Amatria
Sentient Testbed

Philip Beesley, Katy Börner, Andreas Bueckle & Miggy Torres
Living Architecture Systems Group

Amatria is a 2016 sculpture authored by the Living Architecture Systems Group located within Indiana University, Bloomington, Indiana, USA. This folio provides an overview of the technical organization of Amatria.

Amatria is a permanent sculpture offering a delicate quilt-like overhead canopy composed of hundreds of thousands of custom-made transparent components. The experimental research foregrounding this permanent testbed sculpture asks: what if buildings could know us, communicate with us us and care about us? This is an experimental testbed where features will naturally evolve, and existing additional electronic hardware, software and physical features are available for further development.

Amatria, hanging above the stairs in the 4th floor atrium of Indiana University's Luddy Hall, invites visitors into an interactive, ethereal space. Soaring clouds and tangled thickets of 3D-printed formations are embedded with artificial intelligence that produces waves of constantly-changing sound, whispering and calling in response to the movements of viewers. By merging lighting and motion sensors with atmospheric sounds, this "living" sculpture breathes, undulates and shifts in response to the movements of visitors. Expanding the interpretation of the project, Indiana University students and faculty designed custom data visualizations mapping Amatria's sensor and actuator types, their positions, communication flows and complex behaviors unfolding over time.

ISBN 978-1-988366-56-2
9 781988 366562
Riverside Architectural Press

AMATRIA
SENTIENT TESTBED

PHILIP BEESLEY, KATY BÖRNER,
ANDREAS BUECKLE & MIGGY TORRES
LIVING ARCHITECTURE SYSTEMS GROUP

LASG
INDIANA UNIVERSITY
UNIVERSITY OF WATERLOO

AMATRIA
Sentient Testbed

Philip Beesley, Katy Börner,
Andreas Bueckle & Miggy Torres
Living Architecture Systems Group

LASG
INDIANA UNIVERSITY
UNIVERSITY OF WATERLOO

Published by Riverside Architectural Press
www.riversidearchitecturalpress.ca

Published in April 2023

Title: Amatria: Sentient Testbed
Authors: Philip Beesley, Katy Borner, Andreas Bueckle and Miggy Torres
Living Architecture Systems Group, issuing body.

Description: Series statement: Living Architecture Systems Group folio series |
Identifiers: ISBN 978-1-988366-56-2 (Online)

Design and Production: Philip Beesley, Timothy Boll, Adrian Chîu, Kevan Cress,
Lisa Jiang, Alex Paxton, Abida Rahman, Rekha Ramachandran

Riverside Architectural Press
7 Melville Street
Cambridge, Ontario, N1S 2H4
Canada

This book is set in Garamond and Zurich BT.

INDIANA UNIVERSITY
UNIVERSITY OF WATERLOO
Social Sciences and Humanities Research Council of Canada

Contents

Introduction

This folio provides an overview of the technical organization of *Amatria*. *Amatria* is a permanent sculpture offering a delicate quilt-like overhead canopy composed of hundreds of thousands of custom-made transparent components, creating a powerfully evocative presence drawing from interdisciplinary research and technological innovation. *Amatria* evokes the core theme of abiogenesis, the emergence of life occurring at the beginning of the universe. Soaring clouds and tangled thickets of 3D-printed formations are embedded with artificial intelligence that produces waves of constantly-changing sound, whispering and calling in response to the movements of viewers. Hovering, lightweight meshes create delicate canopies filled with quivering, pulsing mechanisms above and around this artificial forest. Densely massed spheres emulating the hollow bones of birds and mammals are lined with bubbling prototype cells filled with self-renewing chemistry. *Amatria* includes core contributions led by Amsterdam-based 4DSOUND, University of Waterloo Psychology Professor, Colin Ellard, Monash University Electrical & Computer Systems Engineering Professor, Dana Kulić, and Indiana University Distinguished Information Science and Computing Professor, Katy Börner. By merging lighting and motion sensors with atmospheric sounds, a "living" sculpture breathes, undulates and shifts in response to the movements of visitors.

The experimental research foregrounding this permanent testbed sculpture asks: what if buildings could know us, talk to us and even care about us? Indiana University students and faculty designed custom data visualizations mapping *Amatria*'s sensor and actuator types, their positions, communication flows and complex behaviors unfolding over time.

Amatria, hanging above the stairs in the 4th floor atrium, is a luminous, forest-inspired landscape of soaring clouds and tangled thickets of 3D-printed formations alive with artificial intelligence that invites visitors into an interactive, ethereal space. '*Amatria* is a near-living sculpture with a delicate canopy of mesh- and frond-like organic structures suspended from the ceiling in Indiana University's Luddy Hall atrium. It gathers information about its environment using light and motion sensors, responding with atmospheric sounds, undulating movements, and changing colors. It is aware of the people who enter its sphere to gaze upon its visual story of abiogenesis: the emergence of life during the earliest stages of development of the universe.

Amatria is an experimental testbed where features will naturally evolve and existing additional electronic hardware, software and physical features are available for further development.

The sculpture is organized in three general sections containing expressive sculpture components: a pair of spheres with automated lights, sound and motion; a canopy containing interactive lights, sound and motion and with infrared sensors and microphones, and a surrounding non-electronic 'cloud' canopy. Six clusters provide interactive functions for sculpture viewers. These are located in a row suspended along the stair-side edge of the interactive canopy. Two further sections contain technical equipment, concealed from public view: a ceiling-mounted control and power cabinet, and a central control closet containing three laptops and power switches that control the four power circuits of the sculpture. The control laptops are configured for secure online access. An additional virtual section, '*Amatria* Unveiled' is configured for public display, accessing and presenting a sensor data stream coming from the interactive system.

The chapter "Tavola: Data Visualizations for *Amatria*" are edited excerpts from "Envisioning Intelligent Interactive Systems: Data Visualizations for Sentient Architecture" by Katy Börner & Andreas Bueckle. The full essay originally appeared in the Living Architecture System Group's *White Papers 2019*.[1]

1 Katy Börner and Andreas Bueckle, "Envisioning Intelligent Interactive Systems: Data Visualizations for Sentient Architecture," in *Living Architecture Systems Group White Papers 2019* (Toronto, Canada: Riverside Architectural Press, 2019), pp. 63-88.

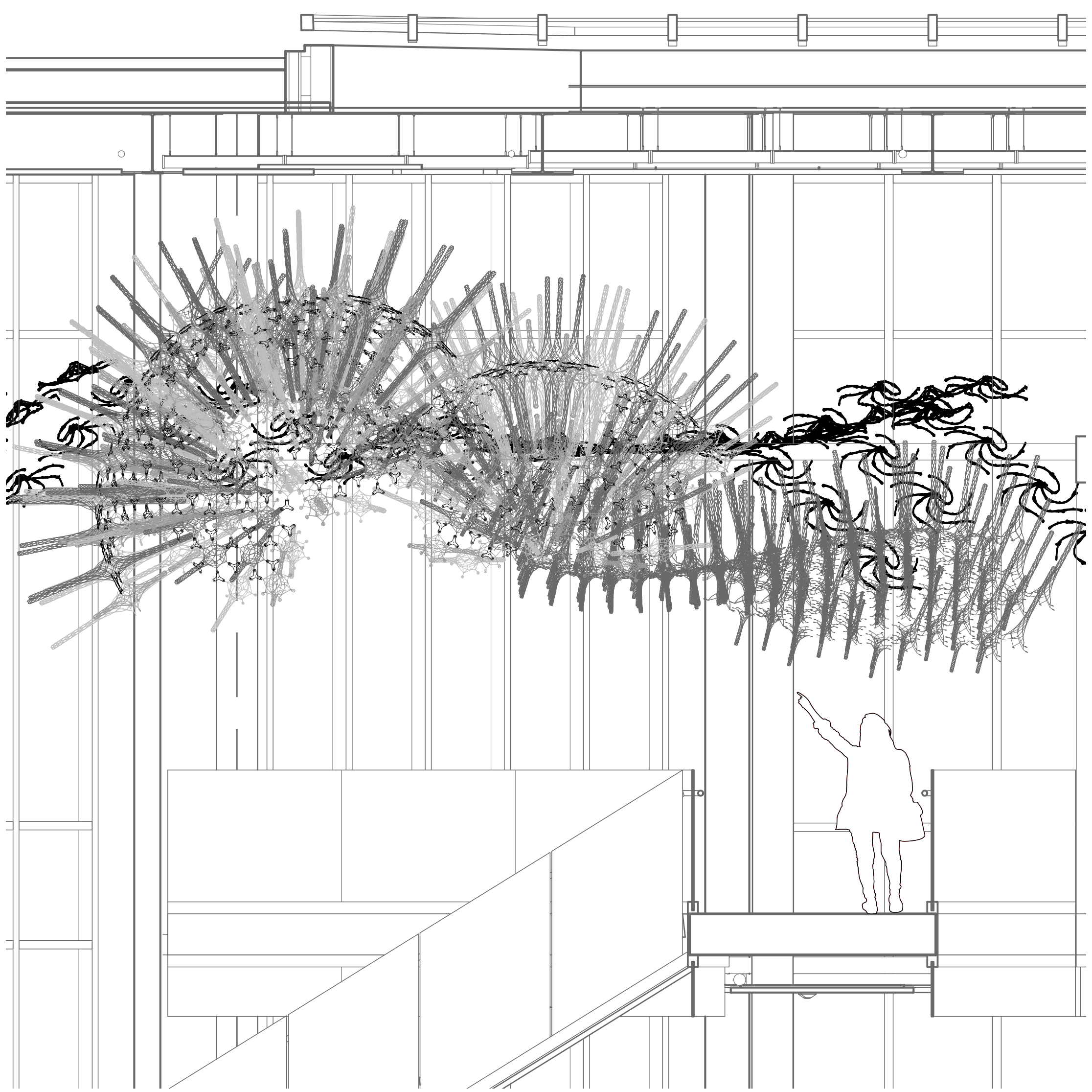
Amatria, Indiana University's Luddy Hall, Bloomington, Indiana (2018)

Amatria, Indiana University's Luddy Hall, Bloomington, Indiana (2018)

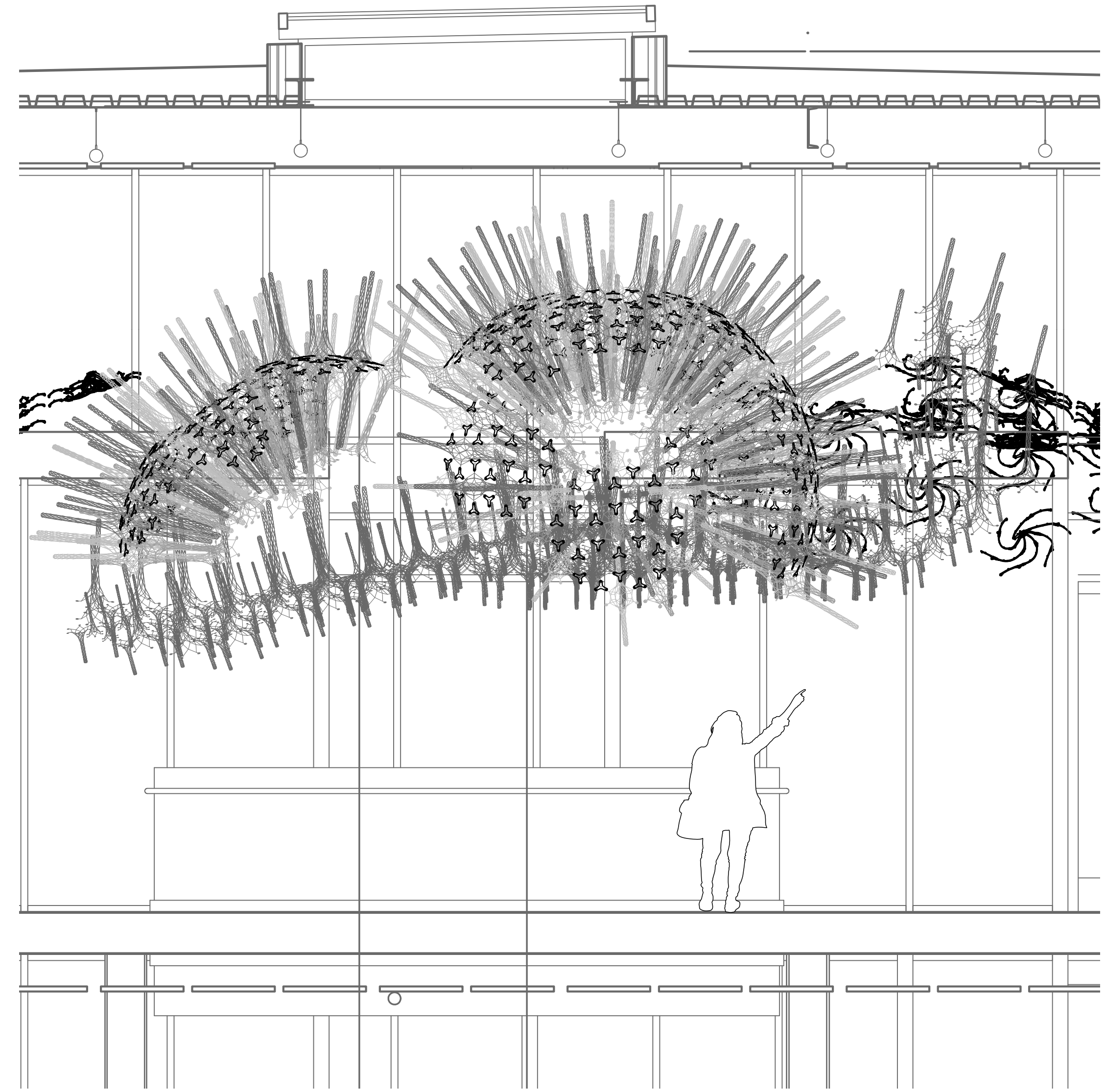
Amatria, Indiana University's Luddy Hall, Bloomington, Indiana (2018)

Amatria, Indiana University's Luddy Hall, Bloomington, Indiana (2018)

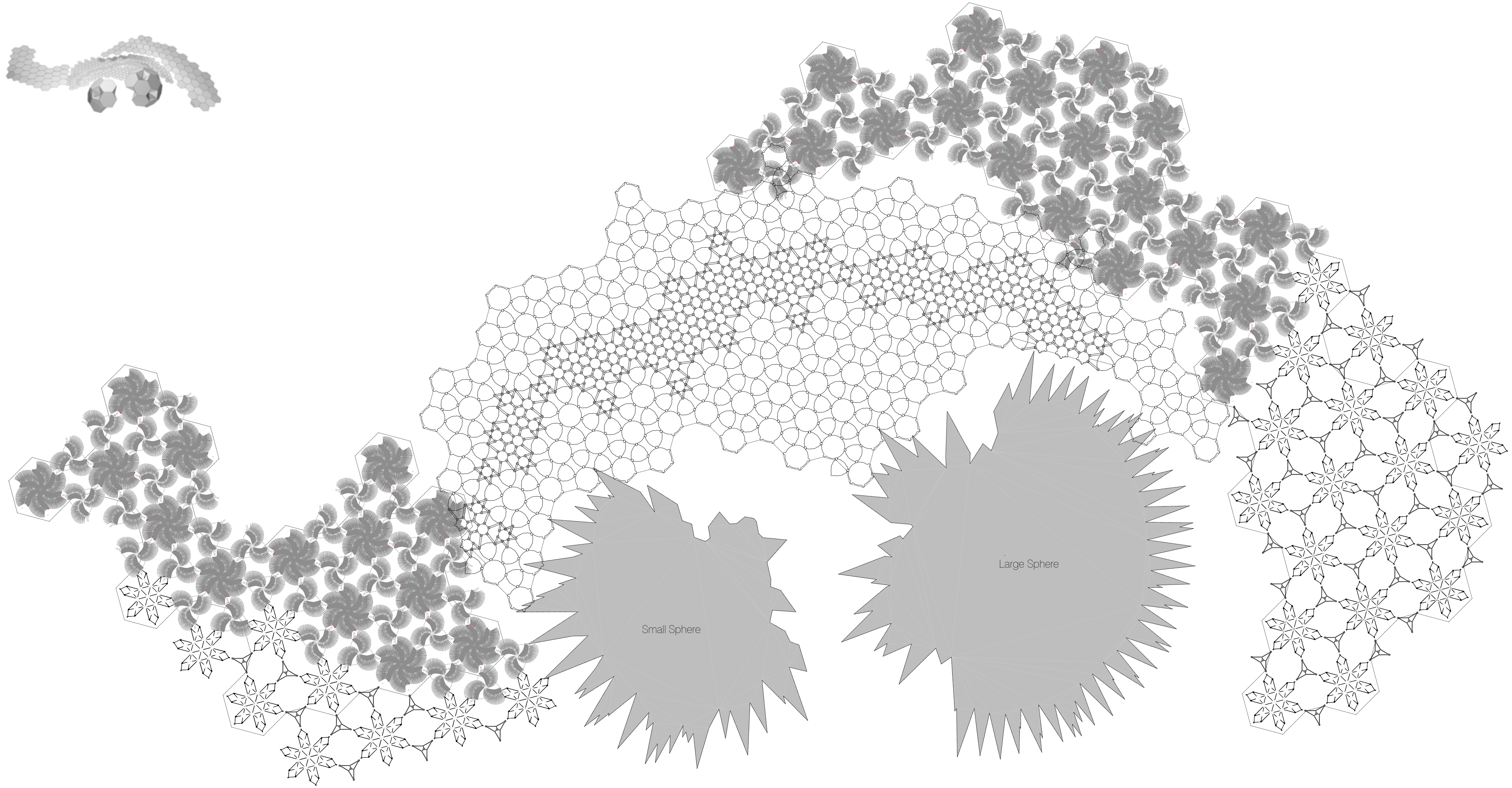
Amatria, Indiana University's Luddy Hall, Bloomington, Indiana (2018)



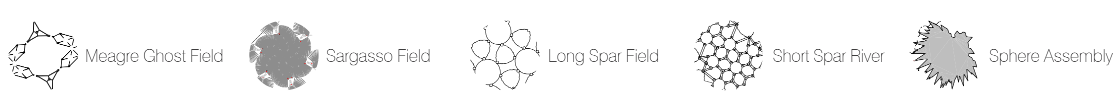
Longitudinal Section



Transverse Section



Plan of Amatria Sentient Testbed



How *Amatria* Is Organized

Amatria exists as an interactive architecture system composed of interconnected microprocessors, sensors and actuators. The interactive system is implemented using a set of networked electrical and electronic components which control high-power LEDs, vibrating mylar fronds with DC vibration motors and LEDs, and speakers. The system employs background behaviour, background sound, and responsive sound, lighting, and vibrating movement. Responsive functions are triggered by infrared sensors and by microphones. Software is configured for managing the network, reading sensors and controlling actuators. Consistent with the evolving nature of this research system, certain additional electronic hardware, software and physical features are available for further development, detailed below.

Amatria contains a series of actuators and sensors controlled by microprocessors. The control system is described below. Clustered electronically active components include Sound Sensor Scouts, high-powered LEDs housed within custom glass and 3d printed housings, and vibrating mylar fronds powered by miniature DC motors. The LED fixtures are positioned in groups within the Large and Small Sphere, within clusters along the front edge of the canopy and in a distributed field along the back edge of the canopy. Vibrating mylar 'Moth' fronds are positioned in repeating arrays that line the interior surfaces of the Large and Small Spheres.

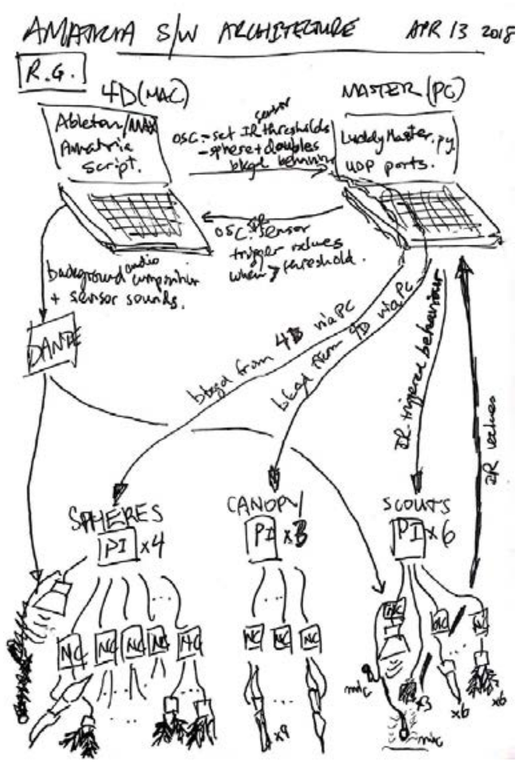
Sensors are grouped together with selected actuators, creating a series of central clusters called Sound Sensor Scouts (SSS). Each SSS contains a speaker, six high-powered LEDS and six pairs of vibrating 'moth' fronds with LEDs. Accompanying these actuators are a set of three infrared sensors oriented to detect gestures and movement by viewers, a pitch-sensing microphone detecting ambient sound and also viewer-generated sound, and an additional microphone mounted to detect sound emitted by the speaker. Sensors include infrared sensors and microphones. Infrared sensors are configured as six groups each containing three sensors. Six pitch-detecting microphones are positioned within.

Controls

Luddy Hall's network infrastructure, three computers in a control closet, and the electronics embedded within *Amatria* form an interactive system of microprocessors, actuators, and sensors. The system runs on a hierarchy of processing units that distribute computing power and data throughout the sculpture. Three laptops located in the control closet perform specialized functions: a PC-based Master Laptop manages and communicates with all microprocessor-based hardware; a Macintosh 4DSOUND laptop receives sensor stimulus, manages master interactive behaviour, plays responsive sound on the distributed series of embedded speakers and passes signals through the ML, and a Webcam Laptop manages the two webcams via the IU network. The ML communicates with all of the Raspberry Pi (RPi) microprocessors over a User Datagram Protocol (UDP) connection, each RPi communicates with a subset of Node Controllers (NC) over a serial/USB connection, and each NC uses its electrically active pins to control and measure its connected actuators and sensors with system-generated voltage pulses and fluctuations.

Sound-based functions are controlled by the 4DSOUND Mac (Mac) connected to two parallel sound-generating systems, both employing the same amplifiers and speakers. The first sound system is central, with a series of recordings sent via Ableton software on the Mac to ten speakers embedded within the sculpture dedicated sound unit NCs within the sculpture using Dante Power Over Ethernet (Dante POE). The second sound system has a distributed control system where the Mac communicates with the ML using an Open Sound Control (OSC) command interface provided by the main system code running on the ML. This interface allows devices to access individual actuators and sensors over the local network by sending formatted OSC packets to the correct IP and port. This second type of sound is emitted from Teensy Audio boards. This second system also includes live sources of sound from six microphones embedded within the system. The Teensy Sound Units record sound from the Mic, transform it, mix it into the audio coming from 4D, and plays it on the speakers. Audible sound within the sculpture is created through a combination of these two sources. Both sources are played through the same set of ten individual amplifiers and speakers embedded within the sculpture.

In addition to this control system, a Virtual Reality/Augmented Reality system providing interpretive real time display of key *Amatria* functions is in development currently configured as a tablet display, *Amatria* Unveiled (AU) tablet. The AU employs a wireless connection and communicates with the ML using OSC.



Left

Schematic diagram of *Amatria*'s control hierarchy

Active features and features available for further development

Core functions including control of all high-powered LED and vibrating Moth fixtures, recorded sound distributed over the Dante POE system, and limited sound processing distributed over the Teensy Sound control system are all implemented and were functioning as of April 2018. IR sensors are configured for basic triggering of the interactive system. Additional functions have been included but are not currently active. These are available for further development including:

- Wireless Xbee component configuration to open wireless communicationbetween Node Controllers for distributed local behaviour control, and to open the option of communication control and visualization by additional devices including Desktop Kit devices.
- Power monitoring sensors, applied to ten active and 4 reserve power supplies in cabinet
- Three IR clusters to provide directional response for interpreting human gestures and following movement
- Teensy Sound microprocessors for recording, processing and playing evolving sound
- Central laptops for machine learning and Curiosity-Based Learning Algorithm
- Pitch-detecting microphones for detecting varying kinds of sound, for flexible application as a sensor stimulus and for machine learning
- Speaker microphones for proprioceptive functions, detecting speaker-based sound for machine learning

Power System

Amatria is powered by 12 MeanWell power supply units (model SP-320-12) which each plug into a 120V 15A AC wall socket. These units output 12V DC to power the devices embedded in the sculpture. The voltages on all cables outside the closet is below 12V DC, posing no danger to humans. Power wires run from each PSU, along the umbilical trunk, and distribute over the surface of the sculpture. Upon entering the sculpture, these wires enter a power splitter PCB which further divides the power to be carried to RPis, Node Controllers, or Device Modules. These devices will drop the voltage further to 5V for certain parts using on-board voltage regulators. The actuators and sensors are powered generally by 5V DC from device modules.

