Creating Hybrid Programs and Predicting Their Evolution Through 4D Parametric Analysis

Architects don't know the future; they design it, by making the connection between aspiration and reality. Reality, though, is changing. The move to a service society has shifted clients' attention from the reality of physical space to social networking and computer application technologies. In the wake, a growing inattention to spatial quality, programmatic combinations and infrastructure has reduced the architect's creative participation in the development of space.

MICHAEL EVERTS
Montana State University

This paper proposes a new interactive parametric predictive diagramming tool to program and link an environment associatively and creatively to the resource and use processes that create space, reversing the trend of spatial triage.

Spatial programs are inextricably linked to the resources that support its use. As such, human resources, spatial resources and the client's organizational mission is the context of program. This suggests program and its context are a co-dependent system. However, most programming is done by deductively determining uses and then sizing rooms based on standards. Instead, the programming tool is used to creatively design programs as activities that positively feedback into resources and expand a business' value.

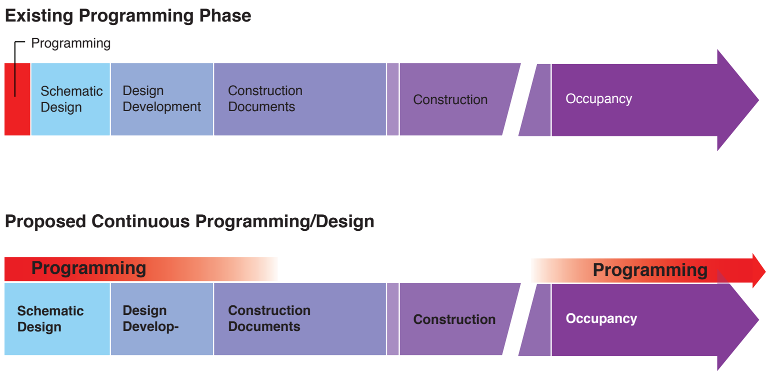
Changing the paradigm of programming from a list of uses, to a process of designing aspirational resource related activities, requires inductive reasoning. The new programming tool uses the framework of the inductive semiotic square (a meaning analysis tool)[1] to register the performance of activities and consider the opportunities of a program on three levels: human resources, spatial resources and use. The tool is modeled in Rhino and Grasshopper as an interactive diagram. It uses tracking data to inform quantitative sliders that in turn dynamically rate the performance of each level. It can be animated over time to test evolutionary developments of program. As a discovery tool, the diagram creates a complex adaptive system for understanding and designing programmatic strategies.

The programming tool creates a process for enriching spatial arrangements and aligning them with creative interpretations of resources. It is not a tool for optimizing space, such as deductive-logic programming matrices do. With the programming tool, space, use and resources can be continuously evaluated and developed. This recasts program as a design activity. The type of space that results is strategic, fluid and evolutionary, offering the client an opportunity to integrate a process of value added spatial development into the project's life.

INTRODUCTION

Traditional programming emphasizes efficiency, aligning spatial adjacencies and criteria in a logical and deductive manner. The formal program is a compilation of the pieces of use, becoming the components for puzzle solving in the next phase: design. In addition, when programming is done, it's done. There is little overlap, or reconsideration of, program in the design phase.

Standard programming is a process of elimination, not imagination. However, if the goal is not an irreducible list, rather, a range of hypotheses, ideas about strategies, relationships and processes of use, then everything changes. Motivations for this type of change are in the rapid evolution of spatial uses, the increasing fluidity of space and the complexity of virtual/real relationships. Inductive reasoning processes embrace this type of complexity. Architectural programming, done inductively, continually and in an integrated manner, could be a process of linking varying dimensions of resources, becoming a predictive technology that simulates, monitors and designs future realities.



1

The proposed inductive programming/design tool is demonstrated in the initial phase of a community engagement service-learning project for a non-profit organization: Eagle Mount, in Bozeman, Montana. They provide therapeutic recreational opportunities for people with disabilities. The project is an outdoor amphitheater-like space. Eagle Mount's need was for a dynamic and participatory event and presentation space. In addition, Eagle Mount's mission is about constant improvement of their clientele's being. These conditions constituted the need for a new approach to programming, one that would be continually engaged with the operations of the space. The programming/design tool that was created, responds to this challenge, and in the process, questions the architect's role in a project, offering an opportunity for continuous architectural services.

