The Siren's Call

Phillip Anzalone, eXtended Office (XO); Pratt Institute

Cory Clarke, eXtended Office (XO); Columbia University; Pratt Institute

Abstract

This paper presents an account of our research and development of processes providing seamless transition from design to fabrication. The narrative of our design, development and prototyping experiments spans seven years, including our current project, the Trusset software/structural system. Trusset is a combined building system and agent-based software design tool. The building system is based on a differential space-truss designed for fabrication entirely with computer numerically controlled (CNC) linear cutting devices, such as laser cutters or three-axis mills. The software component is a set of agent-based design tools for developing surfaces and envelopes formally suitable to be built using the space-truss structure. Developed in parallel, the software and building components combine within the Trusset system to provide a seamless pipeline from design to fabrication and assembly.

The story of the development of software components and structural system, leading to the Trusset, act as a means of discussing the larger issues framing the research – the potential pitfalls and benefits of design and fabrication integration via the computer.

1 INTRODUCTION

The following is an account of our research and development in producing a computer based system for the seamless integration of design and fabrication. The narrative describes a timeline of our varied experiments leading to our current research and development endeavor, the Trusset system. Illustrating how we developed our assumptions, we will objectively outline the shortcomings and successes of our research processes and products. The narrative patches together seven years of intermittent and discontinuous research to demonstrate potential problems and issues in automation of design and fabrication procedures using digital tools.

1.1 The Trusset System

Trusset is a combination building system and agent-based software design tool. The basis of the building system is a two-way space-truss, comprised of struts and nodes in a square-on-square offset configuration. The nodes are constructed using metal plate gussets (gusset + truss = Trusset), folded at 90º angles, which can be fabricated using simple 2D computer-numerically controlled (CNC) laser cutters. The struts are variably cut lengths of uniform aluminum extrusions. The variable length struts and the variable angles produced by CNC cutting the gusset plate nodes allow for almost infinite range of forms that can be constructed using the building system.

To produce the components of the space-truss, we developed a set of custom software plug-ins in parallel with the construction system. The software component of the Trusset system provides tools for designing forms compatible with the construction logic – using the material and formal limits of the construction system as a basis for the design logic of the software. The design tools use a combination of various agent-based design elements. Along with the design tools, the software component provides tools to produce geometry files for the generation of GCODE files for manufacturing.

The strategies of the gusset-plate based construction system, and the combined agent-based system have developed from the successes and failures of a series of research projects. The research can be bracketed into four major groups of experiments. These experiments and their results are outlined below.