Power Cabinet

The power cabinet is located near the ceiling outside the closet room and covered by opaque ceiling covers.

Inside the power cabinet there are 16 power supplies (12 used, and 4 backup) and 16 Green Hall-effect current transducers for monitoring current. The power hot line cable goes through the center hole of the current transducer. The current transducer will generate voltage (0~5V) in proportion to the current (0~10A DC) going in the power hotline. This output voltage is transmitted to the Node Controller and further to RPi in order to permit monitoring of power consumption.

Under normal conditions the *Amatria* current consumption will vary according to system behaviors, such as how many sensors and actuators are activated or the intensity of reaction. This monitoring subsystem offers an opportunity for machine learning research about system behavior and system energy cost.

Above

Mean Well power supply and Hall-effect current monitor

Right

Vinegar battery

Vinegar Battery

Vinegar batteries are small power-producing units made from the chemical reaction between vinegar acid, copper cathodes and aluminum anodes. The current produced from each cell is approximately 5 milliamps. When chained together in series, the vinegar battery could be attached to the Node Controller and used to trigger background behaviour. This function is currently dormant, available for future implementation.

Electronic Boards

This section will describe in detail the electronic boards for *Amatria*. Amplifiers, microphone boards, Teensy and Raspberry Pi microcontrollers are commercially available products. Other boards including Node Controllers, High Current Device Modules, Jack Plates and Power Splitters are custom electronic boards designed and produced by LASG/PBAI.

Raspberry Pi

The Raspberry Pi (RPi) is a commercial single-board computer widely used for embedded systems. It is used to program high-level behaviour of the connected Node Controllers used for processing in *Amatria* and other LASG installations.

Node Controller

A Teensy 3.2 Microcontroller is mounted on this custom PCB board. The board enables batch programming and serial communication between the RPi and Teensy 3.2 Microcontroller.

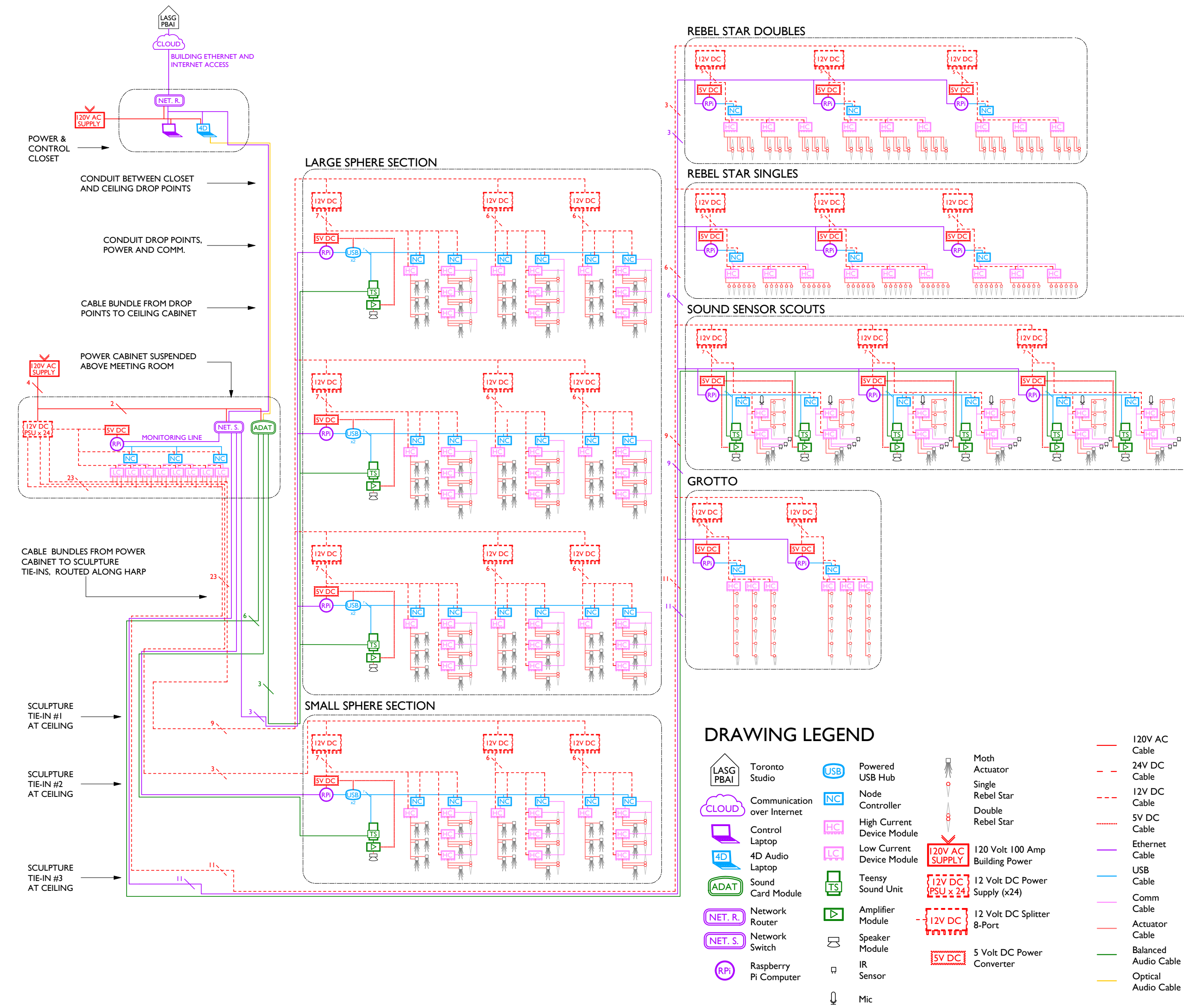
Node Controller boards control the local-level interactive behavior of *Amatria* while RPi is used for its more global behaviors.

High Current Device Module

A peripheral custom board that utilizes I/O pins from the Node Controller to control high current actuators such as lights and motor, and to read sensor values in *Amatria*.

XBee ZigBee Wireless Module

This commercially available module is mounted onto the Node Controller and offers Radio Frequency (RF) wireless communication between the Node Controller and RPi.

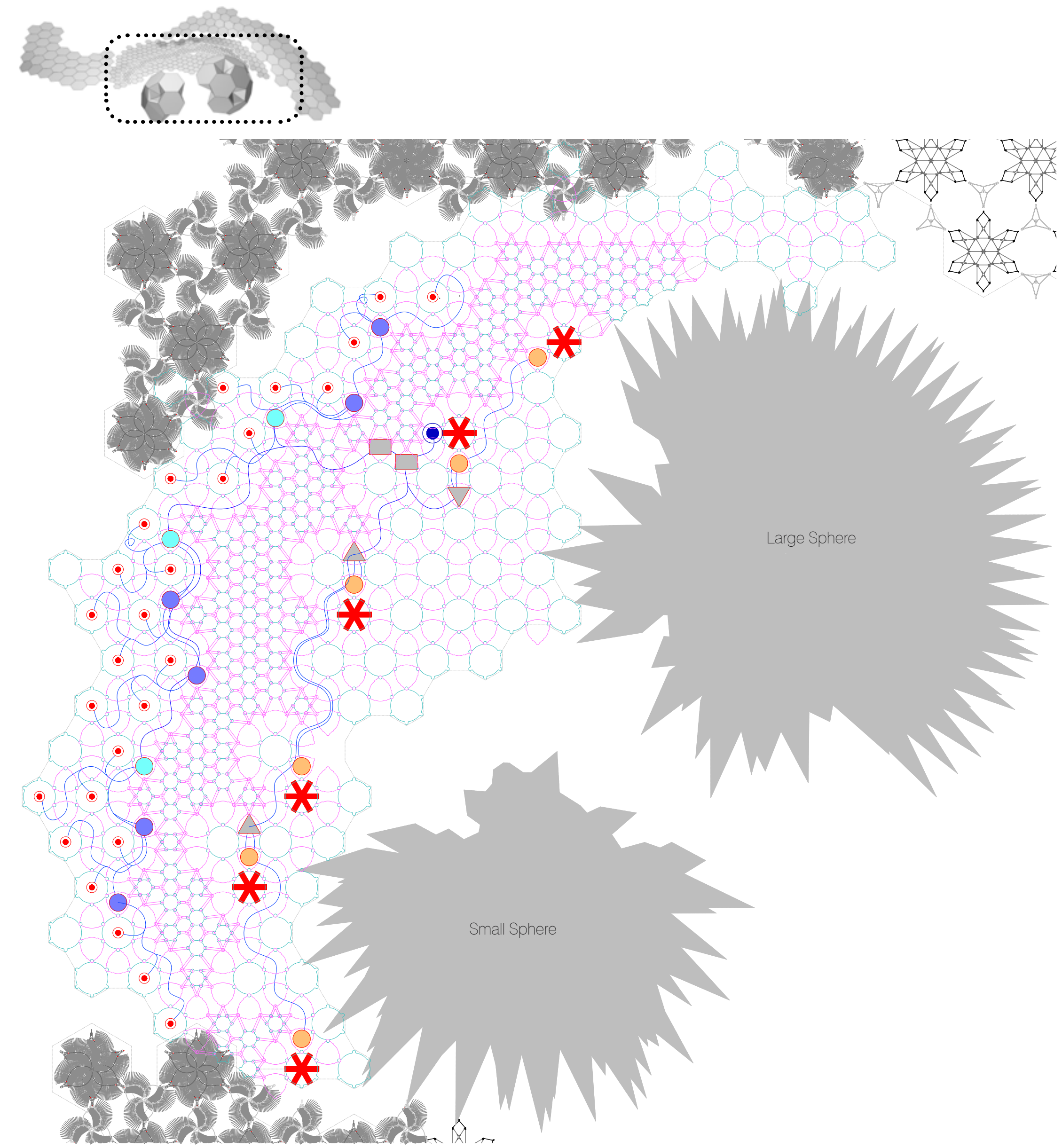


Electronics Block Diagram

Actuators and Sensors

The electronic system in the installation contains a hierarchical network of modules organized in three levels of control, including sensors and actuators connected to Device Modules; Node Controllers containing Teensy microcontrollers, and Raspberry Pi microcontrollers. A separate set of modules is organized for sound, employing Teensy Sound Modules.

The base of the hierarchy is sensors and actuators. Up to 2 sensors and 6 actuators can be connected to each Device Module B High Current, a peripheral board that utilizes I/O pins between actuators and the Node Controller. The next higher level of the network is the Node Controller, which is a custom PCB enabling batch programming and serial communication between RPi, which is the higher level electronic device, and Teensy Microcontroller. Next is the highest level of the electronics device – RPi. It is used to program high-level behaviour of connected Node Controllers used for calculations and control nodes to vary behaviours and systems.



Plan of Active Components in Spar Field

Power Cable · Umbilical Drop Point · Network Switch · Dante Audio Ethernet Adapter · Node Controller with Teensy Sound Unit
Node Controller with High Current Device Module · High Current Device Module · Sound Sensor Scout · Double Rebel Star

Sphere Section

The Large Sphere contains 9 actuated sphere units and the Small Sphere contains 3 actuated sphere units. At the base of this sphere hierarchy, each sphere unit contains 9 moths, 6 Rebel Star LEDs, 1 speaker and 8 custom PCBs and 4 RPi in total.

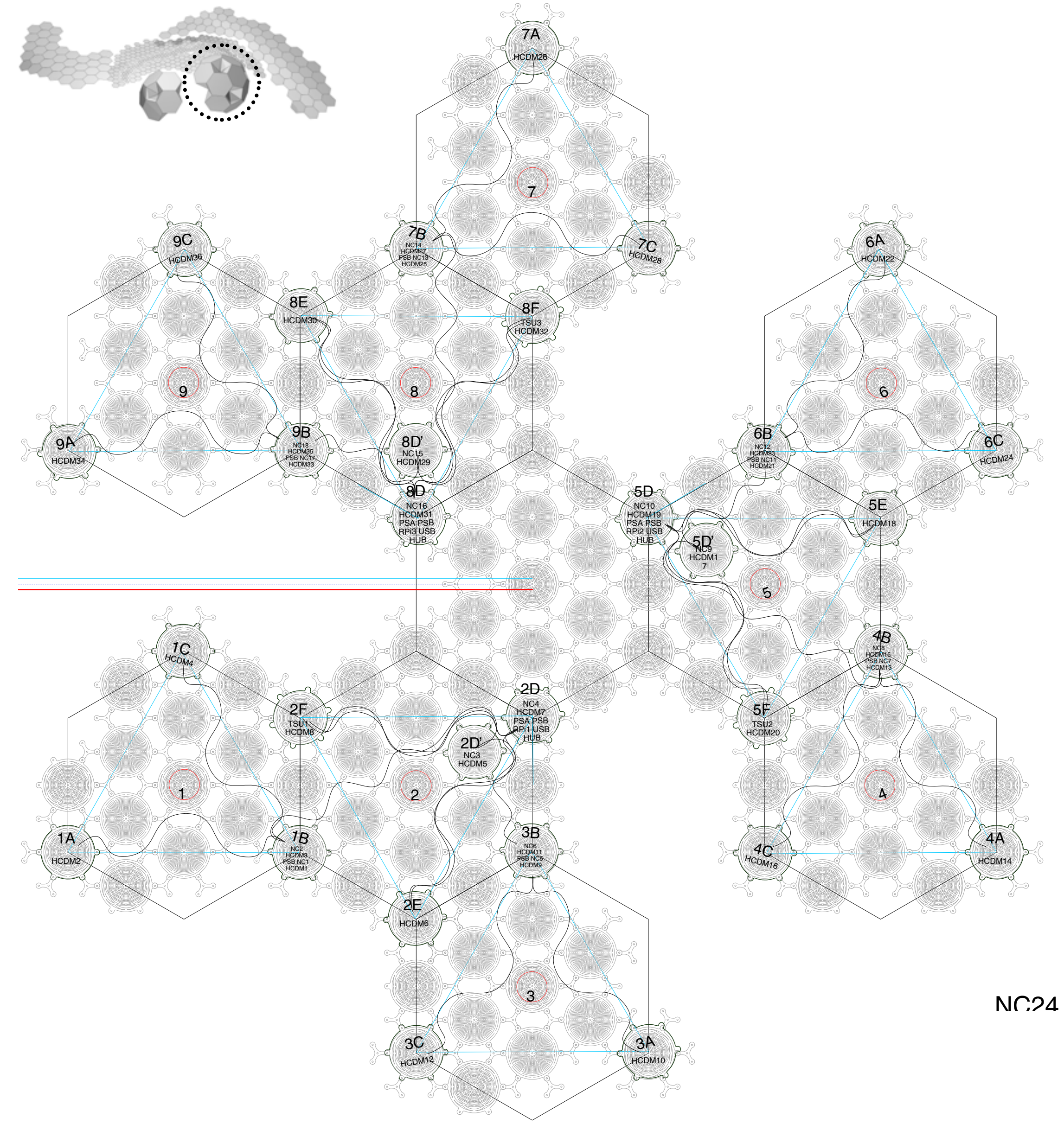
Sound Sensor Scout Section

There are six scouts in total. At the base of each scout there are 6 moths, 6 Rebel Stars LEDs, 3 infrared sensors, 1 Speaker, 1 sound detector sensor, 1 clip microphone, 1 RPi and 5 custom PCBs along with 18 vinegar cells suspended around each scout.

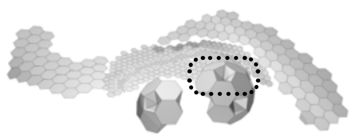
Double Rebel Star Section

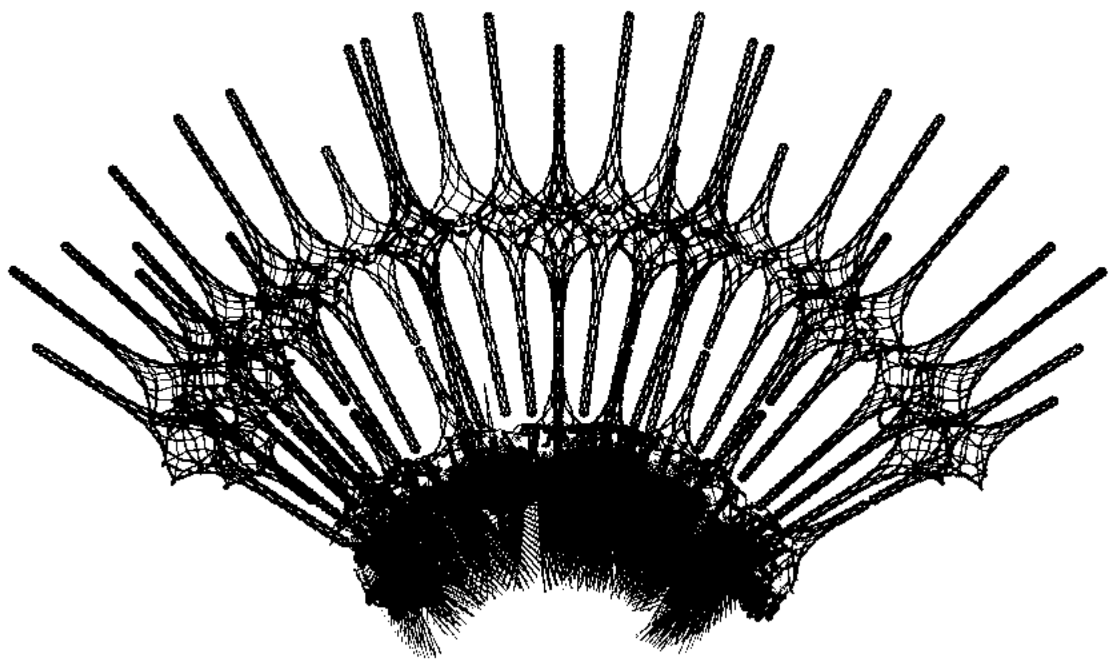
This section contains 54 Rebel Star LEDs, 3 RPi boards and 15 custom PCBs.

Each section in the sculpture is powered from 3 to 4 switching power supplies with a specification of 100-240VAC/5V.

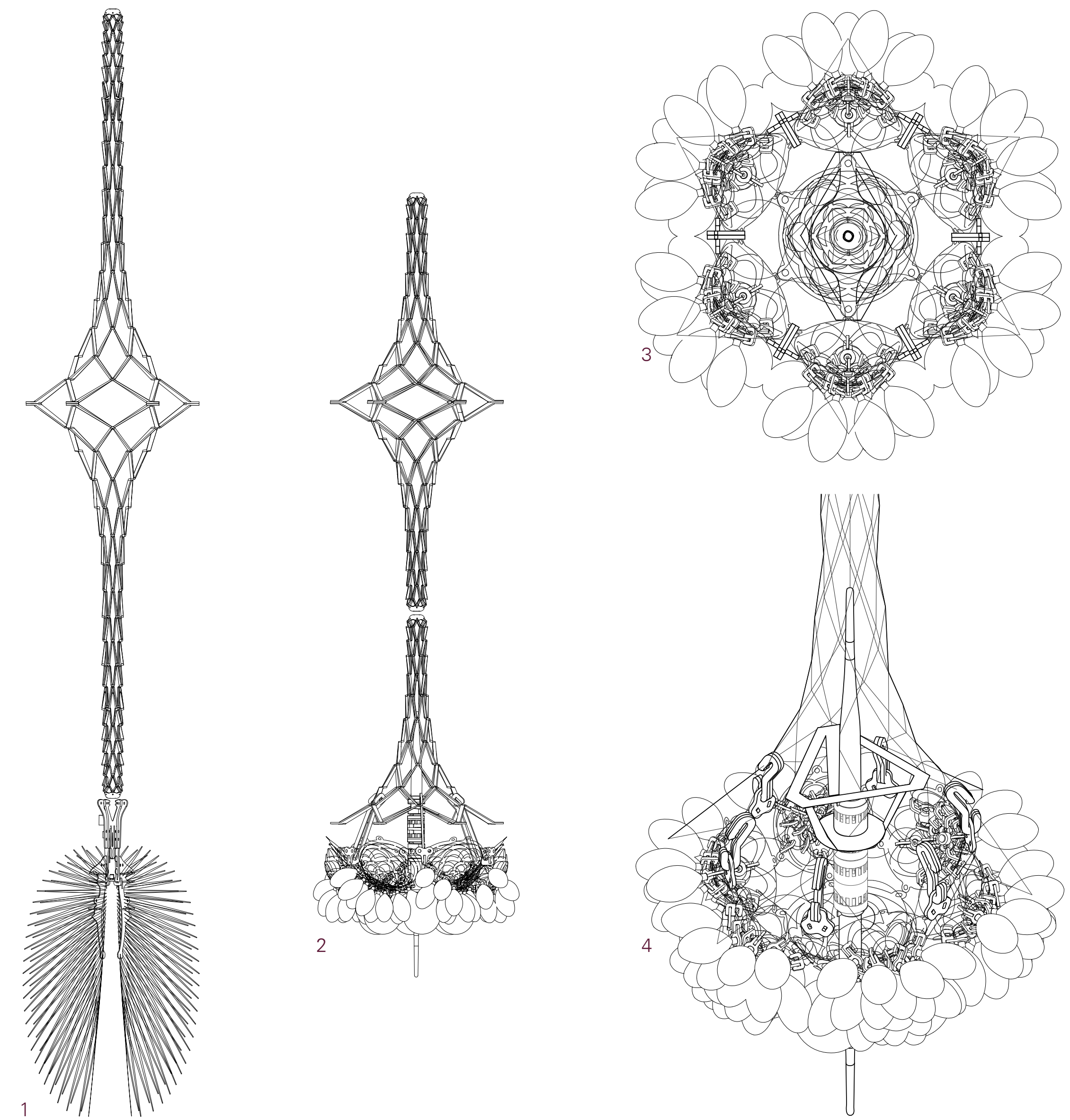


Unfolded Large Sphere Assembly



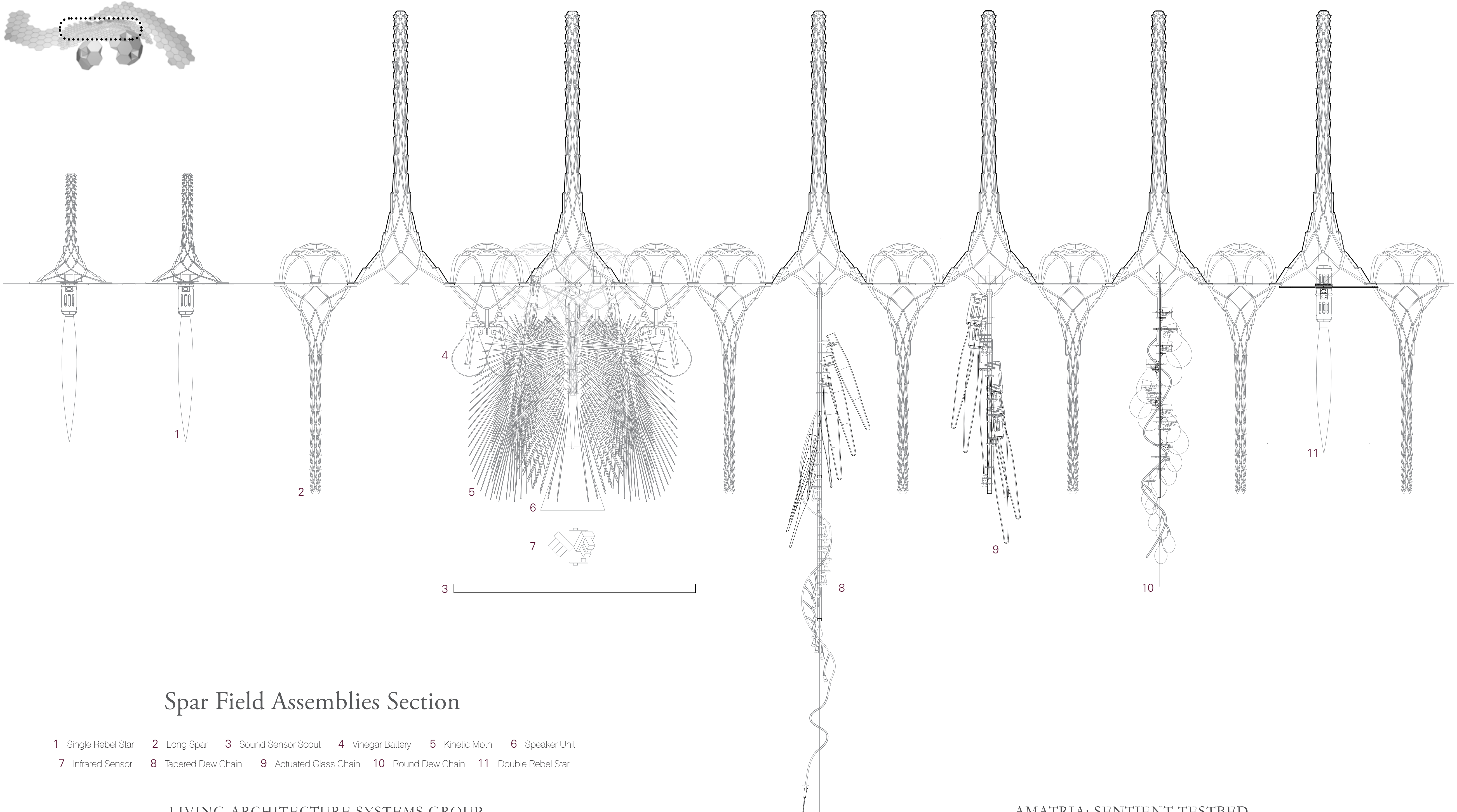


Sphere Section Detail



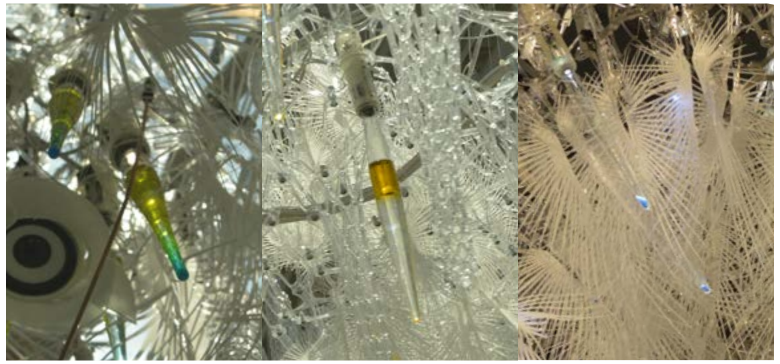
Detail of Sphere Spar Assemblies

1 Long Sphere Spar with Kinetic Moth   2 Short Sphere Spar with Glass Cluster   3 Glass Cluster Plan   4 Glass Cluster Detail



Spar Field Assemblies Section

1 Single Rebel Star 2 Long Spar 3 Sound Sensor Scout 4 Vinegar Battery 5 Kinetic Moth 6 Speaker Unit
7 Infrared Sensor 8 Tapered Dew Chain 9 Actuated Glass Chain 10 Round Dew Chain 11 Double Rebel Star



Left

Three types of Rebel Star LED in *Amatria*. Left: In Sound Sensor Scout; Middle: In Double Rebel Star section; Right: In Sphere section

Rebel Star LED

Bare LEDs are a purchased commercial product, and LASG designed the breakout board PCB to power and control them. Photos of these Rebel Star LEDs are shown below, which have different housing mounts designed by LASG. With respect to electrical power and control, these three types are the same. Liquid and solid color materials are integrated for decorative purposes.

