Integrated Reconfigurable Autonomous Architecture System

Full-process autonomy in the continuous architectural lifecycle driven by interactive platform, intelligent algorithm and robotic material

Tyson Hosmer
Bartlett School of Architecture, UCL

Jiaqi Wang
School of Architecture, South China University of Technology

Wanzhu Jiang
School of Architecture, South China University of Technology

Ziming He
College of Design and Innovation, Tongji University

1

1 The rendering of project TESSERACT developed based on the IRAAS.

ABSTRACT

Advances in state-of-the-art architectural robotics and artificially intelligent design algorithms have the potential not only to transform how we design and build architecture, but to fundamentally change our relationship to the built environment. This system is situated within a larger body of research related to embedding autonomous agency directly into the built environment through the linkage of AI, computation, and robotics. It challenges the traditional separation between digital design and physical construction through the development of an autonomous architecture with an adaptive lifecycle. Integrated Reconfigurable Autonomous Architecture System (IRAAS) is composed of three components: 1) an interactive platform for user and environmental data input, 2) an agent-based generative space planning algorithm with deep reinforcement learning for continuous spatial adaptation, 3) a distributed robotic material system with bi-directional cyber-physical control protocols for simultaneous state alignment. The generative algorithm is a multi-agent system trained using deep reinforcement learning to learn adaptive policies for adjusting the scales, shapes, and relational organization of spatial volumes by processing changes in the environment and user requirements. The robotic material system was designed with a symbiotic relationship between active and passive modular components. Distributed robots slide their bodies on tracks built into passive blocks that enable their locomotion while utilizing a locking and unlocking system to reconfigure the assemblages they move across. The three subsystems have been developed in relation to each other to consider both the constraints of the AI-driven design algorithm and the robotic material system, enabling intelligent spatial adaptation with a continuous feedback chain.

INTRODUCTION

The system we present is situated within a larger body of work related to embedding agency and autonomy directly into the built environment through the linkage of AI, computation, and robotics. The underpinnings of this research are rooted in the transdisciplinary field of cybernetics, largely pioneered in the 1960s and 1970s through the work of Gordon Pask, Cedric Price, Nicholas Negroponte, Christopher Alexander, John Frazer, and Archigram (Steenson 2010). Pask stated, "The role of the architect here, I think, is not so much to design a building or city as to catalyze them; to act that they may evolve (Frazer 1995). Pask challenged the paradigm that architecture is a static material artifact produced through a linear series of processes from design to fabrication to construction and reconsidered it as a composition of interrelated active systems regulated and controlled through feedback within a constantly shifting environment (Pask 1969).

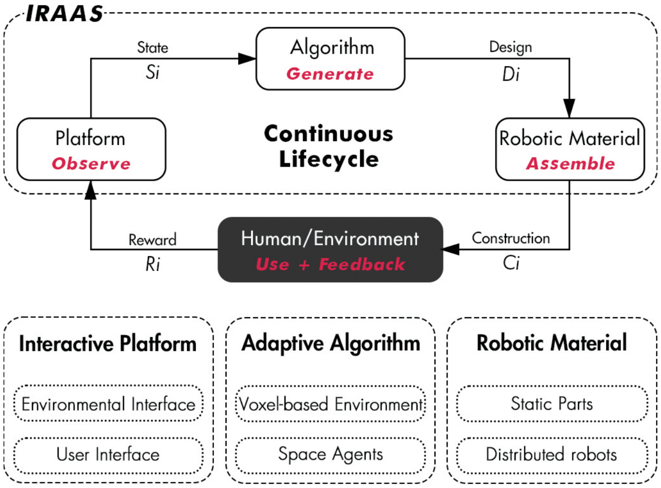
Today, the climate crisis, housing crisis, and covid pandemic are radically transforming the way we live and work, while automation, consumer platforms, and AI are changing how we interact with and personalize our world. As Sean Hanna states, "for the first time our environment is no longer seen as fixed, or shaped by forces beyond our control, but as in constant and noticeable change, and that our relationship with it is one of mutual interaction (Hanna 2020)." Computational design and construction robotics research focus primarily on automation, customization, and optimization within linear, separated processes of design and construction rather than fundamentally changing the interrelationship between them to enable open-ended and physically adaptive architecture. Robotic buildings must consider not only design to production, but design-to-production-to-operation chains from a lifecycle perspective relating to the socio-economic and ecological impacts (Bier and Mostafavi 2018; Bier et al. 2018).

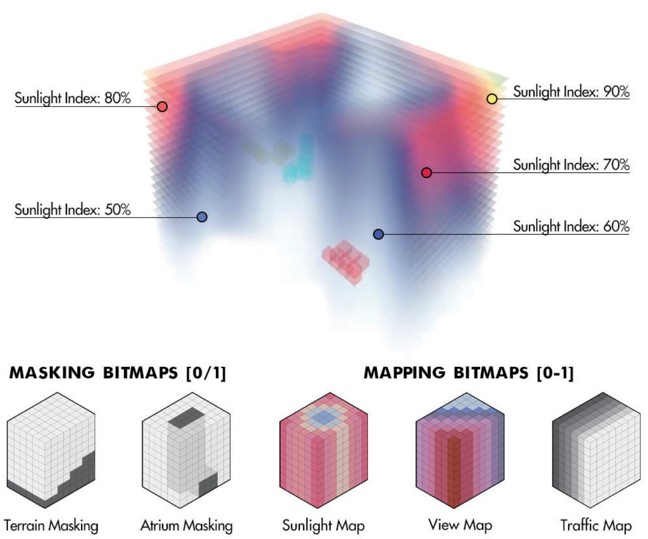
Our research challenges the separation between digital design and physical construction processes through the development of an integrated cyber-physical architectural system with a feedback-based adaptive lifecycle. IRAAS is a semi-autonomous reconfigurable architecture integrating three main components: 1) an interactive platform for user and environmental data input 2) an agent-based space generation algorithm with deep reinforcement learning for continuous spatial adaptation and 3) a distributed robotic material system with a bi-directional control protocol for simultaneous state alignment. The three subsystems have been developed in relation to each other to consider both the constraints of the AI-driven design algorithm and the robotic material system, enabling intelligent spatial adaptation with a continuous feedback chain.

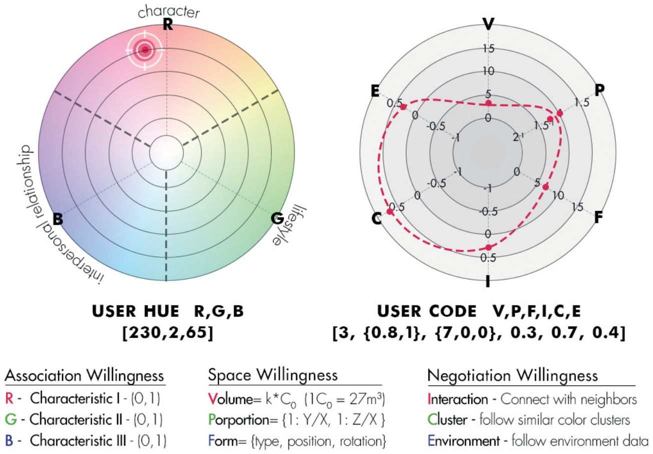
BACKGROUND

Historically cybernetics, defined as the study of communication and control within systems (Wiener 1948), and artificial intelligence (McCarthy 1955), defined as the study of machines that exhibit and simulate intelligent behavior (Oxford English Dictionary reference), are closely interrelated. They have led to two primary threads of architecture research: 1) intelligent computational processes for design and 2) physically adaptive and responsive environments. Nicholas Negroponte was interested in developing a symbiotic relationship between designers and "architectural machines," introducing the notion of an intelligent agent in the design process (Negroponte 1970), while Cedric Price developed systems where "the act of engaging and interacting with the architecture would change the user (Steenson 2010)." Both Pask and Price were interested in managing "indeterminacy", considering architecture's ability to adapt to, be adapted by, and impact its inhabitants (Pask 1965; Landau 1968; Steenson 2010). Development of embodied adaptive architecture requires reappraisal of linear building lifecycles. Rather than automate known, predefined patterns of construction, our overriding aim is to develop an autonomous architecture that continuously adjusts itself to actively maintain a symbiotic relationship between people, the natural environment, and the architectural environment.

