An Agent Based Approach for Evaluation of Free-Form Surfaces

Mahmut Çağdaş Durmazoglu [1], Gülen Çağdaş [2], Hakan Tong [3]
[1] The Institute of Science & Technology, Architectural Design Computing Graduate Program, Istanbul Technical University, Istanbul, Turkey,
[2,3] Faculty of Architecture, Istanbul Technical University, Istanbul, Turkey,
http://www.mimarliktabilisim.itu.edu.tr/index_eng.html
[1] cagdas.durmazoglu@gmail.com, [2] cagdas@itu.edu.tr, [3] hakantong@gmail.com

Abstract. *Designers; architects and artists in general always proceed with one crucial goal in their mind when they design: to be able to materialize the scheme in their mind. There are some tools that can help designers to test their design to see if it will be successfully constructed or functioned as close as possible as the way they imagined or expected; visual simulation models in digital environments of the designs are such important tools for architectural design process, but these models does not provide enough data to determine the design's applicability. Therefore, the need of a tool that would work with the designer in order to determine the possibility of whether the design could be built or not as the way designer imagined, is vital.*
Keywords. *Rain-flow analysis; performance analysis; agent systems; decision support system; free-form bodies.*

Introduction

The aim of this paper is to explain that an agent based system tool – in the sense of a program component – can help designers in the free-form shape generation by analyzing and evaluating the surface in digital media. The problem domain was decided by examining the difficulties within the free-form design domain, such as difficulties concerning the structural components, materials and the form finding process, the research which is explained in this paper only focuses on the difficulties concerning the form finding process of the free-form design.

The desire to achieve an aesthetic condition in design which can only be explained by the irregular curves; which is also used to describe the free-form architecture, have always been haunted the architects throughout the architectural history. The roots of this yearning to capture the flow, movement and to be able to reflect this notion into designs can be traced back to the different architectural eras from Baroque to Art Nouveau. Still, this quest is pursued by a new generation of architects who are bold and experimental in their search of forms and themes like their neo-avant-garde predecessors.

Being able to achieve that aesthetic state in design has become easier with the help of computers. By means of recent developments in the computers and their ascending use in architectural design, there have been serious innovations in the domain of architecture: the most important one among them is the change in the construction techniques, which inevitably led free-form designs to become products of computerized procedures with the benefit of the recent use of digital computer technology. As a result of these developments, most of the contemporary designs are computer generated or about to emerge as a result of the implementations of latest computer hardware and software. With the aid of computer animation software, designers can master complicated forms that they ever dreamed of. Nevertheless, in order to master these irregular forms, first of all one have to be capable of outlining the knowledge about the design and production phases, and some difficulties in this domain. By successfully and clearly outlining and describing the problems in the form- finding process of the artifacts which belongs in the free-form design domain, this lifelong search of architects to design these unusual forms, could finally be realized.

Distribution of the forces on a surface can be used as a key factor in the form-finding process of the free-form bodies to determine the final shape of the design. There are not many practical ways to determine if a design is capable of baring these forces or not. But the most practical one among these approaches is using the rain-flow analysis method to test the performance of the surface under these forces. The hypothesis for the flow of forces in shell structures was used in this research to determine the applicability of the free-form surfaces according to three design flaws: loads on the free-edges, bending moments and drain curves.

Digital form finding approaches

The recent developments in the computer technology have led to an intensive use of digital design environments in architecture. Consequently, architects were started to use these environments not only for their drafting capabilities, but also for developing highly complex geometric forms and shapes. The most common computational design approaches for form-finding can be categorized as: using computational tools to manipulate the shape directly, using external factors to determine the final shape, and using rule structures (Schodek et al., 2005).

Manipulating the shape directly via computational tools in the digital modeling environments, such as, points, lines, splines, lofts etc., was the most common method used by designers to generate shapes. Many digital design environments that widely used in architectural practice can be used in form-finding process by manipulation.

Using external factors to decide the final shape was another interesting approach that was used in form-finding process. These external factors can be a force-field simulation in a computer to generate the form like in BMW Pavilion by Bernhard Franken, forces that simulate the movement and flow of the pedestrians, cars, and busses across the site like Greg Lynn's project for Port Authority Gateway Competition or two systems that are animated to pull against each other in Peter Eisenman's West Side Project.

