Designing for Constant Change: An Adaptable Growth Model for Architecture

Taro Narahara
Designing for Constant Change:
An Adaptable Growth Model for Architecture

Taro Narahara

ABSTRACT

Design of universal components that can tolerate technological, environmental, and circumstantial changes over time is a challenge for an architect. In this paper, I would like to propose a scaled prototype of architectural components that can reconfigure themselves into globally functional configurations based on feedback from locally distributed intelligence embedded inside the component. The project aims at demonstrating a design system that can respond to dynamically changing environment over time without imposing a static blueprint of the structure in a top-down manner from the outset of design processes. The control of the subunits are governed by the logic of a distributed system simulated by the use of multiple microcontrollers, and appropriate geometrical configurations will be computationally derived based on physical-environmental criteria such as solar radiation from various sensors and social-programmatic issues.
1. Introduction

Recently, the physical scale of buildings and the complexity involved in building programs have been increasing at unprecedented levels due to globalization and economic development. For example, housing projects for thousands of inhabitants with multiple occupancy types have been emerging in urban areas. I speculate that there will be more demands for buildings to be able to change over time to adapt to newly emerging needs for different qualities and quantities of architectural information. Within the limited allowable growth areas in dense urban settings, this need to accommodate new demands will increase the pressure to design more adaptable buildings that can avoid unnecessary drastic future demolitions and reconstructions. As a solution for coming future conditions, I propose active incorporation of the “bottom-up” approach into our existing models of building designs.

2. Background

In today’s building design methodologies, we thoroughly study all kinds of future requirements and potential changes for buildings before construction. For some building types we expect fewer future changes but some require more. Besides seeking to minimize the need for few possible future alterations or maintenance work, these design methods aim at producing completed buildings that can tolerate as many future conditions as possible from the outset of their design. I will call this type of design approach “Top-down.”

On the other hand, some buildings do not include a comprehensive solution for all the potential scenarios of the future from the beginning, “as is.” Instead, some of those buildings possess systems that allow them to adapt to future changes over time by altering their designs spontaneously based on simultaneous feedback from a number of simple entities (or agents) inside the system. These locally distributed feedback systems can collectively work to find globally functional building solutions that will cater for many different environments. These methods could be effective where there is no deterministic and analytical means to derive a solution. As a natural consequence of adapting to radical population growth, sometimes these characteristics can be seen in low-cost housing developments in less regulated zones with no supervision by professionals, like the Kowloon Walled City [3] (Figure 1). In general, many spontaneous settlements in Third World cities are often regarded as undirected, chaotic, and negative; however, their informal growth patterns exhibit transient and flexible characteristics that can help us compensate for a lack of robustness in our current top-down planning methodologies. I will call this type of design approach “bottom-up.”
Emergent behaviors display many characteristics that are similar to the aforementioned bottom-up approach. Products of emergence can be found in many natural organisms: flocking of birds, collective constructions by various social insects, pigmentation patterns of cells in animal skin, formation of dunes by sand particles, and so on. Today, many scientists are attempting to simulate, employ and adapt the advantages of these self-organizing systems in nature, (especially their robustness, flexibility, adaptability, and concurrency). In the bio-inspired computation field, the application of ants’ foraging behaviors to network systems and the control of multiple robots (swarms) are a few examples of using distributed controls to gain more flexibility and robustness in systems and to achieve a bottom-up approach [2]. A question arises from the observation of the recent technical developments inspired by distributed systems in nature: is there any place for such applications in architecture?

In this paper, I would like to investigate the potential of actively incorporating this bottom-up approach into our architectural design models. I am aware that this strategy will not be the universal solution for all building types. In fact, many building types have very clear scope for future scenarios and no particular future adaptations are needed. Existing methods seem to provide efficient controls for a majority of current architectural projects. However, beyond the scale of buildings, we have witnessed some bottom-up design processes in many formations of cities over longer spans of time [1]. Many growth patterns of cities display a heuristic approach, and it is difficult (or frankly impossible) to foresee all the consequences and results of each development many steps in advance. It is evident that the bottom-up approach starts to become desirable where we do not have total control of the entire system due to its scale or complexity. Once unpredictability and complexity of systems reach beyond a certain critical limit, top-down deterministic solutions alone may not be able to respond to all the potential future conditions.

3. Methods

3.1. Distributed systems

Distributed control is one technical strategy to realize a feedback process inside a bottom-up system, and this strategy can be applied to the control of multiple structures. Inputs for this feedback system are fed from
separated nodes and can be triggered by participation of independently acting agents with some intelligence. The entire system’s behavior is a result of feedback from multiple distributed intelligent sources, and such a system is often called collective intelligence. In this project, I would like to offer a novel method to formulate a hypothesis by proposing a scale model prototype of architectural components that can assemble themselves into globally functional configurations based on feedback from locally distributed intelligence embedded within. The project aims at demonstrating a design system that can respond to a dynamically changing environment over time without imposing a static blueprint of the structure in a top-down manner. The assembly and control of the subunits are governed by the logic of a distributed system simulated by the use of multiple microcontrollers, Arduino, an open-source electronics prototyping platform ([9] from http://www.arduino.cc), and appropriate geometrical configurations that are computationally derived from locally communicating components.

