2004

Trusset: Parallel Development of Software and Construction Systems for Space-Truss Structures

Phillip Anzalone
CUNY New York City College of Technology

Cory Clarke

Recommended Citation

Clarke, Cory, and Phillip Anzalone. "Trusset: Parallel development of software and construction systems for space-truss structures." International Journal of Architectural Computing 2.2 (2004): 229-244.

Trusset: Parallel Development of Software and Construction Systems for Space-Truss Structures

Cory Clarke and Phillip Anzalone

Trusset: Parallel Development of Software and Construction Systems for Space-Truss Structures

Cory Clarke and Phillip Anzalone

This paper documents our current progress on the parallel development of a building system and corresponding agent-based software design tools; together the two produce a seamless pipeline from design to fabrication and assembly. The building system is a clad differential space-truss designed for fabrication entirely with computer numerically controlled (CNC) linear cutting devices such as CNC laser cutters or two-axis mills. The software component is a set of agent-based design tools for developing surfaces and envelopes formally suitable to be built using our space-truss system.

1. Introduction

A space-truss is a highly efficient structural system employing bi-directional, offset lattices of rods and connecting nodes. These structures are typically built using a catalog of standard rods and nodes, leading to architecture of regular geometric forms, as in the geodetic domes of Buckminster Fuller [8] and I.M. Pei's Javits Convention Center in New York. When the rod lengths and node angles are varied across the structure, in what we call a *differential space-truss*, a diverse formal range of complex doubly curved structures can be realized. Despite their formal potential and structural efficiency, differential space-truss structures have not become prevalent in architecture primarily due to constraints of design, analysis and fabrication. As a response to this we are developing a space-truss system that can be easily and efficiently fabricated, along with a set of software components that provide design tools and automated generation of the parts inventory and digital files for fabrication and assembly. The space-truss developed utilizes a folded gusset-plate for its structural connections, reflected in its name Trusset, which is derived from the terms "truss" and "gusset".

While the *Trusset* structural system has a diverse formal range at the global scale, it has specific formal limits at the scale of the individual rods and nodes. To address these formal limits and rules at a local scale, and allow for design control at the scale of the overall structure, we have adopted an agent-based software strategy. The software uses a collection of "intelligent" agents, each with behavior controlled by an embedded logic embodying the formal and construction limitations of the structural system.

The following document outlines our progress in the development of both the building and software components of the *Trusset* system. The paper proceeds as follows. Section 2 outlines the concepts of the software system and its primary components, the design module (Section 2.1), the structure generation and inventory module (Section 2.2). The next section (3) describes the structural system (3.1), and its components (3.2) and assembly (3.3). The paper concludes with speculation on potential application of the system and an outline of future work.

2. Software components

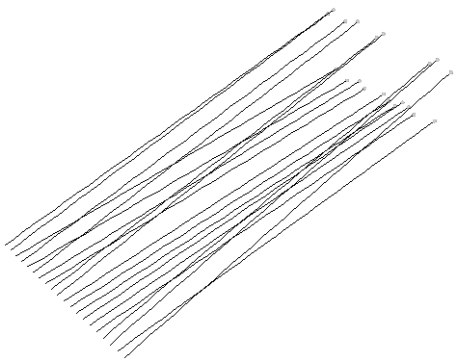
The software component of the *Trusset* system is designed as two primary modules. The first module operates at the design level, assisting an architect in creating surfaces buildable using the *Trusset* structural system. The second module performs the practical task of automating the generation of the structural components, including all the rods and nodes. The division of software modules reflects the process of utilizing the system. The designer uses the primary software module to produce building envelopes; the outcome of this module is a surface that can be input into the second module to generate the structural system and parts inventory used for fabrication.

The software is prototyped utilizing scripting languages in Alias|Wavefront's Maya (Maya Embedded Language) and Maxon's Cinema4d (COFFEE). The use of non-compiled scripting languages allows for simplified rapid prototyping and testing of different algorithms and approaches to the modules. As the logic of the software components is refined and finalized, they are rewritten in C++, a compiled software language, providing for more robust and faster performing modules.

2.1. Design module

The design module of the software component is intended to provide a simple interface for making forms buildable within the construction and fabrication limitations of the space-truss system. To provide a high-level design tool to architects and designers, the logic of the formal limitations of the building system is internalized into the software tool. These formal limitations are internalized in the software in the behavior of interdependent "intelligent" agents. Following current trends in digital construction and fabrication tools, the formal logic of the building system could be embedded within a parametric object. 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 runs the potential of overly limiting the design space. [1][2] We have instead chosen to utilize a bottom-up generative approach using the constraints of the building system itself as behavioral constraints on coordinated autonomous agents. The intelligence of the agents is implemented as a layered logic. Along with the behavioral constraints of the construction logic, the agents have additional capabilities to react to internal and external control structures, providing design control over the system.