State of the art research in autonomous robotic systems for architecture has investigated two primary strategies, 1) small, distributed robots manipulating building assemblies and 2) large, monolithic embodied robotic spaces and buildings. Monolithic systems enlarge the robot to the size of spaces or whole buildings (Oosterhuis 2012; Kilian 2018; Maierhofer 2019; Hosmer 2019). Adaptive spatial behaviors are triggered through the sequencing of actuators integrated into larger bodies. Alternatively, strategies have been developed using distributed robots to collaboratively manipulate building assemblies. Principles of swarm intelligence have been applied through embodied swarm construction (Rubenstein et al. 2012; K. Petersen and Nagpal 2017; Petersen et al. 2011). Full scale collaborative robotic assembly of timber structures is demonstrated using industrial robotic arms hung in a mobile gantry system (Adel et al. 2018). BILL-E robotic platform demonstrates a type of "relative robot in a structured environment", enabling robots to climb on the same structure they assemble and reconfigure (Jenett and Cheung 2017). A similarly symbiotic dependency between distributed robots and static elements demonstrates self-assembly of timber structures (Leder et al. 2019). Monolithic embodiments offer opportunities for controlled adaptation through constraints with less radical shifts in topology while distributed robotic strategies offer more flexibility enabling radical changes in topology at the cost of organizational complexity requiring additional spatial design inputs to be effective.






character
R
B
G
interpersonal relationship
lifestyle
USER HUE R,G,B
[230,2,65]
V
P
F
I
C
E
USER CODE V,P,F,I,C,E
[3, {0.8,1}, {7,0,0}, 0.3, 0.7, 0.4]
Association Willingness
R - Characteristic I - (0,1)
G - Characteristic II - (0,1)
B - Characteristic III - (0,1)
Space Willingness
Volume= k*C0 (1C0 = 27m3)
Porportion= {1: Y/X, 1: Z/X }
Form= {type, position, rotation}
Negotiation Willingness
Interaction - Connect with neighbors
Cluster - follow similar color clusters
Environment - follow environment data

2 Continuous lifecycle and system composition.

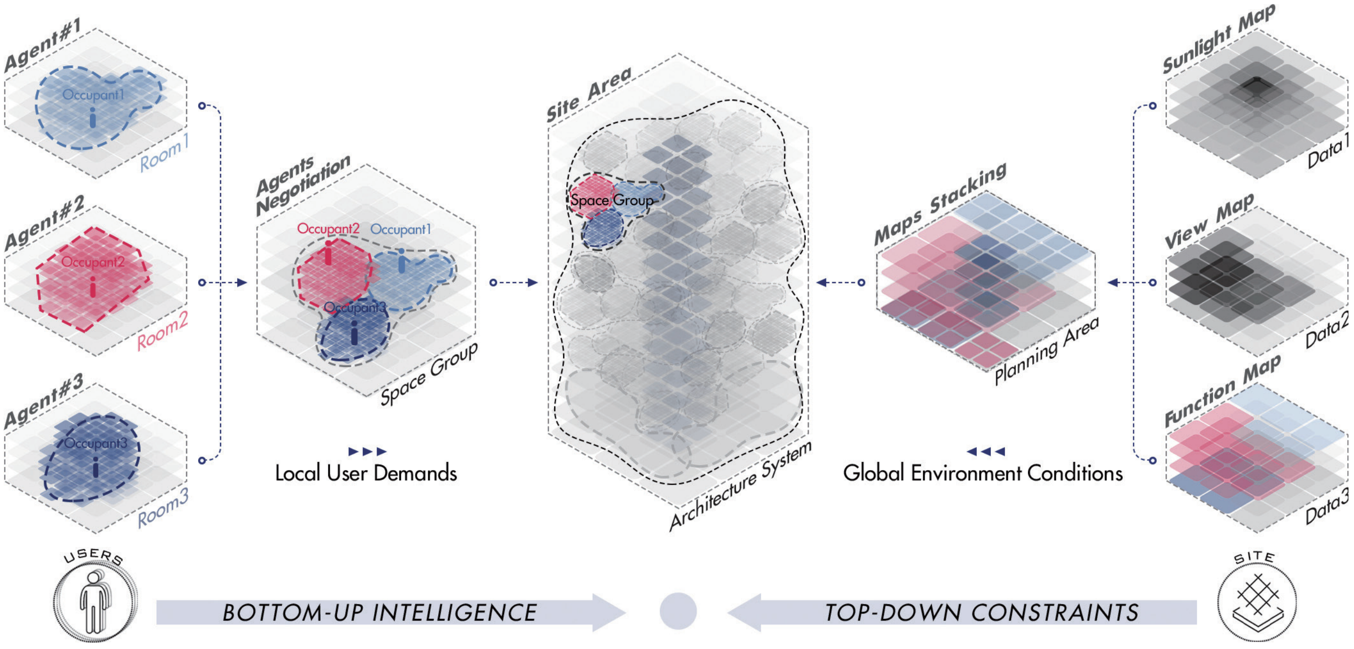
3 The environmental data, types and examples.

4 The user interface, data structure and example.

Generative space planning research focuses on a range of strategies for semi-autonomously computing spatial organization. We divide relevant strategies into procedural geometry, rule-based, and data-based algorithms. Liggett gives a historical overview of automated space planning methods (Liggett 2000). Procedural geometry is used for automating desk layouts (Anderson 2018) while procedural geometry with a multi-objective genetic algorithm was developed for 2D office space planning (Nagy 2017). Rule-driven models have been explored such as shape grammars (Hua 2017), cellular automata (Dinçer et al. 2014), semi-autonomous constraint satisfaction (Hosmer et al. 2020), and various multi-agent systems achieving 2D or 3D space planning by coupling design goals with geometric or topological constraints (Veloso 2019; Meyboom and Reeves 2013). Multi-agent approaches can be divided into three main groups: (1) agents as moving spatial units, (2) agents that occupy a space, and (3) agents that partition a space (Veloso 2019). One multi-agent method learns space planning behaviors in 2D with reinforcement learning (Veloso and Ramesh 2020). Data-based models have been developed using Generative Adversarial Networks (GANs) (Goodfellow et al. 2014). To learn significant features and their relationships across image-based datasets (Isola et al. 2017) and have demonstrated 2D space planning strategies (Zheng and Huang 2018; Chaillou 2019; Chaillou 2020).

Most research in semi-autonomous robotics places emphasis on methods of construction or adaptability through the lens of the robotic constraints without apt consideration for how spaces would be designed, organized, and adjusted. Results of purely bottom-up robotic methods tend to be limited to behaviors for assembling abstract structures. Generative space planning strategies tend to overemphasize methods for spatial organization resulting in digital models without ample alignment with the constraints of the robotic material system nor an appropriate communication protocol between the digital design processes and physical assembly processes leaving gaps that prevent a continuous chain.

Autonomy requires a system with an ability to self-manage through a degree of self-awareness. It can be defined as having an effective interdependency between properties of facilitated variation, situated and embodied agency, and intelligence (Hosmer 2019). This research embeds principals of autonomous architecture by developing an interdependency between the intelligent agency of the space planning algorithm and situated and embodied agency in the design of the robotic material system as a structured environment. Facilitated variation is achieved through the design of effective constraints in the robotic material system to reconfigure elements that simultaneously enable locomotion patterns. The development of the control system protocols enables simultaneous state alignment



5 The basic logic of the Agent-Based Spatial Planner.

between the virtual simulator and physical world in a continuous feedback cycle.

METHODS

System Composition and Lifecycle

IRAAS is a semi-autonomous architecture that operates in a continuously adaptive lifecycle through an "Observe, Generate, Assemble" feedback loop. IRAAS is implemented through a combination of three closely related components: 1) interactive platform, 2) agent-based space generation algorithm, and 3) robotic material system. At a discrete moment (T_i), environment data and user goals are collected by the interactive platform and sent to the space generation algorithm. It observes its current state (S_i) in relation to processed inputs to generate an adaptation as a virtual space design (D_i), guiding the adjustment or reconfiguration of the actual construction (C_i) through the robotic control system. Changes in the environment and user goals provide constant feedback (R_i) in a closed loop (Figure 1).

Interactive Platform

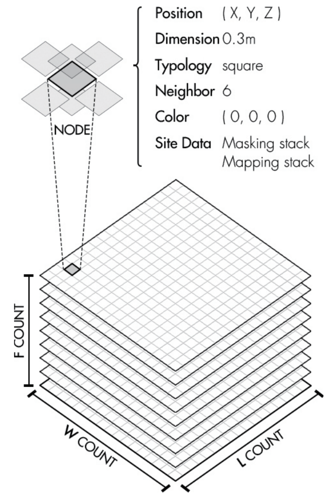
The interactive platform serves as a data collection port that triggers adjustments in the system from 1) environmental data and 2) multi-user space planning goals collected through the interface. Environmental data such as sunlight exposure is collected through digital simulations and processed as 3D bitmaps. Fixed elements in the environment such as exterior facades are also processed through bitmaps. Bitmaps are translated into 3D data matrices establishing global constraints and environmental factors that influence the space planning algorithm (Figure 2).

The user interface is responsible for collecting user characteristics and space planning goals, processed as inputs that drive the behaviors of the space generation algorithm. Each user is translated into a "user code" and "user hue." User hue reflects the user's characteristics in relation to other users. The user code captures their spatial goals and willingness for negotiating their spaces with others (Figure 3).

