

---

# Abstraction Language: Digital/ Analog Dialogues

**Bradley E. Cantrell**

Assistant Professor  
Louisiana State University Robert Reich School of Landscape Architecture

**Natalie A. Yates**

2009 Master of Landscape Architecture Candidate  
Louisiana State University Robert Reich School of Landscape Architecture

---

## ABSTRACT

The connection between biological systems and machines is quickly becoming an important factor in designing the built environment. Sustainable power generation, water management, ecological infrastructures, and other nascent enviro-tech applications require negotiations between digital technologies and analog systems. This paper explores the model of abstraction languages as a method to create communications between biological and mechanical systems, focusing on modes accessible to design professionals.

Developing a mediating language based on abstraction creates an approach that addresses simultaneous issues in enviro-tech interfaces for sensing, responsiveness, automation, and interactivity. This language exists in a liminal realm between mechanical or digital devices and biological or ecological systems. Abstraction addresses simultaneous modes of complexity through a common dialogue. This comes in multiple forms, from a library for design to a logical interface between chaotic components. This lends itself to the development of methods of abstraction, taking infinitely complex organisms and finding threads to abstract information meaningfully.

The development of data and control abstraction in programming is explored in order to develop linkages between physical systems and digital interfaces. This examination looks at current methods of data conveyance for the built environment, and at pushing beyond these current methods to suggest a method of abstraction. The researchers are particularly interested in the ability of abstraction to compress ecological/biological complexity into accessible modules for responsive environments. These modules would be developed to create evolving systems to control, modify, link, or interpret ecologies, biologies, cultures, and societies.

## 1 INTRODUCTION

An increasing attention to sustainability and design vernacular coupled with the availability of new technologies that enable evaluation, efficiency, and adaptability challenges designers to establish meaningful connections between the built environment and users. Sustainable power generation, water management, ecological infrastructures, and other nascent enviro-tech applications already require negotiations between digital and analog systems; however, these negotiations are often insulated from design, inaccessible to designers without specific scientific knowledge or technical skill in electrical or computer engineering. Our environment is saturated with "smart" devices that we use every day. Many of these devices are embedded into our lives in such a manner that we hardly know they exist, functioning in the periphery and blending into the context of our environment (McCullough 2005). As designers of the built environment, we have the responsibility to engage site context and, therefore, take into account this layer of technology, which is increasingly affecting the environment. The need to establish a methodology for communicating between technological, environmental, biological, cultural, and social systems becomes apparent. In this paper, we propose that by examining high-order computing principles and borrowing from the concept of abstraction layers in order to compress complexity, modules for responsive environments that are accessible to designers can be developed to control, modify, link, and interpret ecologies.

## 2 ABSTRACTION

In many cases, collected data is filtered, reduced, and refined in order to become manageable. When data is diminished in this way, valuable pieces of context can be lost. Abstraction, however, separates essential elements from complexity in order to analyze or classify the properties without losing data fidelity. Disparate systems can be arranged and compared when characteristics are separated or details of one or more properties are hidden, allowing one to concentrate on different concepts in isolation from others (Dale and Weems 2007). 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 (Giunchiglia and Walsh 1992; Zucker 2003).

There are two types of abstraction: 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. Subprograms and control flows simulate desired concepts or interactions. Structured programming breaks down complex tasks into many different modules and isolates the implementation details of one component from the others (Dale and Weems 2007). 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 (Dale and Weems 2007). 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.

A system that focuses on defined methods of abstraction must provide methods for separating qualities of each component and classifying those that are similar, creating 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 system that expands the boundaries of programmed systems, which are currently built for specific interactions and which lack the opportunity to create evolving organic interactions. Creating a system that expresses intricate complexities is typically an issue of maintaining high-data fidelity and temporal accessibility.

## 3 SYSTEMS AND COMMUNICATION

Communication between disparate systems requires the development of a mediating language that can address simultaneous modes of complexity through a common dialogue. Contrasting ambient local systems creates context and gives meaning to places, shaping how we use and associate with others in that place. Ecological-biological systems consist of physical structures and living systems, functional processes, ecological mechanisms, and the hierarchical arrangement of these communities. Evolution, interaction, behavior, and development occur within these systems and create communities within which we can monitor water quality, commodity production, environmental problems, and energy production. Social-cultural systems correspond to institutions, organizations, entities, or groups in relation to each other and patterns of behavior. They are based on ideology, knowledge, beliefs, attitudes, values, goals, practices, experience, history, and courses of action. Technological systems negotiate between these, in many cases imperceptibly, and consist of the use and knowledge of tools, methods of organization, techniques, skills, processes, adaptation, and control of the environment.