The agents' basic behaviors are controlled by a three-rule cellular automaton (CA) [3]. The logic of the CA system provides a communication mechanism between agents to produce a 3D patterning of the surface generated by the software. As we have previously stated in our research [4], the discreet composition of cellular automata at a local scale provides a suitable analog to the organizational logic of the space-truss, which is also composed of discreet elements at the local scale. CA systems operate in a regular spatial matrix, the organization of the data offering an easy representation in a regular 'voxel' array; in architectural applications of CAs this has often led to regular geometric compositions. Jonathan Frazer discusses at length potential pitfalls and regarding the formal realization of CA data-structures in *An Evolutionary Architecture* [5]. We adopted a technique suggested by Frazer of using a 'process driven' approach [6] (90). The *Trusset* software system disconnects the data-structure of the CA system (rows of "cells") from its formal expression, using the CA states as an instruction for the movement of the agents. A basic example of the agents exhibiting behavior dictated by the CA rules can be seen in Figure 1,

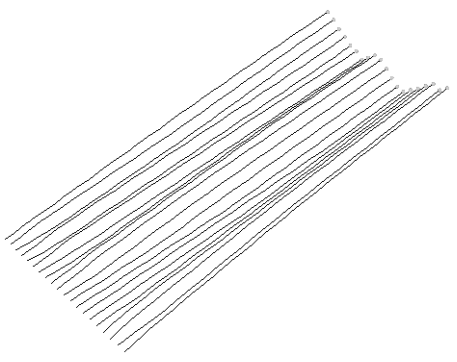


► Figure 1. Agents exhibiting behavior controlled by only the CA system

(for the sake of illustration all of the other behavioral rules outlined below have been disabled).

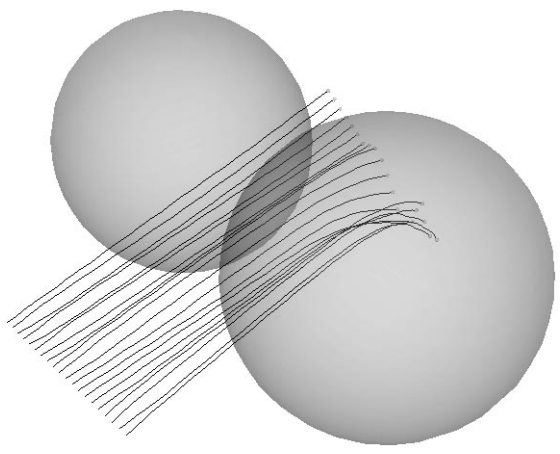
Layered on top of the behavior produced through the CA, the agents employ operating rules to limit their behavior to a range of buildable relationships at a local scale. The building system, outlined in more depth in section 3 below, is comprised of rods and nodes. For fabrication and structural reasons the rods have a limited range of possible lengths between 100cm and 150cm. The agents are programmed with a simple flocking [7] rule instructing them to maintain a minimum and maximum spacing between neighboring agents, ensuring constructional constraints are not violated. The flocking distances can be further adjusted to provide for material efficiency. By raising the limit on the minimum rod length it is possible to get more surface coverage out of a set of agents producing variations in material efficiency. Along with the limitations of rod length, since the space-truss uses an offset geometry (described in 3.1 below), there are limitations to the maximum angle between nodes of the system. The behavior of the agents is further modified by a second rule that attempts to keep the angle between any three adjacent nodes to between 35° and -35°. The behavior exhibited in Figure 2 is the identical behavior as shown in Figure 1, but modified by the flocking rule and angle limits.

The CA, flocking and angle rules work internally to give simple design direction and internalize construction logic. The software design module also



► Figure 2. Agents showing behavior controlled modified by the flocking rule and angle limits

has three families of control elements for exerting external guidance on the agents as they generate surfaces. Designers using standard 3D tools within Maya can distribute these three element types – Attractor, Avoidance and Plateau envelopes – within the agents' operating environment. The Attractor envelopes can be used to define areas of attraction for the agents, operating like local gravitational influences to pull the agents vertically and horizontally out of their paths. The Attractor envelopes are spherical volumes and have a square drop-off of their strength from the center to the outside edge. These Attractors are intended to provide a high-level of guidance and influence over the shaping of the system to designers, and operate in a similar manner to shaping tools in other generative architectural systems [8]. Figure 3 shows the agent behavior from Figure 2, further modified by the presence of two attractors within the modeling environment.