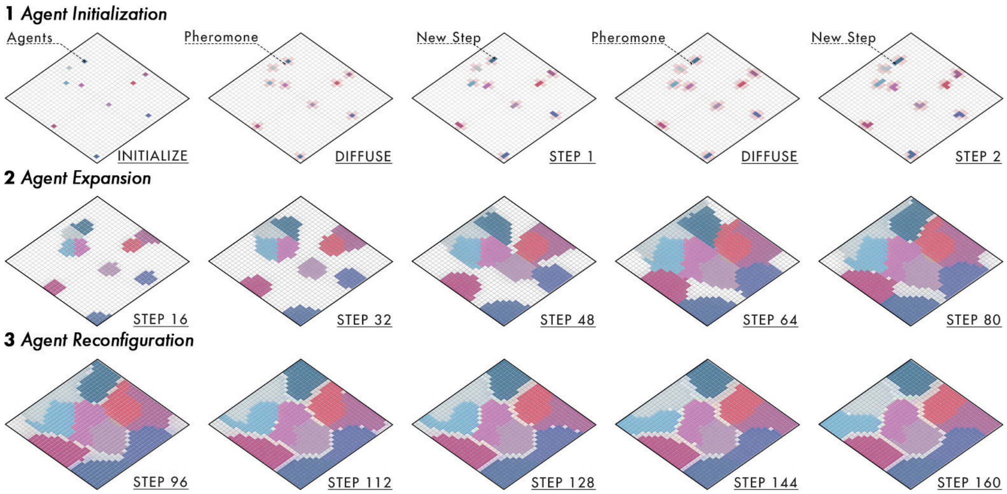
Agent-Based Spatial Planner with Self-Play Reinforcement Learning

The spatial planner component is responsible for generating 3D volumetric space boundaries within a virtual environment. The computational model is trained using reinforcement learning to learn adaptive policies for adjusting the scales, shapes, and relational organization of spatial volumes by processing changes in the environment and user requirements in near real-time (Figure 4). It is designed as a multi-agent system with programmable agents, each representing an independent space within the environment. We extend principals of the Stigmergic Space Adjacency Software for multi-agent space planning with stigmergic communication in a 3D environment (Meyboom and Reeves 2013). Our algorithm generates spatial organization through intelligent negotiation behaviors of agents driven by user and environment data, outputting structured environments readable by the robotic material control system.

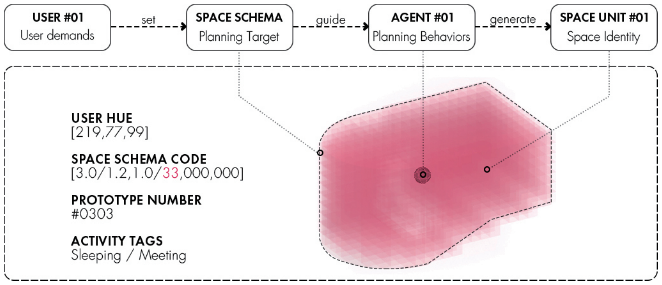
The environment is a 3D grid of voxel arrays. Each node contains metadata, including its position, size, geometry type (cubic, tetrahedral, etc.), color, and layers of site/environmental information imported from the platform. The size and geometry



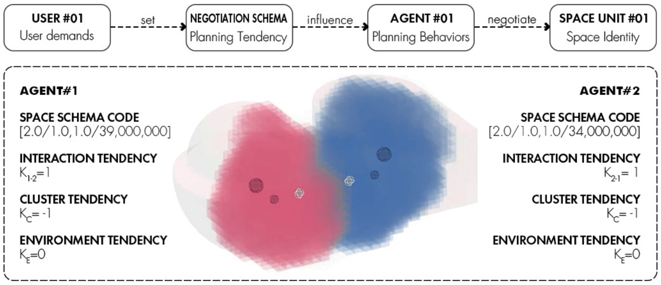
6



7



8



9

type have a consistent relationship with the parts and degrees of freedom in the robotic material system. Color represents a "pheromone", distinguishing an association of agents occupying the node with an initial value of [0,0,0] for unoccupied nodes (Figure 5).

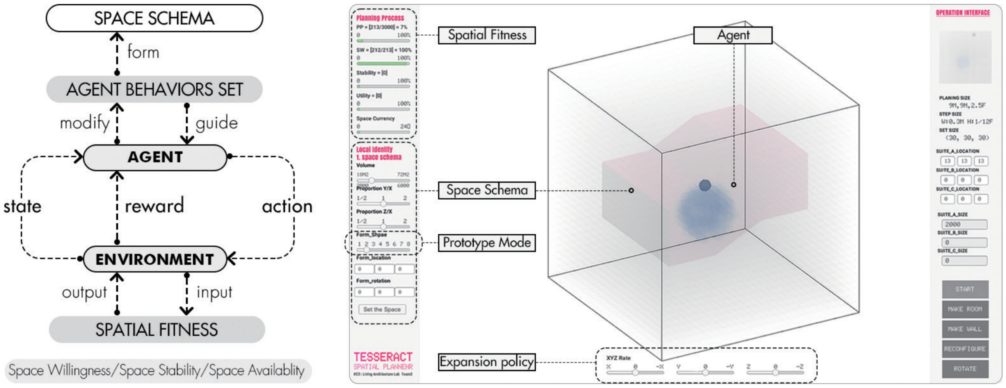
Each spatial agent contains a collection of nodes which form a volume. Its basic parameters include position, color, capacity, diffusion rate, and cohesion rate. Position is the agent's center of mass. Color represents agent type as a degree of association to other agents. Capacity is the maximum territory the agent will contain. Diffusion rate is the speed the agent passes its pheromone through the environment. Cohesion rate is the speed of balancing force of the agent boundary toward its center of mass. The agent expands the three-dimensional territory it occupies from its starting location by adding or releasing adjacent nodes until reaching its capacity in a state of dynamic equilibrium with its neighbors, thereby gradually forming an internally closed volume (Figure 6).

Behaviors defining how the agent moves and adapts its territory are dictated by adjustments to internal parameters in relation to neighboring agents and its environment. To achieve a mapping from user space planning goals to the agents' space planning behavior, we introduce three "Schema" as collections of properties of the agent that influence changes in its behavior in response to design objectives: Relational Schema, Space Schema, and Negotiation Schema.

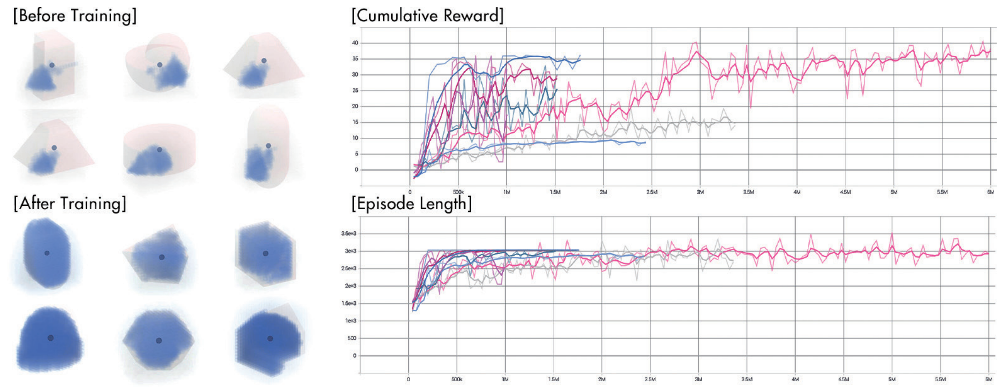
Relational schema defines an agent's degree of relationality or association to other agents through an RGB value. The degree of similarity of each agent's color to other agents forms a closer or more distant association.

Space schema contains parameters that influence the agent's behavior related to the local space it generates and adjusts. These include "Volume", "Proportion", and "Form", which respectively determine the size of the space volume, the aspect ratio of the space bounding box, and the space morphology type (Figure 7).

Negotiation Schema parameters influence its behavior related to negotiating space with its neighbors and environment, including three parameters described as "tendencies." Interaction Tendency represents the degree of resistance or attraction along the boundary with a neighbor. For each adjacent agent, it is expressed as a weight K_{I-A} in the range of [-2,2], applied as a repulsive or attractive force according to the value of K_{I-A}. Cluster Tendency defines the relationship between the agent and spatial clusters together with the Relational Schema. The weight K_C in [-1,1] dominates the agent's tendency to groups with similar colors. Environment Tendency quantifies the agent's demand in relation to environmental resources such as natural light. The set K_E aggregates the agent's reaction to site information,



10



11

6 The discrete planning environment and the metadata in each node.

7 The space planning process.

8 Space Schema, principle and example.

9 Negotiation Schema, principle and example.

10 The principle of reinforcement learning and the training model.

11 Reinforcement learning results, left: comparison of behaviors, right: graphs (pink one success) with the cumulative reward and the episode length.

making agents exhibit specific behaviors such as phototaxis or light avoidance (Figure 8).

The spatial planner is designed to autonomously negotiate multiple users' goals and environmental factors. To manage the high dimensionality of this problem, we leverage Reinforcement Learning with Self-Play to train a neural network to learn adaptive decision-making strategies for the agents (Sutton and Barto 2018). Each agent learns a behavioral policy to continuously maintain a close mapping between its space schema goal parameters and the generated volume while negotiating its territory with other agents through the negotiation schema. Training begins with 600 groups of {V, P, F} random combinations within a limited range as the initial dataset. Adjustments in stigmergic parameters control the agent's three-dimensional growth direction and intensity based on the centroid position and a policy {x, y, z} (Figure 9). Observations include the space schema parameters, the agent's current position, the last occupied node, and its stigmergic parameters. A real-time reward is given when each captured point is inside the target shape, and a penalty is added when it is outside. A staged reward is set to double the real-time reward when the proportion of qualified points (defined as "space fitness") increases beyond a threshold (30%, 60%). When the ratio reaches 90%, the task is considered complete, and a set of global rewards from additional analysis such as "structural stability" and "space availability" are given. After 6*106 episodes, the training curve typically reaches a state of relative stability (Figure 10).