Rebel Star LEDs are high-powered and each one consumes around 3.5W in its typical brightness. While providing sufficient lighting, these LEDs generate lots of heat. In order to increase its lifespan, a heat sink is sourced and designed in contact with the Rebel Star to reduce the temperature and avoid overheating inside the housings.

Kinetic Moth

A photo of a kinetic moth is shown below. It consists of a pair of DC motors, a pair of two low-power LEDs, a cell breakout board and non-actuated components such as a pair of white acrylic feathers.

The DC motor and LEDs are connected in parallel to the cell breakout board for powering and control. An on-board resistor is designed that adjusts the 5V DC voltage from the device module to around 3V DC for the motor and LEDs. The DC motor will create a vibration of the white feathers and LEDs provide interactive lighting responses according to the IR readings.

Above

Kinetic Moth. Location: in Sound Sensor Scout and Sphere sections.section

Above

Speaker Unit in spheres

Speaker Units

Ten speakers are positioned within *Amatria*, each housed within an enclosure that contains a resonating back chamber with folded horns that increase bass register response: six within Sound Sensor Scouts located in the canopy area; three within the Large Sphere and one within the Small Sphere. See below for an illustration of the speakers used. Two wires (signal and ground) connect each speaker to an amplifier positioned within the adjacent electronics cluster.

Above

Speaker Unit, three Infrared Sensors and Sound Detector located in Sound Sensor Scout section

Infrared Sensor

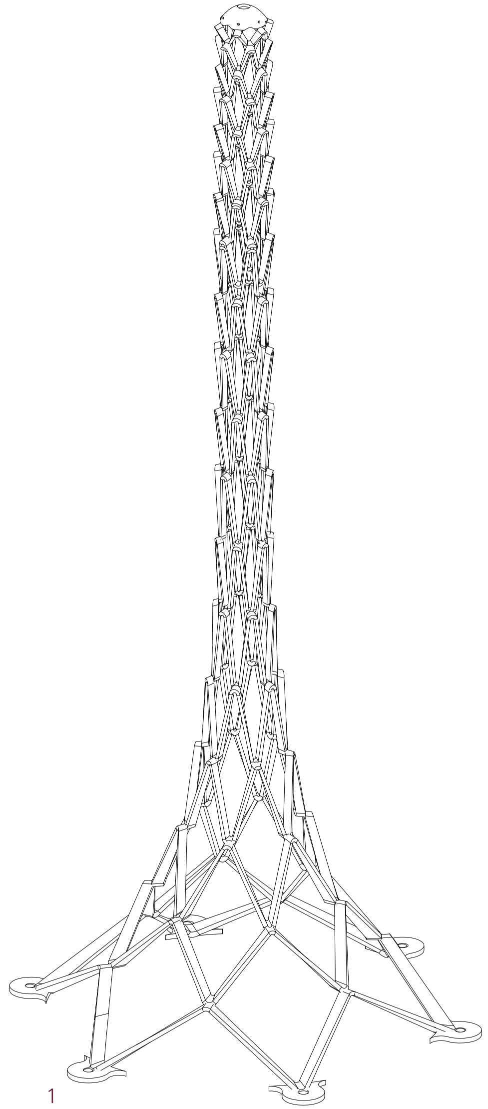
The Infrared (IR) Distance Sensor is a sensor that detects how far an object is in front of it.

Microphone Sound

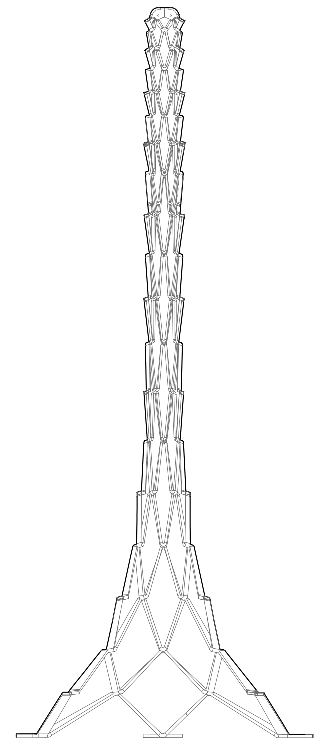
The Microphone Sensor is a sound sensor that detects loudness and can return audio signals to the Node Controller. Onboard pitch-sensing circuitry is also available for future configuration.

Speaker Sound Sensor

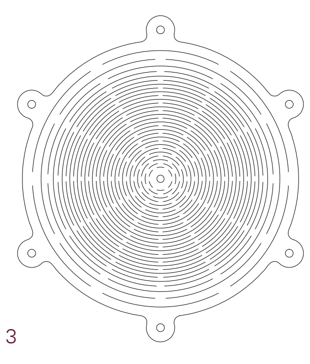
A Lavalier-brand miniature microphone is attached to each Sound Sensor Scout immediately beneath each speaker within that cluster. The device is plugged into the corresponding RPi. This device is positioned for future software configuration, anticipating machine learning. The future implementation of Curiosity Based Learning Algorithm would be supported by this proprioceptive function. The microphone is currently dormant.