THE NEED FOR A NEW PROGRAMMING APPROACH

Exponentially increasing internet development, mobile device use, and other technology advancements have created new mediums of communication, dramatically changing customer relations, productivity, employee expectations, marketing demands, and other fundamental aspects of business. However, these technologies have not fundamentally changed the process and service that architects offer clients. This is perplexing, because the mission of a

Figure 1: Diagram showing strategy of continuous programming.

2

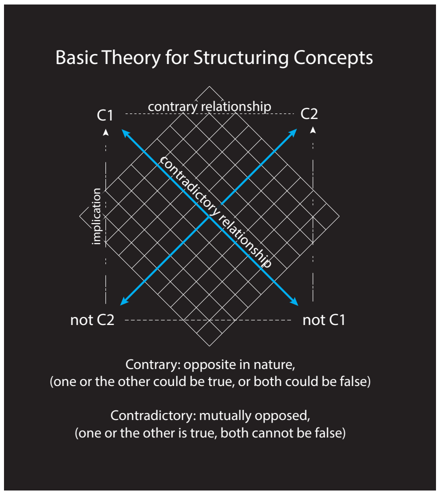

Basic Theory for Structuring Concepts
C1
contrary relationship
C2
implication
contradictory relationship
not C2
not C1
Contrary: opposite in nature,
(one or the other could be true, or both could be false)
Contradictory: mutually opposed,
(one or the other is true, both cannot be false)

Figure 2: The semiotic square.

business is embedded in architectural programming: space is a tool for implementing a business strategy. In addition, these technologies are the main driver of changing spatial uses. The architectural programming process has not embraced the very forces that question its effectiveness. Business, however, has embraced the new environment.[2] They understand and have taken advantage of the new reality, becoming a Complex Adaptive System (CAS): a dynamic system that is able to adapt in and evolve with a changing environment.[3] Architectural programming would benefit by understanding and integrating with this new environment.

MODELING COMPLEX ADAPTIVE SYSTEMS

A CAS and its environment are inextricably linked, changes are coevolutionary with all relevant systems. The environment is relational, as opposed to absolute. Modeling this type of environment requires an open system theory that is inclusive and inductive. The semiotic square is a methodology for understanding these types of relational systems. Greimas and Rastier introduced the semiotic square[4] in 1968 as a method for expanding meaning and interpretation between the relationship of terms. It presents a way of seeing between dichotomies by setting up a framework of gradients and probabilities. The square establishes contrariety, contradiction, and complementarity bilateral relations between selected terms,[5] becoming a frame for the movement of interpretation and speculation.[6]

USE OF THE SEMIOTIC SQUARE IN BUSINESS

Designing or strategizing a service or product, analyzing performance, marketing, and customer service are high “value-added” areas of a business’s service and production chain. Creativity, innovation, and performance in these areas are more beneficial to the businesses performance then increased efficiency in assembly, manufacturing, or implementation.[7] The semiotic square and derivatives of it have been used by businesses as tools to create strategic value. The square creates an inclusive interpretive framework, in which businesses can see an investment’s context, a full range of possible risk ratios, and the investment’s performance.

Visualizing data in the square formalizes a process of matching external environmental circumstances with a business’ internal factors.[8] The growth-share matrix, which uses a type of semiotic square to analyze economic portfolios, helps companies allocate resources and analyze brand marketing.[9] It represents a spectrum of market conditions to situate and analyze stock performance.

Part of a business’ strategizing is uniting and correlating customer values with a business’ offerings: marketing. Persuasion is the main tactic of marketing and it usually involves realignment, or at least a connection, of previously unconnected beliefs and understandings that can be represented by terms. The semiotic square organizes concepts in a manner that creates a field of connecting terms, which makes it a productive framework for creating connections between people’s desires and the services or products that can integrate with them. George Rossolatos, an academic researcher and marketing practitioner, has done studies that suggest the extent to which consumers recognize, internalize and relate to the (virtual) space of a brand is not just an academic question, but a performative approach to financial and customer value.[10]

Use of the semiotic square in business analysis and marketing demonstrates two aspects of how the theory of the square is used in the programming/design tool: 1) it unites a system with its context into an ecology of co-dependent relationships and 2) it allows for varying interpretations and movement in understandings.

Using the semiotic square framework for understanding the systems of resource and use that define program recasts programming as a creative activity.

