Using Interactive Evolution to Design Behaviors for Non-deterministic Self-organized Construction

Mary Katherine Heinrich[1], Payam Zahadat[2], John Harding[3], Mads Nikolaj Brandt[4], Emil Fabritius Buchwald[4], Leonardo Castaman[4], Kit Wai Chan[4], Malte Harrig[4], Stian Vestly Holte[4], Lynn Kieffer[4], Jens Lee Jørgensen[4], Chao-Yuan Liang[4], Erik Lima Skjæveland[4], Sofie Lund Michaelsen[4], Viktoria Millentrup[4], Suna Petersen[4], Martin Petrov[4], Enrico Pontello[4], Fabian Puller[4], Gabriella Rossi[4], Johanne Marie Skalle[4], Sofia Stylianou[4], Alma Bangsgaard Svendsen[4], James Walker[4], Paul Nicholas[1] and Phil Ayres[1]

[1]Centre for IT and Architecture, Royal Danish Academy, Copenhagen, Denmark, mhei@kadk.dk
[2]Artificial Life Lab, University of Graz, Austria, payam.zahadat@uni-graz.at
[3]University of The West of England (UWE), Bristol, UK, John3.Harding@uwe.ac.uk
[4]School of Architecture, Royal Danish Academy, Copenhagen, Denmark

ABSTRACT
Self-organizing construction is an emerging subdomain for on-site construction robots. This not only presents new challenges for robotics, but due to the stochasticity involved in such systems, impacts the modeling and prediction of resulting built structures. Self-organizing models have been explored by architects for generative design and for optimization, but so far have infrequently been studied in the context of construction. Here we present a strategy for architects to design with non-deterministic self-organizing behaviors, using interactive evolution to incorporate user judgment. We introduce our "Integrated Growth Projection" method, having implemented it into a software pipeline for early phase design. We test the software with an initial user group of architects, to see whether the method and pipeline helps them design a non-deterministic self-organizing behavior. The user group creates several hybrid controllers that reliably solve their chosen design tasks.

Author Keywords
Swarm construction; self-organization; artificial growth; artificial selection; interactive evolution.

ACM Classification Keywords
I.6 SIMULATION AND MODELING; I.2 ARTIFICIAL INTELLIGENCE; J.6.1 Computer-aided design (CAD)

1 INTRODUCTION

On-site construction robotics is a broad field of increasing importance for AEC industries [22, 35]. A significant subdomain of that field, presenting unique hardware and control challenges, is autonomous mobile robots for construction [1]. For mobile robots generally, a commonly investigated approach to control is self-organization—e.g., in Swarm Engineering [17]. These systems usually incorporate stochasticity (i.e., non-determinism), to establish needed features like robustness in real-world setups [17]. As a type of on-site construction process, non-deterministic behaviors are quite antithesis to the way AEC projects are designed, communicated, and constructed. As the drive for automation of on-site construction strengthens, it is advantageous to look to swarm robotic systems of construction for certain types of tasks [16], in addition to robotic systems with centralized control. If swarm robotic construction systems are employed, it introduces new challenges to the process of early phase architectural design. Architectural design processes have usefully incorporated non-deterministic self-organizing behaviors as means of generative design [24] and performance optimization [8], but they are less explored when applied to construction processes. In this paper, we look at how the design process of architects could be supported, if designing architecture that will be built by a non-deterministic self-organizing construction process. Specifically, we introduce the method of *Integrated Growth Projection*, to facilitate architects' use of interactive evolution in the design of behaviors for self-organized construction or artificial growth.

1.1 Self-organizing Behaviors for Construction

Hardware in self-organized construction often follows one of two approaches: either 1) using climbing or reconfigurable modular robots to comprise the structure itself (e.g., [16]), or 2) using mobile robots to aggregate transportable material into a mechanical structure (e.g., [33]). For all hardware approaches, the development of controllers to govern the robot swarm remains an open challenge [32]. Control types include, for instance, generalized control to reliably construct any arbitrarily specified artifact (e.g., [33, 36]), as well as

SimAUD 2018 June 04-07 Delft, The Netherlands

control to build a specific type of structure (e.g., [13]) or solve a specific task (e.g., [31]). Across types, controllers are advantaged by incorporating stochasticity (cf. robustness [17]). In purpose-specific controllers, a prominent feature of non-deterministic behavior is likely to be high variability in the resultant built artifact.