1



2

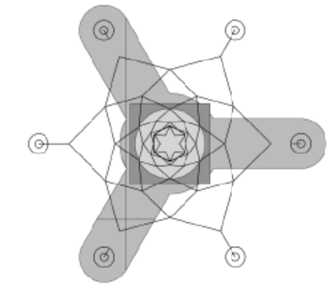


3

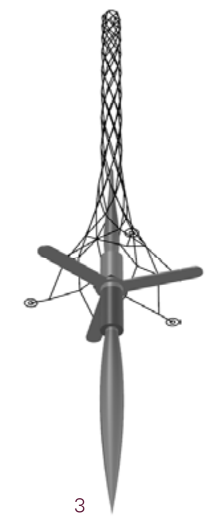
Long Spar

1 Long Spar 2 Long Spar Section 3 Long Spar Plan

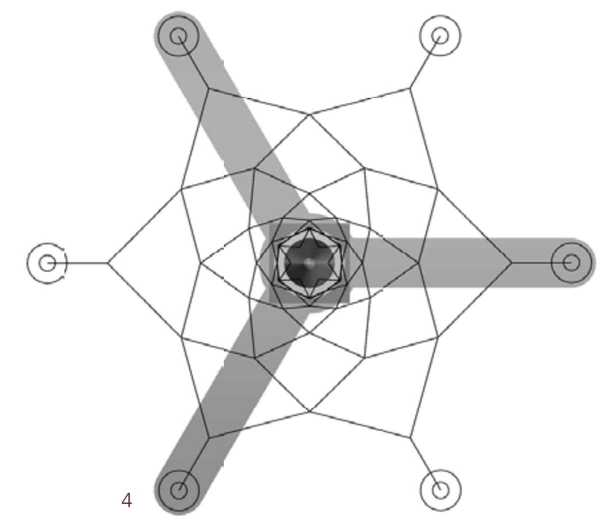
1



2



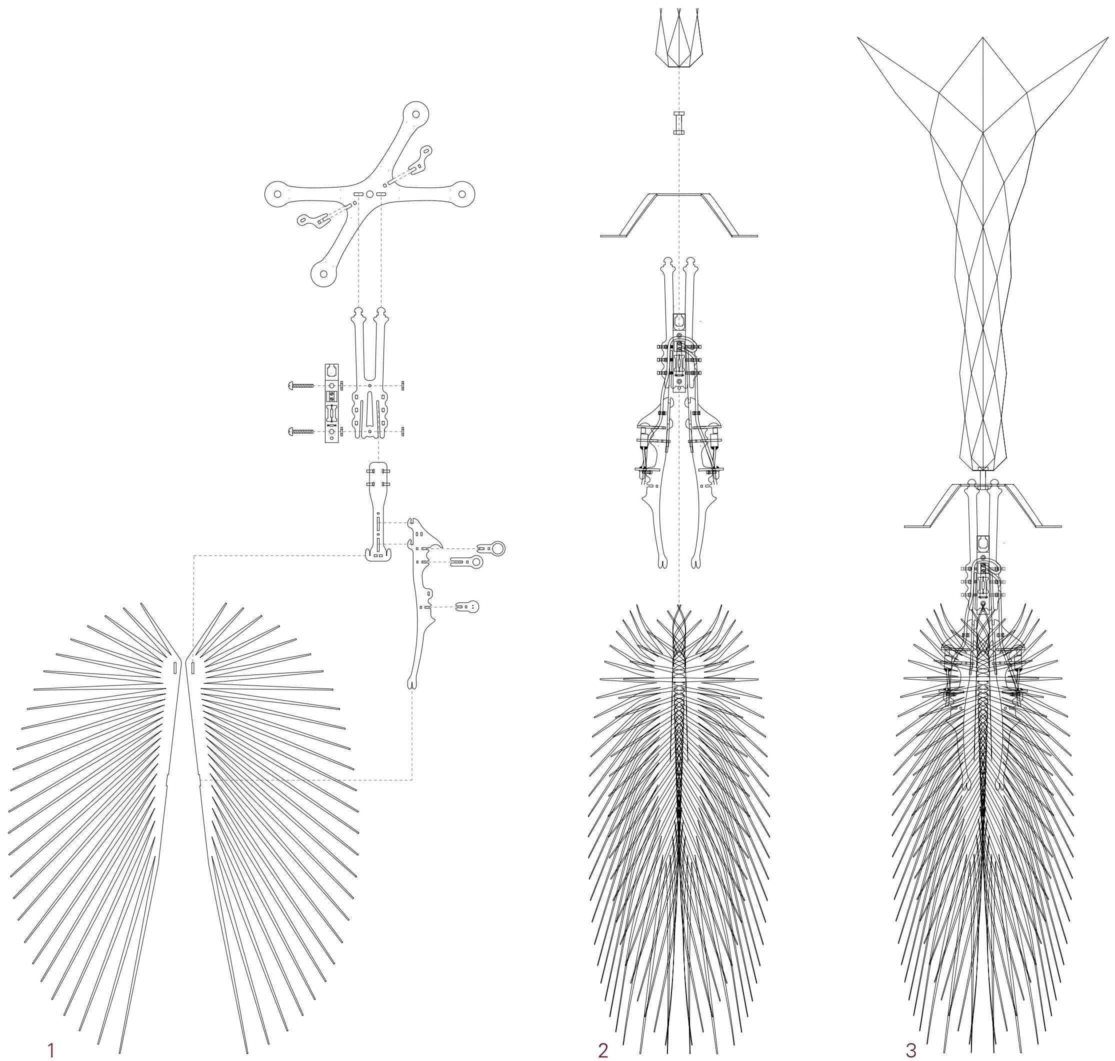
3



4

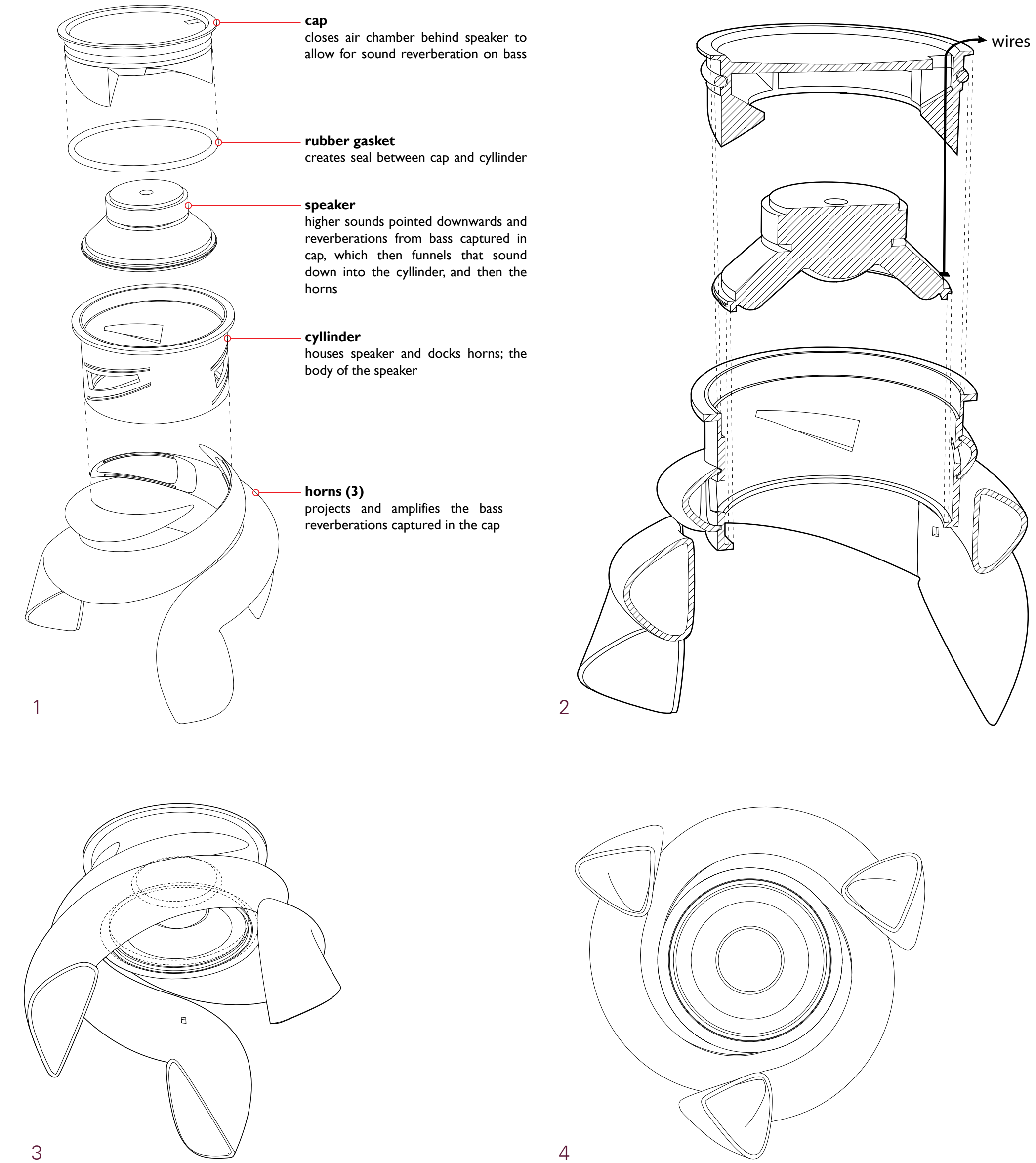
Single and Double Rebel Star

1 Single Rebel Star 2 Single Rebel Star Plan 3 Double Rebel Star 4 Double Rebel Star Plan



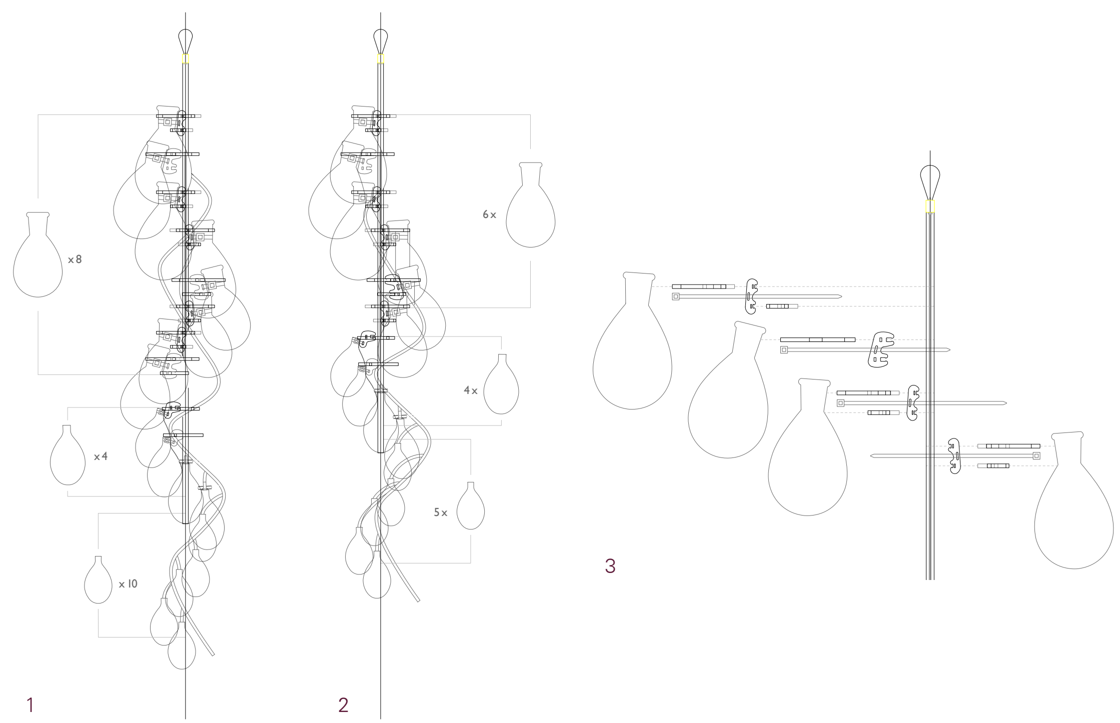
Kinetic Moth Assembly

1 Kinetic Moth Components 2 Kinetic Moth Exploaded View 3 Kinetic Moth Assembled View



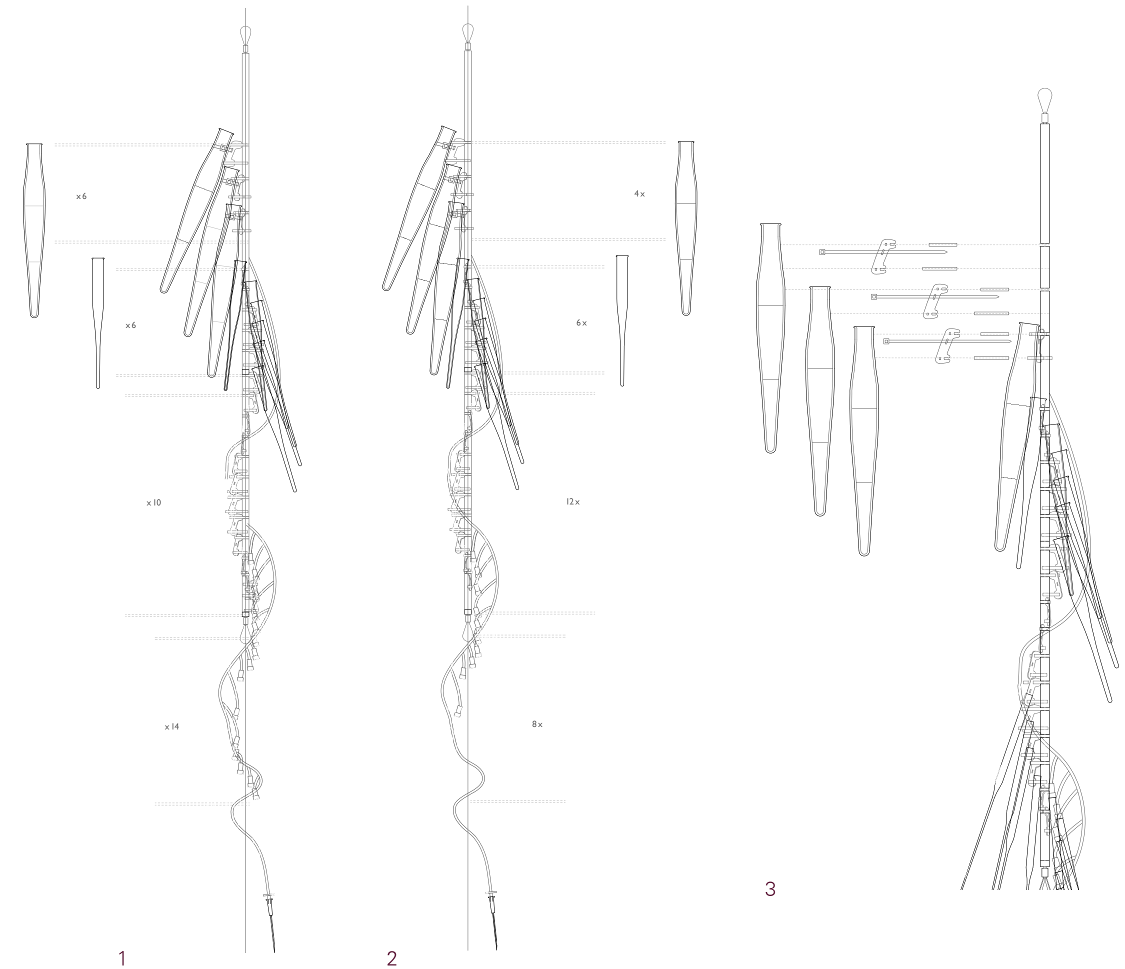
Speaker Unit Assembly

1 Speaker Unit Components 1 Speaker Unit Exploaded Section 1 Speaker Unit Assembled View 1 Speaker Unit Plan



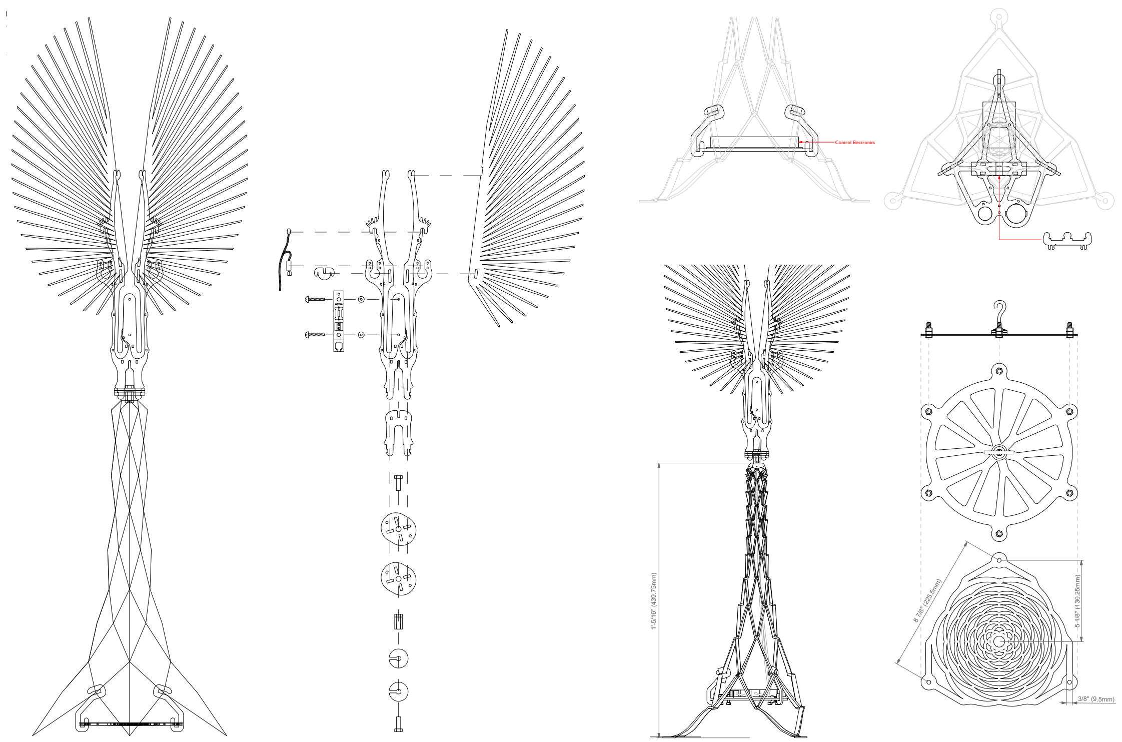
Round Dew Chain Assembly

1 Long Round Dew Chain 2 Short Round Dew Chain 3 Round Dew Chain Assembly Diagram



Tapered Dew Chain Assembly

1 Long Tapered Dew Chain 2 Short Tapered Dew Chain 3 Tapered Dew Chain Assembly Diagram



Amatria Exploration Kits

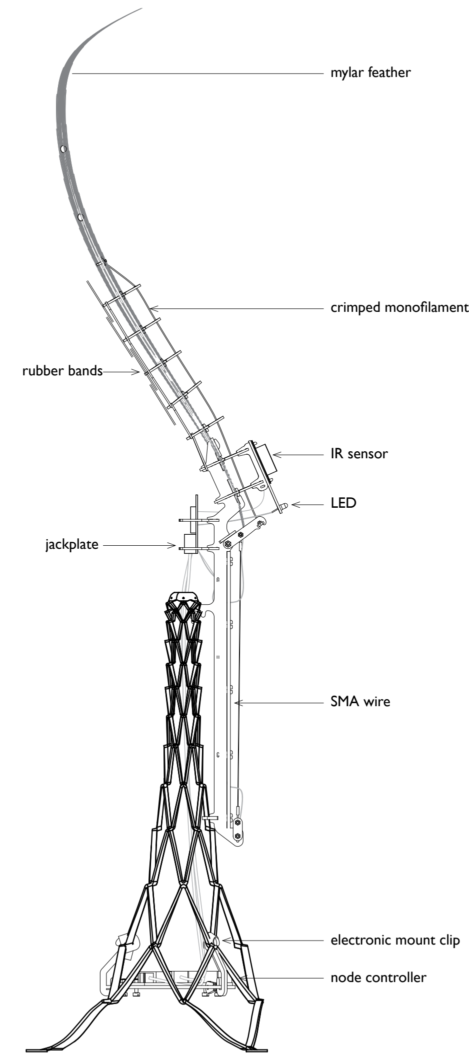
During the installation of *Amatria*, a complementary workshop was held in which local students were invited to participate and learn about *Amatria*. Actuation components found in the installation were adapted into kits that were distributed to participants. By giving the general audience the set of pieces in which to construct these actuators, they get a closer, more hands-on experience on learning about *Amatria* from its fabrication to how it behaves. The kits thus give a DIY introduction to digital fabrication, mechatronics and by interconnecting them, they give an introduction to the Internet of Things (IoT). The following pages show the kits that were distributed, from the pieces and tools needed to the final product.

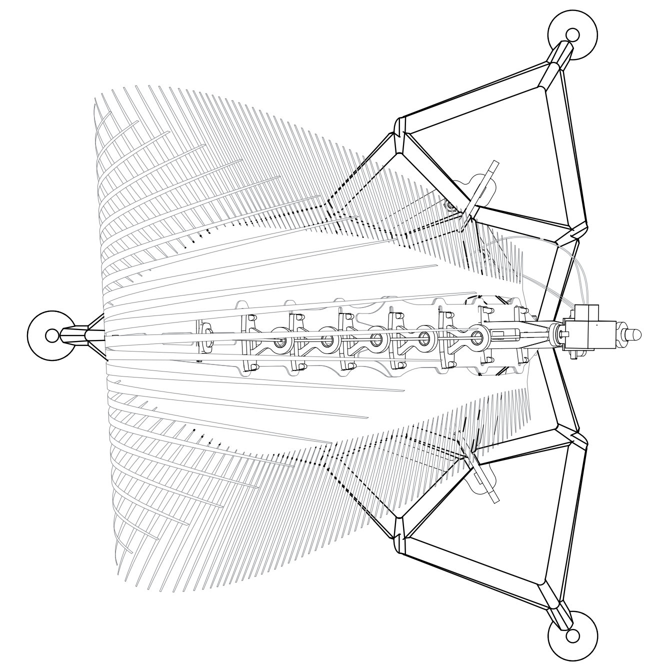
Above

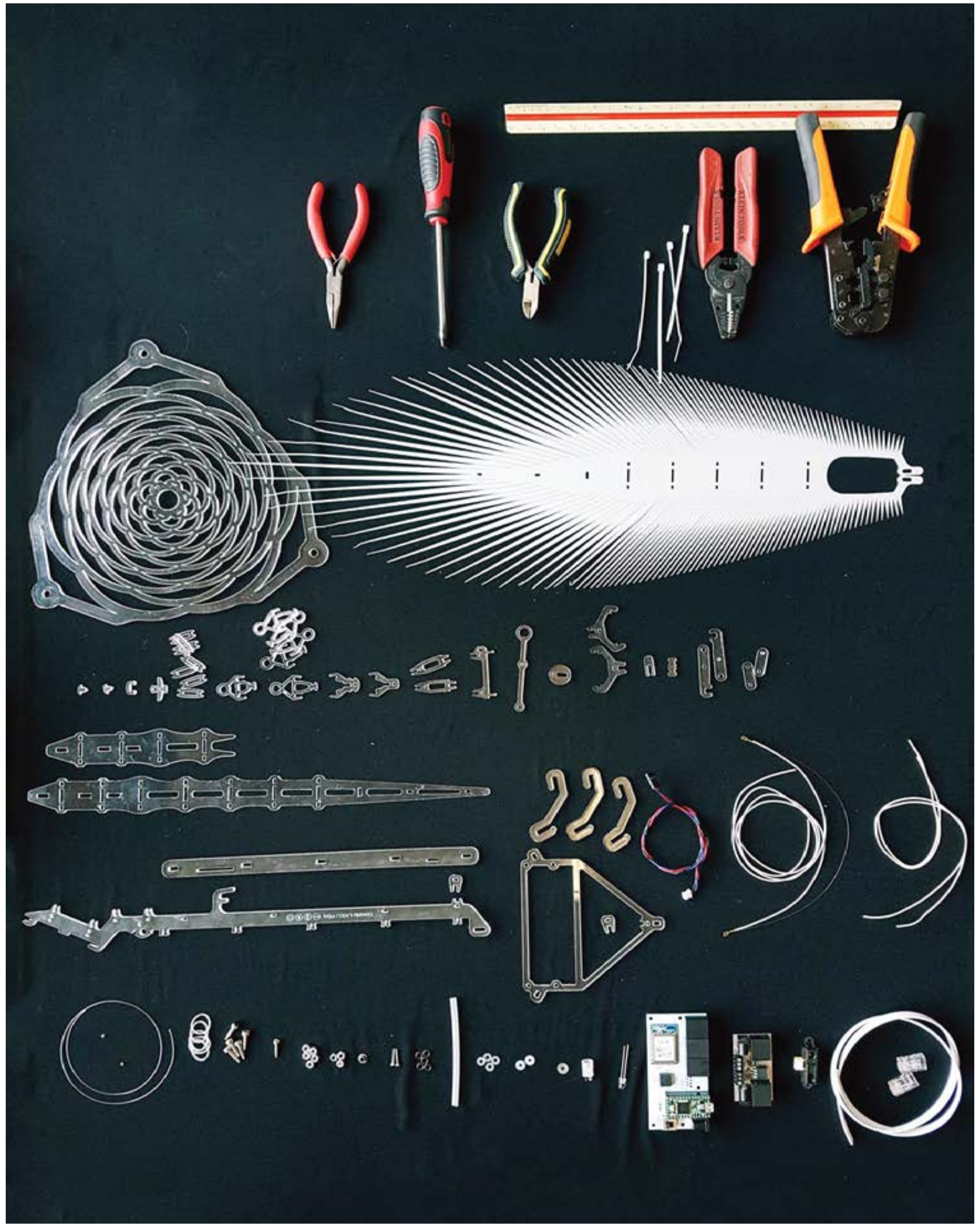
Instruction illustrations from the Moth Kit Manual

Facing

Orthographic illustrations of Dendrite Kit. Each Dendrite comprises one light sensor (the eye) and actuators such as lights and a strand of shape memory alloy that makes the sculpture move via software.