A NEW APPROACH TO ARCHITECTURAL PROGRAMMING

Space needs to reciprocally engage use in order to be effective.[12] A recent report by the classroom design committee at Princeton University, on the effectiveness of the flipped classroom, concluded: 1) because use is more complex, it needs to involve multiple viewpoints in its programming, 2) because uses are more fluid and change more rapidly, technologies that track use should be utilized for allocating space, and 3) spaces of learning need to be a network of spaces that incorporate schedule dynamics. These recommendations are symptomatic of a need for fundamental changes in how programming is done.[13]

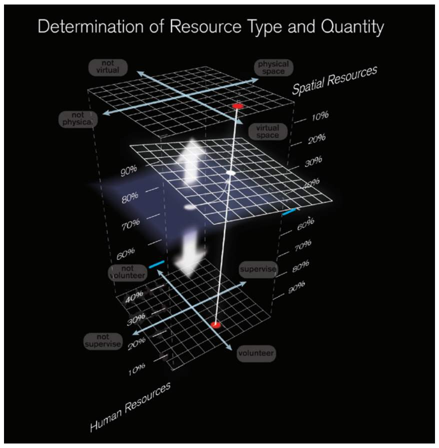
The programming/design tool defines the scope of a project by establishing a semiotic framework for the aspiration of program and resources. The framework takes into account the process of interaction that a user has with the program, the performance of the physical space and human resource approaches.

The tool demonstrates that programming doesn't have to be restrictive. It uses an inductive parametric approach to take into account dynamics of use and multiple perspectives, creating a system that allows for insightful review and creative insight. By using this methodology as a strategy to continually form program that informs design, instead of forming form, it essentially becomes a predictive technology, a tool to design the future.

CHARACTERIZING RESOURCES AND USE

On a basic level, the programming diagram visualizes abstract relationships between resources and use, as a 3D shape. In the diagram, the shapes are constructed by lofting surfaces from points that have been plotted on three levels: human resources, spatial resources, and program performance. As points move on each plane and between planes the shape adjusts. The shape is a cognitive model that conceives resource and programmatic performance.

The identification of values and the field of terms that they reside in, is the first input for each of the semiotic levels. As a human resources example, a company may pay six employees to do a task initially, but development of automation or self-service may, over time, reduce the paid employees to two, shifting the make-up of human resources. The model-space of the semiotic square supports this type of evolutionary interpretation of resources. In the square, the terms of supervision and volunteerism are used as oppositional contraries (S1 and S2). A second binary relationship is set up on a diagonal axis to each of the original terms. For supervision (S1), a contradictory term is set up: not-supervision (~S1); and for volunteer, a contradictory term: not-volunteer (~S2). Not-supervision and not-volunteer are contradictory to their references. The addition of these terms and additional axis introduces the principle of differences: everything is defined by its difference from other elements, i.e., definitions are matters of degree. All three semiotic planes operate similarly, using contrary and contradictory relations to set up a field for interpretive possibilities. The three planes are stacked vertically, resource interpretations are on the top and bottom and use performance is in the middle. The distance between the three planes indicates the ratio of their reliance of use on one resource or another. In the visualization, the closer the use plane is to one resource plane or the other is an indication of how critical that resource is to the activity.



3

Figure 3: Programming/Design tool framework.