There is limited existing work studying how architectural design processes are impacted if artifacts are built by self-organized construction. In such cases where construction is steered by a sensed environment, architects would not be designing a specific artifact, but would rather be steering low-level behaviors of material aggregation to meet their high-level design objectives [12]. For early phase architectural design with self-organized construction, related work includes swarm agents evolved for massing inspiration [30]; incorporation of structural analysis feedback [27, 28]; robot body impacts on collective behavior [20]; and a BIM pipeline for swarm aggregation on a master surface [7].

1.2 Open-ended Tasks in Evolution

Evolutionary Algorithms (EA) for single- and multi-objective optimization have been used heavily in architectural and structural design, to guide design towards given performance criteria [34]. Although multi-objective optimization facilitates user input through the balancing of performance weightings [4, 3], it still assumes fitness functions can be defined for all prioritized design goals. Evolving non-dominated sets is not necessarily suited to the type of unrestricted tasks present in open-ended exploration and early phase architectural design.

Other evolutionary computation methods have been developed to target more open-ended tasks or creative tasks. They include Genetic Programming [18], where the program is evolved instead of the parameters, increasing flexibility; evolution of Compositional Pattern Producing Networks (CPPNs) [25], useful for creating organized groupings of material in open-ended tasks because they tend to produce features like symmetry and repetition [5]; Novelty Search [19], where a search algorithm looks for solutions that have not yet been seen instead of considering performance; and interactive evolution, where human judgment is at the forefront of the evolutionary selection process [26]. Relevant for early phase architectural design, interactive evolution in theory allows designs that may have poor measurable performance to thrive, without the designer necessarily making their reasoning explicit [26].

For creative tasks, interactive evolution has notably been used in combination with CPPNs, to evolve 2D images (PicBreeder [23]) and 3D shapes using voxels (Endless Forms [6]). In both of these projects, the problem of user fatigue was overcome by an accessible online interface allowing large, varied groups of individuals to collaborate on tasks of judgment and selection. This strategy may impact architectural design tasks differently, if the tasks require judgment informed by discipline-specific knowledge.

In architectural design research, evolution with CPPNs have been preliminarily explored for 2D and 3D drawing and structural engineering [29]. Techniques from Genetic Programming have been applied for massing studies in early phase design in combination with artificial selection [11, 10]. A plug-in for interactive evolution—Biomorpher[1]—has been developed for software used in architectural design, and is utilized in this paper.

2 METHOD

Early design phases for architecture typically incorporate both explicit performance criteria and subjective designer judgment. Here we investigate whether architectural designers can use interactive evolution to steer non-deterministic aggregation behaviors towards high-level design objectives. For this, we develop an approach called the *Integrated Growth Projection* and implement it into a software setup. In a workshop structure, we test whether architects can use it for specific design tasks, and whether the setup supports user understanding.

2.1 Software: Setup Overview

For the scope of this paper, we implement our software setup in Grasshopper3D,[2] using IronPython[3] and C# to create user objects. The software setup and example files are available in the repository.[4] Within our setup, we use the existing Grasshopper3D plug-in *Biomorpher* for interactive evolution, and implement changes[5] to support the specific use-case of designing self-organizing behaviors. We also use the artificial growth model Vascular Morphogenesis Controller [38], which we implement into Grasshopper3D to be used by architects[6]. We select it for its features of environmentally responsive artificial growth. These two elements, one for interactive evolution and one for non-deterministic self-organized control, are combined through our *Integrated Growth Projection* method.

Vascular Morphogenesis Controller

We use a self-organizing controller from the literature—the Vascular Morphogenesis Controller [38]—that is designed to generatively grow artificial structures. We modify this controller for use in an architectural design application, and implement it into a Grasshopper3D component.

The Vascular Morphogenesis Controller (VMC) is a limited-resource controller inspired by the branching mechanisms of plants, and is encoded as an acyclic directed graph (i.e., tree), as shown in Figure 1. It uses behavior features such as competition and balancing to grow in response to environmental conditions. The VMC grows in a self-organizing way, meaning that there is no centralized point of decision making, and uses both positive and negative feedback. The edge vertices of the graph (i.e., leaves) sense the environment and send

[1] `https://github.com/johnharding/Biomorpher`

[2] Grasshopper3D is a Visual Programming Language (VPL) platform for use in the CAD package Rhinoceros3D (`https://www.rhino3d.com/download/grasshopper/1.0/wip/rc`).

