The 2011 ACADIA Conference theme positions architecture’s present condition as “post-digital.” This term calls to question the relevance of an organization like ACADIA, whose focus is precisely on the digital in architecture. Paradoxically, it is the work of ACADIA and digitally-oriented designers that has facilitated the gradual dissolution of the digital. In reality, we of course know that the computer has never been more present and relevant to architecture than it is today. The barriers to entry are lower than they’ve ever been. Digital technology is more pervasive and flexible. But, its ubiquitous integration has rendered the computer itself, with its legacy of opaque user interfaces and inaccessible language, ostensibly transparent. Through this transparency, an array of designers and collaborators previously relegated to the sidelines of computation discourse are now active participants in it. The papers in this session point to five ways in which the boundaries between the digital and non-digital, between architecture and non-architecture, are quickly eroding, and thereby allowing each to influence the other in profound and surprising ways.

First, visualization software increasingly thinks like designers think. How we interface with it closely mirrors how we draw and build with our hands, how we think about space and material, and how we manage complexity through the integration of words, numbers, and lines. In their paper “Potentials for Multi-Dimensional Tessellations in Architectural Applications,” for example, authors Celento and Harriss discuss the reasons why theoretical mathematical explorations in multi-dimensional space have not gained traction in architectural venues. One of those reasons is what the authors call “the challenge of visualization.” Today, robust computational programs allow for the visualization of higher mathematical concepts that were previously inaccessible to fields outside of mathematics. Concepts that were opaque and discipline-specific before can now be visualized through software that most designers have installed on their computers. This allows those concepts to infiltrate and contribute to the discourse of other disciplines, like architecture.

Second, there is a transparency to current software that allows architects to understand what is happening under the hood and how to control it. This means that they can focus more on “why” they make certain decisions and less on “how” they get software to work. For example, in “Free-form Grid Shell Design Based on Genetic Algorithms,” authors Dimicic and Knippers develop a methodology for applying a statically efficient, optimized grid structure over any pre-defined free-form shape. This opens building shape back up to the possibility of being defined not by what the computer dictates (typically referred to as “form-finding”), but by issues external to the computer, like program, site, or more qualitative conceptual strategies.

Third, contemporary parametric software allows for a combination of bottom-up and top-down design processes. Up to now, most computationally-intensive design
processes have privileged a bottom-up approach where form is "grown" from a set of mathematical equations. This form may or may not address issues of site or program in a way that works for a particular project. This method's potential efficiency, constructability, and clarity are its advantages. But new developments in software allow for work to emerge from a combination of bottom-up mathematical considerations as well as top-down non-mathematical considerations, where the ratio of one to the other is fluid and project specific. The paper "Irregular Vertex Editing and Pattern Design on Mesh" by Kobayashi develops a tool for designing components for a given form that emerge from a more top-down starting point. It takes a form that may have been developed from a study of prevailing breezes or the shape of a site and imbues it with the same level of geometrical and structural efficiency as a form that was developed from the bottom-up.

Fourth, advancements in design software allow for considerations like function, program, or use pattern to be treated as data. In other words, software can anticipate behavior and model systems after it. For example, in the paper "Tetrahedron Cloud," the author develops a stochastic system that embeds particular tendencies in a component's behavior. Their system eschews a model of optimization, where a "best" solution is pursued, in favor of a series of behavioral studies in which every outcome of a deployed system, with its embedded rules, presents a new scenario of behavioral variation. The architect then chooses an array of outcomes to develop further, rather than a single optimized solution.

Fifth, it is easier than ever for architects to customize software to fit their particular needs or those of a specific project. The ability to program, write scripts, or generate plug-ins that modify existing software capabilities is rendering the computer more relevant. It results in work that can precisely respond to its own particular, nuanced constraints. In their paper "Just Passing Through: Integration in Computational Environmental Design," authors Davis, Tsigkari, Iseki, and Aish conclude that integrated approaches to highly complex design problems require a "significant degree of custom programming to solve a specific class of problems." Further, in "Seeking Performative Beauty," Dominguez, Schimek, and Wiltsche write about the development of a software tool that seeks to reveal the hidden complexities in computational geometry and thereby render the whole process more manageable.

In fact, in virtually every paper from this session, the authors have cracked open their software to either render its processes more visible, or to customize its capabilities, or to bring digitally-challenged collaborators into a position of greater participation with a project. This is perhaps the most intriguing finding presented in these six exceptional papers. The digital interface has eroded to the extent that we can now make software our own. Perhaps architecture is moving into an era that can be categorized not as "post-digital," but as "plus-digital." The definition of what is or is not digital is no longer relevant. It is now the material residing outside the field of computation that is returning to exert its impact on the computer, rather than the other way around.