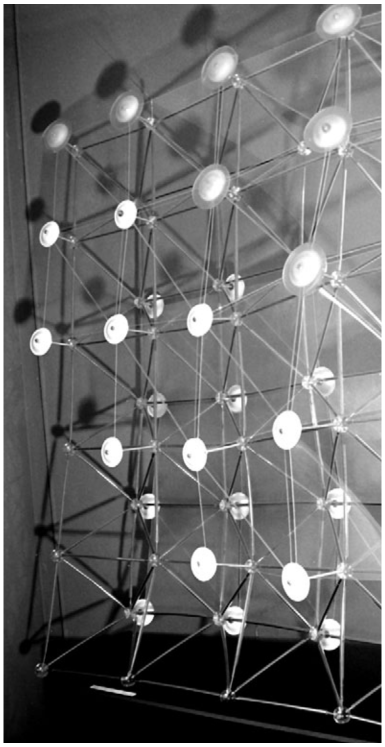

Figure 1 - Prototype of CNC fabricated space-truss

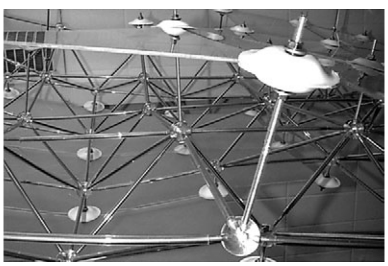

Figure 2 - Space-truss node

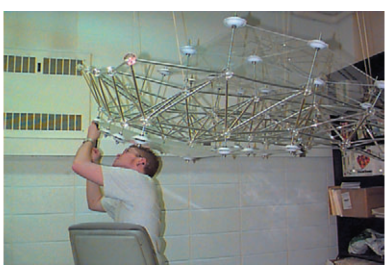

Figure 3 - Assembly of space-truss

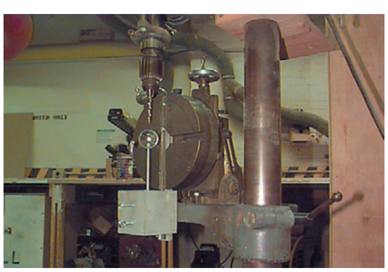

Figure 4 - 3-axis jig for fabrication of nodes

2 EXPERIMENT 1: 1997

2.1 Introduction

Our experimentation with the integration of digital design and fabrication began with the design and prototyping of a spherical node space-truss (Chilton 2000) and custom cladding solution using a 3-axis CNC milling machine and 5-axis jig. The research stemmed from a desire to find a flexible and universal means of building complex curvilinear forms that could be produced with the emerging 3D design software becoming readily available at the time. The research focused on the design and implementation of a differential space-truss, an under-utilized structural system capable of complex, three-dimensional form.

The traditional space-truss is a lattice structure of standard elements, typically leading to architecture of regular geometrical forms (Gabriel 1997 and Pearce 1990). The traditional space-truss employs standard elements due to constraints of design, analysis and fabrication. The differential space-truss, however, uses non-standard elements; by allowing each element to be unique the truss can take on complex three-dimensional curvilinear form as well as basic regular shapes. To produce complex shapes, potentially every element in the system would be unique, causing potential for problems in logistics and production (Ramaswamy, Eekhout and Suresh 2002). This first research project was based on the assumption that these limitations were surmountable through computer-based techniques.

2.2 Process

The experiment was to design a differential space-truss and cladding system, and fabricated a quarter-size prototype using a CNC milling machine. For the prototype, we produced a six-foot undulating facetted wall section (Figure 1).

The research and design in this experiment focused around creating a construction system capable of supporting complex curvilinear form, and details that would be adaptable enough to allow for its cladding using generic sheet material. The design of the details was done in consultation with an engineer and developed using form*Z digital modeling package. The detail (Figure 2) utilized two partially hemispherical caps, fitted with rubber gaskets and held together by a bolt and washers. The caps were prototyped from the digital model by machining molds using the 3-axis milling machine and vacuum-forming styrene over the molds. The gaskets were created using a similar process, where the molds were used to cast silicon rubber. The space-truss struts were fabricated by hand from linear steel rods (Figure 3].

The lengths were determined by manually dimensioning the components of the computer model. The nodes would ideally have been fabricated using a 5-axis milling machine, but due to expense and limited availability of this equipment, they were instead made using a 3-axis jig and a drill press (Figure 4). The angles for the nodes were again determined my manually dimensioning the computer model. The panels for the cladding were cut using the CNC milling machine from files generated by unfolding the surface of 3D model in form*Z (Figure 5).

2.3 Results

Figure 5 - CNC Milling process

The production of a scale prototype of the differential space-truss led to several realizations. The first was a need to reconsider our assumptions regarding fabrication methods. The use of a CNC mill for the 2D elements was successful, but the inability to mill the nodes using a 5-axis mill pointed towards larger problems. Upon further research, we have dismissed the idea of using a 5-axis mill for the production of the nodes because of limited feasibility in mass-production. The investment and maintenance of 5-axis equipment is an order of magnitude more costly than that of 2D/3D machining equipment. For the design of a building system that is to be low-cost and mass-producible, we decided to pursue details and joints designs using lower-cost, more readily available equipment – making it possible for the easy prototyping of design iterations. This insight informed our current Trusset node design, utilizing entirely 2D linear cutting technology.

An additional shortcoming of our experiment was a seemingly obvious issue of tolerance. The illusion of CNC manufacturing is that the accuracy attainable within CAD software can be carried through to the site and construction. While CNC machining does allow for a high degree of accuracy (the CNC machinery used in all our experiments had .001 inch accuracy), absolute precision in all parts does not allow for tolerances necessary for onsite installation. We found that if our nodes and struts were manufactured to exact lengths and angles, it provided no allowance for inaccuracy in our on-site measurements. This problem was compounded by the rigidly networked nature of the spherical space-truss. During installation, a minor adjustment made in one area of the truss to account for inevitable inaccuracies in site measurements, compounded across the entire truss (because of the zero tolerance design), forcing every component out of alignment. This has led to the inclusion of adjustability and a strategy of designing with tolerances in our later versions, countering the hyper-accuracy of CNC manufacturing.