[3] `http://ironpython.net`

[4] Download the software setup: `https://github.com/florarobotica/IGP-for-Grasshopper/releases`

[5] Biomorpher updates contributed by author JH.

[6] VMC implementation into IronPython for Grasshopper contributed by authors MKH and PZ.

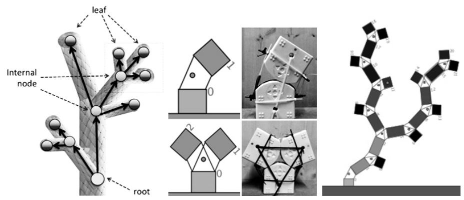


Figure 1. The Vascular Morphogenesis Controller [38] is a graph-based self-organizing controller for artificial growth. (Left) The free edge vertices of the graph (leaves) are possible points of new growth; the root is the vertex where growth was initiated; the dynamic pathways of the graph connect the internal vertices (internal nodes) and host local communication between them, enabling positive and negative feedback. (Center) In an example one-to-one interpretation of the VMC to physical structure [38], each possible graph edge becomes a robot module, resulting in tree-like structures (right).

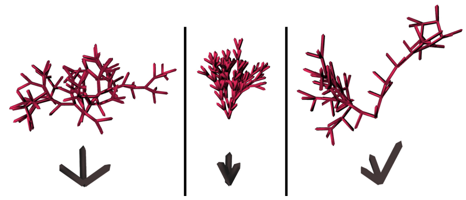


Figure 2. Examples of user-defined branch sets (below), with example resulting structures (above), when used in a non-deterministic 3D VMC setup. The example branch sets are (left) symmetrical with wide proportions, (center) symmetrical with narrow proportions, and (right) uneven. Uneven branch sets tend to create morphologies that have directional bias, forming curved primary axes.

information to their immediate neighbors. This information is sent along dynamic pathways through the graph structure, eventually reaching the root position. Likewise, the root position is aware of resource limitations, and sends information to its neighbors, which then continues through the graph. The dynamic pathways of the graph modify themselves according to bidirectional information they receive locally. Each leaf of the graph makes decisions about whether to grow, based on the information sensed and received. The VMC is able to conduct its behavior deterministically, but can also incorporate stochasticity into many steps of the decision-making process. Stochasticity can help the VMC to more efficiently explore certain environments, and to avoid getting stuck in false optimums.

In our software implementation of the VMC[7], we include the following of its possible parameters [38], to influence the controller's dynamics: 1) *growth style* (probabilistic, randomly chosen from best, or deterministically chosen from best); 2) growth amount or probability; 3) *resource* constant; 4) environment-dependent *success* constant; 5) transfer rate *limit* constant; 6) *competition* constant; and 7) speed of *adaptation*. The parameters can be manually defined, or linked to an evolutionary algorithm. Growth is initiated at arbitrary user-defined root locations on a ground plane, in any quantity.

In a two-dimensional growth plane, the usability of the VMC has been demonstrated in the literature, for tasks such as growing tall in a harsh environment [38] and navigating a maze [37]. Here, we extend the VMC to encompass growth in three-dimensions. In the 3D VMC, each decision to grow results in three new branches from the respective leaf (i.e., edge vertex). Three branches provides the minimum information needed to sense an advantageous direction in a 3D environment. Each new set of branches is oriented around the axis of its parent branch. The relationship of each individual new branch to the parent is user-defined, according to 1) the distance of the new leaf from its parent leaf, and 2) the new branch's angle of inclination from its parent branch.

Not only do new branch sets vary in width-to-height ratio, but they can be radially symmetrical or can be uneven. The user-defined shape of new branch sets can dramatically impact the overall morphology of a 3D structure grown by VMC, as seen in Figure 2. Our implementation of a parametrized 3D VMC controller is combined with Biomorpher through the *Integrated Growth Projection* setup.

Biomorpher

A plug-in for interactive evolution (IE) in Grasshopper3D, Biomorpher[8] uses a version of a Cluster-Oriented Genetic Algorithm (COGA) [21]. COGAs encompass a two-step process, first using a diverse search algorithm to rapidly explore the design space, and second, adaptively filtering the population in order to present the user with concise, digestible clusters of solutions. At each generation, Biomorpher allows the user to choose either evolution by artificial selection, or optimization according to performance objectives. These can be combined into multi-mode optimization, by alternating the two modes between generations. COGAs normally filter and cluster solutions according to fitness value, helping the user to choose high performing regions for refinement [2]. When Biomorpher optimizes according to performance objectives, it uses a version of this approach.

