconnected. Most importantly this interconnected system does not ignore the value of local context but instead exemplifies it and expands its relationships, creating potentials for rich human experience. The objects designed with agency have the ability to imagine infrastructure that is driven by autonomous logics and engages landscape processes through an evolved learning—a constructed landscape that is neither pre-determined nor bounded but requires our engagement. This is a spatial paradigm that disengages with previous computational roles and requires models that are deeply embedded within ecological systems. Models are abstractions, abstractions that grow their relationships to the ecologies they are modeling.

Usman Haque's description of the "collectively designed project" as incorporating "conflicting logics" refers to numerous intentions competing within the constructed environment.[24] In this sense, the feedback loop (for the built environment) should frame unintentional influence over evolved environments. The larger question lies in how feedback evolves to consider these external consequences, the areas that are not pre-determined but are resultant from the actions of the system. As a method this embraces landscape as a composite of intelligences, a biological form of memory and processing that shifts or opens possibilities. In much the same way we are programming and tuning the landscape, other actors—such as sediment or mosquitos—are doing the same thing. We imagine their input as having less intention, although it has every bit as much agency and outcome. In a similar way that intelligence, as described by computation, is an extension of humanity, it is also an extension of ecology. Responsive landscapes point towards a realm where built systems emerge from goals that are not solely concerned with humanity.

Goals are an integral benchmark in design, an outcome that designers and engineers attempt to match or calibrate toward. Without goals there is no known outcome and iteration relies on the moment, there is only history to repeat. At the same moment, goals promote stasis in that they specify an outcome and the system, or landscape, regulates itself to match this outcome. Conceptually the evolving feedback loop refers to methods of interaction that move beyond static goals and system efficiency and embrace divergent or bifurcating futures. The evolving feedback loop leaves room for novel or emergent conditions. In a critique of Landscape Urbanism's "process discourse" to actualize landscapes exhibiting truly novel and emergent conditions, Julian Raxworthy's dissertation "Novelty in the Entropic Landscape" examines methods in gardening to find material-based interaction and real-time responsive behaviors to novel conditions found in the evolving garden landscape. Raxworthy offers the term "tendency" to describe an approach to designing for change or emergence:

> . . . that seeks to aim towards an outcome rather than concretely specify it. Instead tendency promotes flexibility in how an outcome will result, exhibiting novelty in the form of specificity rather than contrast. Feedback describes real-time processes, such as gardening, that allow for a recurrent involvement in the development of projects over time, maximising emergent opportunities. [25]

This concept of feedback points to a need, particular in landscape, to develop fuzzy outcomes that operate within a range—where success is not narrowly defined, but instead connotates novel outcomes that may deviate from a previous hypothesis. A focus on indeterminate systems relies on unorthodox methods of measurement for success, implementation of failure, and modes of resistance. The indeterminacy of site systems and networks constructs a future for networks as political ontologies, which place the material significance of networks as a critical indicator to establishing effective protocol; the ability for political control of networks.[26] This methodology of applying protocol to biological networks, termed "protocological control," to aid in political resistance can be paralleled to the necessary methods for adjusting the complex networks that would exist in a synthetic or responsive landscape. Small adjustments to protocol, termed "counterprotocols," can be envisioned as a set of design rules, management guidelines, or instructions for landscape manipulation to aid in adaptive management. "Resistance" is generally a political term, used to describe tactics for initiating political reform and can be understood as a metaphor for small adjustments to the system. The idea of resistance is especially compelling within the context of designing counterprotocols for ecological systems and networks. Instead of instituting a definite system or network, the counterprotocol takes advantage of an ecosystem's adaptive and generative capabilities to excite change by a catalytic resistance. This target of resistance as a stimulus for protocol's ability to "sculpt" and "inflect" allows for small adjustments to effectively manipulate the overall structure of the network over time.

Conceptually this is an attempt to answer a fundamental question in contemporary landscape architecture theory, "how can we design landscapes that engage dynamic systems and adapt to changing conditions?" Protocological control and modes of resistance are convincing metaphors for manipulating technological networks but they also have significance for ecological networks. As our technologies increasingly interpret landscape as a network or network of networks, the view of ecological networks can be negotiated through an evolved form of feedback. In relation to landscape, "interactive architectural design . . . will enable the relationship between building and program to become a much more subtle and communicative process, embracing a wider, personalised set of functions, desires and experiences."[27] Pulling back from architecture there are several methods of interacting on landscape that look past personalization and desire, instead highlighting ecological fitness and robust dynamics. The methods described in the following chapters—elucidate, compress, displace, connect, ambient, and modify—employ forms of response, actuation, and perception as actors in the landscape. They are ways to conceptualize both virtual and physical transformations of the environment. Landscape shifts are enabled by the manipulation and choreography of landscape phenomena and matter, these are methods that speak to connectivity.