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). In a single-loop conversation, outputs are determined by filtering, and the system feeds information back into itself. In multiple-loop conversation, new information from each system influences future dialogue and depends upon cycles of response (Haque 2006; Dubberly and Haque and Pangaro 2009). In a true multiple-loop scenario, the output is not predetermined but is predicated on the context that continually changes in response to the systems, and expresses the complexity within that environment. By focusing on context, designers can be more equipped to accommodate different and flexible relationships.

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, architecture, or infrastructure, allowing for rich interactions with the environment and the adjacent systems. Systems coagulate and disperse, looking for contextual stability; the underlying framework is in flux with shifting priorities and goals. (Fig. 1, Fig. 2, Fig. 3)

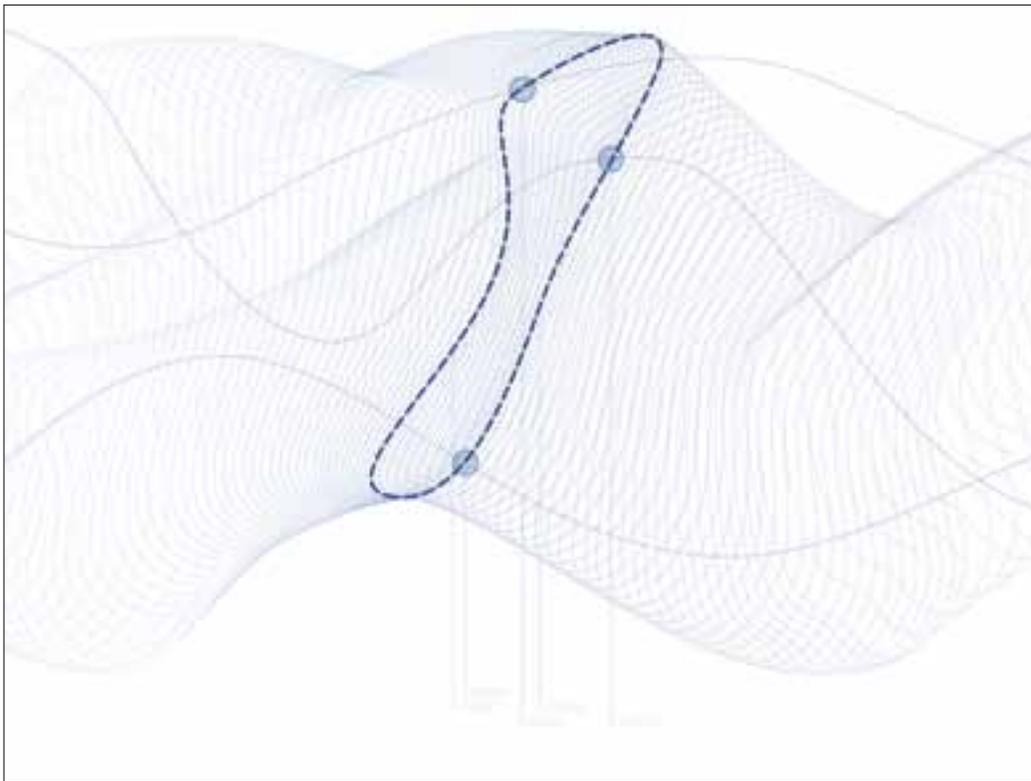


Figure 1 First order loop, singular interaction through temporal field

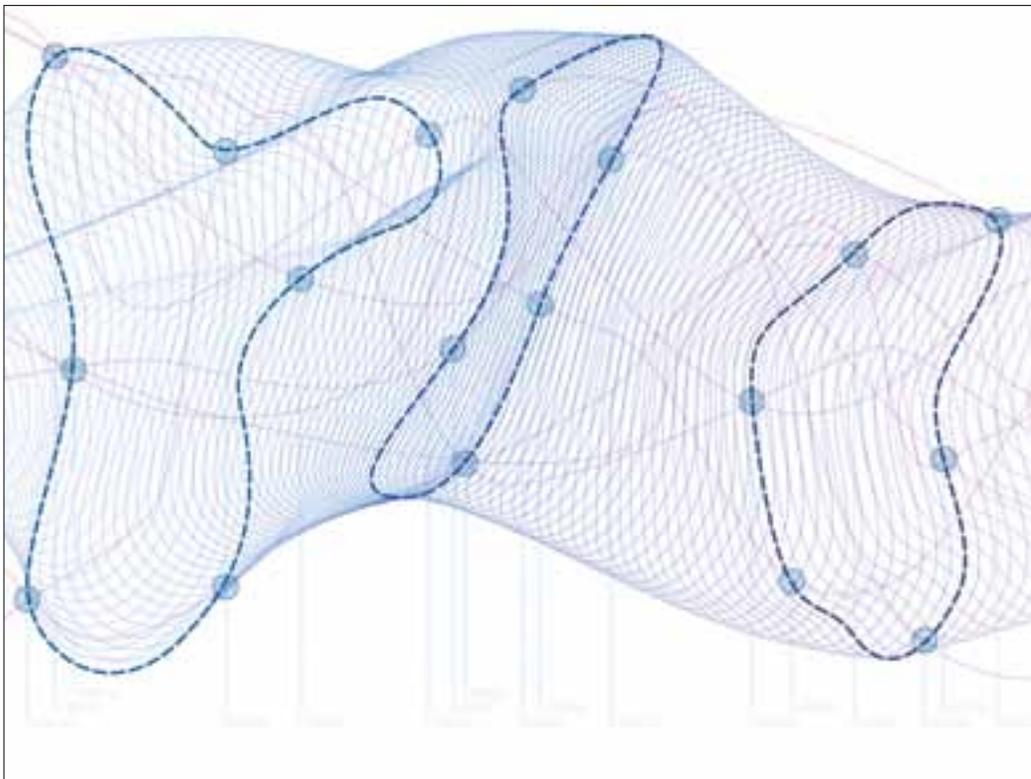
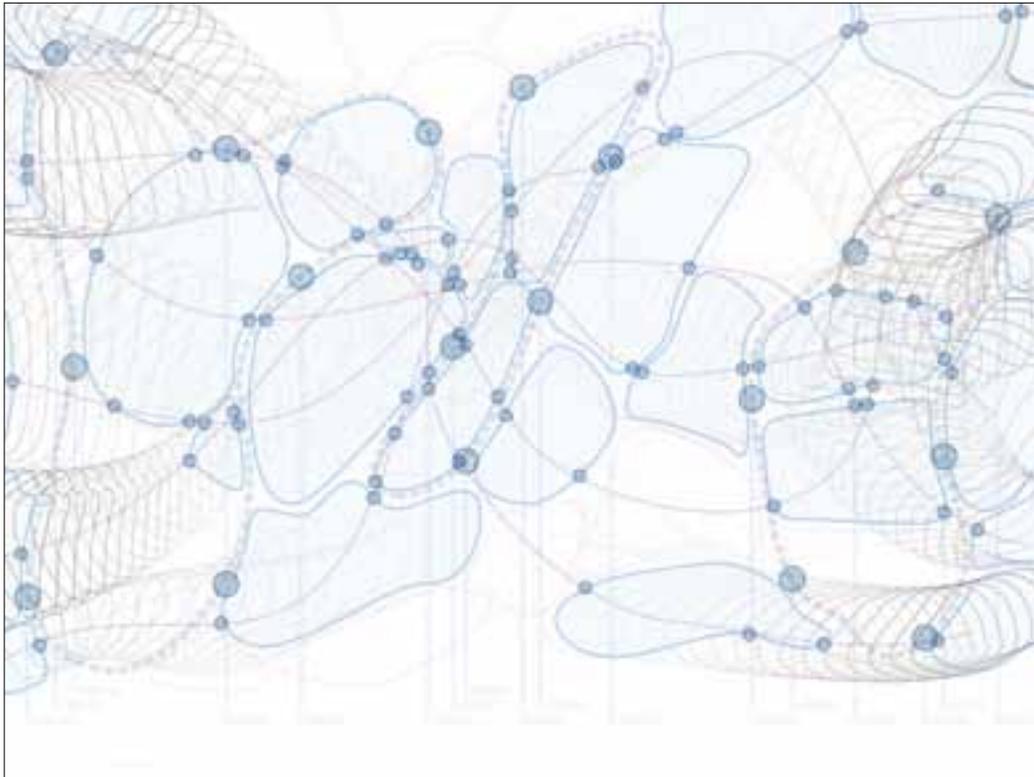


Figure 2 Second order loop, multiple interactions through predefined relationships

MIDDLEWARE

**Figure 3**  
Evolving dialogue,  
interaction fields occur  
with multiple overlaps  
within incongruent tem-  
poral relationships



### 4 CURRENT SCENARIOS

Dialogue creation and 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 that move from action to sensing to comparison to action again, especially between artificial and biological systems (Pask 1972). The investigation between biological and mechanical systems intersects a variety of disciplines, including psychology, ecology, linguistics, anthropology, and visual arts.