Another method to generate shapes is using predefined rule structures. The most common approach to use predefined rule structures is defining the buildings' exterior by series of parametrically defined elements such as structural ribs. Shape grammar formulations that are widely used for the generation of plans are another form-finding method based on rule structures. Additionally cellular automata, growth and repetition algorithms, fractals and pattern tessellations are other methods, which are based on rule structures, used in the form-finding process.

DROP was used for evaluating and analyzing the surfaces according to the distribution of the forces. After the analysis, the results can be used to manipulate the shape directly within the modeling environment to solve the problems that are presented by the model and increase the chance of surface's applicability.

Theoretical model of the DROP

The theoretical model of the DROP is developed by examining the knowledge regarding the agent-based systems used to assist design and the distribution of forces on free-form surfaces. While developing the model, design is perceived as a process of finding the alternative solutions then deciding the most suitable one located in the design domain, which is completely controlled by human designer. Additionally, the most important aspect of the design process is perceived as the ability to comprehend the reasons of the problems, especially in the free-form design domain where very little knowledge is presented for the designers. Therefore, assisting the designers not only in the earlier steps of the design where the form finding is in progress, but also providing them an understanding regarding the sources of these problems and how to avoid them has seen as one of the pillars of the model.

Another important factor that shaped the functionality of the model was developing the model, which has an ability to represent the results of the rain-flow analysis to the designers. Architects are known to consider different aspects of the design always in relation to the other issues. According to Lawson, all different considerations run through the head of the designers simultaneously and the designers currently jump from one design aspect to another (Lawson, 1997). Therefore, the model aims to provide useful design information with the analyses of the surfaces in the form finding process of the free-form design domain so the designers could consider other aspects of the design like materialization aspect. Having results regarding the applicability of the surface early in the design process can make designers more aware of the repercussions of their decisions about the design.

The method that was used in the model to analyze free form surfaces according to the distribution of forces is rain-flow analysis. Rain-flow analysis is based on the hypothesis: "Like a rain flow loads will flow along curves with the steepest ascent on the shell surface to its supports" (Borgart, et. al., 2005). Since the mechanical behavior of free form surfaces, which are irregular curved surfaces, and shell surfaces can be explained by illustrating the flow of forces on the surfaces that are subjected to their own weight, it was decided that, analyzing the free-form surfaces with the help of a tool which has the ability to execute the rain-flow analysis, should clarify the relationship between form and force in the form finding process of free-form surfaces.

DROP was developed to implement the simulation of the rain-flow analysis, in order to obtain the information regarding the analyzed surfaces' mechanical behavior. In other words, the simulation of the rain flow analysis is only illustrated the distribution of the forces on a free-form surface which is subjected to its own weight. Results of the simulation of the rain-flow analysis on a free-form surface are presented as still images, which possess information whether the subjected surface is applicable in terms of its ability to distribute the forces that are emanated by the surface's own weight. Information regarding the surface's applicability that is presented in the still images is obtained by using the rain-flow analogy in a digital environment (Maya) and simulating the flow of the particles on an irregular curved (free-form) surface.

Agent behavior in the model

Knowledge and its representation are the most crucial features of the design systems. According to Brodie, knowledge representation is considered as a prior condition to the development of knowledge support tools (Brodie et al., 1984). Case-based, knowledge-based, expert, and other similar systems, which depend on the definite symbolic representation of knowledge, could be examined in two directions regarding the support they provide for designers; automated design systems and design support systems. Automated design systems; as stated by MacCallum (1990), are the systems whose aim is to provide full or partial automation of the design process; while the human designer is giving the initial requirements, evaluating solutions and building prototypes. Design support systems are the systems whose aim is to assist human designers in their tasks. There are many approaches regarding the types of design support systems' assistance to human designers in their tasks: According to Watson and Perrera, design support systems can assist human designers in their design tasks by recalling past cases (Watson & Perera, 1997), and as stated by Fischer, design support systems can assist human designers by critiquing and navigating (Fischer, 1992). Design support systems' most distinctive feature is that they do not design anything unlike a CAD system, design support systems only attempts to support the designer during the exploration of possible designs, help designer to become familiar with the problems and possible solutions.