4

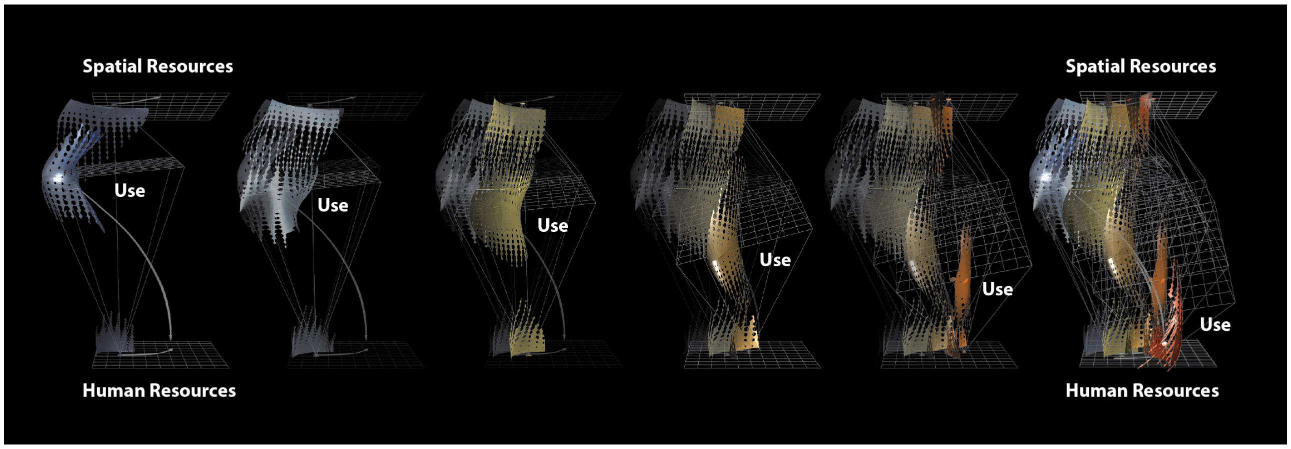


Figure 4: Evolving resources use and programatic development.

The relationship between human resources, spatial resources and use performance is an associative and codependent system. Ideas or interpretations in one category adapt within an environment, or model-space, that the other categories have created. This interdependent triad of relationships creates a 4D framework. It essentially is a scatter plot describing a complex programmatic environment. It unpacks the relationships of program and resources into a comprehendible visualization that displays the component parts in relation to the whole.

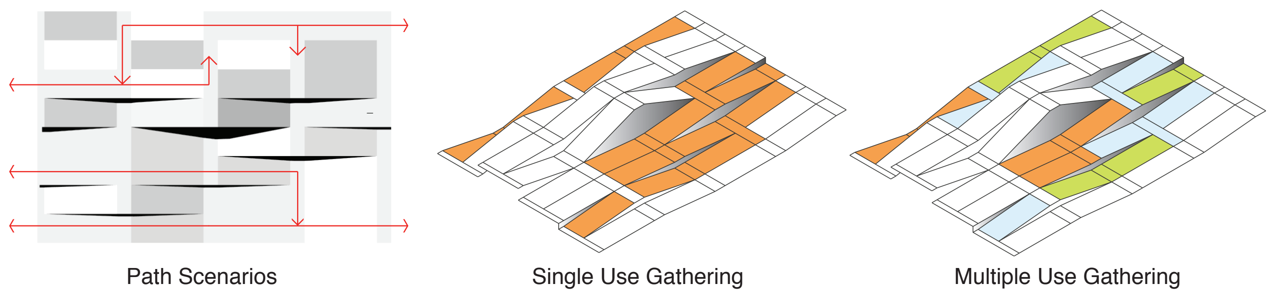
PROGRAMMING AS A DESIGN ACTIVITY

Using strategies instead of lists, casts program differently, instead of puzzling together pieces of program, the architect choreographs four-dimensional sequentially developing space that is integrally linked with resources. After the initial phase of selecting terms and creating strategies of program, the programming/design tool is used to generate spatial diagrams that can inform the physical manifestation of so much strategy. For the Eagle Mount amphitheater-like space, the tool helped conceptualize path strategies. Some paths run through the space, some end in the space and others are re-directed. The color-coded isometric view shows how event gathering (shown in orange) could manifest itself related to these spatial strategies. The diagrams are not images of formal design proposals, rather they are strategies. Instead of lists, inductive programming creates: 1) diagrams of performative connections, 2) references for resource and performance evolution, and 3) design strategy diagrams. These guides enter the design phase, staying open to interpretation and change. After construction of the project, the informed framework becomes a strategy for monitoring the success of the spatial system and direct evolution.

CHARACTERISTICS OF THE DESIGN FOR CONTINUOUS PROGRAMMING

Space that promotes and anticipates creative development is fundamentally different than space that is designed for a limited use and then modified for new uses. The former is inspirational and suggestive, while the later is accommodating and limiting. The benefit of predictive and aspirational space is that it engages the user in a meaningful and collaborative way, because the user becomes a controlling part of the space making. Their input, which may be gathered through surveys, tracking, and monitoring, informs the use area (middle level) of the programming/design tool as an external factor to be balanced with the internal factor of resources. The nature of this approach creates a responsive space, because use isn't compromised, it's composing. Being more aligned and in control of the future encourages a smoother flow of spatial development.