The final major discovery during this experiment was the need for custom software to automate the modeling of the space-truss, generation of CNC files, and production of the inventory of elements. In this initial experiment, we manually created the truss geometry and all the files for machining, consuming a disproportionate amount of time in the process. This, combined with the difficulty of tracking and inventorying all the parts, led us to create our own custom software implementations in later experiments.

2.4 Conclusions

The experiment's premise was testing the possibility of using CNC manufacturing to produce a differential space-truss. This proved relatively successful, as we were able to create a proof-of-concept prototype and develop feasible details for a cladding system for the truss. The earnest attempt to manufacture all the parts using CNC generated several lessons for future experiments, as outlined above. The overall result was the conclusion that to successfully design and build using a differential space-truss, we would need to develop an integrated means of moving from design through to fabrication. This led to our second set of experiments.

3 EXPERIMENT 2: 2001

3.1 Introduction

After the initial experiments with the space-truss, we began research in two separate paths, one continuing development on the construction system and details, the other investigating ways of integrating and automating the design and fabrication process. Along this second path we performed a set of experiments developing custom software plug-ins that generated space-truss geometries for any 3D curvilinear surface.

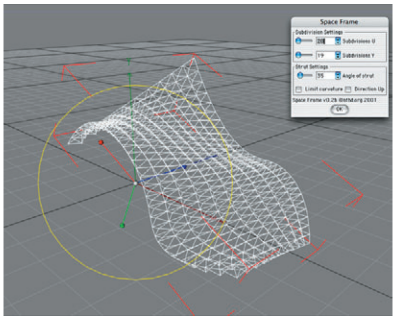

Figure 6 - Software interface for generating space-truss

3.2 Process

For this experiment we focused entirely on the software component, suspending any concerns with the construction system developed from the previous experiment; addressing the space-truss strictly as a geometric abstraction. The software plug-in was developed for Maxon's Cinema4d, utilizing its C++ style scripting language COFFEE. The plug-in was developed to take any surface and tessellate it based on user settings (Figure 6), the tessellation generated quadrangles, which were used as the basis for the square-on-square offset space-truss. The plug-in utilized the quadrangles to additionally generate the struts and the nodes for the space-truss structure. Once the plug-in was developed we tested it with a variety of surface shapes and sizes, finding its computational and geometric limits.

3.3 Results

The plug-in for producing the space-truss geometry succeeded in providing a means of automating the tedious task of manually modeling each strut and node, but its limitations led to another revelation in our process. The development of the plug-in focused entirely on automating the production of the space-truss geometry, but took no direct consideration of the material of the structure (Ramaswamy, Eekhout and Suresh 2002). When running the plug-in on different surface shapes we found that it often generated struts that were either too long or too short to be realistically fabricated, or violated the material capacities of the struts (Aluminum Association 2002 and Thornton 2001). We found in the end, after analysis of the results, that we could model many surface shapes that were not optimal for construction with the differential space-truss system.

3.4 Conclusions

The need for defining the formal limitations of the construction system became apparent in our plug-in experiment, forcing us to the conclusion that we must produce software tools for the design of suitable forms. The initial impetus for making a software plug-in was to automate the tedious task of modeling space-truss geometry, but its partial failure illustrates that, in our case, a seamless system from design to fabrication cannot work entirely in reverse – retrofitting a construction system and fabrication method to the output of a formally arbitrary, at least in terms of construction, design process. While the potential is within the computer to rationalize arbitrary form to match construction logic, there will inevitably be reciprocity between the fabrication and construction on one side, and the design phase of the process on the other. To provide for this reciprocity, we decided to develop design tools that embed the material and construction limitations of the space-truss into the software.

