Procedural Design

Sean Ahlquist
University of Michigan, Taubman College of Architecture and Urban Planning

Procedural design is often classified as a computational approach relying upon a set of instructions that, when used in a particular sequence, are the generators of form. While within this framework certain methods may be iterative and cyclical, procedural design often denotes the construction, conceptually, of a linear solver. The work documented in this section, though, shows a significant evolution of this approach. Intelligent systems are formed in which computation is given the freedom to absorb, interpret, and respond within the sequential set of procedures, thus shifting from linear logics to networked ones. This is addressed through papers that discuss the language through which such processes are enacted and explore the emergence of a built architecture through dynamic logics of design computation.

Ludwig von Bertalanffy established the sequencing of a feedback system as a part of General Systems Theory.[1] This laid the groundwork for the semantics and structure of procedural design. In essence, it is a methodology that is used to test the relationships of parameters through iteration. Bertalanffy classified the components of a feedback system by *count*, *species*, and *association*. These have been superceded as the metrics of design space, since procedural operations allow for the exploration, testing, and refinement of ideal parametric relationships. In this application, the feedback system is an active agent of design exploration.

Traditionally, procedural design has offered means of testing the relationships of parameters, but the work shown in this section demonstrates an evolution of this approach. Procedural processes become an active agent for resolving the relationships of *systems*. In Gerber's "Multi-Agent System for Design" and Savov's utilization of gameplay, logics of fabrication shift from defining constraints to being exploratory agents for design ideation and the construction of architectural systems. Human inflection becomes an operational procedure in Johnson's work with SIFT algorithms and Sanchez's "Combinatorial Design." Both exploit the iterative facet of the feedback mechanism to scan massive datascapes while interjecting the transformational feature of human intuition.

Emergence is an innate function of a properly constructed procedural design process. Through works such as Andréen's "Large Swarms of Simple Robots" and Rusenova's "Aggregate Architectures," it is possible to see a shift from merely seeking emergence to enabling machine intelligence to learn from and respond to specific emergent behaviors. Koschitz, through the visual programming language "Beetle Blocks," proposes a platform that simplifies the construction of procedural processes and the conceptualization of emergent design. Davis's incorporation of building evaluation and Smith's method for building automation subsequently extends, on a grand scale, the scope of procedural design. Data is either a live agent orchestrating building systems or an encapsulation of the live agents—the building occupants—to re-inform successive design explorations.

Collectively, the research in this section brings a valuable ambiguity to the finality of the feedback system. This reverberates into processes and modes of design, where the work provides a clear indication that architectural form is the enmeshment of systems, not just a collection of geometric constructs.

1. Ludwig von Bertalanffy, *General Systems Theory: Foundations, Development, Applications* (New York: George Braziller, 1969.

Procedural Design