Semi-autonomous Robotic Material System

To enable semi-autonomous reconfiguration of building parts in direct relation, a novel voxel-based robotic material system was developed with a structured environment operating in the same voxel grid as the spatial planner. Rather than using a gantry system for locomotion, the robots are designed symbiotically with passive blocks to slide over the dynamically changing host structure while reconfiguring it through a system of tracks, dynamic knobs acting as gates and switches, and locking and unlocking mechanisms. This enables simple collaborative robots with low numbers of degrees of freedom to efficiently adapt to the spatial assembly.

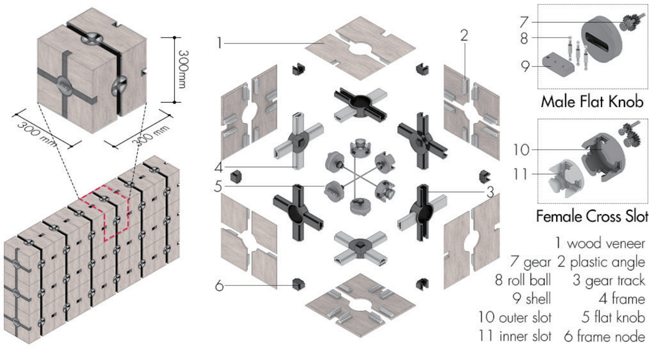
Modular passive parts are cubic with a side length of 0.3m. Each passive part is divided into passive actuation, structural, and panel components. Passive actuation components enable relative sliding and locking, equipped with female cross slots on one side of each axis and male flat knobs on the opposite. Structural components form a frame with tracks for gears in the knobs. Panel components provide the architectural infill for the assemblies (Figure 11).

Modular robots are the same size as passive parts with similar fabrication details, but robotic structural components are filleted to avoid collisions during movement (Figure 12). Physical prototypes are manufactured with 3D printing at a scale of 1:1 (Figure 14). Two servomotors are installed behind each knob for driving and steering. The body of one robot consists of three linked cubic parts forming an L-shaped combination with an action mode for sliding its cubic parts along tracks in its body or sliding itself along tracks in static parts. Robots can slide, change direction, push and pull, and lock and unlock through mutual collaboration for various reconfiguration tasks. In Figure 13, for example, the robot on the top can connect to a passive part, drive the motor and slide it down along the tracks provided by the adjacent passive parts, pushing the object one side length into place.

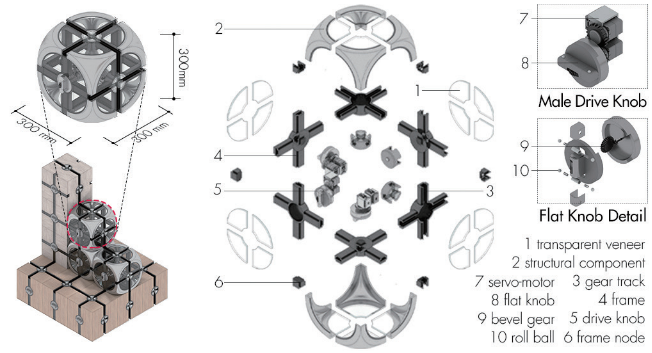
The robotic control system is setup with the principle of simultaneous state alignment. Simple "relative" robots cooperate with a relative localization strategy synchronized through bidirectional communication between the simulation environment and physical environment. The simulation environment is built with Unity3D, a game engine with a comprehensive multi-physics module and high extendibility (reference to Unity). Assemblies of robots and passive blocks are designed as a digital twin with encoded actions computed through multi-physics constraints and inverse kinematics directly related to the actions of the physical system. The actions developed in the simulator include: 1) sliding on static parts 2) changing direction 3) sliding segments of their body 4) locking/unlocking passive blocks 5) pushing/pulling, and 6) carrying passive blocks (Figure 15 and 16).

Communication is established wirelessly with a local network through a UDP protocol between our robotic simulator loaded on a PC and Raspberry Pi microcomputers mounted to the physical robots. Each Raspberry Pi is loaded with custom-built control software developed in Python. Custom commands are streamed back and forth as packets through a wireless port, converted to python functions on the Raspberry Pi and C# functions in the PC simulator. Dynamixel AX-12A servo motors were installed on the robotic knobs with high precision angle control with data feedback including angle position, load, and speed, enabling semi-realtime cyber-physical state alignment. The simulator hosts the server and connects the physical modules as clients. Robotic actions in the simulator are calculated with inverse kinematic functions, mapped to precise motor speeds and rotations, and sent wirelessly as instruction sequences to the Raspberry Pi to drive the servos on the physical robots. Each motor takes 0.196s to rotate 60° in 10V voltage. When the robot slides one unit length (300mm), the bearing set rotates 3.18 circles in 3.75s. Sensors in the servos then collect physical state data (torque, load, speed, and temperature) and send it back to the simulator as observations from the physical environment.

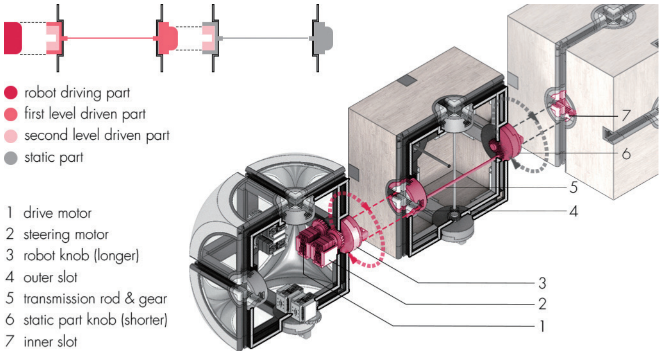
Assembly constraints are introduced through the rotating