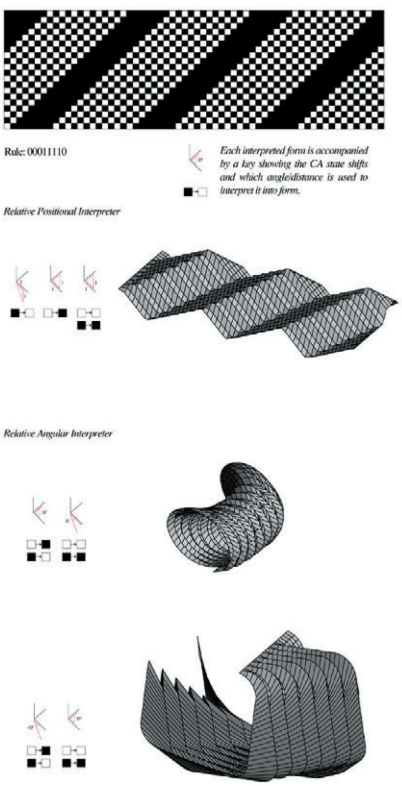

Figure 7 - Formal interpreters for Cellular Automata

4 EXPERIMENT 3: 2002

4.1 Introduction

Our third set of research experiments focused on developing a means of integrating material, fabrication and construction logic into a set of design tools. The software tools for the individual joints and nodes of the structure could obviously be parametric objects, but we felt that tools for the design of the overall form of the structure should not be parametric but generative. While parametric design tools provide suitable means of limiting the formal range of a tool to a set of buildable possibilities, the top-down approach of parametrics overly limits the design space. (Fischer, Burry and Woodbury 2000; Fischer, Burry and Frazer 2003). Working from this initial assumption, we researched computational techniques for embedding material logic into software components..

4.2 Process

The research efforts in this series of experiments focused on computational techniques for modeling complex adaptive systems, primarily cellular automata and flocking algorithms (Neumann and Burks 1966; Renyolds 1987). We felt that the ability of these systems to manage the complex interactions between innumerable individual elements provides a means of managing the complex interrelations between all the elements of a building system. The process began with creating 1D, 2D and 3D cellular automata, and a variety of formal interpreters (Figure 7).

The automata were used as a dynamic data structure for coordinating interrelationships within the system; the formal interpreters were used to give shape to that information as a surface or volume. We found that different mechanisms for formal interpretation matched very effectively with different material or fabrication concepts. For example, we found that by treating the automata as 'virtual pens', drawing lines in 3D space, we could consistently produce forms that were ideal for cutting with a 2D laser cutter or CNC mill by moving the automata at regular intervals in the x and y direction and varying movement in the z based on the information within each automaton (Clarke and Anzalone 2003).

Along with the cellular automata we applied several other algorithms, such as flocking rules and rules for environmental responsiveness. These provided additional means for formal control by the designer and for embedding construction logic. (Figure 8)

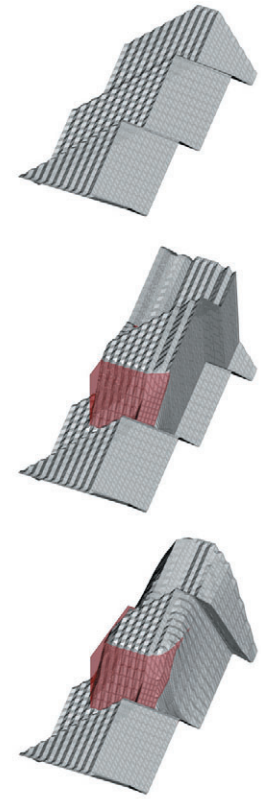

Figure 8 - External control mechanisms at work. The same CA is represented here unconstrained, constained by context and constrained by flocking meachinsms.

4.3 Results

While the cellular automata generated interesting, and potentially constructible forms, they were extremely limited in their formal range, and more importantly in their degree of control. Cellular automata did provide a means of managing the informational complexity of the interrelations of construction elements, but since their overall behavior and form was emergent, based on only a few rules, the formal outcome was unpredictable and ultimately too uncontrollable to be valuable as a general design tool (Figure 9). The additional layers of logic utilized in the research, those of flocking and rules for environmental response were much more adaptable and predictable in their formal effects.

4.4 Conclusions

The successes of this series of experiments, the intercommunication between elements provided by the cellular automata and the balance of predictability and adaptability exhibited by the flocking controls, stemmed from the fact that they were acting as autonomous communicative agents. The agent-based nature of the systems allows for the individual elements to communicate through a network of interrelations, giving them a degree of autonomy responsive to outside influences – a means of design control. The idea of individual shape-generating agents, embedded with a behavioral logic that prohibits them from making un-constructible, yet accepting of external input for design control, acted as the basis for our current set of research experiments.