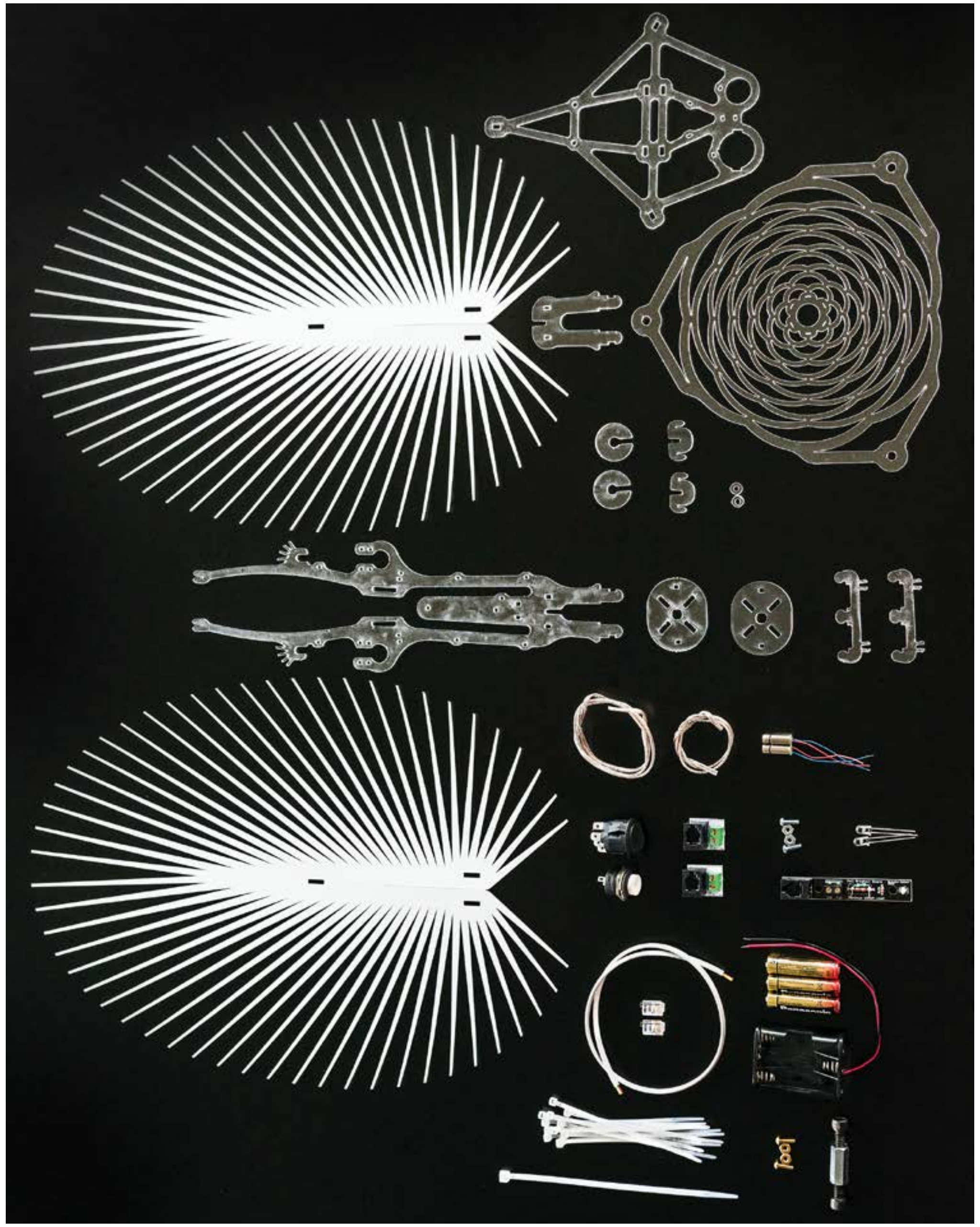






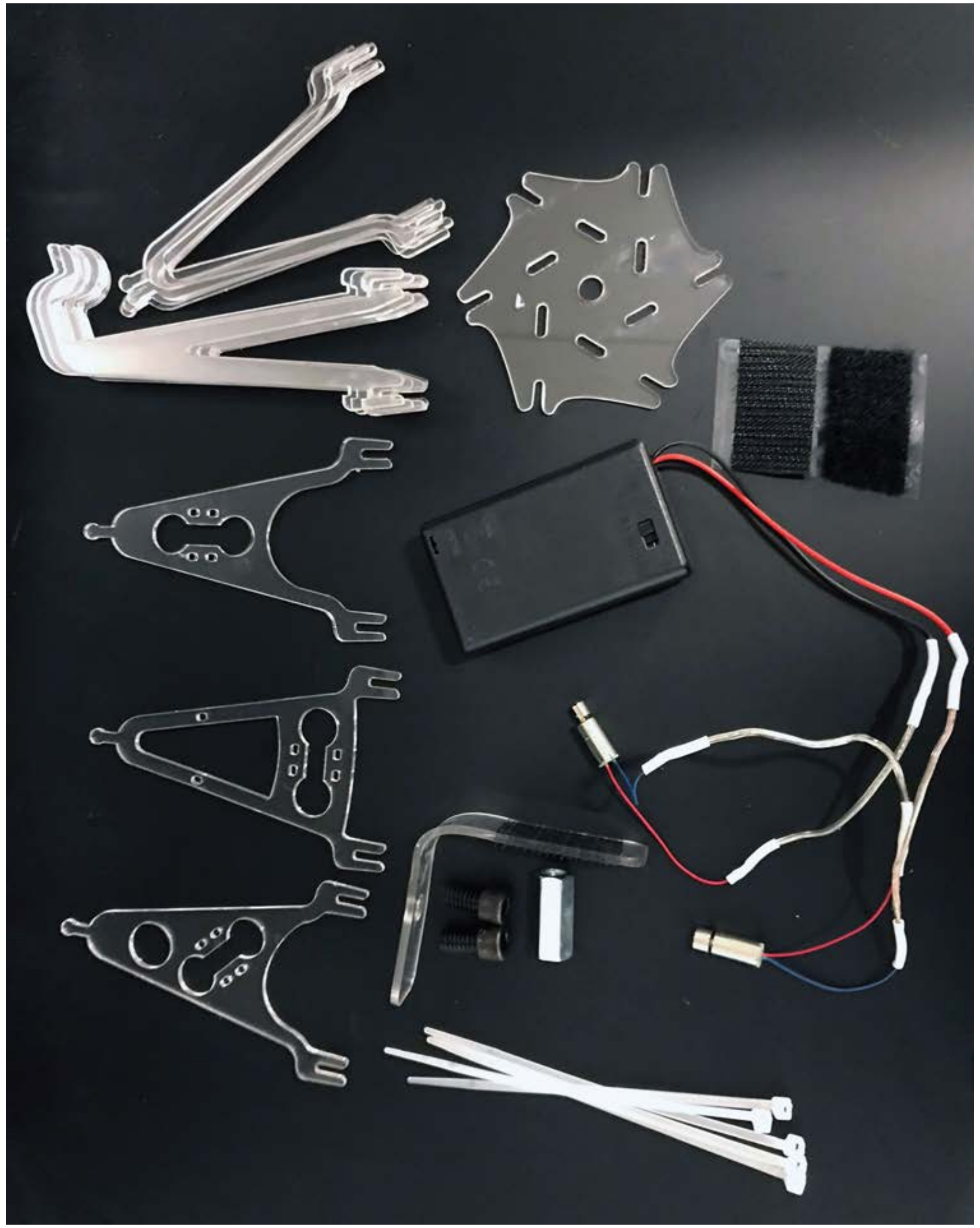
Dendrite Kit Components

Assembled Dendrite

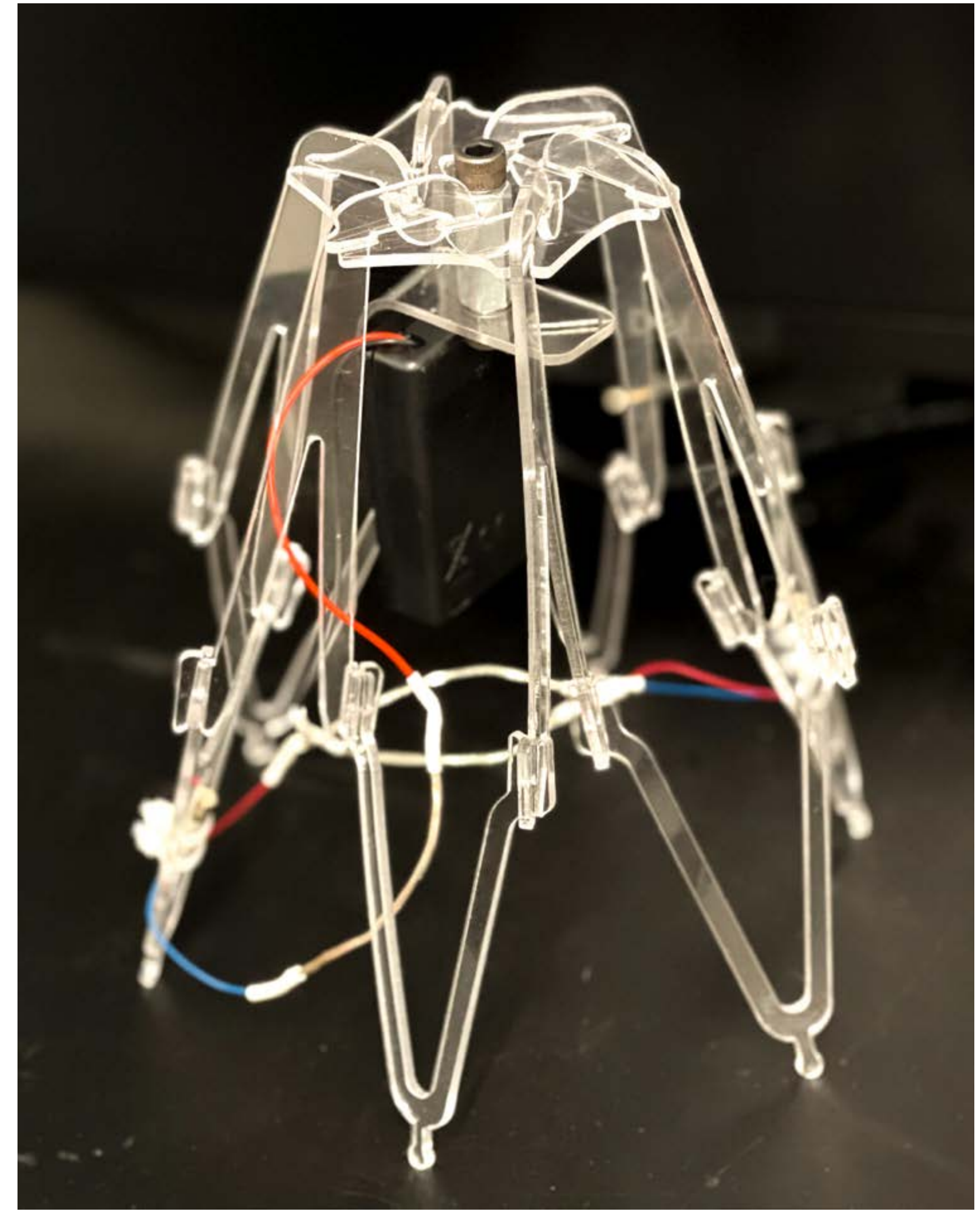


Moth Kit Components

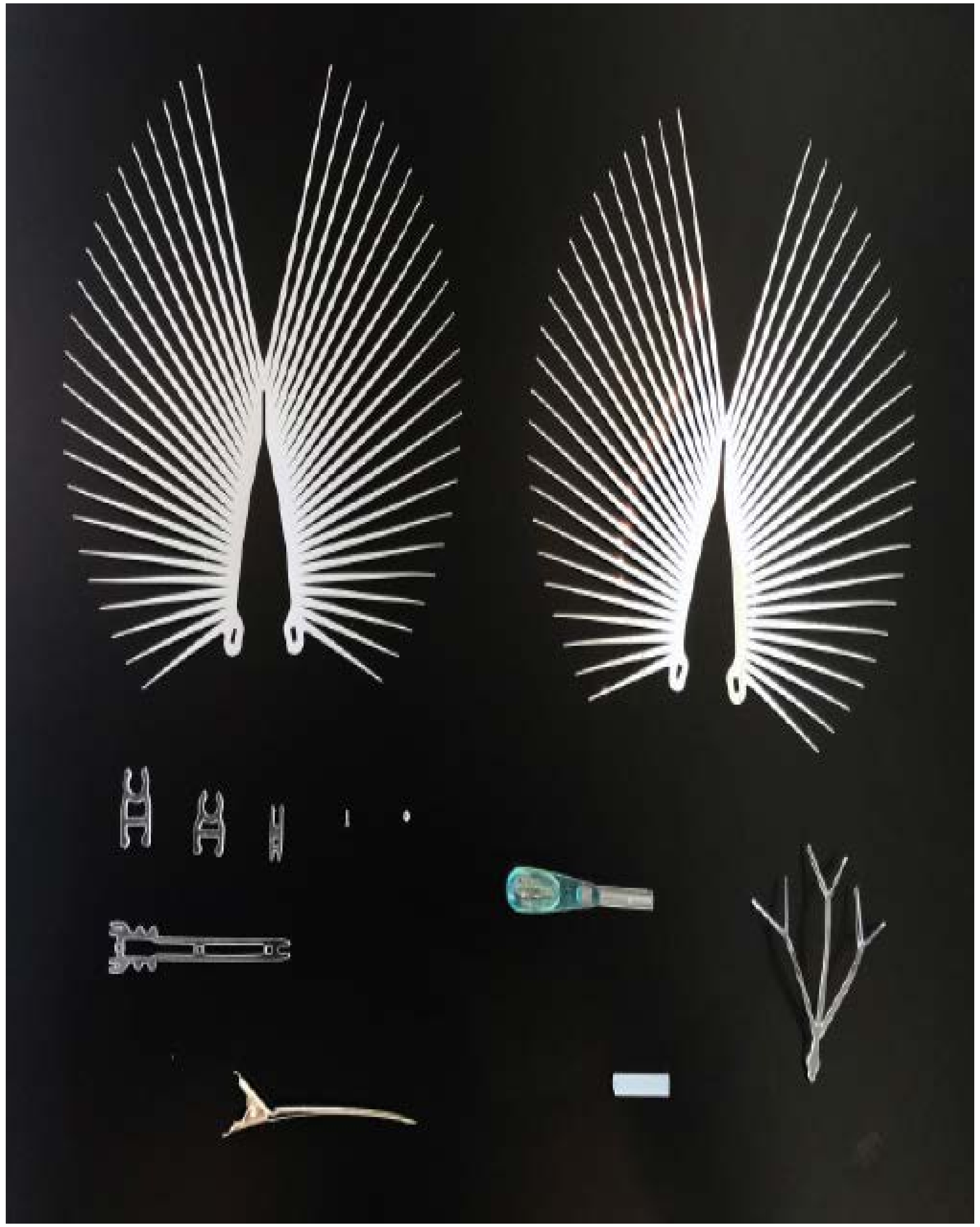
Assembled Moth



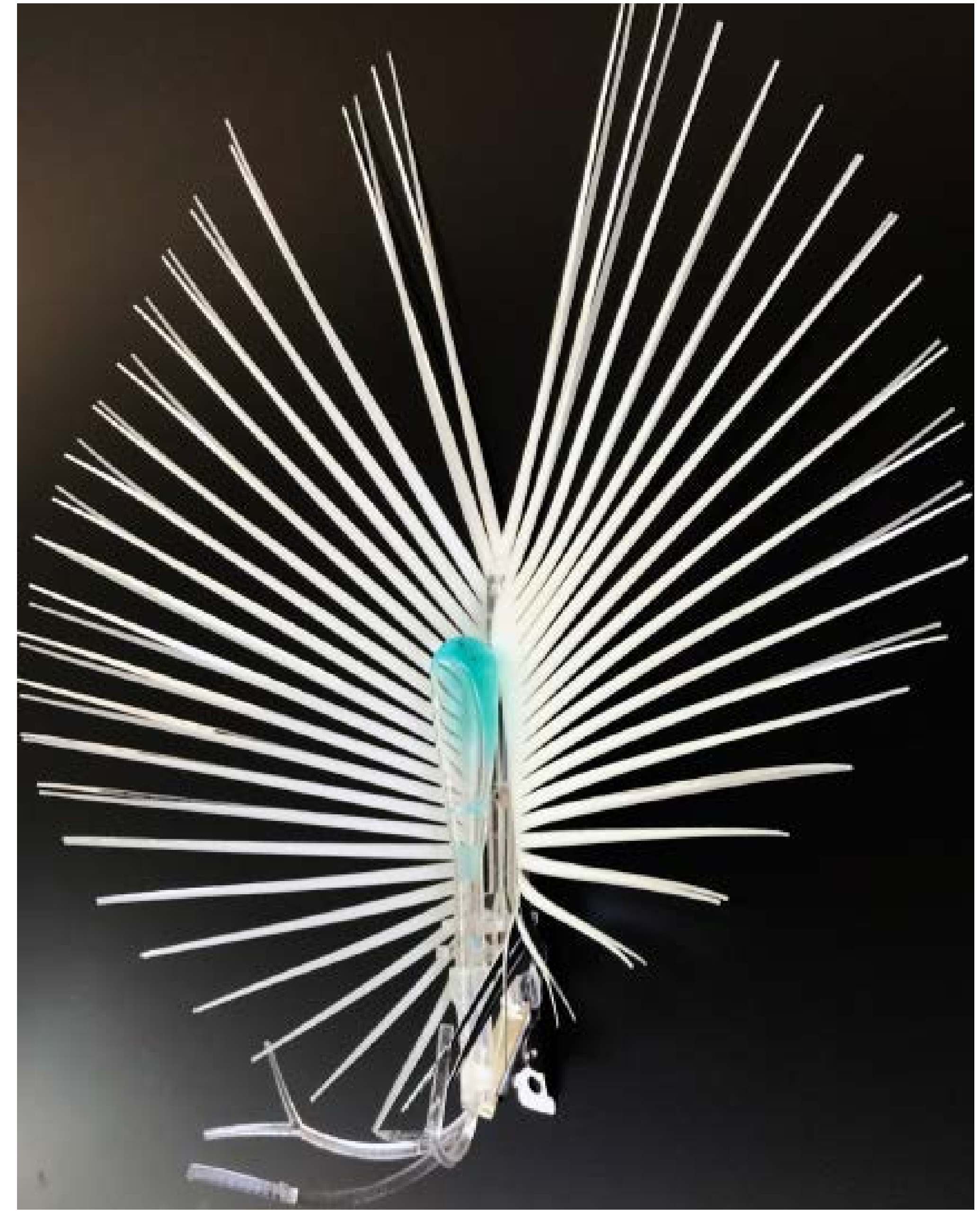
Amaria Kit Components



Assembled Amaria



Fascinator Kit Components



Assembled Fascinator

Autopoeitic Bloom

Autopoietic Bloom is a mood-based gestural improvisation engine for multichannel sentient architecture systems.

A site-specific work for *Amatria*—yet adaptable to myriad environments, installations, and configurations—*Autopoietic Bloom* comprises a suite of musical improvisational schemata and custom software that analyzes these schemata and generates music in real-time.

Autopoietic Bloom allows *Amatria* to improvise upon predetermined parametric improvisational schemata that specify general musical information—such as gestural content, algorithmic musical transformations, and timbral ideas—analogous to what a human would read and interpret in, for example, a more traditional musical score by Modernist composers such as Lutosławski, Xenakis, and Cage.[2]

2 See such works as *Jeux vénetiens* (1961), *Pithoprakta* (1955–56), and *Music of Changes* (1951), by Lutosławski, Xenakis, and Cage, respectively.

Moreover, *Autopoietic Bloom* is able to access and respond to real-world local weather data, which influences the sculpture's "mood" and, thus, how it improvises.

Written in Max, using the Bach and Cage libraries (with a soupçon of Javascript), the software is highly modular, extensible, and extremely versatile, capable of producing music both desperately chaotic and restrictively prescribed (though that would, of course, defeat the work's purpose).

Concept Design

Autopoietic Bloom was born from a deep examination of some of the central questions in the field of Living Architecture:[3]

1. Can architecture integrate living functions?

2. Could future buildings think, care?

3. How do humans respond to these evolving interactions in a process of mutual adaptation?

Taking these questions further, we began to ask:

1. Can living architecture help people create social (parasocial?) relationships with their environments? What would be the nature of those relationships?

2. How can living architecture mediate social relationships among people? What role does sound play in that mediation?

3. How can people connect with each other through a shared sound environment?

These questions served as a point of departure for the project and a set of broad phenomenological desiderata was developed.

Phenomenological Desiderata

Because the primary questions that lay at the core of *Autopoietic Bloom* are centered around the human aesthetic experience, the desirable outcomes are phenomenological in nature, as oppose to ontological. In other words, questions such as *Can a machine improvise music?* were of secondary concern. Rather, questions such as *How could a machine that seemed to be improvising make a human feel?* were of central importance.

3 Living Architecture Systems Group. "About." *Living Architecture Systems Group*. Accessed 5 April 2023. https://livingarchitecturesystems.com/about/.

These desiderata were:

1. Create a musical experience that allows people to deepen their relationship with the architecture of Luddy Hall and with each other, as mediated through their relationship with *Amatria*.

2. Create a musical experience that deepens the sense that *Amatria* is alive.

3. Situate the observer(s) and the sculpture in the same sacred time and place.

4. Has life beyond a single performance.

5. Respects and preserves *Amatria*'s artistic identity as a sculpture and quasi-living thing.

The first desideratum was centered around the idea that unique musical events constitute shared aesthetic experiences that deeply connect people together across space and time: each improvisation is a non-repeatable event, an expression of uniqueness of identity, and an opportunity for human connection. The uniqueness of the aesthetic experience implicitly connects listeners as they experience together a shared musical event that will never happen again.

The second desideratum guided the project toward musical biomimesis, inspired partly by *Amatria*'s non-musical behavior engine. *Autopoietic Bloom* is designed to give *Amatria* agency over the music she creates, and to help personify the sculpture for observers, enhancing people's ability to create social relationships with the architecture they inhabit. Herein lies the central metaphor of the work's title, *Autopoietic Bloom*: as *Amatria* gives life to music, music gives life to *Amatria* in a homeostatic loop.

The third desideratum concerns philosophical and anthropological ideas regarding how humans experience space and time when coming into contact with a work of art. These ideas go beyond the scope of this article, but they are drawn from foundational research of anthropologists Émile Durkheim (1858–1917) and Mircea Eliade (1907-1986) that examines the Sacred-Profane dichotomy,[4] as well as the composer's original research.[5]

The fourth desideratum helped ensure the longevity and artistic depth of the project.

The final desideratum grew from a desire to uphold the excellent work by artists at 4D Sound, who composed the sculpture's original sound design (OSD). Thus, the sonic structure of *Autopoietic Bloom* was conceived to be fundamentally different—both aesthetically and functionally—from the sculpture's OSD.

There are several key differences: *Autopoietic Bloom* is activated by a different set of triggers than the OSD, *Autopoietic Bloom* activates much less frequently than the OSD, and the sounds are organized differently—*Autopoietic Bloom* generates complete musical works with beginnings, middle, ends, climaxes, nadirs, etc., while the OSD is more conversational.

One could conceive the OSD to be *Amatria*'s natural speaking voice, while *Autopoietic Bloom* represents a performance, wherein *Amatria*—much like a singer or storyteller—adopts the performative role of a persona. *Amatria* puts on a kind of aesthetic mask or costume, crates and performs a musical work, and then removes the mask to return to her natural speaking voice.

Weather Data and Sympathetic Mood

In addition to the above desiderata, it was important to enable the software to respond to the overall mood of the inhabitants of Luddy Hall to help stimulate the development of positive social relationships between those inhabitants and the sculpture.

4 Durkheim, *The Elementary Forms of Religious Life*. Translated by Karen E. Fields (New York, NY: Free Press, 1995); Eliade, Myths, Dreams and Mysteries: The Encounter between Contemporary Faiths and Archaic Reality (New York, NY: Harper & Row, 1960).

5 Miggy Torres, "Unknotting Symbols from the Strands of Reality," 2018, accessed 5 April 2023, https://www.miggytorres.com/aesthetics/unknotting.

Torres, "Understanding, Harnessing, and Annihilating Cliché," What Can Choral Music Be?, accessed 5 April 2023, https://www.miggytorres.com/aesthetics/wccmb/cliche-semiotics.

Torres, "Cliché, Singularities, and Infinite Echolalia," What Can Choral Music Be?, accessed 5 April 2023, https://www.miggytorres.com/aesthetics/wccmb/cliche-echolalia.

Torres, "The Death of Cliché." What Can Choral Music Be?, accessed 5 April 2023, https://www.miggytorres.com/aesthetics/wccmb/cliche-the-death-of.

This posed a challenging problem: how would the sculpture be able to sense the mood of the population? Where would this data come from? How would it be interpreted?

Gathering direct biometric data was out of the question—both due to privacy and cybersecurity concerns, as well as the negative psychological impact it would likely have on a population for whom the specter of Big Brother still hangs thick in the zeitgeist.

Improvisation Engine Technical Overview

Instead, a kind of mood-proxy was used: local weather data. While the weather may seem banal—something that we experience every day without giving it too much thought, excepting extreme circumstances—the weather, in fact, has a large psychological impact on people's mood.[6]

Thus, it was decided that *Autopoietic Bloom* would use local weather data as the basis for what would become *Amatria*'s musical improvisations—connecting meaningfully, yet indirectly, with the inhabitants of her environs.

The improvisation engine that lies at the core of *Autopoietic Bloom* is composed of four primary modules:

- *Donatoni* – Gesture-based Note Generator
- *Syrinx* – Multichannel Synthesizer
- *Quincy* – Dynamic Channel Mixer
- *Zoë* – Interpolation Engine

Donatoni – Gesure-based Note Generator

Donatoni—named for Italian Modernist composer Franco Donatoni (1927-2000)—is where the rhythm and pitch-based improvisation occurs. The module uses the *bach library* for Max[7] —developed by Andrea Agostini and Daniele Ghisi—to represent musical structures as Lisp-like Linked Lists (LLLLs) and apply algorithmic transformations to those LLLLs in real time.

6 Perhaps the most striking example of this is Seasonal Affective Disorder (SAD). also known as Seasonal Depression.

7 Andrea Agostini, Daniele Ghisi, "A Max Library for Musical Notation and Computer-Aided Composition," *Computer Music Journal* 39, no. 2 (Summer 2015): 11–27, DOI: https://doi.org/10.1162/COMJ_a_00296.

These musical representations are as yet devoid of actual sound; they simply describe the *idea* of music.

Donatoni supports up to four separate voices capable of improvising independently from one another. Each voice supports an *ur-gesture* or background gesture—a small musical motif upon which all the improvisation is to take place—and up to eleven distinct musical transformations:

1. Pitch Transposition (within any arbitrary musical mode)
2. Pitch Inversion
3. Pitch Retrogression
4. Pitch Rotation
5. Pitch Scrambling
6. Rhythmic Augmentation or Diminution
7. Rhythmic Retrogression
8. Rhythmic Rotation
9. Coupled Pitch & Rhythmic Rotation
10. Rhythm Scrambling
11. Bi-directional Truncation

These transformations are analogous—if not identical—to the same kinds of operations performed by human composers from Bach to Berio when developing musical ideas with nothing but a pencil and paper. A central goal in designing these functions was to enable the software to *think like a composer*. Thus, while the transformations are mathematical in nature, they are conceived first as musical processes that are then translated into mathematical representations, as opposed to existing first as mathematical ideas that were shoehorned into a musical context.

Each of the eleven transformation modules is ensconced within an associated phrase generator that generates values for these transformation functions, either according to an explicitly defined pattern or, more often, according to a subjective variable called Improvisational Flux.

Improvisational Flux

All of *Autopoietic Bloom*'s improvisable parameters are coupled with a subjective variable—Improvisational Flux—that broadly defines two parameters about the behavior of the associated parameter's phrase generator:

1. How often will a given parameter change? (Frequency)

2 How much will a given parameter change? (Amplitude)

Every phrase generator in *Donatoni* interprets this variable differently for each transformation module. Through these processes of interpretation—amid various layers of abstraction—coherent, life-like improvised musical structures begin to emerge.

These musical structures are generated by *Donatoni* during performance in real-time, converted into MIDI data, and immediately sent to *Syrinx*.

Syrinx – Multi-channel Synthesizer

Syrinx—named for the organ in some species of birds that allows them to sing multiple pitches at once—consists of a bank of four semi-modular synthesizers, one for each *Donatoni* voice. Each synth converts the abstract musical ideas expressed by *Donatoni* into a digital audio signal.

The backbone of *Syrinx* is *Absynth 5*, a semi-modular synthesizer by Native Instruments, here implemented as a VST plugin. *Absynth 5* forms the timbral core of *Autopoietic Bloom*, wherein standard processes for sound synthesis are available (e.g., FM/AM synthesis, modal synthesis, ADSR, LFOs, filtering, etc.). Sampled sounds can also be used and transformed in similar ways, allowing for virtually limitless coloristic possibilities.

In addition, *Syrinx* supports input for up to four control signals per voice, so sounds can be shaped by *Amatria* over the course of an improvisation.

In practice, sounds for each improvisational scheme are carefully crafted in advance by the composer and assigned to a library of presets that *Syrinx* can call instantaneously before or during a performance. This allows *Syrinx* to change instruments on the fly with minimal data throughput.

With abstract music now converted into a digital audio signal, that information is sent to the *Quincy* module.

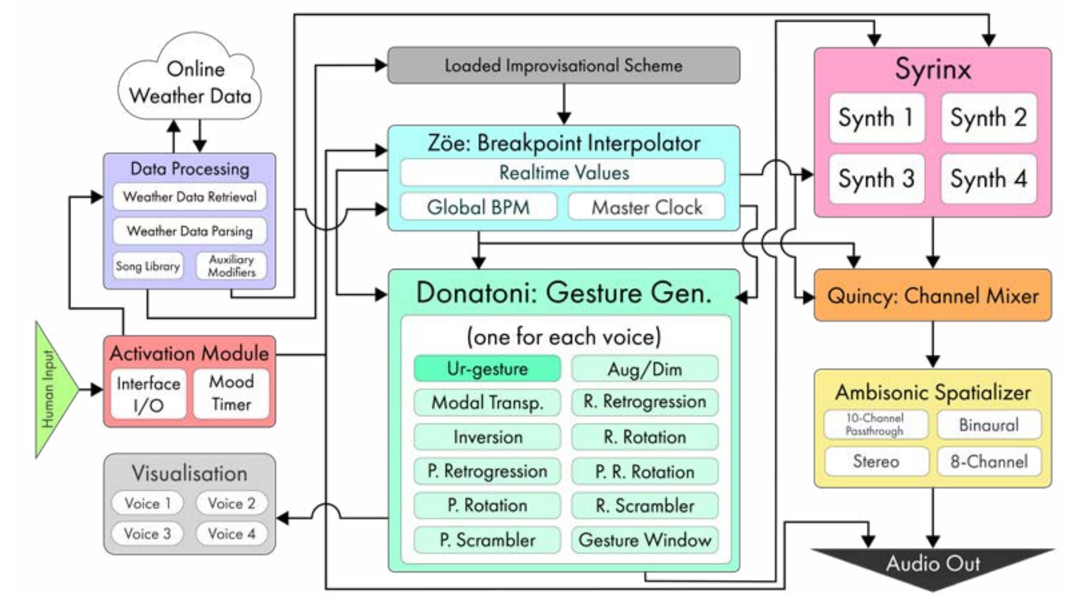
Quincy – Dynamic Mixer

Quincy—named for American record producer Quincy Jones (b. 1933)—dynamically distributes the four channels of digital audio produced by *Syrinx*'s four voices to *Amatria*'s ten speakers. These voice-speaker mappings are able to change seamlessly over the course of a piece, shifting algorithmically using similar modules to those found at the core of the *Donatoni* note generator. Thus, through *Quincy*, *Autopoietic Bloom* not only enables *Amatria* to improvise upon the rhythmic, pitch, and timbral components of a work, but upon the spatial configuration of the sound as well.

Zoë – The Interpolation Engine

Donatoni, *Syrinx*, and *Quincy* work together to enable *Autopoietic Bloom* to create immersive, kaleidoscopic sound-worlds, but these three components alone would generate sounds-worlds that remain relatively static. To create compelling *music*—to turn soundscape into narrative, stasis into drama, recitative into aria—*Autopoietic Bloom* requires a fourth component to inject dramatic trajectory, transformation, and teleology into its works.

Enter *Zoë*—named after the Greek word for life— *Autopoietic Bloom*'s interpolation engine. *Zoë* allows all 235 parameters of an improvisation—either defined explicitly as individual values or implicitly as Improvisational Flux—to vary over the course of the piece, thereby giving each song its own individualized shape. Through *Autopoietic Bloom*'s *Zoë* module, *Amatria* can improvise music that has climaxes, beginnings, middles, ends,



Above

Diagram of *Autopoietic Bloom*'s Information Flow

narratives, nadirs, accelerandi, crescendi, and more. Thus, each unique song is imbued with artistic intent, motivation, growth, development, metamorphosis, and *autopoiesis*.

Within an improvisational scheme, breakpoints are defined architecturally, specifying values for various parameters at major time-points across a piece. *Zoë* is able to read these improvisational schemes and use that information to fill in the gaps between breakpoints in an intuitive way. The number of possible points between which *Zoë* can interpolate any parameter is virtually endless.

Improvisational Schemata

Improvisational Schemata are composed in an idiosyncratic syntax and formatted as a JSON structured data file. Upon activation, *Autopoietic Bloom* reads live weather data for Bloomington, Indiana and uses the data to select an improvisational scheme from its internal library. This selected scheme is then loaded into memory and read by *Zoë*, which then sends interpolated data throughout the various modules of *Autopoietic Bloom*'s improvisation engine.

Above

Mood Oracle's touch-screen interface for *Amatria*, showing which of *Autopoietic Bloom*'s four voices are playing at which speaker in the sculpture

Virtualizing *Autopoietic Bloom*

Mood Oracle – Visualization System

Autopoietic Bloom's auditory and spatial experience is enhanced visually by *Mood Oracle*, a visualization system for *Autopoietic Bloom*—created specifically for its implementation with *Amatria*—installed at the *Amatria* touch-screen kiosk under the sculpture in Luddy Hall. *Mood Oracle* will display the what is being sounded from each of *Autopoietic Bloom*'s four voices using standard musical notation. Each voice color-coded in red (1), yellow (2), green (3), and blue (4).

A top-view graphic of *Amatria*'s speaker configuration is also displayed. As different voices emerge from different speakers, the graphic changes color to reflect which how the voices are mapped onto the sculpture's audio outputs.

Ambison – Ambisonic Spatializer

Autopoietic Bloom is a site-specific work, created for the *Amatria* sentient sculpture. However, the software has two broad modes of audio output: 10-channel Passthrough and Ambisonic.

In its 10-channel mode, *Autopoietic Bloom* simply sends raw audio data to the ten channels corresponding to *Amatria*'s ten speakers.

In ambisonic mode, however, audio data is sent into a virtual sound-model of *Amatria*—containing information about the sculpture's speaker placement and acoustic environment. This model can then be projected onto an arbitrary number of speakers, arranged in any arbitrary configuration to produce an acoustic experience very similar to a live, in-person viewing of *Amatria* and *Autopoietic Bloom*.

Ambison—Autopioetic Bloom's ambisonic spatializer—leverages Spat5—an ambisonic spatialization suite developed by IRCAM[8]—to enable the software's adaptability to myriad audio configurations and environments.

In tandem with *Mood Oracle*, *Ambison* has allowed listeners to experience a binaural version of *Autopoietic Bloom* remotely via Zoom, as well as in a quadraphonic format at arts festivals across the United States, away from the physical sculpture itself.

8 Thibaut Carpentier, "A new implementation of Spat in Max." In *Proc. of the 15th Sound & Music Computing Conference (SMC)*, Limassol, Cyprus, (July 2018): 184–191, DOI: https://doi.org/10.5281/zenodo.1422552.

Learn More

More information on *Autopoietic Bloom*—including technical documentation, recorded talks, and a demonstration of the software—and on composer Miggy Torres can be found on the composer's website at https://miggytorres.com/projects/AutopoieticBloom

Amatria's Behaviour

System behaviour is dictated simultaneously by the ML and Mac. The ML is responsible for global management of system states and coordination of the system network, and responds to sensor triggers with actuation patterns. The Mac outputs all sound and determines background actuation. AU accesses and visualizes a sensor data stream coming from IS.

General behaviour patterns may include:

i. Sensor Triggers

Individual sensors may be given a sensor value threshold. A NC controller will detect when any of its connected sensors have passed their threshold.

ii. Sensor Sampling

A NC samples the sensor values of connected sensors at a set interval and sends the information up the hierarchy.

iii. Reflexive Response

The actuators surrounding a sensor respond immediately to a sensor trigger.

iv. Neighbour Response

Groups of actuators respond to the reflex response of neighbouring groups.

v. Global Response

The system reacts globally to a significant level of sensor-detected activity.

vi. Background Behaviour

Ambient behaviour that continues without sensor-detected activity.

Behaviour Patterns Summary

The control of this installation is organized into different groups of output Rebel Star LED sequences, moth vibration using DC motors and audio track play through speakers. Each of the RS LED/moth sequence behaviours is defined using the terminology “Shot” and “Wave”. Each Shot initiates a chain of RS LED/moth activations. Each Wave defines a distance-based change to a specific Shot sequence, propagating out through groups of actuators adjacent to the location of the original IR sensor stimulus.

Sensor Response Behaviour

Sensor Response Behaviour is the primary behaviour in the sculpture. Rebel Star LED and Moth groups located within the Sound Sensor Scout (SSS) are triggered when the infrared distance sensor at the bottom of the SSS detects a change in its field. When the system detects this stimulus, the outputs perform a “Shot” sequence along with a “Wave” sequence on its neighbours. This “Wave” consists of a chain of actuates, progressively modified by delaying and slowing down each successive group, as well as reducing the number and brightness of used outputs on each group.

A Double Rebel Star spar and Rebel Star LEDs in the SSS itself will have the similar “Wave” sequence behaviour once it is triggered by the sensor in the corresponding SSS, which is fading-in-out through the whole spar field. Moth vibration behaviour is also triggered on a given Sound Sensor Scout point when the infrared distance sensor at the bottom detects a change of the pre-set detection value. Depending on the direction of the detected move, the moth pairs in Sound Sensor Scout and sphere units will follow that direction and start to vibrate.