The idea of autonomous self-generated architectural design comes from the search for embedding fragments of intelligence and design sensibility within architectural objects so that they might learn how to search through design possibilities autonomously (Krause, 1997). When developing the DROP, embedding intelligence within architectural objects was one of the most essential features that tried to achieve. The focus of this research was on the free-form design, therefore it was decided that surfaces must be the architectural objects that intelligence and design sensibility would be embedded. Hence, the surfaces could have the means to have a control over their geometry, executing the analysis by themselves and suggest different design possibilities within the solution domain according to the results of the rain-flow analysis.

Krause's and most of the other researchers' applications were started after selecting the agents' behaviors and the agents that would be used in the simulation. Agents' behaviors and properties are heavily depended on the level of AI they have and differ according to the type of Artificial Intelligence. A common agent model has sensors, effectors, and different kinds of reasoning mechanisms, for example: sensation, perception, conception, hypothesizer, and action. Behavioral complexity measures the richness of the reasoning process that produces the dynamics of an environment, Behavioral complexity in agents can be measured in five modes: reflexive, reactive, reflective (Maher and Gero, 2002), autonomous (Steels, 1995) and proactive (Wooldridge and Jennings, 1995).

DROP was developed with an agent that shows reflexive behavior. Reflexive behavior is a pre-programmed response to the state of the environment - a reflex without reasoning. Agents that have reflexive behavior need familiar or recognized states to produce a response. The most common way to achieve this kind of behavior in agents is using a scripting language to implement behaviors associated with the 3D objects in an environment. These scripts define behaviors that are triggered by predefined patterns of events. In our case, reflexive behavior could be best explained by the behaviors of the surfaces. As mentioned above, surfaces developed as architectural objects that have intelligence and design sensibility in the model. Henceforth, any surface that resides in the model's environment would have the means to have a control over their geometry, executing the analysis by themselves according to the results of the rain-flow analysis and suggest different design possibilities within the solution domain. For example, in the current version of the model, any surface, whether it was imported into the environment or created in it in the first place, have the ability to execute the rain-flow analysis and calculate and represent the directions and the intensity of the particles used in the analysis. In conclusion, this representation was used to determine whether the forces distributed successfully or not. In addition, the data extracted from the rain-flow analysis could be used to make other analyses and having multiple specific results like areas that are suitable for bending moments to occur on the surface or solely visualizations of the drain curves.

Emergence is another important issue that must be taken into consideration within an agent-based system used to assist design. Emergence could be both a desirable or undesirable state according to the developer of the system's point of view. Even though emergence in agent-based systems to assist design, could be resulted in novel design solutions in some cases, current version of the model stated on a different ground regarding the emergence in design. Generating forms, which emerges through, was not a feature of the current version of the model. After the analysis of the initial shape (surface), the surface would reason about the problem areas and present the results to the user. The aim that shaped the current version of the implemented model was placed more on the designer as she/he decides on the form rather than the agent system.

Phases of the model

As explained in the previous sections, the model that executes the simulation of the rain flow analysis is developed to illustrate the flow of the forces on a free form surface that is subjected to its own weight to determine the applicability of the surface by examining the distribution of forces. The implementation of the model is done by simulating the flow of the raindrops on a surface in a digital environment. Raindrops that are used to explain the flow of the forces on the surface are replaced by particles in the implementation environment. The flow of the forces on a free-form surface are illustrated by observing the flow of the particles on the surface in the implementing environment and extracting the information regarding the position of each particle during the simulation of the rain flow analysis, which will be eventually used to calculate and illustrate the trajectory of the particles. Hereby, results of the simulation of the rain flow analysis in the implementation environment are presented as still images, which have the information regarding the distribution of the forces on the subjected surface that is obtained by calculating and illustrating the trajectory of each particle.

The simulation of the rain flow analysis in the implementation environment starts with the recognition of the subjected surface by the developed model. The model duplicates the initial surface and positions the duplicated surface above the initial surface in order to use it as a source, which particles emit. After the duplication and relocation of the duplicated surface, required data groups such as, gravity, amount of resilience of the particles, particle type, line, and tail widths of the particles, duration of the simulation must be entered into the model.

The steps that actually take place before the simulation of the rain flow analysis are controlled by the model in order to accelerate the processes before the simulation. Additionally, the whole simulation and the variables are controlled by the implemented model in order to decrease the required amount of time compared to the case that requires manual adjusting of all these procedures.