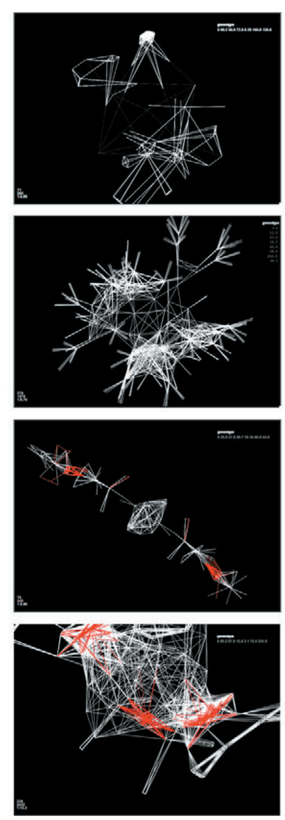

Figure 9 - Sample of some structures resulting from software system

5 EXPERIMENT 4: 2003

5.1 Introduction

The current focus of our research is developing the Trusset system, one potential derivative of the series of experiments we have been conducting. Following our work with complex adaptive systems for generating build-able form, we decided to focus on developing an agent-based design system. Critical to the development of the software agents is the creation of a rigorous set of rules for their behaviors; these rules are the means of embedding the formal limits of the construction logic into the software. In order to create a valid set of rules we re-kindled our efforts to developing a feasible differential space-

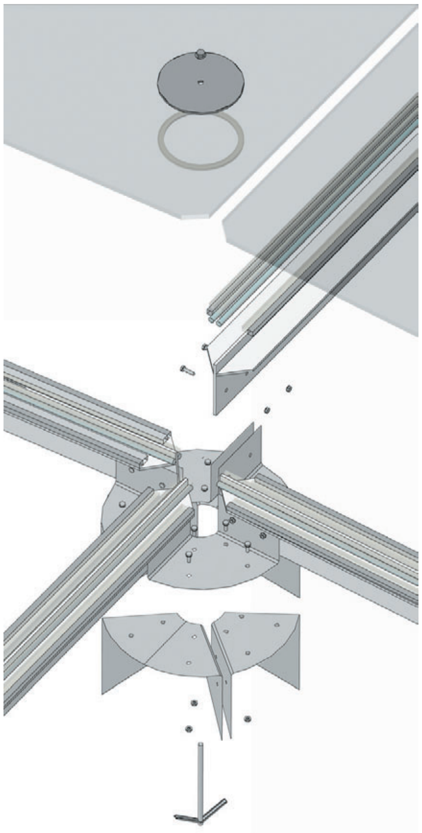

Figure 10 - Exploded axonometric of node detail

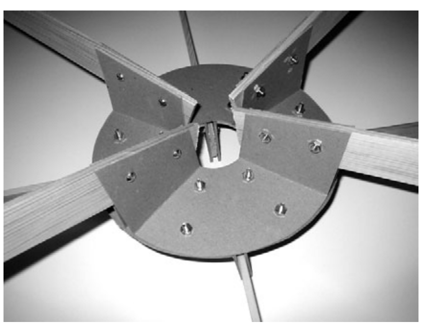

Figure 11 - Photograph of node prototype

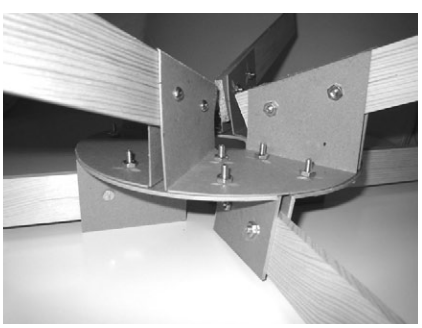

Figure 11 - Photograph of node prototype

truss system to be fabricated with CNC fabrication technology. The construction parameters of the nodes and struts of the building system provide the limits for the agent-based software.