The behaviour of track playing through the speakers is triggered when the Mic sound sensor in the Sound Sensor Scout point detects the volume of the ambient sound goes over the preset range.

Background Behaviour

This behaviour happens after a random amount of time has passed from the last sensor event anywhere in each Sound Sensor Scout. Once the set amount of time has passed, a random double RS group is selected to execute the "Shot" behaviour followed by a "Wave" behaviour across all the neighbouring groups. Once a sensor has been triggered in any of the six Sound Sensor Scouts in the sculpture, this behaviour will be disabled.

Evening Mode

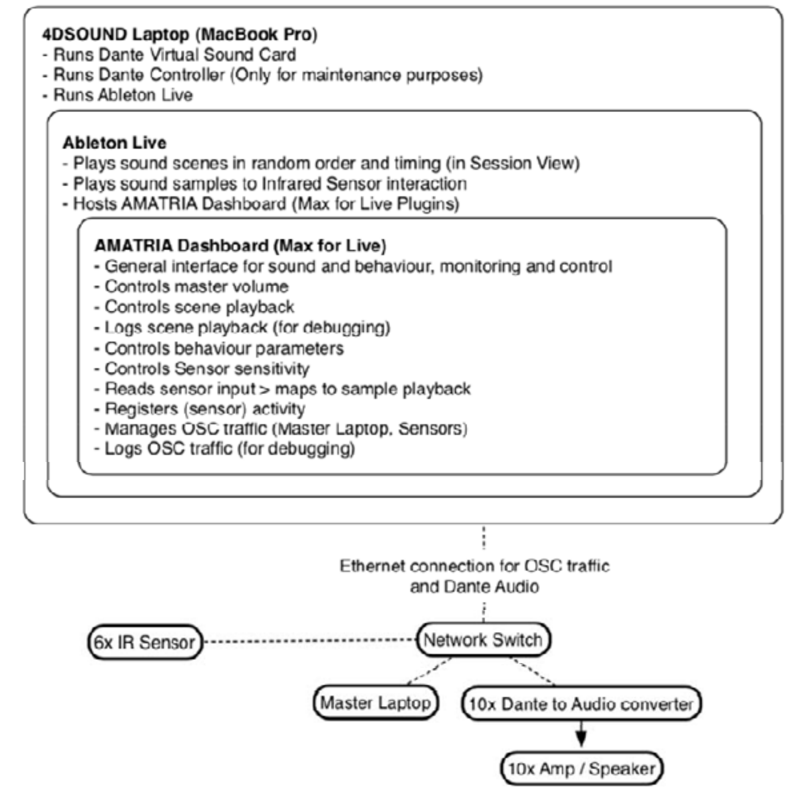
The installation is configured to move into Evening Mode if there has been no sensor activity for a customizable period of time (e.g. after the building is closed at night and no one is moving below the artwork). If a sensor is triggered at any point during evening mode, the installation will return to normal operating conditions. During Evening Mode, the background behaviour is disabled for all Nodes.

Sleep Mode

The installation will move into dormant mode if there has been no sensor activity for a customizable amount of time since switching into evening mode. If a sensor is triggered at any point during such mode, the installation will return to normal operating conditions.

Launch Mode

This mode is designed for demonstration purposes. It shows strong behaviour intensity.



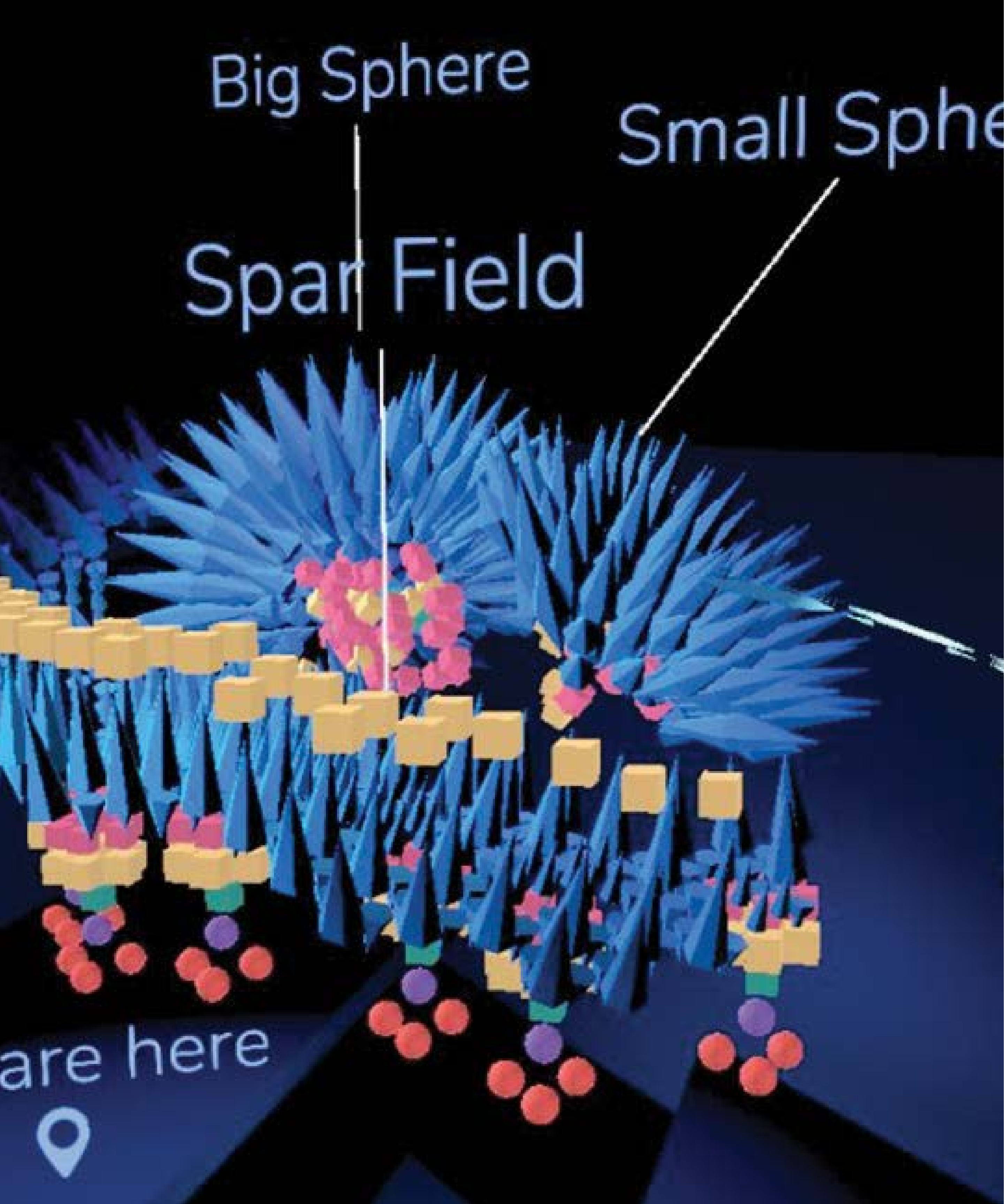
Above

Diagram of *Autopoietic Bloom*'s Information Flow

4D Sound System

Another key subsystem of the audio system is the 4DSOUND system, which receives IR trigger information from the master laptop, and sends signals to the customized TSU (Teensy 3.6 sound) board to play sounds.

The master laptop communicates with a 4D Macbook through OSC communication with a pre-set format of message protocols. The message contains the location of IR sensors that have been triggered. The master laptop sends such IR information to the 4D Macbook and the 4D Macbook will send back the location of actuators (moth, Rebel Star LED) that need to be triggered by Teensy. At the same time, 4D Macbook sends audio tracks via PoE network switch and Ethernet cables to the Teensy Audio Board to control the speakers to play the sounds.



Envisioning Intelligent Interactive Systems

Data Visualizations for Sentient Architecture

Katy Börner & Andreas Bueckle
Indiana University, Bloomington, USA

Cover Image Tavola, scene 1: *Amatria* overview.

This paper presents data visualizations of an intelligent environment that were designed to serve the needs of two stakeholder groups: visitors wanting to understand how that environment operates, and developers interested in optimizing it. The visualizations presented here were designed for *Amatria*, a sentient sculpture built by the Living Architecture Systems Group (LASG) at Indiana University Bloomington, IN, USA, in the spring of 2018. They are the result of an extended collaboration between LASG and the Cyberinfrastructure for Network Science Center (CNS) at Indiana University. We introduce *Amatria*, review related work on the visualization of smart environments and sentient architectures, and explain how the Data Visualization Literacy Framework (DVL-FW) can be used to develop visualizations of intelligent interactive systems (IIS) for these two stakeholder groups.

Image 1 ***Amatria*** Philip Beesley (Bloomington, Indiana, 2018).

1. Introduction

Increasingly, our everyday environments are becoming smarter and more connected. IIS is an umbrella term to describe environments that can process data and generate responsive behaviors using sensors, actuators, and microprocessors in order to bridge the gap between humans and machines (1, 2). Sentient architecture implements IIS through artful, imaginative, and engaging architectural artifacts with which we can experiment and observe human behavior and capabilities when confronted with IIS (3, 4). Due to the transformation of the physical environment achieved through scaffolds, sensors, actuators, and microprocessors, new experiences are possible for humans inhabiting these spaces.

Amatria is one such sentient architecture piece on display in Luddy Hall at Indiana University.[1] Foreshadowing a future where the Internet of Things (IoT) is omnipresent, *Amatria* stands as an artful interpretation of a data-driven environment adapting to what it senses about its physical surroundings. She demonstrates how IoT technologies can affect humans—be it with amazement, unease, mystery, or curiosity. Luddy Hall is a 124,000-square-foot public school building, visited by hundreds of students, faculty, staff, and external guests every day for work, study, talks, and conferences.[2] Conspicuously placed on the fourth floor of Luddy Hall overlooking the atrium, *Amatria* is not simply a piece of art seen by hundreds of eyes every day; more significantly, *Amatria* is embedded into the lives of students, faculty, and staff who strive to develop self-driving cars, explore the social aspects of machine learning, or engineer next-generation robots. In short, *Amatria* embodies a future rich with data where machines and humans live and work together.

1 "Amatria: Sentient Architecture," Cyberinfrastucture for Network Science Center, https://cns.iu.edu/amatria

2 "Luddy Hall," Indiana University Bloomington School of Informatics, Computing, and Engineering, https://www.sice.indiana.edu/about/luddy-hall.html

The presence of *Amatria* raises the question of how her human cohabitants perceive and understand her. *Amatria* tours have introduced her to elementary children, retirees, invited speakers, advisory council members, wealthy donors, and the president of IU. In order to enrich the experience of each of these stakeholder groups, it is essential to understand their insight needs, meaning what information these groups need to enhance and/or satisfy their interest in the sculpture.

This paper presents research and development efforts on data visualizations that facilitate the communication and exploration of IIS in the form of sentient architecture such as *Amatria*. First, we will introduce related work on the visualization of IIS, and on sentient architecture. Second, we will explain how the Data Visualization Literacy Framework (DVL-FW) can be used to develop visualizations of IIS for two stakeholder groups: visitors and developers. Specifically, we will introduce the Tavola (Italian for "table") 3D visualization for visitors along with two graph-based visualizations for developers, plus informal user study results that informed the current version of Tavola. Finally, we will discuss lessons learned and planned future work.

2. Related Work

2.1 Visualization for IIS

Previous research on data visualization for IIS is diverse and broad, but a common goal across many lines of research is to provide insights into the observed system, necessitating visualization as an integral part of the system architecture. Common challenges identified in the literature are: system infrastructure, data acquisition, and end-user usability.

Roalter et al.[3] implemented a "Cognitive Office" using the Robot Operating System (ROS)[4] as middleware between a physical office space and a virtual model of it, allowing users to view temperature, humidity, and other data in a 3D model in real time. On the other hand, Jeong et al.[5] opted for a web-based IoT framework and WebGL[6] to allow users to control a virtual home.

3 Roalter, Luis, Matthias Kranz and Andreas Möller. "A Middleware for Intelligent Environments and the Internet of Things." International Conference on Ubiquitous Intelligence and Computing, (Berlin/Heidelberg, Germany: Springer, 2010), 267 - 281.

4 ROS http://www.ros.org

5 Jeong, Yuna, Hyuntae Joo, Gyeonghwan Hong, Dongkun Shin and Sungkil Lee. "AVIoT: Web-based interactive authoring and visualization of indoor internet of things." IEEE Transactions on Consumer Electronics 61.3 (2015): 295 - 301.

6 "WebGL: 2D and 3D graphics for the web," MDN web docs, https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API

While the former developed their setup alongside a physical space, the latter simulated input and output of their system, i.e., sensation and actuation, in code only. Furthermore, they presented a two-way communication solution where an end user can both monitor the system and send commands to it. In a user study with 24 participants (14 of whom had some experience in programming), they compared two modes of authoring actions: text-based and visual. They found that user satisfaction, user understanding, user efficiency, and user preference are significantly greater for the visual programming mode. In the area of health management, Hassanalieragh et al.[7] surveyed health monitoring frameworks, particularly noting the need for comprehensive, integrated solutions as a result of the increasing availability of wearable sensors and their data. Similarly to Jeong et al.[8], they emphasize that visualization is needed for end-users (physicians, in this case) to gain actionable insights from heart rate and blood pressure data. Belimpasakis et al.[9] propose a framework for home entertainment using augmented reality, employing cell phone cameras as an interface through which a user can have a two-way communication with an IIS. SensMap[10] features an outdoor view, an indoor view, and a topological view of sensors in a building. This allows the user to view sensor location and data, and link quality to other sensors on a Google Maps basemap in different levels of detail to facilitate data interpretation. As opposed to Raolter et al.[11] and Jeong et al.[12], SensMap uses 2D visualizations only.

Existing visualizations of IIS data help a broad range of stakeholders understand data streams. What's needed now, however, is a method for systematically designing such visualizations. In this paper, we aim to provide guidelines for designing effective data visualizations that empower different stakeholder groups to explore and optimize IIS.

2.2 Sentient Architecture

Sentient architecture is a type of interactive art. Various approaches from different disciplines exist, including architecture, robotics, and data visualization. In their *Hylozoic Ground* series, Beesley et al.[13] investigate if and how architecture can be perceived as alive by visitors. Building on this, and interested in creating such life-like behaviors, Chan et al.[14] detail the Curiosity-Based Learning Algorithm (CBLA) that implements Intelligent Adaptive Curiosity, as described by Oudeyer et al.[15] As the purpose of CBLA is to increase a system's behavioral adaptivity through self-experimentation, the team presented a prototype interactive sculpture featuring an implementation of CBLA using Teensy microcontrollers with ambient light sensors, IR sensors, and accelerometers alongside LED lights and kinetic actuators. Subsequently, Chan et al.[16] tested interactions between human subjects and the testbed. In a user study with 10 participants, they created two test conditions: one with pre-scripted behaviors for the testbed, the other one with CBLA running, switching between modes halfway through the experiment. Performing a quantitative analysis, they found no significant difference in self-reported interest score by the participants between the conditions and noticed only a weak correlation between average activation value and participant interest level. Elsewhere, Börner et al.[17] presented *Lifting the Veil*, a 3D visualization of the *Sentient Veil* sculpture in the Isabella Stewart Gardner Museum in Boston, MA, USA. The purpose of the work was to apply data visualization to sentient architecture in order to satisfy visitor insight needs and help them identify the structure, dynamics, and state of a sentient architecture piece. This work also informed the research and development in this paper.

We consider sentient architecture an instantiation of IIS where two stakeholders exist: visitors who consume (and hopefully enjoy) the art piece, and developers (usually engineers or architects) who build and maintain it. In the following chapter, we will present the various data visualization types implemented for *Amatria*, and we will detail how theory from the DVL-FW was applied during every step of development for each stakeholder group.

3. Sentient Architecture Visualizations

The development of Tavola, a 3D tool for visualizing *Amatria's* structure and data flow (see *Images 4 - 5*), was guided by a visualization framework, which we will detail in the following section. But the story of Tavola's creation is also the story of close interdisciplinary collaboration between architects, engineers, software developers, designers, and visualization experts, highlighting the diverse skill requirements needed for data visualization in IIS and in general.

Visualizations usually support two different paradigms of functionality: communication and exploration. Communication refers to instances where visualizations convey insights from data upon which a user can act in an informed manner (such as polls on the news or graphs in a textbook). Exploration, on the other hand, refers to an iterative process where insights from one

7 Hassanalieragh, Moeen, Alex Page, Tolga Soyata, Gaurav Sharma, Mehmet Aktas, Gonzalo Mateos, Burak Kantarci and Silvana Andreescu. "Health Monitoring and Management Using Internet-of-Things (IoT) Sensing with Cloud-Based Processing: Opportunities and Challenges." 2015 IEEE International Conference on Services Computing, (Washington, DC: IEEE Computing Society, 2015), 285 - 292.

8 Jeong et al, "AVIoT: Web-based interactive authoring and visualization of indoor internet of things."

9 Belimpasakis, Petros and Rod Walsh. "A Combined Mixed Reality and Networked Home Approach to Improving User Interaction with Consumer Electronics." IEEE Transactions on Consumer Electronics 57.1 (2011):139 - 144.

10 Simek, Milan, Lubomir Mraz and Kimio Oguchi. "SensMap: Web Framework for Complex Visualization of Indoor and Outdoor Sensing Systems" International Conference on Indoor Positioning and Indoor Navigation (Washington, DC: IEEE Computing Society, 2013), 1-5.

11 Roalter et al. "A Middleware for Intelligent Environments and the Internet of Things."

12 Jeong et al, "AVIoT: Web-based interactive authoring and visualization of indoor internet of things."

13 Beesley, Philip, Rob Gorbet, Pernilla Ohrstedt and Hayley Isaacs. Hylozoic Ground: Liminal Responsive Architecture (Toronto: Riverside Architectural Press, 2010).

14 Chan, Matthew TK, Rob Gorbet, Philip Beesley and Dana Kulić. "Curiosity-Based Learning Algorithm for Distributed Interactive Sculptural Systems." International Conference on Intelligent Robots and Systems (Washington, DC: IEEE Computing Society, 2015), 3435 - 3441.

15 Oudeyer, Pierre-Yves, Frédéric Kaplan, and Verena V. Hafner. "Intrinsic Motivation Systems for Autonomous Mental Development." IEEE Transactions on Evolutionary Computation 11.2(2007): 265 - 286.

16 Chan, Matthew TK, Rob Gorbet, Philip Beesley and Dana Kulić. (2016) "Interacting with Curious Agents: User experience with Interactive Sculptural Systems." IEEE International Symposium on Robot and Human Interactive Communication (Washington, DC: IEEE Computing Society, 2016), 151-158.

17 Börner, Katy, Andreas Bueckle, Philip Beesley and Matthew Spremulli. "Lifting the Veil: Visualizing Sentient Architecture." Future Technologies Conference (Vancouver, Canada: 2017). acc Jan 22, 2019 https://cns.iu.edu/docs/publications/2017-Lifting-the-Veil.pdf.

visualization lead to questions in need of more insights. Exploratory data analysis, for example, is the process of descending into a dataset by means of multiple visualizations to uncover new questions. Communication is certainly a function of Tavola, particularly in its ability to reveal the hidden structure of *Amatria* to the untrained eye. But even more significant is Tavola's ability to encourage *Amatria*'s visitors to explore the sculpture more deeply. Tavola lifts the seemingly impenetrable veil of technology for untrained visitors, drawing them into the experience of interacting with a sentient sculpture made out of acrylic, sensors, and microprocessors, forming a coherent and highly technical art piece. In summary, the main goal of Tavola is to encourage visitors to leave their comfort zone and peek ever so slightly behind the curtain of high-tech.

3.1 DVL-FW – Typology and Workflow Process

Our development of sentient architecture visualizations followed the Data Visualization Literacy Framework (DVL-FW) introduced in the *Atlas of Knowledge*[18] and detailed in a forthcoming paper. The DVL-FW is based on an extensive review of literature from more than 600 publications documenting 50-plus years of work by statisticians, cartographers, cognitive scientists, visualization experts, and others. The DVL-FW process model is shown in Image 2.

The visualization process starts with investigating the insight needs (sometimes also called "task types") of one or multiple stakeholders. Insight needs dictate which data is to be acquired, what analyses need to be made, and what visualization types are to be chosen. Understanding one's stakeholders is essential to creating a visualization with actionable insights. Operationalization is needed to guide both data acquisition and the requirement for data cleaning and refinement that comes with it. Analysis then allows the developer of a visualization to reveal patterns and trends in the data, and to refine insight needs and associated questions to the data. Only after these steps are completed can the actual visualization process start. The developer chooses one or multiple visualization types, such as table, chart, graph, map, or network.

Deployment refers to the act of exporting a visualization to a particular medium, such as a piece of paper, a touch screen, a 33-million-pixel video wall, or a virtual-reality headset. Deciding which method of deployment to use is often done early on in the process as some media offer affordances

18 Börner, Katy. Atlas of Knowledge: Anyone can Map (Cambridge: MIT Press, 2015).

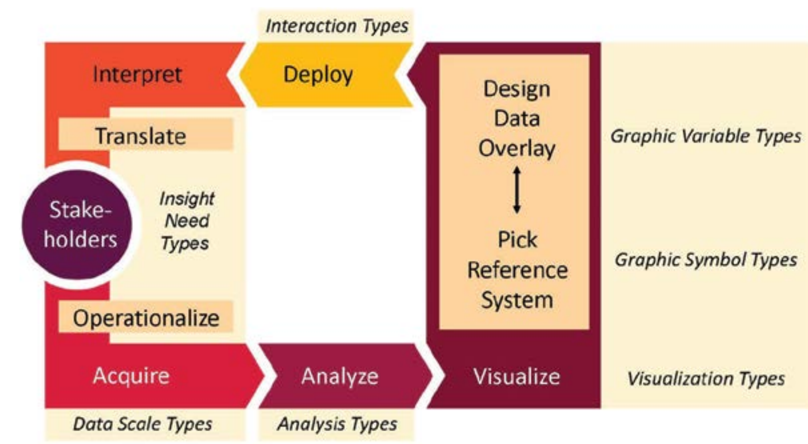


Image 2 Data Visualization Literacy Framework (DVL-FW) process model.

more suitable to satisfy user insight needs. For example, visualizations deployed to paper are usually static, fit within a standard book page, and feature annotation in close proximity. Touch screens or keyboard-mouse interfaces, on the other hand, allow users to manipulate their view of the data, or even the data itself. Interactions with data visualizations are only possible if an input device is available and the visualization can receive input in the first place. Interpretation, finally, requires stakeholders to assess what insights can be gained from the visualization with regards to the originally identified insight needs. Very often, results of this interpretation are communicated with a write-up or in annotations to the visualization. After the interpretation is completed, the workflow begins again.

Due to its iterative nature, the DVL-FW process model captures repetition and refinement of insight needs and data visualization not only for the whole series of workflow process steps but also for cycles between just a subset. For example, developers might acquire data based on a perceived insight need, analyze it, and realize that the dataset is not a good fit to provide answers to the stakeholders' questions—in which case they will likely ask for different data. Similarly, developers might deploy a dynamic visualization and realize that it could be enhanced by adding a second, static visualization. They would then revert to the "Analyze" step to prepare a new chart, graph, map, etc.

3.2 Stakeholders and Insight Needs

While the different stakeholders are quite diverse, we can identify two main groups for which we discuss visualization work here: visitors and developers. Visitors are individuals either unfamiliar with sentient architecture in general, or *Amatria* in particular, who come to the sculpture either on purpose (such as for a tour) or by chance. Developers are engineers or architects who are interested in how to debug or improve a sentient architecture sculpture.

As outlined in the DVL-FW process model, identifying the rather different insight needs for both types is an essential first step in the DVL-FW process model. While our visualization tool Tavola is primarily geared towards visitors (see section 3.4.1), the data collected in the development process can also be used by developers for debugging purposes (see section 3.4.2).

In order to provide value for visitors, Tavola aims to empower them to explore physical and imaginary spaces. This is called *geospatial* exploration, a superset of various insight needs usually satisfied with maps. Tavola presents geospatial information from *Amatria* in a 3D map layout. Maps are among the oldest, most established, and most readable types of visualizations, and Tavola enables visitors to explore *Amatria* as a physical space. Another insight need addressed by Tavola is *comparison*, which we achieve by adding a real-time element that shows the consequences of a visitor's interaction on *Amatria*. In this iteration, Tavola separates these two insight needs and presents them in different scenes. Finally, Tavola can also be used to identify *trends* over time, albeit in small units.