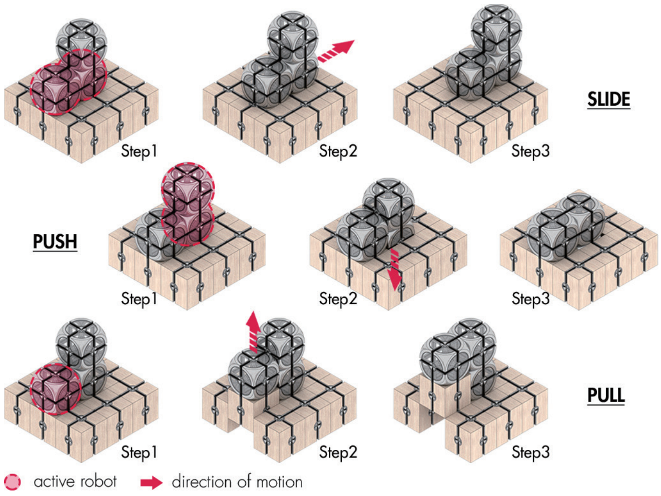
12



13



14



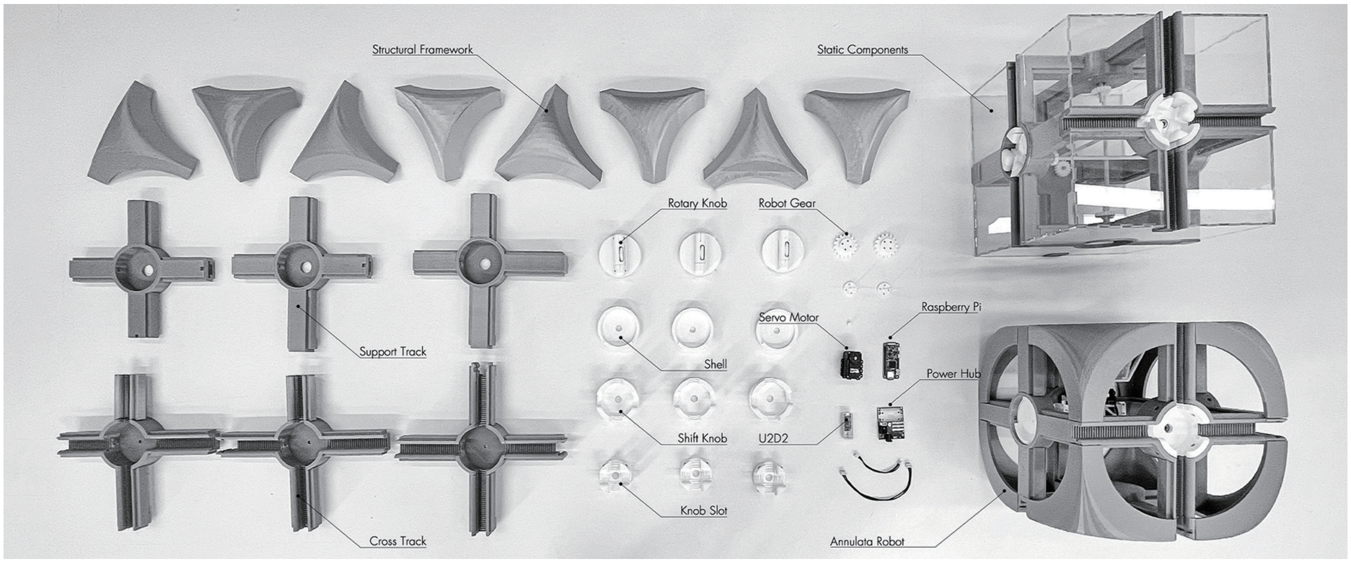
15

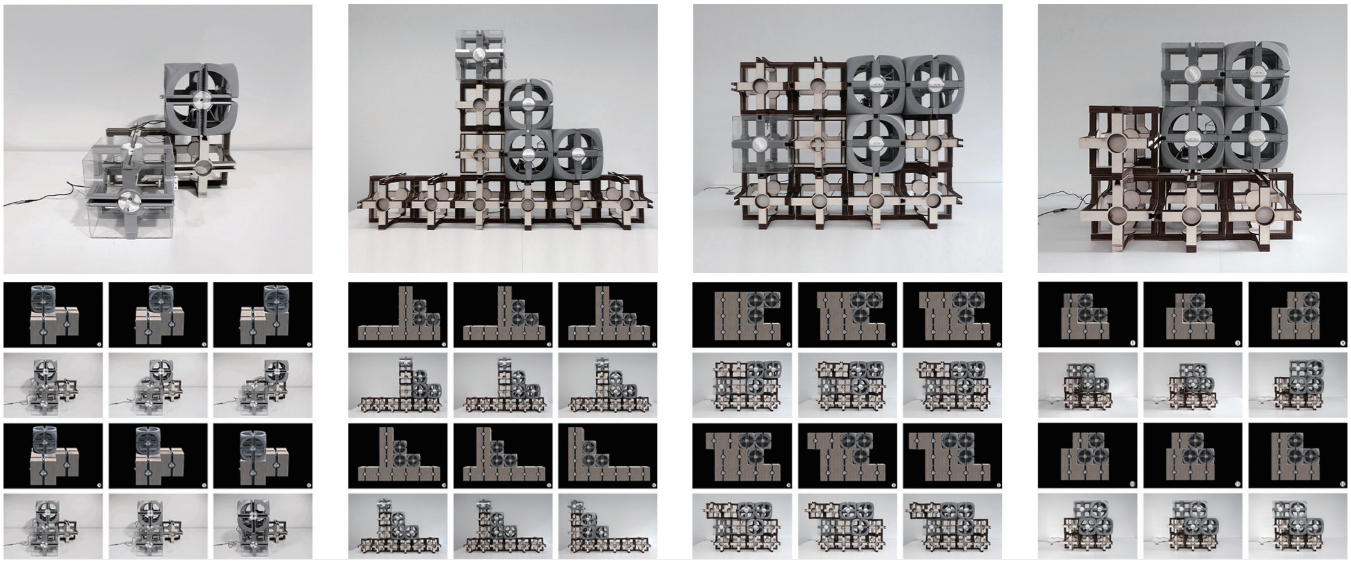
12 Static parts and the construction details.

13 Distributed robots and the construction details.

14 Joint details for lock and unlock.

15 Robot actions.





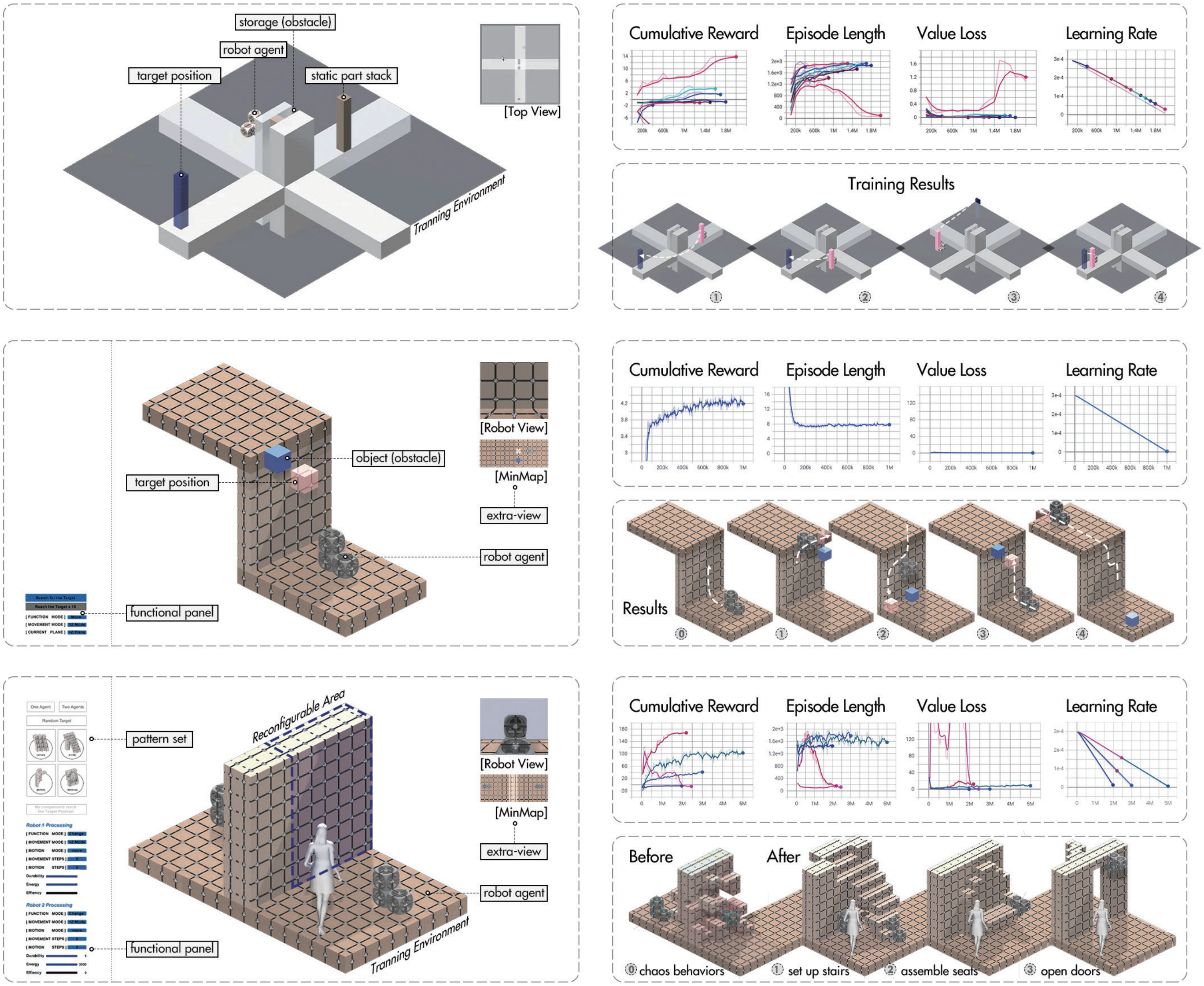
16 The component list and physical prototypes.

17 The robotic actions: 1) sliding on static parts and changing direction 2) sliding segments of their three-part body 3) pushing/pulling passive blocks, and 4) carrying passive blocks reinforcement learning with self-play for collaborative robotic behavior

male-female connections in the passive parts while constraints of locomotion and actuation are introduced through the robots three-part body and male-female knob and groove system. While pathfinding algorithms such as A* (Hart 1968) and Dijkstra's algorithm (Dijkstra 1959) have proven to be efficient for solving simple path planning problems through weighted graph traversal, we introduced deep reinforcement learning in our simulator to formulate adaptive strategies involving collaboration between multiple robots coordinating the biased and constrained behaviors in dynamically changing structured environments. The simulator was setup with L-shaped robots composed of three independent embodied agents taking observations of the current state of each of the six faces of each part. A curriculum learning strategy was developed as three stages of increasing difficulty:

Stage 1 is carried out in an environment over a 2D plane with a range of 8m*8m. The goal is to transport the static parts from an initial position to a target position. Observations include the robot sensors and positions of the agent, starting point, and target point. If the robots or the static parts fall out of the boundaries of the environment, the agents receive a penalty of -1. The reward is set to be inversely proportional to the distance from the target, and 10 extra points are added when the target position is reached (Figure 17).

Stage 2 is performed in a Z-shaped 3D environment composed of 22*6 static parts to train the agents to reach a designated position while avoiding obstacles. The position of the obstacle is added to the observations. When agents collide with an obstacle, they receive a penalty of -1, and the other rewards



18 The training environment and training results of Stage1, Stage2, and Stage3.

remain the same (Figure 18).