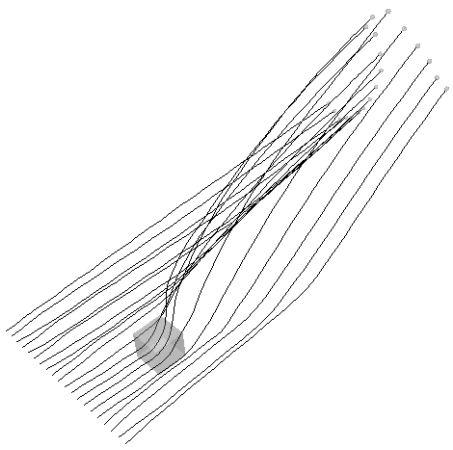


◀ Figure 3. Agents behavior further modified by attractors within the modeling environment

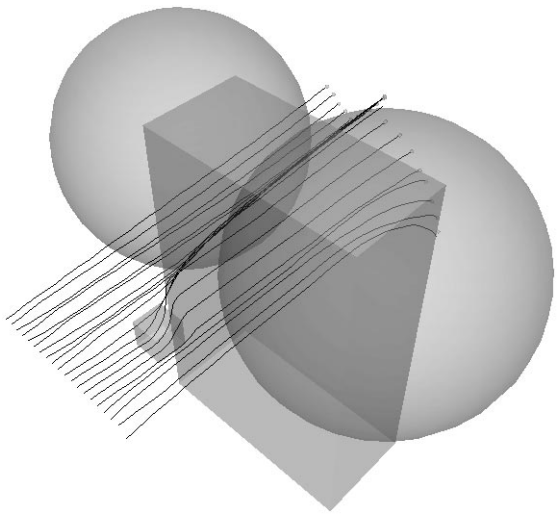
The Avoidance envelopes define rectilinear volumes of space, which the agents attempt to avoid. The Avoidance envelope can be used for defining spatial volumes around which the agent-based surface is shaped, giving a simple means for designers to produce building envelopes. Since the Avoidance volumes are thought as internal spaces around which the surface is shaped, the agents avoid them by moving upwards in 3D space and tend towards producing roof conditions. Figure 4 shows the behavior of agents after the introduction of an Avoidance element into the environment.

The Plateau envelopes define rectilinear volumes within the environment; whenever an agent passes through the Plateau envelopes it is encouraged towards producing a flat horizontal surface (in the current iteration of the software this occurs in one-direction only), or "plateau". The intention of the Plateau envelopes is to provide a simple means for designers to provide inhabitable areas, or areas for mechanical equipment, within the doubly curved surfaces generated by the agents. Figure 5 shows the behavior of a set of agents with internal behaviors and external influences of Avoidance, Attractor and Plateau envelopes.

► Figure 4. Behavior of agents altered by an Avoidance element



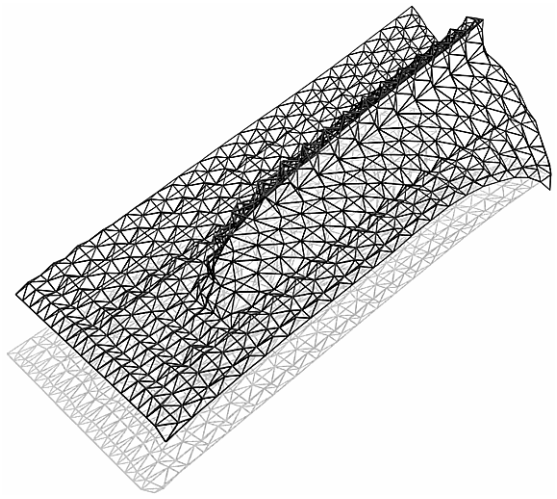
► Figure 5. Agent behavior resulting from combination of internal and external controls



2.2. Structure/inventory module

Since the design module internalizes the formal and fabrication logic of the building system, the structure/inventory module only performs the practical task of converting the surface geometry into a space-truss and cladding. This module can be used independently of the design module, if it is given a surface within the formal limitations of the *Trusset* system. The software does a preliminary check to ensure that the surface meets the basic formal constraints of rod lengths and angles outlined in section 2.1 and rejects any surface that is unsuitable. After verifying the validity of the surface, the application produces a model of the rods and nodes and then derives an inventory of rods and node angles. Since the module only accepts suitable surfaces, the algorithm for translating the surface into a *Trusset* construction works with no distortion of the shape. We feel that this "What-You-See-Is-What-You-Get" approach to the structure/inventory module is critical

because it does not sacrifice formal expression of design for effectiveness of the structure and constructability. The results from the design module shown in Figure 5 were fed into the structure/inventory module to produce the truss structure shown in Figure 6.