Just as theories span disciplines, current interface and interaction scenarios range from the most commonplace to the very fantastic. Environmental monitors control temperature, detect changes in environment quality, and gather scientific data. We have become accustomed to light and motion detectors, thermostats, and smoke alarms. Interest in virtual spaces has given us immersive displays. More recently, the use of GPS has become increasingly pervasive.

Not yet quite hidden in the periphery, designers, architects, and artists challenge the notion of static structures and passive objects. Smart, responsive spaces and reactive objects populate the growing discipline of interaction design. Projects such as Aleph by Begnt Sjolen and Adam Somlai-Fischer and the reactive facade at the Zeil Gallery in Frankfurt gather data and visualize changes in the environment. Bubbles by Fox, Franklin, and Kilian and Tulipomania by Invivia sense and reconfigure to user or environmental presence. These projects represent reactive, single-loop communications where one system influences another, yet the changes are merely a complex set of prescribed outcomes. Abundance by Camille Utterback and Deep Walls by Scott Snibbe record user movement and loop the movements back into the context. The exchange affects both systems yet does not change the intent of either system. In essence, there is communication but the dialogue is static. These projects feign complexity by allowing variation, yet the outcomes are not shaped by process. Haque Design + Research is currently exploring dynamic, responsive, and conversant architecture and multiple projects that also explore operating systems and programming languages as methods for designers to follow. (Open) Burble and Sky Ear both use helium balloons, sensors, and networking to allow the environment and participants to control and reconfigure the structures (Bullivant 2005). Both ecological and social systems are influencing the structure. The social system is influencing the ecological by activating the sensors, but the ecological does not change its intent. Moving toward more process-driven applications, another project of Haque Design + Research, "Siphonophora" consists of devices that monitor context, both ecological and social. As the context changes, the devices react in unanticipated, unscripted behavior.

In all of these scenarios, technology takes the form of an object mediating between systems. The emphasis on object instead of context or environment reduces the complexity of each system and loses 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 architecture shift focus from object to field conditions and address the form between things instead the form of things (Allen 1997). As we design the built environment, addressing the need to control, modify, and interpret ecologies in an effort to enable sustainability and adaptability, we need to be able to compress and exploit the intricacy of context in order to create connections between ecological and social systems.

## 5 ABSTRACTION METHODOLOGY

To understand how abstraction may affect the interrelation between systems it is necessary to discuss a variety of modes of reaction/interaction including sensing, storage, communication, analysis, learning, mechanics, and visualization. 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 they 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?

### 5.1 INPUT

Input provides a stream of data, updating in real time or read from a stored location, typically starting the cycle of sense, process, and output. It should be noted that, based on the aforementioned discussion of conversation, this cycle may continually update or inform its own output, creating an evolving relationship.

Typically, sensing methods are direct interfaces between environmental phenomena and explicit binary data. Sensing is accomplished through a range of technologies created with specific purposes, for example, image, orientation, or temperature recording. The specific purpose for which the device was designed is often expanded through clever modifications to either sensing ranges or application in the environment. An example might be a low-resolution video camera that typically captures images, but which may be used to sense movement or change through an analysis of the pixels between frames, expanding the sensing capabilities through a small tweak in the processing of the input. This versatility is a product of clever modifications to the input/processing relationship, but it is not part of a robustly designed framework for rich environmental systems.

For brevity, this discussion will forgo storage systems, as the relationship between ecological, biological, cultural, or social phenomena and input (sensing) technologies is the real concern. The sensing of each phenomenon must maintain data fidelity across time, location, and content (level). In each instance, these are the recordings that can be abstracted with direct relationships to human or environmental interactions. Through ecological, biological, cultural, or social phenomena, time and location form a basis that allows for meta-relationships to form across systems. For inputs (sensing), it could be concluded that content, location, and time are the abstraction points necessary to maintain a link between ephemeral data and the built environment. Each input type would then have a range of levels for sensed data, such as temperature, and an associated location and time, allowing data to be accessed based on these properties, masking underlying complexities.