Trend reading, done by analyzing, and predicting the use of space, is done instead of the more standardized approach of responding to change requests from users, which ultimately generates dissatisfaction because of the resources and time involved to make unanticipated changes. The concept of trend reading and its implications, on a theoretical level, are demonstrated in the use of pitot-static system aviation instruments. The airspeed indicator, altimeter and vertical speed indicator all operate by measuring the difference in static air pressure and "ram air" pressure, using a series of corresponding air capturing ports located on the exterior of the plane. When reading the instruments you have to be aware that they are registering a past condition, not the present situation. Therefore, you have to monitor the instruments and read them as trend indicators. Projecting the vector of change, you then adjust the plane accordingly, flying in a smooth consistent manner. Reading the instruments as an indicator of a current condition and then trying to adjust, produces continually futile adjustments and uneven flight. Trend reading and anticipatory action makes processes smoother because you are creating the future you are going into.



5

One of the basic concepts of the physical space that is designed using the programming/design tool is that it incorporates spatial margins for contingent activities. The periphery related activities occurring on the edges of the main space, whether they are overflow or staging, can be monitored, analyzed and used to forecast an evolutionary direction for the main space.

ADDING VALUE TO THE CLIENT'S BUSINESS

One of the benefits of the programming/design tool is that it creates intellectual capital for the architect. The new architectural capacity generated by the approach can be offered to the client as a value-added service. The Stan Shih Smile Curve and the Cobb Value Curve offer a strategic way of thinking about where this service can be of the most value to a client. The Smile Curve posits that the lowest value-added area in the value chain of a business' operation is the manufacturing or implementation phase. The real value areas for the client is in the initial research/design phase and the end use phase. In his book *Implementing Value Pricing: A Radical Business Model for Professional Firms,* Ronald Baker demonstrates how these curves apply to the services that an IT firm offers.[14] The phases of IT service are: Determining Value to Solve; Scope Development; Implementation; Go Live; and Ongoing Support. Determining Value and Ongoing Support are the highest "value to customer" areas of the Smile Curve. The services of an IT organization are conceptually similar enough to architectural services that it is a useful reference. Therefore, an architectural programming/design approach that emphasizes these value areas of the curve will create the most value for the customer and most need for the client to have an architect, using the programming/design tool, involved.

Figure 5: Spatial strategy diagrams.

Although developing the scope of a project is typical for an architect, the client has usually predetermined that a capital project will add value to their offerings. The programming/design tool, because of its big picture and comprehensive nature, offers an opportunity for the architect to be involved in the pre-scoping phase of the project. It can be used to analyze various scenarios at the "Determining Value to Solve" stage of a project because it helps inductively consider the widest range of possibilities. Similarly, at the "Ongoing Support" phase of a project, the programming/design tool utilizes customer and use monitoring (qualitative and quantitative) of a space, holistically linked with resources to progressively and beneficially evolve space.

CREATING A NEW SERVICE FOR THE ARCHITECT

The architectural services contract for value-added services related to continuous programming/design would be fundamentally different than traditional fee based services. Delivery of a service to produce a product (building) is not the same as a service to design, continually monitor and develop resource integrated spaces that increase the value of space for people. The success of the first, building design as product, is measured by how well the process of its creation goes and how well it works for program. The success of continual programming/design would be measured by progressive improvements in performance related to established benchmarks of value. Since the programming/design tool is closely tied to the areas and methods that the client adds value in their business, it makes sense that an architectural contract for the service would relate to the value added. Hourly billing contracts and fixed fee contracts have little relationship to the value of the benefits received by the client.[15]

The typical architectural fee for services is a percentage of the construction cost (fixed fee). Fixed fee contract structures incentivize reduced resource spending and increased fee, creating an architectural business environment that is constantly balancing the quality of service with service delivery expenditures. Additionally, the value of the service is usually based on speed of delivery and budget savings. Interestingly, the value added by architectural services is mentioned only once in the 2007 AIA B101 contract,[16] as a benefit of reviewing design criteria with the owner.

Fee arrangements for services related to the programming/design tool would be structured to incentivize improved spatial quality, strategic resource use, and the beneficial development of other influential components of the designed environment. A fee based on a relationship with performance would have several advantages. First, it would reduce risk for the client and second, it would realign economic incentives for the architect, the fee structure would advocate for the beneficial performance of space. The alignment of fee and profit to the performance of space with regards to use and resources would recast how the value of service is measured and architects are compensated.