Developers, as the second stakeholder group, need to make sure each sensor and actuator functions properly by *comparing* expected and observed values and behavior. To that end, we developed a real-time bar graph visualization (see section 3.4.2, *Image 9*) for on-site comparison of IR sensor values. However, developers are also interested in monitoring behavior *trends over time* in order to find bugs, optimize code, and create improvements based on observations. To this end, we used the same real-time data to develop a graph showing IR sensor values over a 24-hour period (see *Images 7 – 8*). As the preceding discussion illustrates, it is quite common in the process of data visualization to find that different stakeholder groups are best served by different views of the same data.

3.3 Data: *Amatria* Setup, Software, Networking

In the previous subsection, we investigated insight needs for visitors and developers. This subsection discusses the data acquisition required to implement visualizations for both stakeholder groups.

To satisfy visitor *insight need geospatial*, a highly detailed parametric model of *Amatria* was shared with the Tavola team. While the original *Amatria* 3D model featured hundreds of thousands of individual pieces, the simplified model was reduced to primitive shapes and 3D objects to represent only the scaffold of *Amatria*, significantly decreasing the performance requirements for whatever target device Tavola would run on. Since *Amatria* took several weeks to install, and since the venue was on the same floor as the offices of the Tavola team, hours of close cooperation and informal exchange were possible. The *Amatria* team helped us understand and prepare for visualization both the model and the real-time data stream (see *Image 3*, right); additionally, close onsite collaboration during the installation process allowed the Tavola team to gain valuable insights into how *Amatria* functions before it was even finished. Tavola uses the *Amatria* model in two capacities: as a whole, embedded in an additional 3D model of Luddy Hall (also developed by the *Amatria* team based on blueprints from the architects of the venue), and as more high-resolution subset of the model with only parts of the sculpture visible. Planes with opaque material were inserted to replace the transparent walls in the Luddy Hall model, and a highly eclectic, ambient deep-blue light was added to set the building into the background and focus the user's attention on the sculpture (see *Images 4 – 5*).

In order to satisfy *insight need comparison* for both visitors and developers, state information about parts of the sculpture needed to be made available for visualization. IR sensor data was chosen to be streamed from *Amatria* to Tavola by means of universal datagram protocol (UDP). The master laptop (see *Image 3*) sends commands, retrieves sensor information, and transmits data values to a set of specially assigned IP addresses on the *Amatria* virtual local area network (VLAN). A device running Tavola will get one of these IP addresses upon signing into the network, then receive and parse the data stream. Note that in the current implementation of Tavola, no data is shared about other hardware such as the Raspberry Pis[19] and Teensy node controllers[20] that are an integral of *Amatria's* data infrastructure (see *Image 3*).

19 Raspberry Pi https://www.raspberrypi.org/

20 "Teensy USB Development Board." PRJC. https://www.pjrc.com/teensy/

Specifically, Tavola runs a C# script, creating a UDP client that listens to incoming messages. The master laptop sends out values for all 18 IR sensors at three values per message, with a frequency of 2 Hz. This results in 12 messages per second. The values are packed into a string message and then received in Tavola, where the values are parsed at runtime.

The values arrive in bundles of three because each message contains data for all IR sensors on an individual sound sensor scout (SSS), a distinct part of the sculpture that carries multiple sensors and actuators. Thresholds are set for each IR sensor, which, when reached, trigger a neighbor behavior where the IR sensor's SSS plays a scripted sequence of actuations. Subsequently, the neighboring SSS perform the same action until the actuation wave has reached the outermost SSS (#1 and #6). SSS #1, specifically, is the focus of scene 2 in Tavola (see *Image 5*). The message format is to be interpreted the following way: "b" is a prefix added by the Python shell

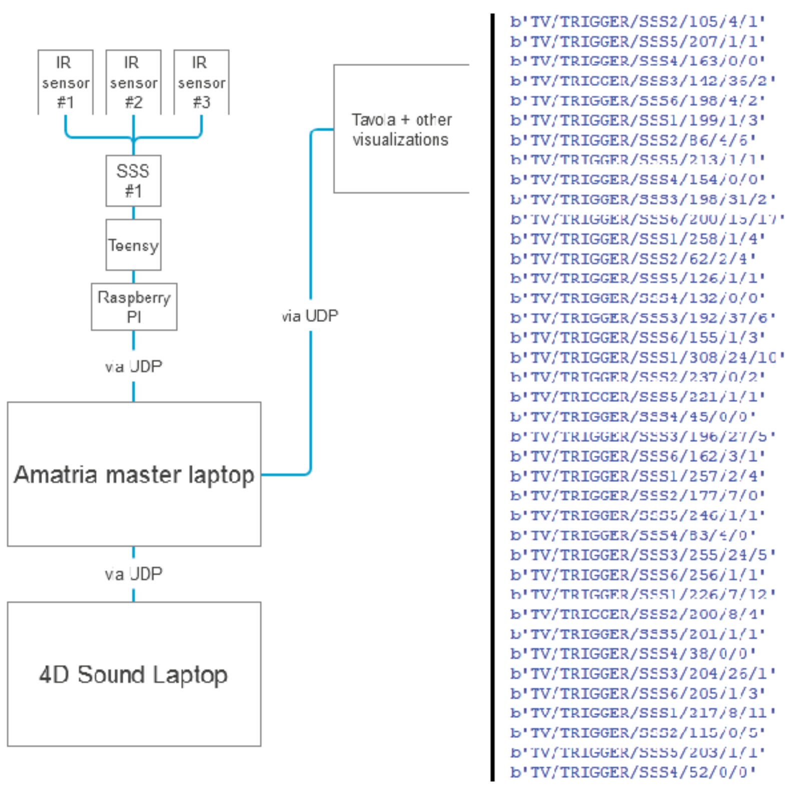


Image 3 Simplified system architecture (left); raw data stream from Amatria master laptop as logged in Python shell (right). The three numbers at the end of each message are IR sensor values.

when printing the incoming bytes and can be ignored. "TV" is a callsign for Tavola, distinguishing visualization-related UDP messages from others on the network. Then follows the SSS for which values are reported, identified by its number (1 - 6). Finally, values for the three IR sensors attached to the SSS are shown.

In order to address *insight need trends over time*, the same real-time UDP data stream was used. However, while Tavola uses sensor data for live updates, visualizations geared towards developers require data aggregation for an extended period of time. While Tavola parses incoming data at runtime and does not store any data (locally or online), a time series for developers required data storage. Section 3.4.2 details our implementation.

3.4 Analysis, Visualization, and Deployment

In the previous subsections, we investigated insight needs and data acquisition, and illuminated challenges faced along the way. In the following subsections, we detail the visualization process by stakeholder group.

3.4.1 For Visitors

In order to enhance visitors' understanding of *Amatria*, we decided to build two scenes in Tavola (see *Images 4 – 5*). This allowed for an easier-to-maintain application to which we can add more scenes in the future while providing a minimal amount of variety. *Scene 1* serves as a structural overview; *scene 2* prompts users for physical interaction with the sculpture and enables them to see their own behavior visualized on a small scale. This two-step gave users a gentle introduction to the use of visualization for sentient architecture, while allowing the small visualization team to keep the workload manageable.

We used Unity 3D[21], a video game engine increasingly used to develop all sorts of interactive 2D and 3D experiences such as games and architectural visualizations.

21 Unity. https://unity3d.com/

Scene 1 addresses *insight need geospatial* by allowing users to control a virtual camera pointing at the *Amatria* model, using the physical layout of the sculpture as a reference system. A data overlay consisting of graphic symbols encodes the location of sensors and actuators using graphic variables

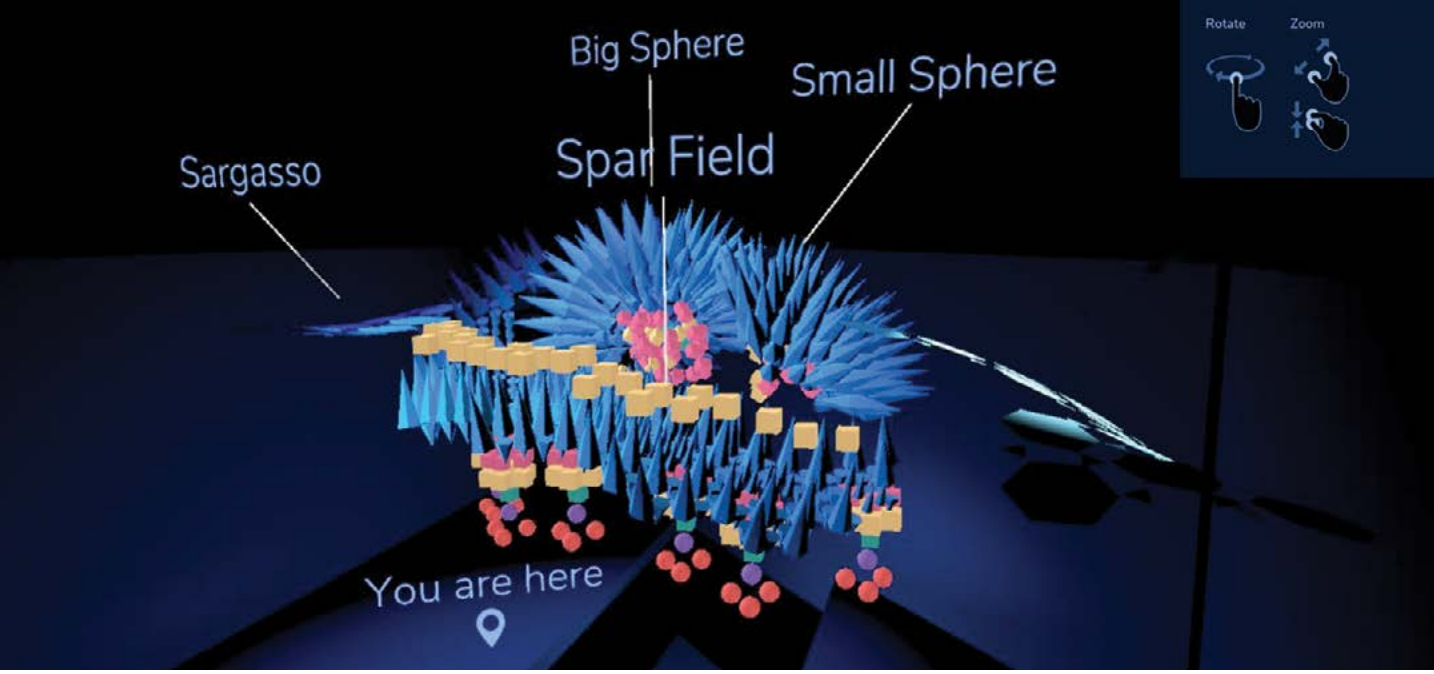


Image 4 Tavola, scene 1: *Amatria* overview.

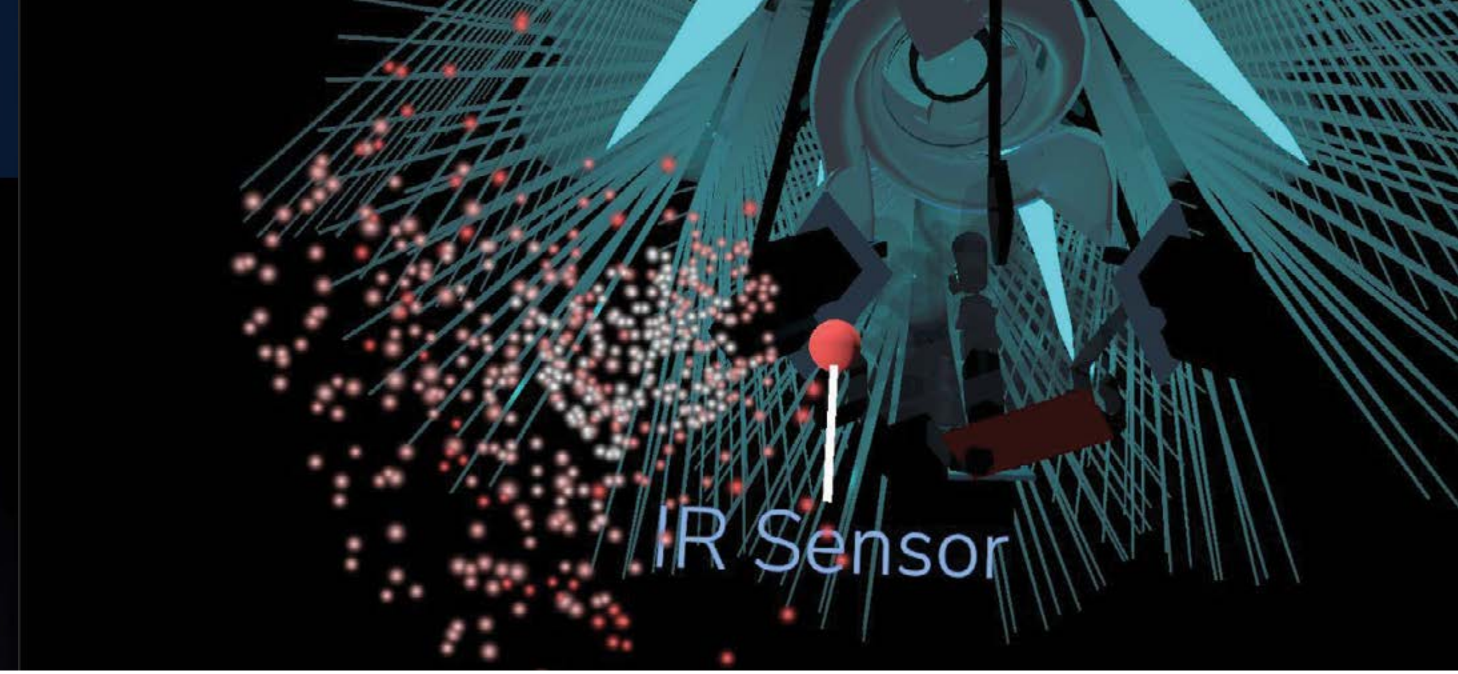


Image 5 Tavola, scene 2: *Amatria* Sound Sensor Scout.

(see *Table 1*). Scene 2 addresses *insight need comparison* where a live sensor value data overlay is updated twice a second over an SSS, allowing visitors to compare that particular sensor's states based on their own input when waving their hand in close proximity. Similarly, *trends over time* can be identified as the visualization visualizes each data point for exactly 5 seconds, allowing visitors to see whether values increase or decrease.

On the SSS model, apart from the three IR sensors the SSS carries, one microphone sensor, six rebel star lights, six vibration motors, and one speaker are also visible, although no data for them is streamed or visualized at this point. At this point, only sensor readings from IR sensor #2 on SSS #1 are visualized in real-time, encoding human proximity to the sensor via various graphic symbols and graphic variables (see *Table 2*). A sensor value over approximately 400 indicates the presence of a foreign object such as a human hand; a number under 200 shows the lack thereof. We created a data overlay with different encodings: a particle cone and a proximity index (see *Image 5*, center-left and lower-left respectively). Tavola is used under SSS #1 (see the location marker in *Image 4*) to ensure that visitors using the application without a docent understand the real-time character of scene 2, where their physical proximity to the marked IR sensor produces state changes in Tavola. While we only visualize data from one sensor, we are planning to scale up and include more sensor data in the future. Visualizing just one sensor was a good step in developing a visual encoding that can then be applied to other parts of the sculpture. In a significant investment of time and resources, a dark blue user interface (UI) was developed to complete this as a professionally made and user-friendly visualization.

Graphic Symbols & Graphic Variables

Readily available common graphical elements of interactive experiences simplified the implementation of Tavola (e.g., highly adaptable particle systems, various types of lights (point, spot, area, directional) can be created with a few mouse clicks). Similarly, materials can be assigned before and at runtime, prompting us to leverage Unity's great functionality to create highly custom visualizations with templates for simple and advanced 3D graphics.

In scene 1, to encode the structural data of *Amatria's* physical layout, we use one graphic symbol (volume) and six graphic variables (see *Table 1*).

Sensors and actuators are encoded by shape (sphere vs. cube). Each type of sensor and actuator is encoded by one color hue. Each element has a unique position in 3D space. The graphic symbols in the spar field were placed manually. Those in the large and small sphere of *Amatria* (only actuators) were laid out using a custom algorithm that assigned every graphical symbol a 3D location within an invisible spheroid, the center point of which was aligned with the middle of the sphere. Since scene 1 does not contain any real-time data, the visual encoding is static. Additionally, note that we use color intensity to encode qualitative data (whether parts of the data overlay are turned on or off). Usually, color intensity encodes quantitative data.

In scene 2, to encode the real-time sensor data stream, we use one graphic symbol (volume) and six graphic variable types (see *Table 2*).

Angle, color intensity, and speed are used for quantitative data encoding. The minimum and maximum values for each graphic variable might be improved by user testing. White and red were chosen as the start and end point of the color intensity gradient, both to ensure maximum visibility against the dark background of the scene and to reflect the red color of the IR sensors in scene 1. The use of x, y, and z-coordinates to encode the location of IR sensor #2 in SSS #1 is analog to how these graphic variables are used in scene 1.

		Graphic symbol types				
		Volume				
Graphic variable types	Shape*	Sphere: sensor		Cube: actuator		
	Color hue*	#EF5350 (red): IR sensor	#9575CD (purple): microphone sensor	#FFCC80 (yellow): light	#26A69A (green): speaker	#f06292 (pink): vibration motor
	Color intensity*	Opacity: 0%: graphic symbol turned off Opacity: 100%: graphic symbol turned off				
	x-position**	Location of sensor or actuator in 3D space				
	y-position**					
	z-position**					

Table 1 Graphic symbol types vs. graphic variable types in scene 1 of Tavola. *qualitative **quantitative

It is, of course, debatable whether the particle system should be considered a volume (seen as a whole) or as a scatter graph (seen as individual graphic symbols of type point). In the latter case, the graphic variables used would be velocity, color intensity, and speed. We concluded that humans likely perceive changes in the direction of the moving particles as a whole rather than as a collection of points. However, a focused user study could help confirm this.

Also in scene 2, in order to guide visitors to interact with this specific IR sensor on SSS #1, its current value is visualized within a 2D proximity index, using the graphic variable types of line and pictorial symbol (see *Image 5*, bottom-left corner).

We compute the x-position of the pictorial hand symbol based on the current IR sensor value. There is no name for this type of visualization, but the 2D reference system and the use of position as graphic variable type for lines of different size could qualify this proximity index as a bar graph. The color intensity gradient of the bars corresponds to the color scheme of the particles. Notably, the proximity index is a 2D visualization within a 3D visualization and introduces purposeful redundancy so that a visitor's interaction with IR #2 on SSS #1 (in the form of the IR sensor value) is encoded multiple times to maximize visitor understanding when using Tavola unsupervised.

		Graphic symbol types	
		Volume	
Graphic variable types	Angle**	0°: low sensor value	90°: high sensor value
	Color intensity**	#FFFFFF (white): low sensor value	#D50000 (red): high sensor value
	Speed**	0.2 scene units per second	0.4 scene units per second
	x-position**	Location of IR sensor	
	y-position**		
	z-position**		

Table 2 Graphic symbol types vs. graphic variable types for particle system in scene 2 of Tavola. *qualitative **quantitative

In addition to reference systems and data overlays, annotation plays an essential role in allowing the user to extract meaning from a visualization. In the case of Tavola, annotation is provided in the form of explanatory text and symbols (such as information on how to use the camera, acknowledgements, etc.), accessible via the UI.

Interaction Types

To satisfy visitor *insight need geospatial*, Tavola uses the interaction types *overview, zoom, rotation & panning*, and *filter*. Many interaction types support one of two functionalities: either manipulating the data or manipulating the view. Tavola users can do both. They can manipulate the data by *filtering* which parts of the data overlay should be displayed (by means of the toggles in the UI). But they can also manipulate the view of the data. For example, *overview* and *zoom* can be used to switch between viewing the whole sculpture and displaying just a small portion of it. *In scene 1, rotating & panning* the camera enables the user to see *Amatria* from different viewpoints, especially those that may not be reachable (such as overhead or very up-close). We support these interaction types by putting a gaze lock on the camera, causing the camera to always face an invisible vantage point in

		Graphic symbol types			
		Line		Pictorial (hand)	
Graphic variable types	Color intensity**	#FFFFFF (white): low sensor value	#D50000 (red): high sensor value		
	Size**	Width: 31 pixels ***: low sensor value	Width: 6 pixels***: high sensor value		
		Height: 0 pixels***: low sensor value	Height: 14 pixels***: high sensor value		
	x-position**	min (relative): low sensor value	Max (relative): high sensor value	Min (relative): low sensor value	Max (relative): high sensor value

Table 3 Graphic symbol types vs. graphic variable types for proximity index in scene 2 of Tavola.
*qualitative **quantitative ***pixels measured in original vector graphic

the middle of the sculpture, between the spheres and the spar field. Also, the camera is at a fixed distance from the vantage point and moves on the surface of an invisible spheroid, facing inward (a similar gaze lock is in place in *scene 2*). We should also note that *rotating & panning* is not part of the DVL-FW but would need to be included to capture the full breadth of 3D visualizations for IIS.

Aside from visualization-related interactions, both scenes require different kinds of real-world interaction between visitor and sculpture due to the fact that this is a visualization of a physical object: observation (*scene 1*), observation and providing sensor input (*scene 2*). Through the use of the proximity index and explanatory graphics in the UI, we hope that visitors feel prompted to interact with *Amatria* and see their influence on the sensors visualized. Whether this works as intended, however, needs to be shown by user studies.

Deployment

As mentioned in section 3.1, interactions often create requirements for preceding steps in the DVL-FW process model (see *Image 2*). Because Tavola allows the user to control camera movement, it was designed to run on touch-enabled devices. Initially, we tested Tavola on a 9.7” tablet. By using a Unity companion app, we could develop the touch interface without needing any time for compilation between versions. As of January 2019, deployment is underway for constructing a kiosk with a 32” touch screen to be placed under *Amatria* for visitor engagement. Users will be prompted to explore the sculpture in a way that organically adds to the experience of walking under it. This large display will run Tavola for visitors without any supervision or guidance from docents.

Informal Usability Study Results

In order to inform our development process, we frequently conducted informal user studies with team members and visitors during *Amatria* tours. While subjects had different levels of familiarity with Amatria, none understood the sensors and actuators in the sculpture. Broadly speaking, there were two categories of feedback: *feedback on design* and *feedback on functionality*.

The most common remarks we received were on *functionality*. Upon seeing Tavola running on a tablet, many visitors assumed it was possible to control the sculpture; they interpreted the presence of a 3D model as an affordance for a two-way communication. While this is not part of Tavola, we are in the process of developing such an interface, named “Encefalo” (Italian for “brain”), which is, however, still in a very early stage as of January 2019. We received lots of variations of this particular feedback, such as “I want to see more consequences of my actions on the sculpture,” and “Is the stand *Tavola* what controls *Amatria*/acting as her brain?” and “I want to interact with the sculpture via the app.” Comments on design were usually a lot more specific. We readily implemented feedback such as “I should be able to turn specific data overlay elements on and off by touching the corresponding entry in the legend,” and “Attach lines to bottom middle of text labels,” and “Remove [the 3D models of the] focus rooms [next to the *Amatria* model].” Testing the usability of Tavola is an ongoing effort and will most certainly figure in future research and development.

Image 6 Early prototype of Tavola where the IR sensor on the right reports consistently high proximity values.

3.4.2 For Developers

When creating Tavola, we did not have developers in mind as potential stakeholders. For the *Amatria* engineering team which implemented the UDP stream, observing behavior in the sculpture was essential, but individual sensor values were of limited importance, and the team used the Python shell while observing behaviors in the sculpture for debugging.

In an early prototype of Tavola, we visualized IR sensors #1 and #2 on SSS #1 (as opposed to just #2 in the current version). Image 6 shows this prototype in a state when no one is present under the sculpture. The particle cone on IR sensor #2 on the left is white, at a small angle, and the particle speed is at its minimum value, which is expected behavior. The cone for IR sensor #1 on the right, however, seems to indicate the close proximity of a foreign object.

Working with the developers to investigate this behavior uncovered an issue with the configuration of that sensor. This led to the realization that data visualization tools can also be helpful for developers, allowing them to "look under the hood" of a complex system.

While Tavola was developed for visitors, in this instance, we could satisfy *insight need comparison* for developers using the same data, which prompted us to consider this stakeholder group in our visualization design. In the following paragraphs, we will detail two visualizations for developers: a trends-over-time visualization (*Images 7 – 8*) and a real-time bar graph (*Image 9*), along with the visual encoding used for both (see *Tables 4 – 5*).