Within the scope of this paper, we employ exclusively Biomorpher's artificial selection mode. When Biomorpher evolves with artificial selection, it uses k-means clustering [15] to group solutions according to parameter state similarity. For each of 12 defined clusters, a representative solution is presented to the user.[9] The user selects one or more of these representatives to define parents for the next generation of evolution. When using artificial selection, each cluster representative is accompanied by an indicator of performance for supplied criteria, allowing the user to incorporate quantitative feedback into their judgment and selection.

In order to support the specific use-case of the *Integrated Growth Projection* described below, we implement a modification to Biomorpher[10], enabling the visualization of user-defined mesh colors in the selection preview windows.

2.2 Software: Integrated Growth Projection

The *Integrated Growth Projection* setup is comprised of two major features to support architectural design of self-organizing behaviors. The first deals with extending the user

[7] VMC implementation into IronPython for Grasshopper contributed by authors MKH and PZ.

[8] `https://github.com/johnharding/Biomorpher`

[9] See an illustrative video demonstration: `https://www.youtube.com/watch?v=EM6uoXW7Yeo`

[10] Biomorpher updates contributed by author JH.

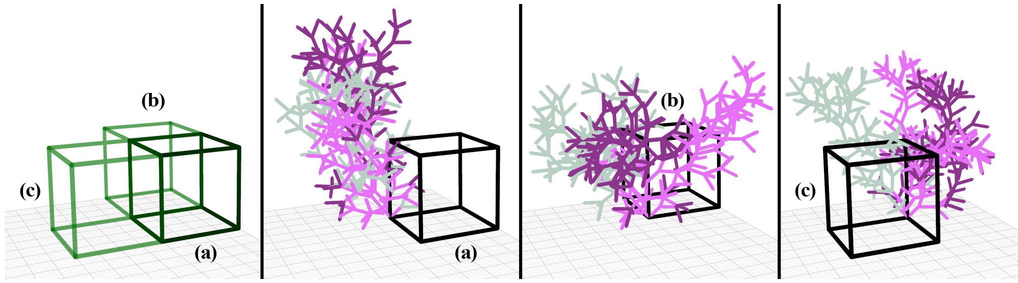


Figure 3. Visualization of the environment to which the controller is currently responding during growth. Three example environments are shown here, for the task of obstacle avoidance. Each environment (a,b, and c) contains a box-shaped obstacle in a different location (all three obstacles shown, left). At the right and center, the VMC growth can be seen responding to each respective environment. Also seen here is simultaneous viewing of multiple possible growth results (VMC structures), differentiated by color.

interface of interactive evolution (in this case, in Biomorpher) to support the user's understanding of non-deterministic self-organizing behaviors—and therefore support their ability to make informed judgments during artificial selection. The second deals with the interpretation of a self-organized controller's logic (in this instance, the VMC's graph topology) into a physical structure in simulation. We illustrate our description of the *Integrated Growth Projection* setup with the example design task of a behavior for obstacle avoidance.

User Interface for Self-organization in IE

In Biomorpher, the parameter states are supplied separately from the mesh geometry that will display in the respective preview window for user selection. As such, the *Integrated Growth Projection* setup can control what the user will see during selection.

The *Integrated Growth Projection* provides two IE user interface functions: 1) simultaneous viewing of multiple possible results, and 2) visualization of the environment.

In the user interfaces of prominent IE projects for creative production [23, 6], the user is endeavoring to evolve a static image or 3D shape. In these cases, the visualization of evolved solutions is straightforward—each preview window shows the image or shape created by the respective parameter state. In our case, however, visualization is less straightforward. Because non-deterministic behaviors will generate variability and unpredictability in resultant structures, any given parameter state might produce a range of results. Every time a non-deterministic controller runs, it will produce a different result. The degree to which the results can vary will depend on the behavioral properties of that individual controller. In the *Integrated Growth Projection* setup, we therefore display multiple possible results simultaneously in each IE preview window (see Figures 5, 3). Each displayed result is given a separate color so the user can differentiate. The number of results shown at a time is user-defined (here, three solutions are shown). This interface feature performs several functions. First, it promotes user understanding of the inher-