Each component is connected at the joint with two degrees of freedom made of a pair of servo motors, which offers sufficient variations in configurations when multiple components are forming clusters. This mechanical setting allows them to configure all possible patterns in orthogonal geometry as long as all components are contiguous in series. Each microcontroller is responsible for the control of several adjacent components (only two controllers are used for the current experiments), and neighboring controllers can, in principle, send and receive their states, such as their orientations in degree, levels of solar radiation, thermal conditions from various sensors, architectural programs of components, and so on. Based on feedback among the neighbors (informational exchanges

Figure 2. Left: Concept of distributed control and 2 degrees of freedom at joints. Right: Established physical and virtual synchronization using the Virtual Controller Software written in Java/Processing.
among the components), each microcontroller will send a signal to change its components’ states in order to locally optimize its condition. Consequently, multiple interactions among the locally defined actions lead us to gain globally functional configurations rather than a final form being imposed in a top-down manner.

To construct the present system, I first developed a graphic user interface which can display and control four components’ movements via Java/Processing. This allows bi-directional communication between physical and virtual environments (Figure 2).

As a starting point of this experiment, each component was populated with a photodiode (light sensor) to measure the level of solar radiation at the panel surface (Figure 5). Sensors returned the values to assigned microcontrollers based on the current orientations of the components, which can be varied by rotations of the motors in tandem at the joint. If components change their configurations with different rotation angles, then obviously the results from the four sensors will have different values. There are four panels connected in series at three joints, so that there are, in total, six motors to govern all the configuration patterns. In order to find better configurations to maximize average solar exposures for each component’s panel surfaces, we have to find ways to derive better combinations of the six rotation angles of the motors. This framework for the problem resulted in the use of multi-dimensional optimization algorithms.

3.2. The nelder-mead method: Physical implementation

The Nelder-Mead method is a commonly used nonlinear optimization algorithm and is often used for minimizing an objective function in multi-dimensional space. In the case of this experiment, the search is in
six-dimensional space formed by independent variables of six angles of motors, and the mechanical components literally become a physical objective function to provide values (average values of four light sensor outputs) which need to be minimized. Firstly, the algorithm configures seven different physical configurations to sample light values for each case, which will form a polytope of 7 vertices in 6 dimensions (using the simplex concept). Then, the algorithm will rank them based on sensor values' feedbacks from the physical machine and search for vertices which provide

![Figure 4. The process of reconfiguration and the search.](image)

![Figure 5. Components autonomously find better configurations to maximize lighting exposure at 4 sensor nodes at the middle of the panel using two different optimization algorithms.](image)
better configurations of new motor angles. The robot will show the best configuration and turn the LED indicator on. The algorithm will repeat the above processes until it stops improving the value above a certain minimum. This method is also nicknamed “the amoeba method” (see figure 6) since the way the polytope finds the new vertices and moves towards a better solution inside the multi-dimensional space is similar to the movements of amoebas. In case the directions of light sources are altered, the system will dynamically react to the changes and will run the algorithm based on the values returned from the new condition.

For searches in two-dimensional space, a polytope forms triangles (3 vertices), and Figure 6 shows examples of amoeba processes applied to a simple 2-D objective function. Each vertex of the triangle (B, G, and W in Figure 7) has a different value for the function F (i.e. \( F(B) < F(G) < F(W) \)). In each iteration of the Nelder-Mead method, the worst point, W, with the largest value from the function will be replaced by a point with a better value by using the Nelder-Mead algorithm. In this case, the objective is to minimize the value of function F. Acquisition of the new vertex is based on reflection, expansion, contraction, or multiple-contraction of the current triangle in 2-D space based on the function’s returning value of the newly defined vertex. The algorithm finds a better vertex using one of the geometrical transformations of the triangle listed above (Figure 7).

There are many types of multi-dimensional optimization methods similar to Nelder-Mead method (NM), such as the Levenberg-Marquardt algorithm. The choice of NM algorithm for the project was merely based on its simplicity of implementation and its geometrically intuitive logic. Other nonlinear optimization methods can be selected for different frameworks of problems in order to gain optimally better performances.

Prior to the Nelder-Mead method, the simpler random search method was tested. In this relatively simpler random search method, an algorithm...
rotates each joint continuously in one direction until it stops improving the assigned sensor values for the joint, compared to its former state. Then it rotates the joint in the other direction to test the improvement. It is a simpler strategy for preventing the system from stagnating at local optima. The results show that the use of the Nelder-Mead method reduces the number of trials needed to find better configurations compared to the simple random search. In addition, responses of reconfiguration to dynamic changes of light source directions would be better to use the Nelder-Mead method (which means that fewer trials are required to obtain appropriate orientations of panels for newly defined lighting conditions in changing environments).