Graphic Symbol & Graphic Variable Types

Interested to understand if this was an isolated bug or the same for all 18 IR sensors distributed over the six SSS, we logged data for all IR sensors to answer other questions about *Amatria's* optimal configuration. Over the course of 24 hours in May 2018, we stored UDP messages from *Amatria* carrying IR values to address the insight need of identifying *trends over time*. Aiming to see if the thresholds for IR sensors need to be set dynamically based on the time of day (which varied by amount of people in the building, levels of ambient light during the day vs. at night, etc.), we created the graphs in Image 7-8. Here, time is plotted on the x-axis where the leftmost point is the beginning of the observation period (May 7th, ~7pm EST) and the rightmost point is the end thereof (May 8th, ~7pm EST). In order to maintain a manageable dataset size, we plotted only 1% of messages from *Amatria*, resulting in a message frequency of ~7.2 messages per minute. The data was then grouped into individual tables for each SSS and imported into Processing.[22] In order to create the final visualizations, graphs for all 18 IR sensors were created, and grouped into packs of six to get three visualization sheets. Each of these visualizations show six IR sensors with the same label, consisting of "IR" plus a number from 1 to 3.

Two *trend*-related insights can readily be gained from these graphs: No threshold change for IR sensors seems to be needed based on differences

22 Processing. https://processing.org/

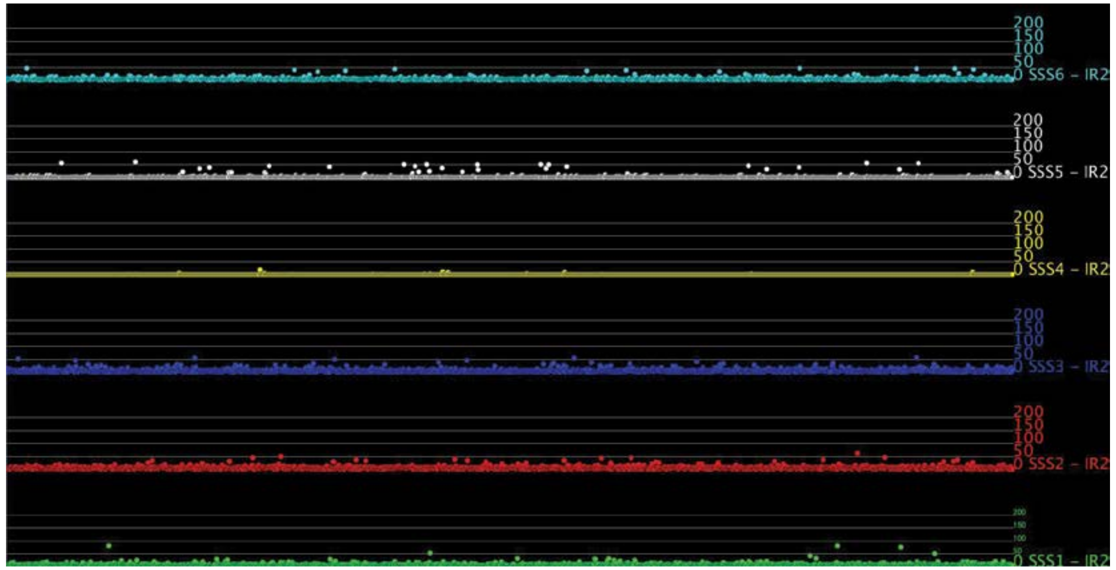


Image 7 Raw sensor data from six *Amatria* IR sensors (labeled "IR2") plotted over 24 hours

between day and night. But more importantly, we found that the consistently high sensor value for IR sensor #1 on SSS #1 is not an isolated incident; in fact, we encountered the same problem for all 6 SSS, specifically in their IRs #1. This actionable insight hinted at a challenging problem for the *Amatria* and Tavola team that is going to be tackled in the future.

To satisfy the developer *insight need comparison*, we wanted to allow an on-site team to see state change at sensor level during a maintenance visit in October 2018. This is why we created a real-time bar graph visualizing each IR sensor, grouped by the SSS to which they are attached. Image 8 shows this bar graph at a moment with no person underneath the sculpture; normally, all bars would be expected to hover at 0 or slightly above. What becomes clear is that all IR sensors labeled "IR1" report constantly high values.

Tables 4 and 5 below present the data mappings to graphic variables and graphic symbols used for the visualizations in Images 7 - 9.

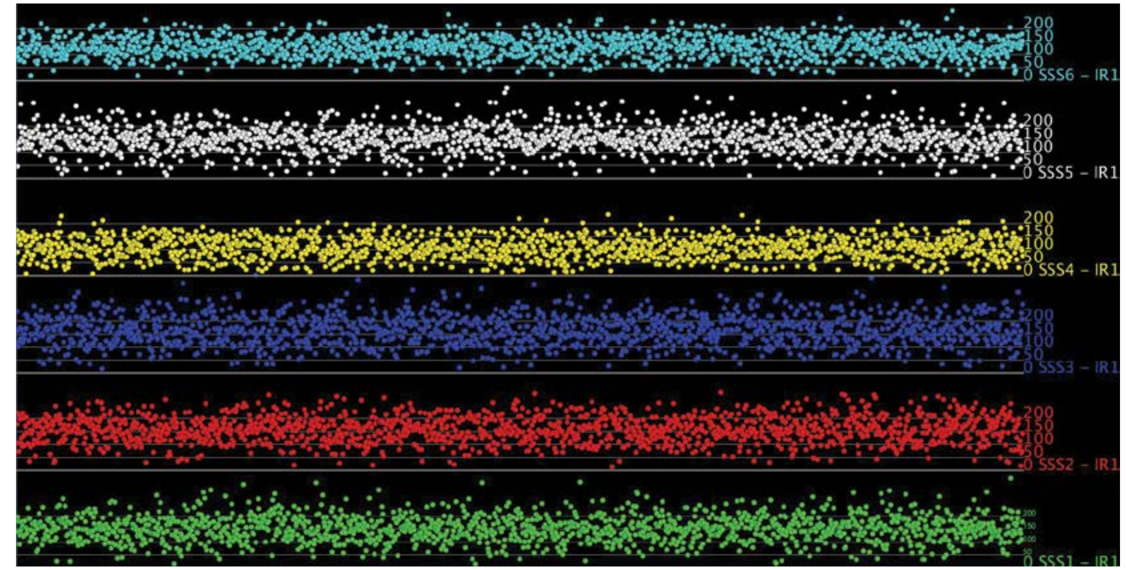


Image 8 Raw sensor data from six *Amatria* IR sensors (labeled "IR1") plotted over 24 hours

Interaction Types

Neither the trends-over-time visualization nor the real-time bar graph require interaction to satisfy developer *insight needs comparison* and *trends over time*.

Deployment

To keep the use of the trends-over-time visualization and the real-time bar graph simple, they were deployed on the same 9.7" tablet used for Tavola's initial development, and a laptop, respectively. The tablet can be carried around under the sculpture when testing IR responses, and the renderings of the trends-over-time visualizations can easily be shared via email or shown at talks and workshops. Thus, deployment meets insight needs.

3.5 Interpretation

The data visualizations presented here differ greatly by stakeholder. User studies will help us evaluate if the visualizations help explore and communicate *Amatria's* state and functionality.

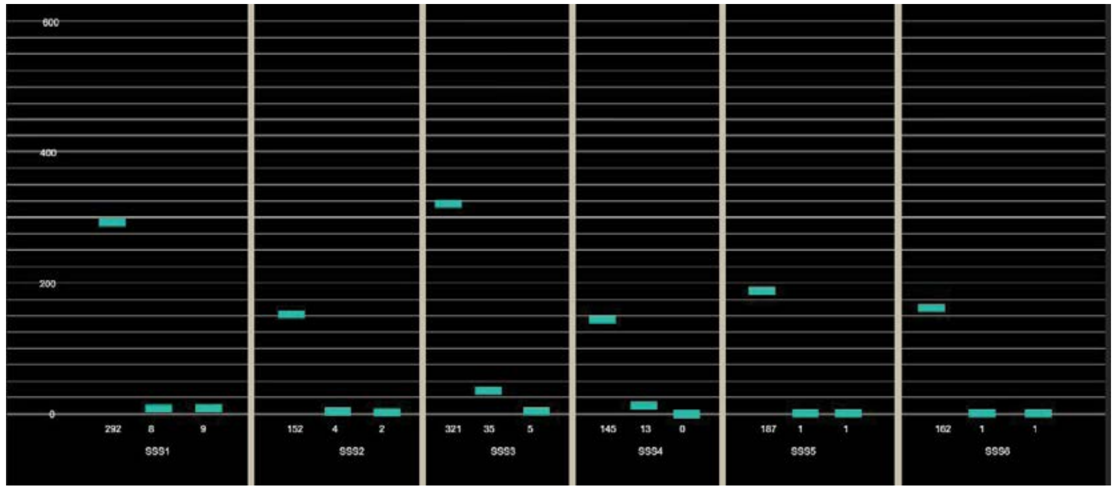


Image 9 Bar graph visualization showing all 18 IR sensor values in real-time.

Visitors are enabled to explore *Amatria* geospatially, compare IR sensor states based on their own input, and see trends over time for a fraction of the data traveling through *Amatria's* network. The expectation is that that visitors who used Tavola will understand *Amatria* measurably better than those who did not. Since no formal user studies have been conducted yet, we are enthused about the possibility of investigating user interpretations of Tavola (see Section 4).

Developers share some of the insight needs of visitors (*comparison* and *trends over time*), but their requirements for data visualization are quite different. Developers need to ensure *Amatria's* systems work properly and must check whether observed behavior is expected behavior. Plus they already have a firm *geospatial* understanding of the sculpture. While seeing values for all 18 IR sensor simultaneously in real-time (see *Image 9*) is a convenient way of ensuring proper functionality of sensors, developers must make decisions based on more information than just sensor values. Hence, to enable developers to make interpretations of the whole sculpture, visualizations with more data need to be implemented in the future.

Table 4 Graphic symbols and graphic variables for trends over time graph
*qualitative **quantitative

		Graphic symbol types	
		Point	
Graphic variable types	Color hue*	To distinguish all IR sensors with the same label by which SSS they are attached to	
	x-position**	min: recording time (start) May 7th, ~7p EST	min: recording time (end) May 8th, ~7p EST
	y-position**	min: 0 no proximity of foreign object	max: ~800 high proximity of foreign object

Table 5 Graphic symbols and graphic variables for real-time bar graph of 18 IR sensors. Note that x-position is qualitative in this case.
*qualitative **quantitative

		Graphic symbol types	
		Point	
Graphic variable types	x-position*	To separate bars areas for different sensors (no quantitative relationship)	
	y-position**	min: 0 no proximity of foreign object	max: ~800 high proximity of foreign object

4. Discussion and Outlook

This paper discussed different visualizations of sentient architecture data: Tavola for visitors and simpler graphs for developers. For all visualizations, we detailed the graphic variable and graphic symbol types, as well as interaction and deployment types. In this last section, we discuss planned research and development.

Going forward, we plan to run formal user studies where a control group of visitors would get a basic oral introduction to *Amatria* and would then be asked questions about its structure, such as "How many IR sensors are there in *Amatria*?" or "Which part of the sculpture is not actuated?" An experiment group would also get this introduction but then explore Tavola before answering the very same questions. Comparison of both user groups—quantitatively in terms of the time needed to complete the questionnaire and accuracy of answers, but also qualitatively based on verbal feedback—will help us understand the utility and usability of Tavola and its impact on how visitors understand and interact with Amatria.

In addition, we are interested to visualize more of *Amatria's* real-time data, more specifically the interactions between various parts of the sculpture that are not sensors or actuators (e.g., Teensies and Raspberry Pis. During a behavior workshop with *Amatria* and Tavola team members in Toronto in December 2018, alternative data transmission protocols were tested. Open sound control (OSC), already used for communication between the master laptop and a 4D Sound Laptop (see *Image 3*), has been found to facilitate the formatting of messages. While investigating OSC for Tavola, we discovered that the workload needed for custom parsing in Tavola can be reduced. Additionally, message queuing telemetry transport (MQTT), already in use in many IoT setups, shows promise in terms of the scalability of data streams out of *Amatria*. Rather than sending data to specific IP addresses, the master laptop would publish data to a broker that can be accessed by another machine, eliminating the need for the master laptop to be provided with the IP addresses of the destination for the data stream. Another reason for testing OSC and MQTT was to check the feasibility of two-way communication between *Amatria* and Encefalo, a proposed software to control the sculpture from a graphical user interface. This would enable various stakeholders not only to see *Amatria* data visualized but also to send input back to the sculpture, resulting in a lot of potential interaction between humans and sentient architecture.

To further advance our line of study, we need to also align our research and development to projects in the realm of the Industrial Internet of Things (IIoT), as well as validate findings with applications and user studies across application domains. While sentient architecture is a fascinating artistic view on smart environments, there are many areas of data-driven innovation that could greatly benefit from data visualization.

As we move forward, we will need to address questions of efficacy: for instance, is there a measurable difference in IIS understanding between groups who use visualizations and those who do not? Questions arise, as well, regarding the DVL-FW. As mentioned, the DVL-FW draws on much previous research on graphic symbols and variables, data scale types, and insight needs: how, for example, can we use 2D visualization within 3D applications (e.g., what is the optimal visualization of the IR sensor proximity in *Image 5*), or what graphic variables and graphic symbols are useful in 3D visualization? Which interactions are uniquely suited for 2D visualization, and which ones for 3D visualizations, and which can be translated between the two? Furthermore, how can we add interactions that only work on very specific deployment modes (e.g., camera rotation & panning) to the DVL-FW without overloading it?

Additional References

Spaulding, Aaron, Krzysztof Z. Gajos, Anthony Jameson, Per Ola Kristensson, Andrea Bunt and Will Haines. (2009) "Usable Intelligent Interactive Systems: CHI 2009 Special Interest Group Meeting." CHI'09 Extended Abstracts on Human Factors in Computing Systems (Newy York: ACM, 2009), 2743 - 2746.

Winograd, Terry "Shifting viewpoints: Artificial Intelligence and Human–Computer Interaction". Artificial Intelligence 170.18 (2006): 1256 - 1258.

Hassanalieragh M, et al. (2015) Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges. 2015 IEEE International Conference on Services Computing (IEEE Computing Society, Washington, DC), pp 285-292.

Acknowledgements

The authors would like to thank Rob Gorbet and Philip Beesley for their expert support of the work presented here; Rob Henderson, Lynne Crohn, Mark Napier for *Amatria* network IT support; Adam Francey for implementing the *Amatria* data pipeline; Chauncey Frend for Unity support; Leonard Cross for graphic design support; Todd Theriault for proofreading. Victoria Fard and Timothy Boll kindly provided the *Amatria* and Luddy Hall 3D model. This work was partially funded by CNS at IU and LASG in Toronto, ON, Canada.

Katy Börner is the Victor H. Yngve Distinguished Professor of Engineering and Information Science in the School of Informatics, Computing, and Engineering, Adjunct Professor at the Department of Statistics in the College of Arts and Sciences, Core Faculty of Cognitive Science, Founding Director of the Cyberinfrastructure for Network Science Center at Indiana University, Bloomington, IN. She is a curator of the international Places & Spaces: Mapping Science exhibit that features large-format maps and interactive data visualizations. She holds a MS in Electrical Engineering from the University of Technology in Leipzig, 1991 and a Ph.D. in Computer Science from the University of Kaiserslautern, 1997. She is a member of ACM and IEEE and is an American Association for the Advancement of Science (AAAS) Fellow and a Humboldt Research Fellow.

Andreas Bueckle is a PhD candidate in Information Science in the School of Informatics, Computing, and Engineering at Indiana University. Coming from a background in video journalism and media, Andreas performs research and development of data visualizations in augmented and virtual reality, exploring the possibilities to visualize and allow for immersive interactions with data in 3D worlds. He holds a B.A. in Media Studies from Eberhard Karls University in Tuebingen and an M.A. in Communications in Economic and Social Contexts from Berlin University of the Arts (Germany).

About the Living Architecture Systems Group

The publication forms part of a series of work-in-progress reports and publications by Living Architecture researchers and contributors. The Living Architecture Systems Group is an international partnership of researchers, artists, and industrial collaborators studying how we can build living architectural systems— sustainable, adaptive environments that can move, respond, and learn, and that are inclusive and empathic toward their inhabitants. "Smart" responsive architecture is rapidly transforming our built environments, but it is fraught with problems including sustainability, data privacy, and privatized infrastructure. These concerns need conceptual and technical analysis so that designers, urban developers and architects can work positively within this deeply influential new field.[i] The Living Architecture Systems Group is developing tools and conceptual frameworks for examining materials, forms, and topologies, seeking sustainable, flexible, and durable working models of living architecture.

Living Architecture Systems Group research is anchored by a series of prototype *testbeds*: accessible, immersive architectural sites containing experiments and proof-of-concept models that support living architecture as a practical model for our future built environment. These testbeds act as *boundary objects*[ii] that help researchers answer ethical, philosophical and practical questions about what living architecture means and who it is for within our societies and environments, creating sites of collaborative exchange that act both as research ventures and as public cultural expressions.

A series of far-reaching critical questions can be explored by using the tools and frameworks that are described within this specialized publication series: can the buildings that we live in come alive? Could living buildings create a sustainable future with adaptive structures while empathizing and inspiring us? These questions can help redefine architecture with new, lightweight physical structures, embedded sentient and responsive systems, and mutual relationships for occupant that provide tools and frameworks to support the emerging field of living architecture. The objective of this integrated work envisions embodied environments that can provide tangible examples in order to shift architecture away from static and inflexible forms towards spaces that can move, respond, learn, and exchange,[iii] becoming adaptive and empathic toward their inhabitants.[iv]

i Kas Oosterhuis and Xin Xia, *iA #1, Interactive Architecture* (Rotterdam: Episode Publishers, 2007).

Nicholas Negroponte, *Soft Architecture Machines* (Cambridge, Mass.: MIT Press, 1975).

Lucy Bullivant, *4dsocial: Interactive Design Environments* (London: AD/John Wiley & Sons, 2007).

Neil Spiller, *Digital Architecture Now: A Global Survey of Emerging Talent* (London: Thames & Hudson, 2009).

Michael Fox and Miles Kemp, *Interactive Architecture* (Princeton: Princeton Architectural Press, 2009).

ii LASG uses Bowker and Star's terms *boundary object* and *boundary concept* to frame the design and analysis of the multiple physical armatures and architectural constructions that make up LASG testbeds. In order to act effectively, a boundary object needs to be both concrete and abstract, both precisely defined and fluid, offering an "object that is part of multiple social worlds and facilitates communication between them; it has a different identity in each social world that it inhabits": Geoffrey C. Bowker and Susan Leigh Star, *Sorting Things Out: Classification and Its Consequences* (Cambridge, Mass: MIT Press, 1999): pp. 409.

iii For example the Living Architecture (LIAR) next-generation, selectively programmable bioreactor developed by LASG Metabolism Stream Lead Rachel Armstrong, Newcastle, uses microbial processes to generate electricity, oxygen, fertiliser, and other life-sustaining outputs from waste (carbon dioxide, grey water) that would otherwise be ejected from a building: "Living Architecture LIAR," accessed February 2, 2022, https://livingarchitecture-h2020.eu/.

iv Bullivant, *4dsocial*.

References

Agostini, Andrea, and Daniele Ghisi, "A Max Library for Musical Notation and Computer-Aided Composition," *Computer Music Journal* 39, no. 2 (Summer 2015): 11–27, DOI: https://doi.org/10.1162/COMJ_a_00296.

Beans, Carolyn. "Sentient Architecture Promises Insight into Our Evolving Relationship with AI." *Proceedings of the National Academy of Sciences* 115, no. 30 (2018): 7638–40. https://doi.org/10.1073/pnas.1809390115.

Börner, Katy, and Andreas Bueckle. "Envisioning Intelligent Interactive Systems: Data Visualizations for Sentient Architecture." Essay. In *Living Architecture Systems Group White Papers 2019*, 63–88. Toronto, Canada: Riverside Architectural Press, 2019.

Bullivant, Lucy. *4dsocial: Interactive Design Environments*. London: AD/John Wiley & Sons, 2007.

Carpentier, Thibaut. "A new implementation of Spat in Max." In *Proc. of the 15th Sound & Music Computing Conference (SMC)*, Limassol, Cyprus, (July 2018): 184–191, DOI: https://doi.org/10.5281/zenodo.1422552.

Durkheim, Émile. *The Elementary Forms of Religious Life*. Translated by Karen E. Fields. New York, NY: Free Press, 1995.

Fox, Michael and Miles Kemp. *Interactive Architecture*. Princeton: Princeton Architectural Press, 2009.

"Living Architecture LIAR," accessed February 2, 2022, https://livingarchitecture-h2020.eu/.

Negroponte, Nicholas. *Soft Architecture Machines*. Cambridge, Mass.: MIT Press, 1975.

Oosterhuis, Kas and Xin Xia. *iA #1, Interactive Architecture*. Rotterdam: Episode Publishers, 2007.

Spiller, Neil. *Digital Architecture Now: A Global Survey of Emerging Talent*. London: Thames & Hudson, 2009.

Credits

Project Leads

Philip Beesley
Katy Börner
Andreas Buekle
Mark Francis

PBAI Core Team

Production Manager Gabriella Bevilacqua
Design Lead Timothy Boll
Design Lead Victoria Fard
Design Nathanael Scheffler
Engineer Wenlong Meng
Engineer Amber Ma

PBAI Team

Mehreen Ali
Lorena Almarez
Siddhant Chandgadkar
Charlotte Dyck
Adam Francey
Maxime Gordon
Joey Jacobson
Luke Kimmerer
Richard Mui
Anne Paxton
Jordan Prosser
Lily Rae
Adam Schwartzentruber
Aleksandra Simic
Matthew Spremuli
Amir Rostami
Padmini Unni
Mike Zheh Hu

Collaborators

Robert Gorbet
Miggy Torres
Salvador Breed, 4DSOUND
Poul Holleman, 4DSOUND

Indiana University Team

Brian, IU physical plant staff
Greg, IU physical plant staff
Will Bewley
Don Carlberg
Kallah Gallapoo

Production (Toronto)

Rasha Abou-Hawach
Ous Abou Ras
Parvin Atayee
Ozlem Bektas
Brandon Bernard
Avijit Chakrabarty
Eric Co
Chelsea Chisim
Jianghong Dai
Gene Duterte
Marla Goubran
Chenyu Huang
Justin Hung
Marina Ibrahim
Chris Kang
Jocelyn Kent
Shamim Khedri
Dohyon Kim
Michelle Lan
Yena Lee
Meikang Li
Noor Mah
Donna Ming Liu
Maha Mubeen
Zargham Nasir
Pierre Ng
Lila Nguyen
Junior Osei Wireko
Christian Paez Diaz
Louell Palparan
Pooja Patel
Sanjana Patel
Lidia Proykina
Zheohuan Qiu

Production (Indiana)

Maryam Adeniyi
Janice Bagwell
Ian Bailey
Anna Bednarski
Wil Bewley
Steve C
Kate Caldwell
Dan Caldwell
Owen Caldwell
Marshall Caldwell
Chelsea Campbell
Zhi Chai
Yanlin Chen
Scott Dauer
Deepthi Devireddy
Kailen Dobias
Tia Donaldson
James Duana
Hunter Durkin
Abin Francis
Gamaliel Garcia
Lindsay Giacomino
Dalton Gray
Hiroko Hanamura
Chris Hebb
Emerson Hobson
Joey Huang
Mark Jackman
Qinwen Jiang
Jeremy Jin
Avryn Leo Justin
Jimmy Kacius
Anna Keune
Diane Kewley-Port
Amanda King
Skyler Kolli
Sahiti Korrapati
Sara Laughlin
Yuqian Li
Kirstine Lindemann
Sarah Lloyd
Cathy Lu
Wenxuan Ma
Binita Madaiah
Ann Marie Matheny
Dominic Matthys
Lindsey Maynard
Michael McDaniel
Xandria McDowell
Kate Messing
Connie Michalopoulos
Stasa Milojev
Rachel Morris
Kathy Morrison
Benjamin Moschea
Hamid Nadir
Krishan Narsinghani
Andrew O'Conor
Jay Patel
Nate Pellant
Max Potter + Parent
Bunardy Pranata
Kiarra Pratchett
Tim Pratt
Andre Quan
Sierra T. Reed
Nova Richardson