◀ Figure 6. Truss structure generated from agent behaviors show in Figure 5

This module is currently in a rudimentary state as the exact detailing of the individual nodes and rods is still under development, but the intention of the software is to output three sets of CAD/CAM files for fabrication. The first set of files will be of G-code and inventory files describing the unfolded geometry of the nodes, the former suitable for direct control of a CNC laser-cutter or 3-axis milling machine. The second set of files will be a schedule of lengths for rods, which will be cut from linear stock of aluminum extrusion directly at the factory. The third set of files is G-code and inventory files describing the infill panels. The fabrication method is described in more length in section 3.1 and 3.2 below.

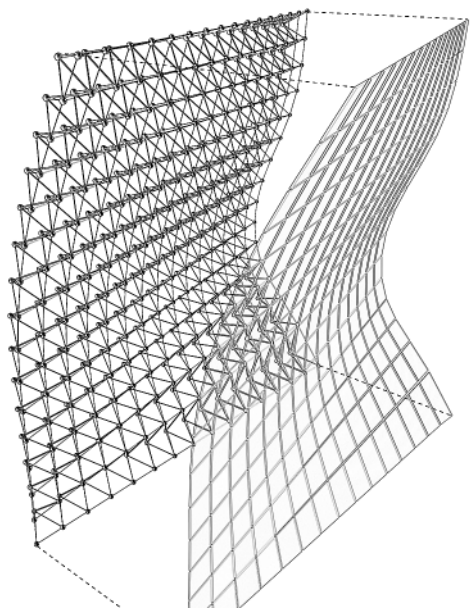
3. Building components

The building component of the space-truss system and the software component are being developed in tandem; innovations in the fabrication and construction method informing the constraints and logic of the software, and the digital methods of the software driving the formulation of the structural details. The primary intention behind the development of building components for the *Trusset* system is to create a direct relationship between the digital components outlined in the software section above and the physical manifestation of the elements. We found in our previous research, including real and virtual prototyping, that the space-truss provides a wide formal range and is well suited for translating into a virtual logic [4][9]. An understanding and qualitative description of the elements of the building system, as they relate to the embedded logic of the software agents, is described below.