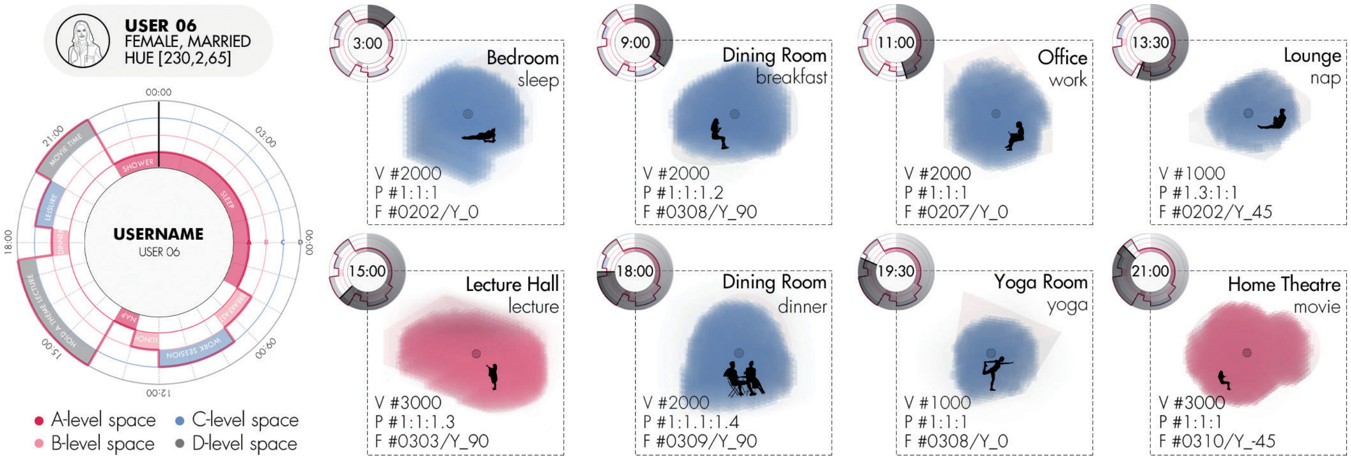
Stage 3 aims to train two robotic agents to collaborate in assembling passive parts into 3D design goals. The training uses four spatial forms (bookshelf, stairs, seat, and door opening) as target goals. Observations are the robot positions, sensor activation, and all static parts' current and target positions. In addition to the rewards and penalties mentioned above, the energy consumption of the robots and the accuracy of the result in relation to the target goal are included as rewards (Figure 19).

RESULTS AND REFLECTION

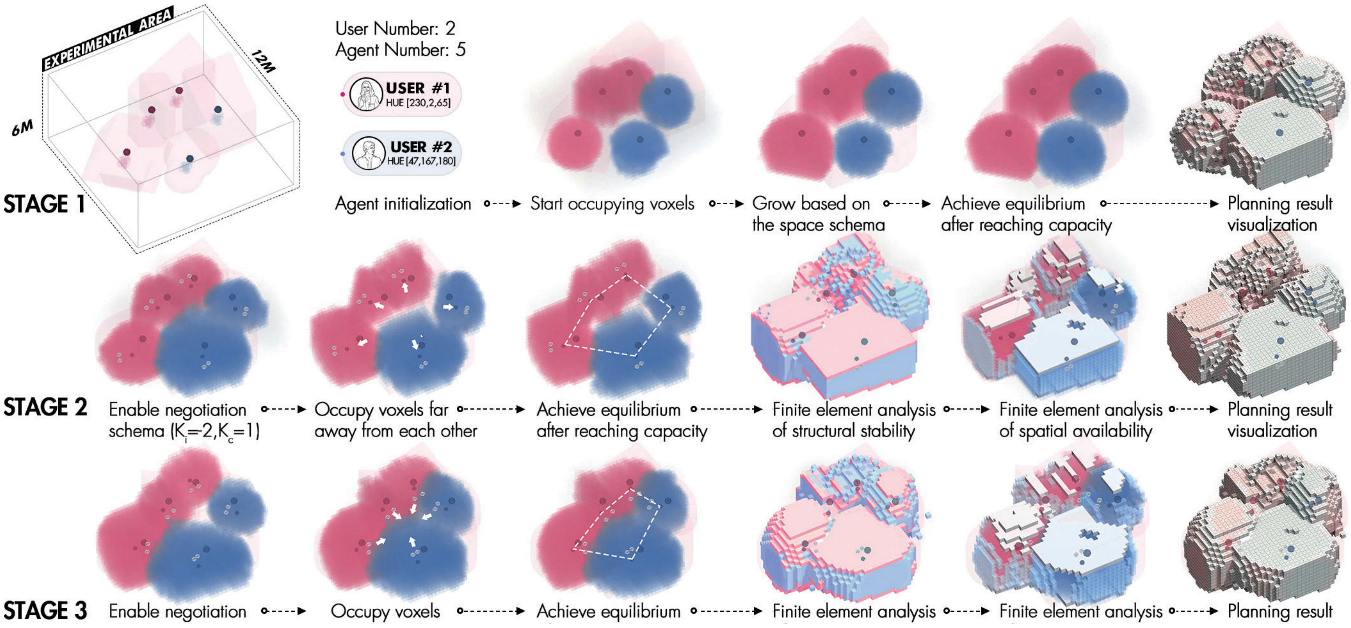
Implementation of this methodology was tested through a case study project for a community called TESSERACT, which proposes a new paradigm of distributed living and working by providing spaces that continuously adapt and reconfigure in near real-time according to the changing requirements of the shared community of users. Within the context of the global Covid-19 pandemic, this enables people to enjoy a variety of activities in a tiny scope of life.

The interactive platform was developed with interfaces for the user to input their preferences. A "user hue" is extracted from three aspects, including personal characteristics, lifestyle, and interpersonal relationships. A "user code" is generated by entering the personal schedules, interaction needs, and spatial and environmental preferences. These are processed as inputs for each agent's relational, spatial, and negotiation schema.

The algorithm was tested through a series of experiments. Individual agents were successfully trained through curriculum learning with to achieve policies first for space schema goals for volume, proportion, and shape only (Figure 10, Figure 20), and next to achieve space schema goals while adapting through relational and negotiation parameters with multiple agents.



19


EXPERIMENTAL AREA
12M
6M
STAGE 1
User Number: 2
Agent Number: 5
USER #1
HUE [230,2,65]
USER #2
HUE [47,167,180]
Agent initialization
Start occupying voxels
Grow based on the space schema
Achieve equilibrium after reaching capacity
Planning result visualization
STAGE 2
Enable negotiation schema (K$_i$=-2,K$_c$=1)
Occupy voxels far away from each other
Achieve equilibrium after reaching capacity
Finite element analysis of structural stability
Finite element analysis of spatial availability
Planning result visualization
STAGE 3
Enable negotiation
Occupy voxels
Achieve equilibrium
Finite element analysis
Finite element analysis
Planning result

20

19 Single-user continuous space planning experiments.

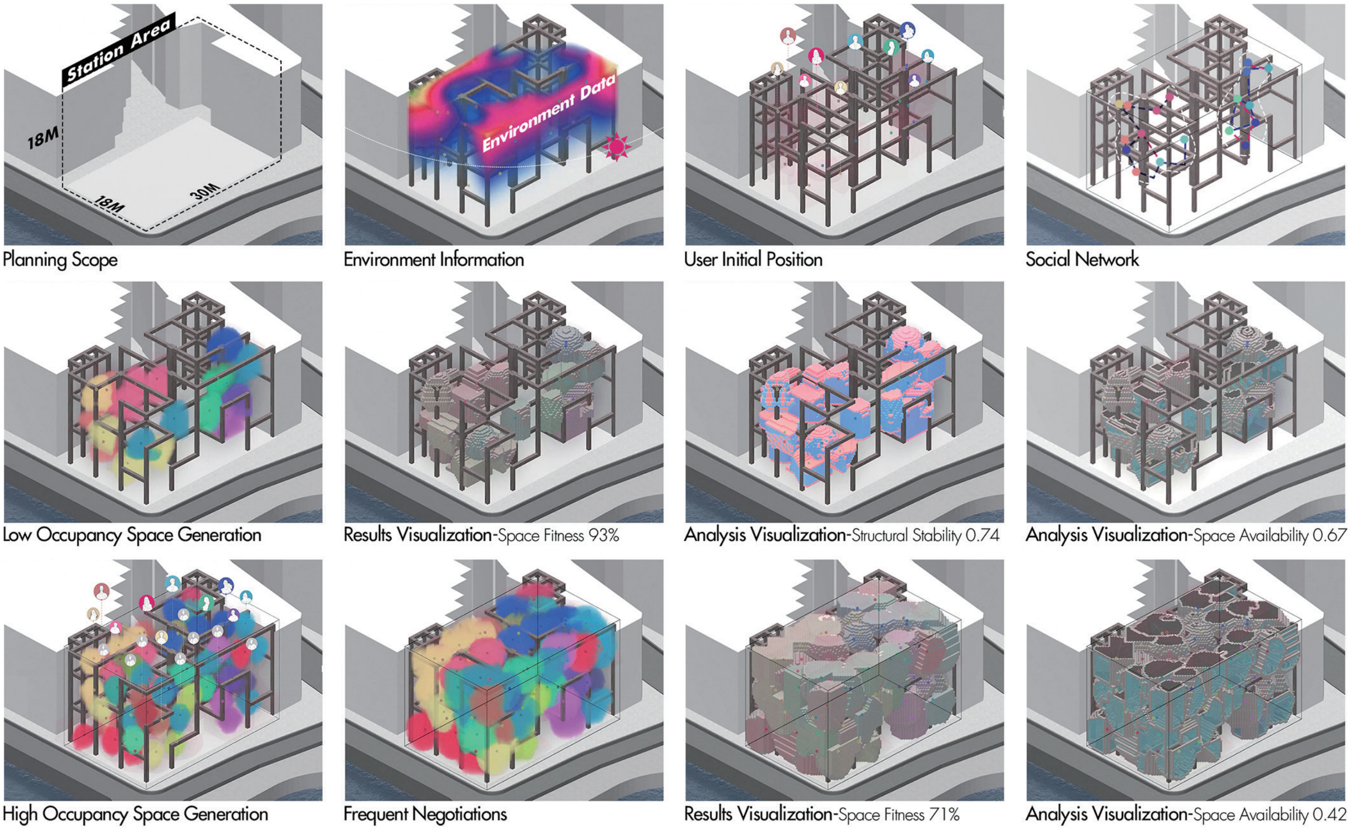
20 Multi-user continuous space negotiation experiments.