5.2 Process

The majority of effort in this set of experiments was spent developing a space-truss system constructible with 2D CNC fabrication technology. Stemming from problems with our earlier experiments in designing and building a differential space-truss, we decided that limiting ourselves to 2D cutting devices would allow us to readily prototype and test our designs, as well as produce a system less expensive to mass-produce. With 2D cutting devices as the primary means of manufacture, we researched and designed different types of space-truss structures, settling on a gusset-plate type, which uses folded plate metal for its nodes.

Using this starting point, we designed several iterations of folded plate details as computer models, building all aspects of the detail including sealant, bolts, nuts, plates, struts, panels, end-caps and adhesives (Figure 10).

These details were then virtually disassembled and the nodes were unfolded to make files for CNC manufacturing. The initial prototypes were made with a laser-cutter and chipboard, testing issues of geometry and assembly, not structural feasibility. (Figure 11 and 12). After several test models, we developed a design that can be cut with 2D devices, and for ease of assembly, folded using only 90º bends.

Using this detail as a basis for the design, we made structural calculations to determine the necessary material thicknesses and sizes (Ambrose 2002). This information, coupled with the angle limits of the node, act as rules for the agent-based software. The software for the system is designed in two major modules; the first being the agent-based design tool set. This tool assists the architect or designer in creating surfaces buildable using the Trusset structural system. The second module, adapted from our earlier research, takes the surface generated from the first module and produces and inventory of angles and node positions for a space-truss. The agent-based design tools have several layers of logic, similar to that of the previous experiments. The primary controls are a flocking logic, ensuring the agents maintain proper distance, and angle relationships, ensuring the generated form is buildable. The second layer of logic is a series of shaping controls (Clark and Anzalone 2004) (Figure 13), providing a means for the designer to direct the agents and impose certain formal limits, such as required areas of flat horizontal runs. There is an optional cellular automata logic layer, adding an undulating and/or repeating pattern to the surface, which is purely aesthetic in nature.

5.3 Results

The success of the software component and the building system can be judged separately in this experiment. The work with developing the building components was much more successful than our previous iterations in that the details are more resolved, universally usable and realistic in their means of fabrication. The detail is less sophisticated than the spherical node construction, yet this aesthetic inelegance does have practical benefits. Since it is limited in material type and thickness, the strut spans are limited to a few feet, and the bends are fixed at 90º, the resulting system can be shipped in a flat, unassembled state in standard crating. Even though it uses sophisticated technology to manufacture, the parts can be easily shipped and assembled on-site without any heavy machinery or technology beyond the complexity of a wrench and small metal break. The scale assures that shipping containers holding a large number of components are transportable by two persons, even over rough terrain.

The agent-based software components were successful in embodying the formal and material limitations of the construction system and providing a flexible intelligent design tool. In its current state, the software only deals with the problems of build-ability as a local issue, ensuring that each node and strut connection is possible. Global issues, such as the structural feasibility of the entire form, are at present not addressed, and most likely will require the addition of software algorithms that operate at a meta scale - monitoring the overall shape of the form, or incorporating finite element analysis.

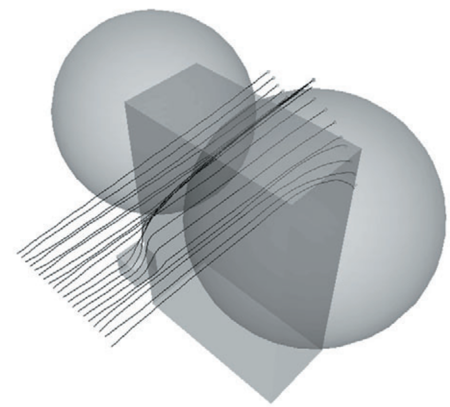

Figure 13 - Software agents controlled by shaping elements

5.4 Conclusion

The parallel development of the construction system and the software, and the iterative nature of the entire series of experiments, led to an extremely modular system. The software component consists of modules for design and for translation of that design into geometry files for fabrication. The building system is considered its own module within the structural system. Each of these modules are interoperable, the information from one feeding into the others, yet they are also capable of standing on their own. This aspect of the overall system, which is primarily a resultant of our approach, has led to an unexpected flexibility. This modularity allows for us to easily change components of the system. For example, the inelegant Trusset can be replaced with a more elegantly designed spherical node space-truss in the future, and the only change necessary within the design software would be an adjustment of the control parameters for the behavior of the agents. Conversely more innovative software approaches, such as the application of genetic algorithms for designing optimal structural efficiency or form-finding, could replace the agent based design system, but still utilize the other software components and the Trusset construction system.