The simulation of the rain flow analysis can be executed after required the data groups were entered into the model. Agent that is responsible for the extracting the information regarding the positions of the particles during the simulation, obtains the position of each particle during the simulation of the rain flow analysis. When the information regarding the position of each particle during the simulation is obtained, trajectory of each particle during the simulation, which represents the flow of the forces on a free-form surface that is subjected to its own weight, is illustrated by the agent.

Case studies

In order to test the tool's reliability, Eero Saarinen's Kresge Auditorium in MIT Campus and Felix Candela's Restaurant Los Manantiales in Xochimilco, Mexico are selected as case studies for which the tool tested out according to previously mentioned design flaws regarding free-form surfaces.

Kresge Auditorium

The model was tested on Eero Saarinen's Kresge Auditorium in MIT campus to determine its reliability. The fact that forces were not distributed along the surface properly resulted in the cracks on the surface of the Auditorium. Because of its structural faults, Kresge Auditorium does not look like the way it was designed – a thin white concrete shell structure with approximately height of 15.24 meters and divided by translucent glass walls so that only three points of the shell structure touches the ground (http://libraries.mit.edu/archives/exhibits/saarinen/). Instead, it was coated with lead panels to cover the cracks on its concrete surface.

Kresge Auditorium is modeled in MAYA, and the rain-flow analysis was executed with the model of the auditorium. A number of still images that shows the results of the rain-flow analysis were presented to the user at the end of the simulation of the rain-flow analysis (Figure 1).

Los Manantiales Restaurant

The model was also tested, regarding its ability to correctly represent the results of the rain-flow analysis, on Los Manantiales Restaurant in Xochimilco, Mexico City. The reason of choosing Los Manantiales Restaurant that designed by Felix Candela was that fact that unlike the Kresge Auditorium, the Restaurant was considered as a successful shell structure.

Los Manantiales is an eight-sided groined vault composed of four intersecting hyperbolic paraboloid saddles. The canted parabolic edges of the Los Manantiales Restaurant display the buildings most striking feature: the thinness of the shell with 40 mm.

After the 3D model of the Los Manantiales Restaurant had been completed in the modeling program, rain-flow analysis was executed on the digital model. The hypothesis for the flow of forces in shell structures - like a rain flow, loads will flow along curves with the steepest ascent on the shell surface to its supports – was tested on the Los Manantiales Restaurant, and the results of the analysis can be seen in the figure below (Figure 2).

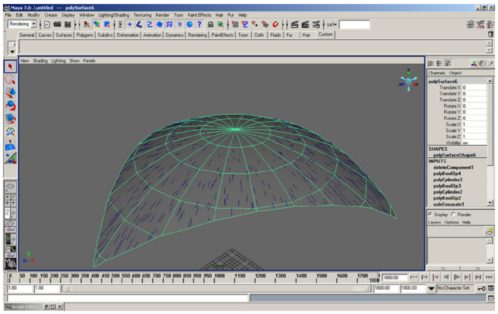

Figure 1
Results of the rain-flow analysis conducted on the Kresge Auditorium

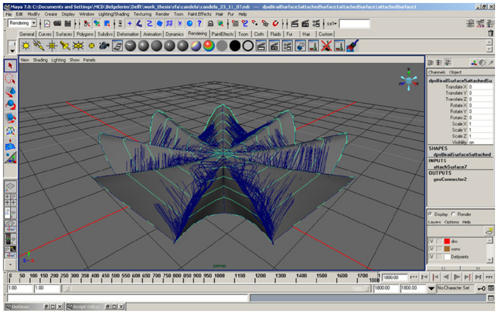

Figure 2
Results of the rain-flow analysis conducted on the Los Manantiales Restaurant

The hypothesis has been tested on the Kresge Auditorium, and the flow of the forces on the surface can be as seen in the results of the rain-flow analysis. On both left and the right sides of the surface, loads run through the edges instead of along the supports.

Unlike the Kresge Auditorium, results of the rain-flow analysis, which was conducted on the Los Manantiales Restaurant, shows that the loads were distributed along the groins properly. Additionally there is not any indication of the loads run through the free edges as shown in the results of the same analysis conducted on the Kresge Auditorium. Whilst the two most stress critical areas of the shell structures – the groin and the edges – were not shown any indication of an unsuccessful distribution of loads, it can be easily suggested that Felix Candela's choice on form is successfully abolishes the possibility of compromising the structural integrity or safety of the shell. A detailed analysis about the Los Manantiales Restaurant and findings of the FEM (Finite Element Method) analysis that was conducted on the form, which are in favour with the results of the model, can be found in the paper written by Noah Burger and David P. Billington (2006).