The proportion of overlap between the resulting territory and the target territory was able to reach more than 85% typically after 6*106 episodes. In further experiments, we input dynamically changing user requirements and the algorithm proved successful in autonomously adjusting its behavioural patterns accurately (Figure 21).

The trained model was tested in a 30m*18m*18m site volume. Environmental data and constraints including fixed structure, daylight intensity, and view analysis were loaded as maps that agents responded to. User codes were auto-generated as virtual user demand parameters to initialize various types of agents, forming a community social network. We modelled a 50% occupancy rate resulting in the agents easily achieving spatial goals at over 90% accuracy with lots of open space and little negotiation behaviours. Next, we tested a 100% occupancy rate with much more negotiation behaviours required which achieved approximately 70% spatial accuracy (Figure 22).

Finally, the robotic material system was iteratively developed through a series of prototype configurations directly controlled by the virtual simulator which successfully demonstrated simple reconfiguration sequences combining sliding, locking/unlocking, and pushing/pulling behaviours (Figure 14,15,16). The system has been successfully tested to convert agent boundaries as voxels directly linked with the agent simulator sending instruction sequences to multiple robots while receiving sensor data back wirelessly. Self-play reinforcement learning was tested with two simulated robots to coordinate reconfiguration of a wall into a series of goals, successfully improving from random outcomes to efficient sequences closely matching the goals (Figure 19).

CONCLUSION



Planning Scope

Environment Information

User Initial Position

Social Network

Low Occupancy Space Generation

Results Visualization-Space Fitness 93%

Analysis Visualization-Structural Stability 0.74

Analysis Visualization-Space Availability 0.67

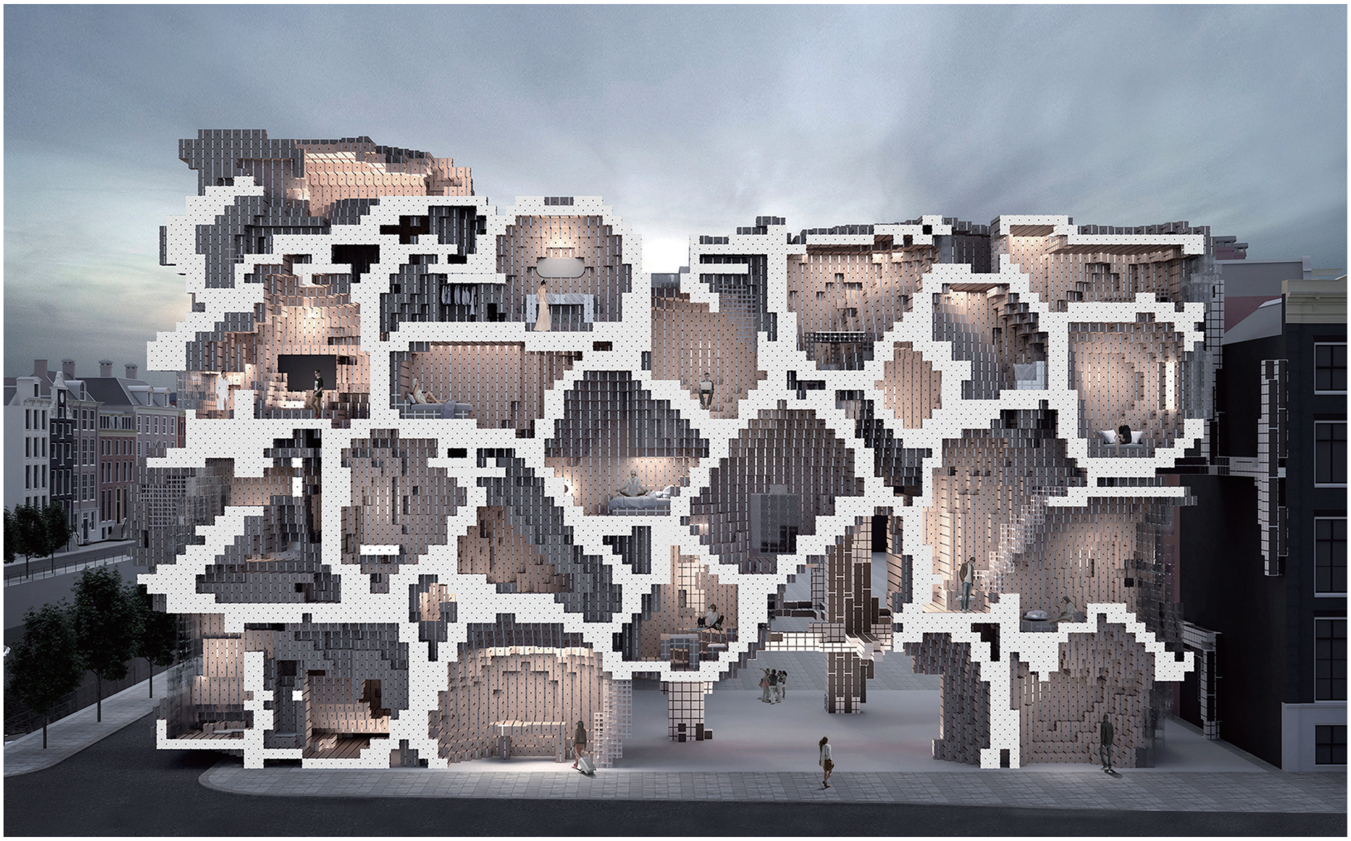
High Occupancy Space Generation

Frequent Negotiations

Results Visualization-Space Fitness 71%

Analysis Visualization-Space Availability 0.42

21



22

21 The space generation of project TESSERACT in different occupancy rate.

22 The section of project TESSERACT at a certain stage.

IRAAS shows enormous potential for negotiating the changing spatial requirements of multiple users in a dynamic environment by learning to adapt itself in near real time. Rather than simply automating known construction patterns, this system begins to leverage the potential for intelligent robotic architecture. Through the agile development of the three components of our system and interrelational communication protocols between them, we have successfully demonstrated version one of a semi-autonomous adaptive system.

While our physical prototypes successfully demonstrate simple reconfiguration behaviours, scaling up the system for building construction poses the next challenge requiring a more robust material and locking system along with stronger mechatronics. In future work, we intend to add sensing technology to the robotic system to increase its degree of self-awareness. Additionally, we are developing more sophisticated spatial adaptation behaviours with reinforcement learning with a focus on multi-agent collaboration, user feedback as a fitness criteria, and integration of structural stability prediction analysis. This model has the potential to disrupt and reorganizes the connection between architecture, humans, and the environment, defining a new paradigm for a self-regulating living environment that continuously adapts itself in a dialogue with its users.

ACKNOWLEDGEMENTS

The project TESSERACT is conducted in UCL, The Bartlett AD Research Cluster 3 (RC3), Living Architecture Lab.
Team members: Jiaqi Wang, Wanzhu Jiang, Ying Lin, and Zongliang Yu.
Tutors: Tyson Hosmer, Octavian Gheorghiu, Philipp Siedler, Ziming He, Panagiotis Tigas, Baris Erdincer

REFERENCES

Adel, Arash, Andreas Thoma, Matthias Helmreich, Fabio Gramazio, and Matthias Kohler. 2018. "Design of Robotically Fabricated Timber Frame Structures." In ACADIA 2018: Recalibration: On Imprecision and Infidelity [Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)], Mexico City, Mexico, 18-20 October 2018, edited by P. Anzalone, M. del Signore, and A. J. Wit, 394-403. CUMINCAD.

Anderson, Carl, Carlo Bailey, Andrew Heumann, and Daniel Davis. 2018. "Augmented Space Planning: Using Procedural Generation to Automate Desk Layouts." International journal of architectural computing 16(2): 164-177.