3.1. Structural system

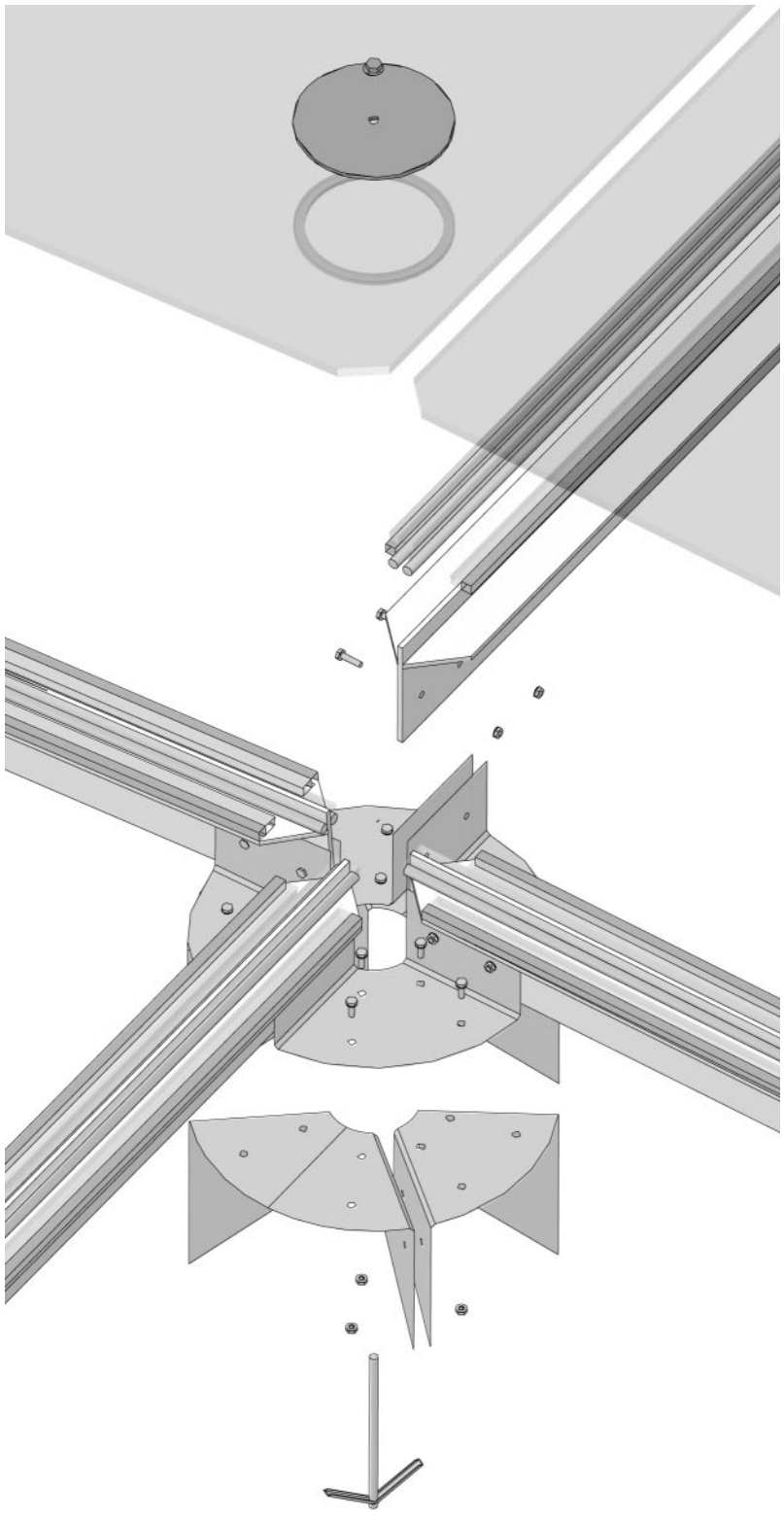
Fundamental aspects of costs, digital fabrication, efficiency and ease of assembly were driving ideas behind the development of the *Trusset* structural system. The system builds on the advantages of the traditional space-truss, modularity and efficiency, and through refinements in detailing and engagement of CNC manufacturing process we have worked to surpass the limitations typically associated with this method of construction, namely that of cost and form.

► Figure 7. Space-truss with square on square offset geometry



We chose to use a square on square offset geometry, as it is the most commonly used and tested configuration, and exhibits a high degree of material efficiency [10] (Figure 7). The system is a two-way space-truss structure, capable of spanning long distances with minimal material due to networked load sharing throughout the entire system. Due to its global rigidity, the necessity for predetermined support conditions is relaxed, while the depth of the space-truss allows for natural insulation and infrastructure raceways, creating a structure that allows for multiple scales of open space to be enclosed with a single envelope.

Curved structures generally exhibit a higher global rigidity than their planar counterparts, therefore requiring less thickness of structure and consequently less material. An aspect of the *Trusset* system is that the software agents, seeking optimal configurations, are encouraged to developed curvature in the surfaces. The *Trusset* system can undulate between flat spaces where needed, controlled by the 'Plateau' elements outlined in section 2.1, and curved interstitial spaces, in order to give the structure a global rigidity. Since the *Trusset* systems would not cost