This realization has led to an adjustment in our research process going forward. The initial plan was to merge all the software aspects, creating a universal tool to design and produce fabrication files. We are now continuing the modular approach that emerged from the process described. One of the next steps in the continued development of the Trusset system is to create a parametric version of the node joint. This parametric model would operate independently from the other software components, taking information from the plug-in that generates node angles and strut lengths. As we develop more refined construction systems, and corresponding parametric details, these could all plugin to the overall software pipeline.

6 CONCLUSION

The original siren's call of CNC fabrication was the possibility of creating a seamless pipeline from design through to construction, facilitated by digital technology and tools. While we started with a premise of the possibility of a universal system – a space-truss that could be used to build any complex 3D curvilinear form and could be fabricated directly from a design model using CNC technology – we have instead come to develop a set of discreet tools and technologies with specific, often singular, applications. While this seems to diverge from the original desire, it offers the potential of a more flexible and adaptable application, in lieu of a singular universal tool.

While envisioning a utopian ideal of a single universal solution to design and fabrication can be attributed to foolish youth (of both the authors and the still youthful field of digital fabrication), the full potential of developing a flexible set of component tools, both physical and virtual, has yet to be fully exploited in architectural practice. The utility of developing software-based tools using an approach of interoperable modules has vastly reconfigured the field of software development, as object-oriented programming logic has led to the proliferation of open-source projects. Modular and standardized building components reshaped the practice or architecture and construction in the post-war era, but the ideas of modular design have yet to bleed into the development of architectural software tools. We feel the potential of the approach of Trusset system, where building components and software components are part of a spectrum of physical and virtual modules, presents a novel and potentially fruitful avenue of approach to design and fabrication.

The Trusset system (Experiment 4) has been previously published in an independent paper entitled "Trusset: Parallel development of software and construction systems for space-truss structures" (International Journal of Architectural Computing)

7 REFERENCES

Chilton, J. (2000). Space Grid Structures, Oxford: Architectural Press.

Gabriel, J. F. (1997). Are Spaceframes Habitable In Beyond the cube: The Architecture of Spaceframes and Polyhedra., ed J. F. Gabriel. New York: John Wiley & Sons, Inc.

Pearce, P. (1990). Structure in Nature is a Strategy for Design, Cambridge: MIT Press.

Ramaswamy, G. S., Eekhout, M. and Suresh, G. R. (2002) Analysis, Design and Construction of Steel Space Frames, London: Thomas Telford.

Clarke, C. and Anzalone, P. (2003) Architectural Applications of Complex Adaptive Systems. In ACADIA 22 Connecting – Crossroads of Digital Discourse, ed. K. Klinger, 324-325, Indianapolis: Ball State University.

Clarke, C. and Anzalone, P. (2004) Trusset: Parallel development of software and construction systems for space-truss structures. International Journal of Architectural Computing, Volume 2, Issue 2, June 2004.

Aluminum Association (AA). (2002). Aluminum Design Manual, Washington D. C.: Aluminum Association, Inc.

Thornton, W. (2001). AMA Institute of Steel Construction and American Institute of Steel, Manual of Steel Construction: Load and Resistance Factor Design, 3rd Edition, Chicago: American Institute of Steel Construction.

Fischer, T. Burry, M. and Woodburry, R. (2000). Object-Oriented Modeling using XML in Computer-Aided Architectural and Educational CAD in CAADRIA2000 Proceedings, eds Tan, Beng-Kiang, 144-155, Singapore: CASA.

Fischer, T. Burry, M. and Frazer, J. (2003). Triangulation of Generative Form for Parametric Design and Rapid Prototyping, in 21th eCAADe Conference Proceedings, 441-448, Graz, Australia.

Neumann, J and Burks A. W. (1966). Theory of Self-reproducing Automata. Urbana, Ill: University of Illinois Press, 1966.

Reynolds, C.W. (1987). Flocks, Herds and Schools: A Distributed Behavioral Model in Computer Graphics, 21(4), SIGGRAPH '87 Conference Proceedings, 25-34.

Ambrose J. (1994) Design of Building Trusses, New York: John Wiley and Sons.