Bier, Henriette, Alexander Liu Cheng, Sina Mostafavi, Ana Anton, and Serban Bodea. 2018. "Robotic Building as Integration of Design-to-Robotic-Production and -Operation." In Robotic Building, edited H. Bier, 97-119. Cham, Switzerland: Springer.

23 The rendering of project TESSERACT.

Bier, Henriette and Sina Mostafavi. 2018. "Robotic Building as Physically Built Robotic Environments and Robotically Supported Building Process." In Robotic Building, edited H. Bier, 253-271. Cham, Switzerland: Springer

Chaillou, Stanislas. 2019. AI + Architecture: Towards a New Approach. Master's thesis. Harvard School of Design.

Chaillou, Stanislas. 2020. "ArchiGAN: Artificial Intelligence x Architecture." Architectural Intelligence. Springer, Singapore. 117-127.

Chaillou, Stanislas. Archigan: a generative stack for apartment building design, https://developer.nvidia.com/blog/archigan-gen-era-tive-stack-apartment-building-design/ 2019/7.

Dijkstra, E.W. 1959. "A Note on Two Problems in Connexion with Graphs." Numerische Mathematik 1(1): 269-271.

Dinçer, Ahmet Emre, Hakan Tong, and G. ÇAGDAS. 2014. "A computational model for mass customized housing design bu using cellular automata." A/Z ITU Journal of the Faculty of Architecture 11(2): 351-368.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. 2014. "Generative Adversarial Networks". arXiv:1406.2661.

Frazer, John. 1995. An Evolutionary Architecture. London: Architectural

Association, 1995.

Hanna, Sean. 2020. “Architecture as Agent.” In Architecture as Agent. Georgia Institute of Technology School of Architecture.

Hart, Peter E, Nils J Nilsson, and Bertram Raphael. 1968. “A Formal Basis for the Heuristic Determination of Minimum Cost Paths.” IEEE transactions on systems science and cybernetics 4 (2): 100-107.

Hosmer, Tyson, and Panagiotis Tigas. 2019. “Deep Reinforcement Learning for Autonomous Robotic Tensegrity (ART).” In ACADIA 2019: Ubiquity and Autonomy. Proceedings of the 39th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), ed. K. Bieg, D. Briscoe, and C. Odom. 16-29. Austin, TX: ACADIA.

Hosmer, Tyson, Panagiotis Tigas, David Reeves, and Ziming He. 2020. “Spatial Assembly with Self-Play Reinforcement Learning.” In ACADIA 2020: Distributed Proximities. Proceedings of the 40th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), ed. B. Slocum, V. Ago, S. Doyle, A. Marcus, M. Yablonina and M. del Campo. 382-393. Online + Global: ACADIA.

Hua, Hao. 2017. “A Bi-Directional Procedural Model for Architectural Design: A Bi-Directional Procedural Model.” Computer graphics forum 36(8): 219-231.

Huang, Weixin, and Zheng Hao. 2018. “Architectural drawings recognition and generation through machine learning.” In ACADIA 2018: Recalibration: On Imprecision and Infidelity. Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), ed. P. Anzalone, M. del Signore and A. J. Wit. 156-165. Mexico City: ACADIA.

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros. 2017. “Image-to-Image Translation with Conditional Adversarial Networks”. CVPR. arXiv:1611.07004.

Jenett, Ben, and Kenneth Cheung. 2017. "Bill-e: Robotic platform for locomotion and manipulation of lightweight space structures." In 25th AIAA/AHS Adaptive Structures Conference, p. 1876.

Kilian, Axel. 2018. “The Flexing Room Architectural Robot.” In ACADIA 2018: Recalibration: On Imprecision and Infidelity. Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), ed. P. Anzalone, M. del Signore and A. J. Wit. 232-241. Mexico City: ACADIA.

Landau, Royston. 1968. New Directions in British Architecture (New York,: G. Braziller, 1968), 11.

Leder, Samuel, Ramon Weber, Dylan Wood, Oliver Bucklin, and Achim Menges. 2019. “Distributed robotic timber construction.” In ACADIA 2019: Ubiquity and Autonomy. Proceedings of the 39th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), ed. K. Bieg, D. Briscoe, and C. Odom. 510-519. Austin, TX: ACADIA.

Liggett, Robin S. 2000. “Automated facilities layout: past, present and future.” Automation in construction 9(2): 197-215.

Maierhofer Mathias, Soana Valentina, Yablonina Maria, Suzuki Seiichi, Koerner Axel, Knippers Jan, Menges Achim. 2019. “Self-choreographing network: towards cyberphysical design and operation processes of adaptive and interactive bending-active systems.” In ACADIA 2019: Ubiquity and Autonomy. Proceedings of the 39th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), ed. K. Bieg, D. Briscoe, and C. Odom. 654-663. Austin, TX: ACADIA.

McCarthy, John, Marvin L Minsky, Nathaniel Rochester, and Claude E Shannon. 2006. “A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence: August 31, 1955.” The AI magazine 27(4): 12-.

Meyboom, AnnaLisa, and Reeves Dave. 2013. “Stigmergic Space.”

Nagy, Danil, Lau Damon, Locke John, Stoddart James, Villaggi Lorenzo, Wang Ray, Zhao Dale, and Benjamin David. 2017. Project Discover: An Application of Generative Design for Architectural Space Planning. 10.22360/simaud.2017.simaud.007.

Negroponte, Nicholas. 1970. The Architecture Machine. Cambridge, MA: MIT Press.

Oosterhuis, Kas, et al. 2012. Hyperbody: First Decade of Interactive Architecture. Heijningen, the Netherlands: Jap Sam.Pask, Gordon. 1969. “The Architectural Relevance of Cybernetics.”Architectural Design 39: 494-496.

Pask, Gordon. 1965. Comments on an indeterminacy that characterizes a self-organising system. Cybernetics of Neural Processes, Consiglio Nazionale delle Richerche (1965):1-30.

Pask, Gordon. 1969. The Architectural Relevance of Cybernetics. Architectural Design 7(6): 494-6.

Petersen, Kirstin Hagelskjaer, Radhika Nagpal, and Justin K. Werfel. 2011. “Termes: An autonomous robotic system for three-dimensional collective construction.” In Robotics: science and systems VII, ed. D. Hugh, N. Roy, P. Abbeel, H. Bai, D. Hsu, M. J., W. S. Lee, S. Bhattacharya, M. Likhachev, and V. Kumar. The MIT Press. 257-264.

Petersen, Kirstin, and Radhika Nagpal. 2017. “Complex Design by Simple Robots: A Collective Embodied Intelligence Approach to Construction.” Architectural design 87(4): 44-49.

Rubenstein, Michael, Christian Ahler, and Radhika Nagpal. 2012. “Kilobot: A Low Cost Scalable Robot System for Collective Behaviors.” In 2012 IEEE International Conference on Robotics and Automation, 3293-3298.

Sutton, Richard S, and Andrew G Barto. 2018. Reinforcement Learning: An Introduction. The MIT Press.

Veloso, Pedro, Rhee Jinmo and Krishnamurti Ramesh. 2019. “Multi-agent Space Planning A literature review (2008-2017).”

Veloso, Pedro, and Ramesh Krishnamurti. 2020. “An Academy of Spatial Agents-Generating spatial configurations with deep reinforcement learning.” In Proceedings of the 38nd eCAADe Conference, 191-200. Berlin: eCAADe.

Wiener, Norbert. 1948. Cybernetics: Or Control and Communication in the Animal and the Machine. Cambridge, Massachusetts: MIT Press.

Wright Steenson, Molly. 2010. Artificial Intelligence, Architectural Intelligence: The Computer in Architecture, 1960-80. Ph.D. diss. Proposal, Princeton University.

IMAGE CREDITS

All drawings and images by the authors.

Tyson Hosmer is an architect, researcher, and software developer working at the intersection of design, computation, AI, and robotics. He is the Director of the Architectural Design masters program and a Lecturer at the Bartlett School of Architecture in London, where he directs the Living Architecture Lab (RC3). He is a Senior Associate Researcher with Zaha Hadid Architects, leading grant-funded research development of cognitive agent-based technologies and machine learning for design. His 15 years of experience in practice include working in Asymptote Architecture, Kokkugia, AXI:OME, and serving as Research Director with Cecil Balmond Studio for over six years. Tyson was previously a Course Tutor with the AADRL for seven years and has been a visiting professor in several institutions internationally.

Jiaqi Wang is now a PhD student and researcher at the South China University of Technology, focusing on artificial intelligence aided architectural design, generative design, and robotics. He got his master’s degree from UCL in Bartlett BPro RC3, guided by Tyson Hosmer.

Wanzhu Jiang is now a PhD student and researcher at the South China University of Technology, whose research direction is artificial intelligence and autonomous architecture. She got her master’s degree from UCL in Bartlett BPro RC3 in 2021, supervised by Tyson Hosmer.

Ziming He is a designer and a software developer focusing on computational design, generative design, architectural visualization, and robotics. He is currently working as a lead designer in Zaha Hadid Architects and is a technical tutor with RC3, Bartlett School of Architecture, UCL.degree from UCL in Bartlett BPro RC3, guided by Tyson Hosmer.