◀ Figure 8. Exploded axonometric of *Trusset* assembly

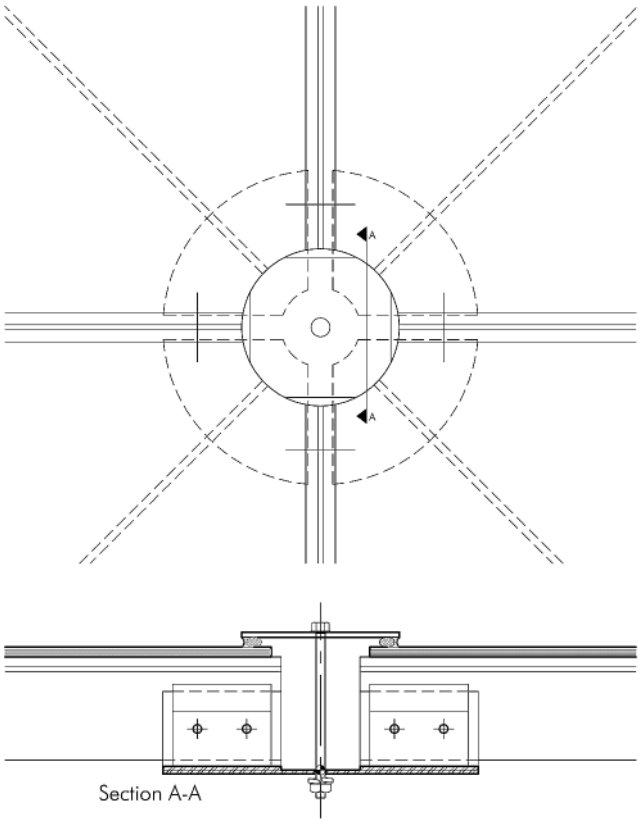
significantly more when used for curved geometry over flat (due to the opportunities of digital design and fabrication) we engaged this potential as an opportunity to expand the form-making potentials of the software.

The *Trusset* system is comprised of nodes, rods and panels (Figure 8). The nodes, as they are located at the highest concentration of shear and moment forces, are composed of gusseted, folded steel plates connected with high-tensile bolts. The rods are composed of extruded aluminum profiles, as the span is modest for a typical space-truss, and the loads are primarily axial. The diagonal rods are simple aluminum bars, and the panels can be of a wide variety of materials.

3.2. Components

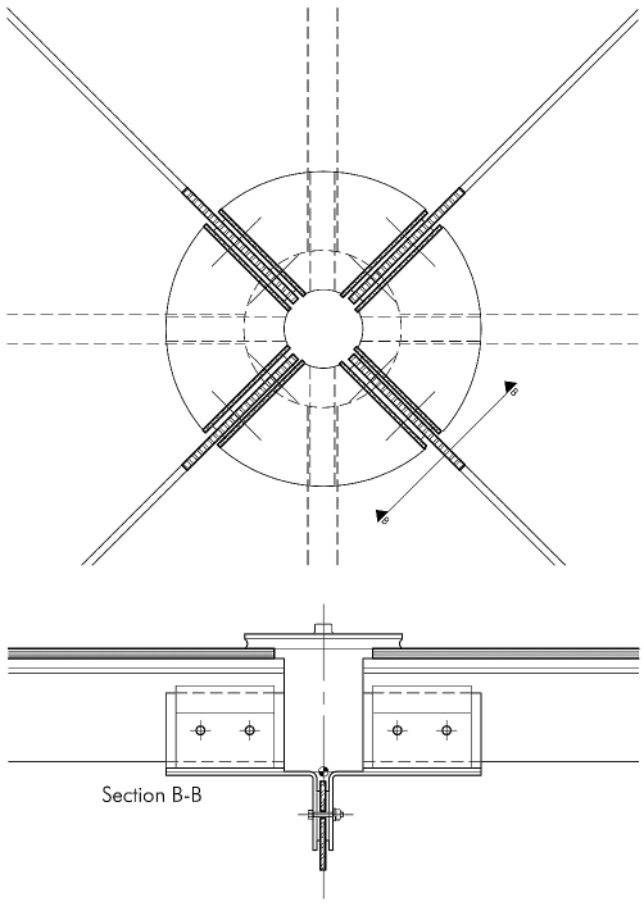
The *Trusset* System employs a node element composed of semi-standardized components, allowing for unique configurations of the structure through complex digital cutting methods available with current CAD/CAM technology. By controlling the parameters such as material type and thickness, spanning limits, and by fixing bends of the plates to 90°, we have developed a pre-manufactured component with nearly infinite configurations that can be shipped in a flat, unassembled state. The complexity of the global structural geometry is resolved in local configurations through the *Trusset*

► Figure 9. *Trusset* upper chord configuration



software, where each node is developed based on its relationship to its neighbors. In turn the capabilities of the design were developed as constants embedded in the logic of the software to allow for realistic modeling.

The *Trusset* node, once modeled digitally and interpreted for fabrication, is laser cut from steel plate, laser etched as necessary for assembly, and then stacked flat for shipment. As can be seen in Figure 9, the steel has been sized and the node designed to handle the shear and bending forces typical with a space-truss system [10][11]. On site, workers simply bend 90 degrees, as illustrated in Figure 10, at the pre-marked locations and bolt the eight bent plates together to form a complete *Trusset* node. Each plate is laser etched with a unique identifying number and attachment notations during fabrication to assure simple on-site assembly.