This is not a new conversation for landscape architecture; pragmatically it examines new ways to shape the environment and is not centered on new forms of human experience. This is a form of landscape that conceptualizes a cyborg—an integrated whole that is formed from integrated processes that are biotic and abiotic. The cyborg speaks to a smartness that goes beyond an environment laden with ubiquitous computing devices:

> The cyborg is the new contemporary archetype, altering existing ecologies by overlaying new sets of relationships between organism and environment. Despite the unprecedented proliferation of ubiquitous computing in daily life, "smartness" in architecture eludes the synthetic promise of the cyborg. [28]

This is a landscape embedded with intelligence and agency, human actions weighed within a matrix of competing interests. A landscape that is difficult to discern humanity from some form of other. It is a network of actors and requires designers to engage this system with methods that embrace adaptive and resistant management scenarios. The promise of a responsive landscape aspires to a methodology that is on one side technological and the other integrative.

NOTES

1 Gordon Pask, *Encyclopaedia Britannica* 14th ed. s.v. "cybernetics," 1972.

2 Stan Allen, "From Object to Field," in *Practice: Architecture, Techniques and Presentation* (London/New York: Routledge, 2008), 251. Revised and expanded from Stan Allen, "From Object to Field," *AD: Architecture After Geometry* 127 (1997): 24–31.

3 McCullough, *Digital Ground*, 154 (see chap. 2, n. 15).

4 Gordon Pask, *Conversation Theory: Applications in Education and Epistemology* (Amsterdam and New York: Elsevier Publishing Co., 1976), cited by Usman Haque in "The Architectural Relevance of Gordon Pask," *Architectural Design* 77, no. 4 (2007): 54–61.

5 Usman Haque, "Hardspace, Softspace and the possibilities of open source architecture," 2002, http://haque.co.uk/papers/hardsp-softsp-open-so-arch.PDF.

6 Dubberly, Usman, and Pangaro, "What is interaction?" (see chap. 1, n. 16).

7 Adriaan Geuze and Maarten Buijs, "West 8 Airport Landscape: Schiphol," *Scenario* 04 (Spring 2014), http://scenariojournal.com/article/airport-landscape/.

8 Dubberly, Usman, and Pangaro, "What is interaction?"

9 Levi R. Bryant, *Onto Cartography: An Ontology of Machines and Media*, in *Speculative Realism Series*, Ed. Graham Harman (Edinburgh: University Press, 2014).

10 Bullivant, *Responsive environments,* 1 (see chap. 1, n. 20).

11 Nell B. Dale and Chip Weems, *Programming and problem solving in Java* (Boston: Jones and Bartlett Publishers, Inc., 2007).

12 Fausto Giunchiglia and Toby Walsh, "A Theory of Abstraction," *Artificial Intelligence* 56 (1992): 323–390.

13 Fuller and Haque, "Urban Versioning System 1.0," 19 (see chap. 2, n. 13).

14 Dale and Weems, *Programming and problem solving in Java* (see chap. 3, n. 11).

15 Wark, "Abstraction/class," 213 (see chap. 2, n. 4).

16 Usman Haque, "Architecture, interaction, systems," *Archiquitetura & Urbanismo* 149 (August 2006).

17 Andrew Ross, "The New Smartness," in *Culture on the Brink: Ideologies of Technology*, Eds. Gretchen Bender and Timothy Druckrey (Seattle: Bay Press, 1994), Series: Discussion in Contemporary Culture, no. 9.

18 Alan Turing, "Computing Machinery and Intelligence," *Mind: a Quarterly Review of Psychology and Philosophy* 59, (January 1950): 433.

19 The passing of the Turing Test is a debated topic and several forms of computational intelligence have passed different benchmarks. The most recent has been an artificial intelligence, Eugene Goostman, created by a group of hackers out of St Petersburg, Russia. The Goostman bot successfully passed the Turing Test by convincing 33% of the judges at a competition hosted by the Royal Society at the University of Reading on June 7, 2014. However, this is not seen as having reached the benchmark set.

20 OpenWorm (website), http://openworm.org/.

21 Nick Bostrom, "A History of Transhumanist Thought," *Journal of Evolution and Technology* 14, no. 1 (January 2005). http://nickbostrom.com/papers/history.pdf. Accessed February 21, 2006.

22 Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: an Introduction* (Cambridge, MA: MIT Press, 1998), 267.

23 Avrim Blum and Tom M. Mitchell, "Combining Labeled and Unlabeled Data with Co-training" (Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998), 92–100, doi:10.1145/279943.279962.

24 Usman Haque, "Hardspace, Softspace and the Possibilities of Open Source Architecture," Haque Design Research, 2002, www.haque.co.uk/papers/hardsp-softsp-open-so-arch.PDF January 19, 2015.

25 Julian Raxworthy, "Novelty in the Entropic Landscape: Landscape Architecture, Gardening, and Change" (Doctoral Dissertation, University of Queensland, 2013), 20–21, (espace) http://academia.edu/8879242/Raxworthy_Julian._2013._Novelty_in_the_Entropic_Landscape_Landscape_architecture_gardening_and_change_School_of_Architecture_University_of_Queensland_Brisbane, July 2015.

26 Alexander Galloway and Eugene Thacker, "Protocol, Control, and Networks," *Grey Room* 17 (2004): 6–29.

27 Bullivant, "Introduction," 7 (see chap. 1, n. 1).

28 Yoon and Höweler, *Expanded practice*, 132 (see chap. 1, n. 18).