CONCLUSION - PROGRAMMING AND DESIGNING THROUGHOUT THE LIFE OF THE PROJECT

The life of a project is influenced by a myriad of internal and external factors. Internally, there are changes in leadership, employees, and business missions. Externally, a change in the economic environment, customer and patron needs, competitor forces, and other dynamics directly affect the performance, requirements, and direction of a project. The dynamics of these forces occur on several dimensions: as changing hierarchal ratios of cause to effect; as factors that the business has the ability or inability to engage and respond to; and as territorializing or deterritorializing factors,[17] making the business stronger or weaker. All of these forces and factors have a relationship, more or less direct, with a real space, a real time, and actual people. Therefore, the design of performative space is critical to its success and enjoyment. This is also why continuous inductive programming would be beneficial. The tool is a framework for a continuous engagement with internal resources and external customer interactions. The tool is opportunistic in its use of linked data sets, that can then play large roles in generating new insights. Also, because it is a creative way of visualizing data, it provides understanding for humans, who are far better than computers at seeing patterns, enabling a process of creating new knowledge.[18]

Although some programmatic components of a project may be straightforward and easily developed using deductive programming, an increasing number of programs are not adequately addressed with historical spatial typologies and traditional architectural methodologies. In addition to design issues, the time and effort allowed for the programming phase is being reduced as a result of tighter time frames and reduced budgets for infrastructure. These forces are creating a need for, and no time to, creatively program a project. The programming/design tool has the capacity to double down on creativity and time because it expands the performance of program and re-conceives the activity of programming as a progressive and continual design strategy.

ENDNOTES

1. A.J. Greimas and F. Rastier, *The Interaction of Semiotic Constraints* (Yale French Studies, no. 41, 1968), 86-105.
2. Vinnie Mirchandani, *The New Polymath: Profiles in Compound-Technology Innovations* (Wiley; 1 Edition, June 2, 2010), xviii.
3. Serena Chan, *Complex Adaptive Systems* (EDS. 83 Research Seminar in Engineering Systems, October 31, 2001/November 6, 2001), 2.
4. Louis Herbert, *Tools for Text and Image Analysis: An Introduction to Applied Semiotics* (online eBook, Texto! 2006).
5. Timothy Lenoir, *Was that Last Turn A Right Turn? The Semiotic Turn and A.J. Greimas*, Configurations Vol. 2 (1994): 119-136.
6. Herbert, *Tools for Text and Image Analysis: An Introduction to Applied Semiotics*.
7. Deborah K. Elms, *Global Value Chains in a Changing World* (World Trade Organization with the Temasek Foundation and the Fung Global Institute, July 17, 2013), 37.
8. Philip Selznick, *Leadership in Administration: A Sociological Interpretation* (Row, Peterson, Evanston Il. 1957), 68.
9. Bruce Henderson, *The Product Portfolio* (Consulting Group, 1970), 89.
10. George Rossolatos, *Applications, Implications and limitations of the Semiotic Square for Analyzing Advertising Discourse and Discerning Alternative Brand Futures, Signs*, International Journal of Semiotics Vol. 6 (2012), 78.
11. " Layouts: Giovanini Commons at Mendoza College of Business," last updated May 21, 2007, http://www3.nd.edu/~gcommons/use/layouts.html.
12. Mung Chiang, Paul LaMarche, Diana Fuss, Alison Gammie, Maria Garlock, Bo Honoré, Alison Isenberg, Simone Marchesi, and Chris Skinner, *Report of the Classroom Design Committee* (Princeton University, 2013), ix.
13. Chiang et al., *Report of the Classroom Design Committee*, x.
14. Ronald J. Baker, *Implementing Value Pricing: A Radical Business Model for Professional Firms* (John Wiley & Sons, 2010), 68-70 .
15. Richard Reed, *Billing Innovations, New Win-Win Ways to End Hourly Billing* (American Bar Association, 1996), 14-16.
16. AIA 2007 B101, *Standard Form of Agreement Between Owner and Architect*, Article 3.1.1.
17. Manuel DeLanda, *A New Philosophy of Society, Assemblage Theory and Social Complexity* (New York: Continuum, 2006), 13.
18. Jonathan Shaw, *Why "Big Data" Is a Big Deal, Information Science Promises to Change the World* (Harvard Magazine, http://harvardmagazine.com/2014/03/why-big-data-is-a-big-deal, March – April 2014).