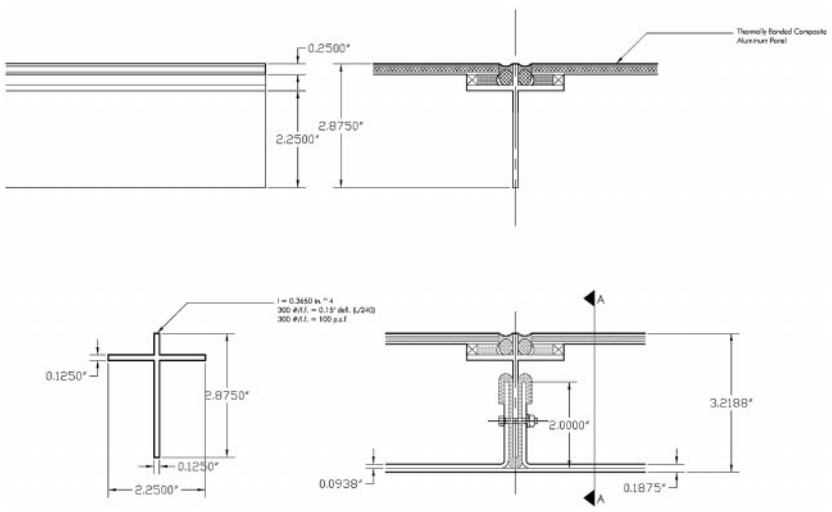


◀ Figure 10. *Trusset* lower chord configuration

The rods are standardized aluminum extrusions in lengths (Figure 11) designed to accept a variety of cladding options including aluminum panels, composite panels, glass and fabric. This multi-performing design allows for inexpensive fabrication while allowing for variation based on local conditions in availability and cost of materials. The software limits lengths of

the extrusions to those capable under the loading conditions input, in the case of proposed aluminum alloy material 100cm to 150 cm [12], assuring that one extrusion will handle any condition. The software can specify any additional milling of the extrusions to be performed during extrusion cutting.

The software, according to the material limitations of the paneling material selected for the design, calculates the size of the panels that provide the spatial enclosure. In this way the material type acts to adjust the software design parameters; different material limitations that would affect the thickness of the cladding panels as well as the spanning capabilities can be provided for by changing the basic parameters within the software agents. The details of the structural system are designed to provide for panels of typical thicknesses. The panels are laser cut and scored in a similar fashion as the nodes, or could be site cut as necessary using local materials as available. Possible panel types include composite aluminum, glass, plastics, fabrics or even plywood.



► Figure 11. Schematic of *Trusset* dimensioning and performance criteria

3.3. Component assembly

As the intention is to create a system that could be easily erected by local labor without the use of heavy machinery, the elements are designed to be small and light enough to be manipulated by hand. This constraint limits the average rod length to approximately 1 meter in length, fixing the cladding panel sizes to typically 1 square meter in size. The scale assures that even shipping containers holding a large number of components to be transportable by two persons, even over rough terrain.

This use of gusset folded steel plate in the *Trusset* system not only allows for simple CNC fabrication, but is also intended to produce a lightweight and flexible structural system with a concern for ease of transport and

erection. The inventory software could potentially calculate efficient cutting of materials, as well as packing of components for shipping. The structure is designed for assembly on site with a minimum of tools by unskilled labor. Therefore, a typical disadvantage of the space-truss increased erection time is mitigated through the ease of assembly and low-skill level necessary for assembly.

Assembly of the *Trusset* system is no more complicated that building a balloon frame house from traditional two-by-fours. All components are marked directly on the material for ease of site fabrication and assembly. The process is as follows:

1. The steel plates are folded and attached to compose a node.
2. Rods are inserted along with the waterproofing sheets and bolted to the nodes in approximate locations.
3. A panel is placed in site with the appropriate spacing blocks, allowing the structure to be firmly bolted in the correct configuration.
4. The panel is then sealed in place with a one-part silicone sealant that provides waterproofing and structural adhesion.

In this method, a module of five nodes with its accompanying panel and rods are assembled, and this module is repeated in the sequence developed by the software and ordered in the packing method.

As the space-truss is a rigid structure, the components are self-supporting within limitation, allowing for the structure to be assembled without the use of excessive scaffolding. In addition, modifications to the position of the components are possible as the structural sealant is flexible and the panel mounting method allows for a range of setting conditions. Demounting the *Trusset* structure is as easy as deployment, and affords the opportunity of recycling and/or reusing some of the materials, relocating the structure, or reusing the structure at a later date, providing additional cost and sustainability efficiencies rarely realized in traditional space-truss structures.