Conclusion

The digital tool DROP was developed and used to exhibit the ways that an agent based tool could be used to assist designers for free-form shape generation and evaluation. Potential users of the tool that was developed in the extent of the research are architects whether they are novice or expert, Architects have limited knowledge regarding the structural applicability of the surface when designing free form surfaces, therefore the designers can face several obstacles in the later phases of the design and forced to revise the form in order to ensure the building's safety. As mentioned before, there are several ways and some computer programs to analyze the free-form surfaces but these methods were used in the later phases of the design and they do not give a clear insight about these surfaces. By using the tool that was developed, architects can obtain the knowledge about surfaces that they design at the early phases of the design and have more time to work on the other aspects of the design in the time they save deciding on the form.

In the future, we aim to develop other agents. Reasoning agents in the DROP were planned to have reactive behavior. Reactive behavior can be best explained as reasoning about responses within a fixed set of goals. In the prototype that fixed set of goals were decided to be the stable state of a surface, which the forces could distributed properly. This mode of behavior is achieved by using agents to control one or more objects in the environment. In the DROP, agents were decided to have control over the surfaces by making changes to them in order to achieve their goal. Reactive behavior of the reasoning agents was not supported by the current version of the model. However, with having the results of the rain-flow analysis and being able to extract the data about the analysis, strongly indicates that, developing agents that have reactive and then reflective behaviors would be the next additions in future versions of the model.

Acknowledgements

I would like to acknowledge Andrew Borgart and Dr.ir. Rudi M.F. Stouffs from Delft University of Technology, for their generous support and guidance in the development of this model and for being my mentors during the period when I was in Delft University of Technology as an exchange student.

References

Borgart, A., Leuw, M., and Hoogenboom, J. P.: 2005, The relationship of form and force in (irregular) curved surfaces, in E. Ramm, W. A. Wall, KU Bletzinger, and M Bischoff (eds.), Book of abstracts of the 5th International Conference on Computation of Shell and Spatial Structures, pp. 223-223.

Brodie, M. L., Mylopoulos, J. and Schmidth, J. W.: 1984, On Conceptual Modelling: perspectives from artificial Intelligence, databases, and programming languages, Springer-Verlag, New York.

Burger, N., Billington, D. P.: 2006, Felix Candela, Elegance and Endurance: An Examination of the Xochimilco Shell, Journal of the International Association For Shell and Spatial Structures: IASS 47(3), December n.152.

Lawson, B.: 1997, How designers think: the design process demystified, Completely rev. 3rd ed., Architectural Press, Oxford; Boston.

MacCallum, K. J.: 1990, Does Intelligent CAD Exist?, Artificial Intelligence in Engineering, 5(2), pp. 55-64.
Watson, I. and Perera, S.: 1997, Case-Based Design: A Review and Analysis of Building Design Applications, Artificial Intelligence for Engineering, Design, Analysis and Manufacturing, 11, pp. 59-87.
Fischer, G.: 1992, Domain Oriented Design Environments, Proceedings of the 7th Knowledge-Based Software Engineering Conference IEEE Computer Society.
Krause, J.: 1997, Agent Generated Architecture, Design and Representation, ACADIA '97 Conference Proceedings, Cincinnati, pp. 63-70.
Maher, M. L. and Gero, J. S.: 2002, Agent Models of 3D virtual worlds, ACADIA 2002: Thresholds. California State Polytechnic University, Pamona, pp. 127-138.
Schodek, D., Bechthold, M., Griggs, J. K., Kao, K. and Steinberg, M.: 2005, Digital Design and Manufacturing: CAD/CAM Applications in Architecture and Design, Wiley & Sons, New Jersey.
Steels, L.: 1995, When are robots intelligent autonomous agents? Journal of Robotics and Autonomous Systems, 15, pp. 3-9.
Wooldridge, M. and Jennings, N. R.: 1995, Intelligent agents: Theory and Practice, Knowledge, Engineering Review 10(2), pp 115-152.