### 5.2 PROCESSING

Processing provides the greatest range of possibilities for abstraction between data and systems. Typically, this creates opportunities for analysis or the transformation of data such as comparisons between moments in time or translations from one mode to another (i.e., temperature to location of objects in space [infrared]). In most instances, this transformation or analysis tends to flatten the dataset and reform it for a specific, pre-defined purpose. In the examples mentioned above, interaction may have been high, but the system's framework is one dimensional, creating a defined expression of outcomes.

Processing must also be based around the same abstraction model where data complexity is funneled through content, location, and time. When information is processed, whether through analysis or transformation, each bit of information is abstracted through the previously outlined physical attributes. This mode of abstraction allows data to be accessed at any point during processing, with specific properties used to develop larger-scale contextual relationships. If we assume that processing would occur through control abstraction, similar to sensing, we must also conceive of a fourth point, translation. Translation refers to the transfer of information between disparate systems, and it must be accomplished through a singular model based on data abstraction. This model would create consistency across data, as it is transferred between disparate systems.

## 5.3 OUTPUT

Output is often the most tangible mode of the sensing, processing, and visualizing stream. This may occur in many forms, such as in the creation of imagery, morphological changes to forms, or the automation of systems. The tangible nature of outputs creates a moment that guides or alters interaction, creating perceivable changes in the environment. Because of the physical nature of output, time and location are the critical abstraction points, with context becoming an underlying factor in the equation. While output can also be embedded within the processing stream in order to create multiple transformations, in this discussion, it will be overlooked for the sake of simplicity.

As the output drives a visualization or change in form, location and time data are used to access underlying system complexity and to relate change back to the larger system. By contextualizing output based on time and location, we create a method to relate the moment of system and human interaction back to sensing and processing moments in the same or disparate systems. These output systems may be one-to-one visualizations or large-scale infrastructures for power generation.

## SYNTHESIS/CONCLUSION

Systems that find common modes of abstraction begin to create opportunities for larger interrelationships between disparate ecological, biological, social, and cultural modes. This common layer of abstraction creates an overarching opportunity to create, analyze, or accentuate patterns of linkages, replacing the design of self-contained devices or architectures. Each element expresses itself through physical data and 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 and adaptation and modification across subsets of systems.

A system or infrastructure based on mutual abstraction would be contextually aware through local interactions, such as through responsive buildings, traffic monitoring, and/or environmental sensors. The system would have predetermined 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.

## REFERENCES

- ALLEN, STAN. (1997) FROM OBJECT TO FIELD. IN ARCHITECTURAL DESIGN: ARCHITECTURE AFTER GEOMETRY, EDS P. DAVIDSON AND D. BATES, 24-31. LONDON: WILEY-ACADEMY PRESS.
- BULLIVANT, L. (ED). (2005). 4DSPACE: INTERACTIVE ARCHITECTURE. ARCHITECTURAL DESIGN 1(75): 5-11, JANUARY/FEBRUARY 2005. GREAT BRITAIN: JOHN WILEY & SONS LTD.
- DALE, N.B. AND CHIP WEEMS. (2007) PROGRAMMING AND PROBLEM SOLVING IN JAVA. BOSTON: JONES AND BARTLETT PUBLISHERS, INC.
- DUBBERLY, H. AND U. HAQUE AND P. PANGARO. (2009) WHAT IS INTERACTION? ARE THERE DIFFERENT TYPES? INTERACTIONS, JANUARY 2009. NEW YORK: ASSOCIATION FOR COMPUTING MACHINERY.
- GIUNCHIGLIA, F. AND T. WALSH. (1992) A THEORY OF ABSTRACTION. ARTIFICIAL INTELLIGENCE 56: 323-390. NEW YORK: ELSEVIER.
- HAQUE, U. (2006). ARCHITECTURE, INTERACTION, SYSTEMS. ARCHITECTURA & URBANISMO. BRAZIL: AU149.
- MCCULLOUGH, M. (2005). DIGITAL GROUND. CAMBRIDGE, MA: THE MIT PRESS.
- PASK, G. (1972). CYBERNETICS. ENTRY IN ENCYCLOPAEDIA BRITANNICA 1972.
- ZUCKER, J. (2003). A GROUNDED THEORY OF ABSTRACTION IN ARTIFICIAL INTELLIGENCE. IN PHILOSOPHICAL TRANSACTIONS: BIOLOGICAL SCIENCES, ED. L. SAITTA, 358(1435): 1293-1309. BRITAIN: ROYAL SOCIETY PUBLISHING.