4. Conclusion

Although the development of both the software and building components of the systems are in a rudimentary state, we can make some projections about the potential of the development process. As we continue the development of the components of *Trusset* system, we are convinced this method of parallel production of software and building system is a fruitful direction. The trend is appearing within the industry, as typified in the work of Frank Gehry, to apply parametric software development along with innovations in CNC fabrication to produce innovative design solutions. Our experimental project of developing a more general use application such as the Trusset has served to magnify the potentials of this integrated co-development.

We have found the development of a software design application, calibrated to a specific construction and structural methodology, produces more robust design tools than developing applications in abstract, because of its linkage to physical reality. The process of the development of the *Trusset* system so far involved only small-scale physical models and virtual prototyping. However, the capability of producing and testing physical forms with a clear linkage back to specific parameters and algorithms within the software provided a rapid feedback mechanism within the software development process typically unavailable in software development for architectural applications. Conversely, the development of the building system in parallel with the software provided for innovations in design of the fabrication process and construction logic. The ability to test algorithms for the generation of building components, and use computational methods as a foil to building methods, provided a critical means to surpass many of the limitations of the building system we have confronted so far in its development.

The development of both systems is still in its preliminary phases and the future goal of the project is to continue development of both the software and building components of the *Trusset* system. For the building component, the next crucial step is for full-scale prototyping of structural details and a small-scale structure, structural testing and analysis, as well as hypothetical development of several design applications. Since the *Trusset* system is designed for high material efficiency, low cost, and simple assembly, we are hoping for potential application in humanitarian missions, or usage by government agencies building small and large-scale temporary and permanent shelters. The development of full-scale prototypes will obviously necessitate the development of the currently under-developed G-code and parts schedule components of the structure/inventory software module. A potential avenue of long-range software development being considered is the implementation of a genetic algorithm within the design module that would allow for the iterative development and testing of formal options for particular design problems.

References

1. Fischer, T. Burry, M. and Woodburry, R., "Object-Oriented Modeling using XML in Computer-Aided Architectural and Educational CAD", in Tan, Beng-Kiang et al (eds). *CAADRIA 2000 Proceedings*, pp. 145-155.
2. Fischer, T. Burry, M. and Frazer, J., "Triangulation of Generative Form for Parametric Design and Rapid Prototyping", in *21th eCAADe Conference Proceedings*, Graz, September 2003, pp. 441-448.
3. Wolfram, Stephen. *A New Kind of Science*. Wolfram Media Inc, Champaign IL., 2002.
4. Clarke, C. and Anzalone, P., Architectural Applications of Complex Adaptive Systems, in: Klinger, K., ed., *ACADIA 22 Connecting – Crossroads of Digital Discourse*, Ball State University, Indianapolis, 2003, pp. 324-335.
5. Frazer, J. *An Evolutionary Architecture*. London: Architectural Association, 1995, p. 86
6. ibid. p 90

7. Reynolds, C.W., "Flocks, Herds and Schools: A Distributed Behavioral Model", in *Computer Graphics*, Vol. 21, No. 4, SIGGRAPH '87 Conference Proceedings, pp. 25-34.
8. Testa, P., O'Reilly, U.M., Kangas, M., Kilian, A., "MoSS: Morphogenetic Surface Structure – A Software Tool for Design Exploration" in *Proceedings of Greenwich 2000: Digital Creativity Symposium*, London: University of Greewich, 2000, pp. 71-80.
9. Chilton, J., *Space Grid Structures*, Architectural Press, Oxford, 2000.
10. Ramaswamy, G. S., Eekhout, M. and Suresh, G. R., Analysis, *Design and Construction of Steel Space Frames*, Thomas Telford, London, 2002.
11. Thornton, W., AMA *Institute of Steel Construction and American Institute of Steel, Manual of Steel Construction: Load and Resistance Factor Design, 3rd Edition*, American Institute of Steel Construction, Chicago, 2001.
12. Aluminum Association, *Aluminum Design Manual*, Aluminum Association, Inc., Washington D. C., 2002.

Cory Clarke[1] and Phillip Anzalone[2]
[1]XO (eXtended Office), 457 State St #4B, Brooklyn NY 11217
[2]XO (eXtended Office)

[1]cory@nthd.org
[2]phil@a-node.net