Having physical mechanical components be an objective function providing fitness values is a unique and original approach in this experiment. However, it is debatable whether this approach is practically feasible for large-scale architectural applications or not. Beyond a certain physical scale of application, moving architectural units physically to test different configurations will be inefficient as the weight of the units becomes prohibitive. Furthermore, the numbers of trials that are required to find optimum configurations will exponentially increase as the numbers of components grow. The virtual controller in this project reports the physical orientations of the components. For future explorations to find more practical applications, more comprehensive simulation environments that can virtually estimate structural, programmatic, and environmental fitness, including energy calculations, without physically moving the components, would be desirable. Bi-directional control combining the use of both physical and virtual objective functions will allow error corrections between the two environments and appears to be a more promising approach.
4. Discussion and critique

Development of flexible and adaptable architecture has been a recurrent theme among practitioners. There were several inspirational projects by architects in the past. During the 60’s in Japan, Metabolists introduced megastructures that could constantly grow and adapt by plugging prefabricated pods onto the infrastructural core; however, original visions of metabolic growth and adaptation were rarely realized physically, as the sizes and weights of the pods were practically very difficult to reconfigure. In the 90’s, construction automation [7] by general construction companies in Japan shed light on the concept of self-reproduction in architecture: architecture that can produce architecture. However, there was still a clear division between assembler and assemblee relationships. Mechanical components that could produce buildings were far from actual livable architectural spaces. Thus they could only repeat, producing an identical or similar building at a time, and no future adaptation was available. Finally, some of the speculative researches by computer scientists in recent years have started to show viable prototypes representing self-reconfigurable systems using swarm robotics [4], [5]. In architecture, it is our responsibility to consider how these noble technical concepts can be applied to enhance our living spaces, and our design processes may well be on the brink of a necessary transition from conventional methods to methods that require evolutionary processes.

For this project, it is worth noting that the Nelder-Mead method is a heuristic. A heuristic method is a solving of a problem by iterative processes of trial and error and is intended to find optimal solutions rather than to find a single deterministic solution. Traditionally, we have a tendency to seek and construct an analytical problem-solving framework due to the invisible pressure to find the final and the best solutions. Finding a single solution, static in time, that satisfies various clients’ needs has been a typical architect’s responsibility, and generating comprehensive plans as a blueprint is normally anticipated. Conventional design problems in architecture may be reducible to an analytical problem-solving framework more easily, compared to finding optimal solutions over time every step of the way. As can be seen in this project, calculations of dynamic reconfigurations for gradual growth of structure can be fairly extensive. This fact implies that the deterministic analytical means are less adequate where we need concurrent solutions for dynamically changing conditions, and we may need to rely on heuristic search as the complexity of the project increases.

As for the implementation in architecture, it is extremely important to consider not only physical and quantitative issues but also internal and qualitative issues. Environmental issues such as lighting can be quantified and may be resolved to some degree; however, more programmatic issues relating to logistics of architectural planning will need to become a new focal point of research among our profession. An aim of this paper is to
show how the self-reconfigurability can be incorporated into architectural design processes in order to realize an adaptable growth model through an extremely simplified working conceptual prototype.

The experiments in this paper are not at the stage of providing a direct application to existing architecture. In principle, the numbers of components can grow and reconnect to expand the structure to respond to increasing and differentiating spatial demands. Solar radiation was one criterion selected for the reconfiguration; however, various different criteria can be technically implemented in the system. For instance, affinities among various occupancy types and their adjacency relationships can be used as a selection and morphing process of various architectural programs. This selection process can be achieved by cellular automata-based logic, similar to the method introduced in [9]. Allocations of different architectural programs such as residence, office, and retail spaces can also be optimized virtually by the use of various simulation programs and physically by tracking the movements and behavioral patterns of occupants in the future. Further investigation will be required for implementations of additional architectural applications. The intention of the project has been to clarify the concept of dynamic form-finding technique based on the bottom-up approach through a relatively simple and clear form of prototype.

5. Conclusion

In architecture, few structures have ever been built or conceived based on the active application of distributed systems. Excluding some of the emergent formations of cities on larger scales over longer spans of time, adaptation of distributed systems and collective intelligence to architectural creations is an uncultivated area of study worthy of investigation. This project is one such effort to demonstrate a novel design system through a conceptual physical prototype that simulates the concept of dynamic adaptation in architecture.

Acknowledgements

I would like to sincerely thank my academic advisers, Professor Kostas Terzidis and Professor Martin Bechthold, both at Harvard University Graduate School of Design, and Professor Takehiko Nagakura at the Massachusetts Institute of Technology, for their insightful guidance and constant support during the development of this project.

References


Taro Narahara
Harvard University
Graduate School of Design
48 Quincy Street, Cambridge,
MA 02138

narahara@gsd.harvard.edu