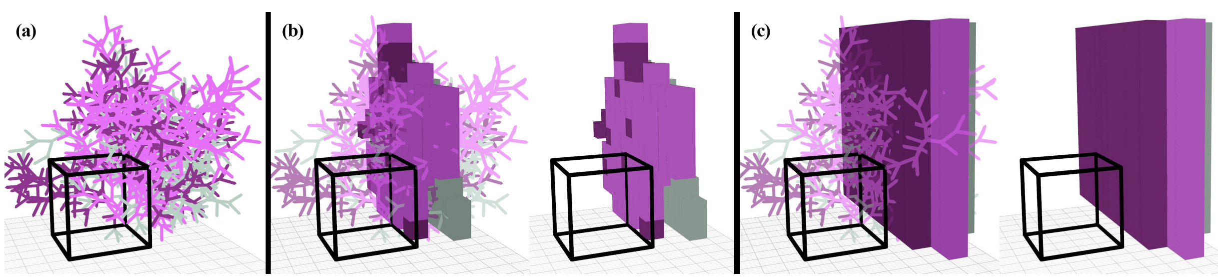


Figure 4. (a) VMC structure without a second layer of control for material aggregation, (b,c) the same VMC output, with its respective structure when incorporating simple rule-based control for material aggregation, with either (b) control for placing bricks, or (c) control for placing a solid wall.

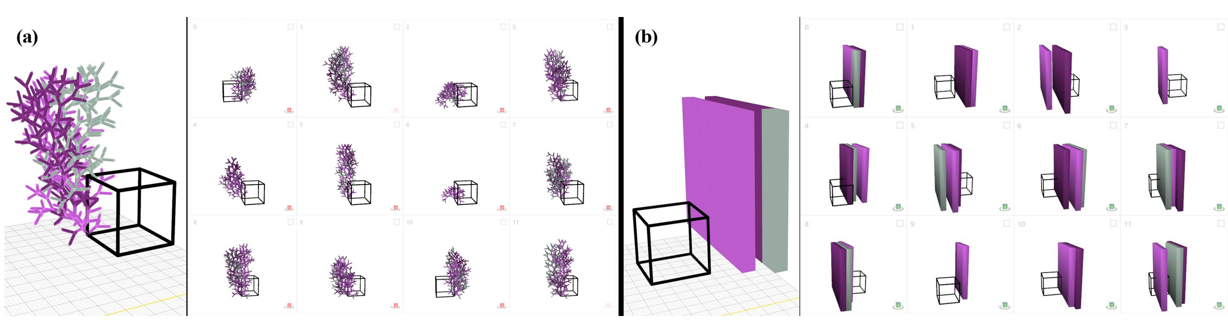


Figure 5. (a) Direct representations of graph-based structures grown by VMC; a generation of such growths in the Biomorpher IE preview windows. (b) Solid wall structures defined by hybridizing the VMC with rule-based control for material aggregation; a generation of such structures in Biomorpher.

ent variability present in non-deterministic self-organization. Second, showing the user a range of possible results helps them avoid the selection of a parameter state that solves the task by coincidence, as opposed to solving it reliably through a feature of behavior. Third, seeing a range of results allows the user to evolve not only the behavior itself, but the degree of variability in that behavior's results (see Figure 5).

Simultaneous viewing of multiple results in the IE preview window is accomplished by consecutive[11] independent simulations of controller behavior, for each parameter state queried by the EA.

When designing a behavior that grows a structure in response to the environment, a visualization of the respective environment is evidently useful. In our case, showing the environment is intended to support the user in making informed judgments about the behavior or response, rather than only the artifact. Importantly, the inclusion of an environment visualization in the IE preview window also allows the environment to be randomized at each generation without confusing the user. In our *Integrated Growth Projection* setup, at each query of a parameter state, the environment data used in simulation of the controller's behavior is randomized from a predefined list or external simulation (see Figures 3, 5). Randomizing the environment at each generation helps the user avoid getting stuck in false optimums by evolving towards a controller that can coincidentally solve the task only in one environment, as opposed to solving it reliably in any environment of the relevant type.

The environment in which the structure grows in simulation is represented in by an *Environment Data File*, containing a 3D matrix of the values that would be obtained by sensors in a matching reality setup. The wireframe box shown here (see Figures 3, 4, 5) does not represent the process by which the controller responds to environment in simulation, but rather is a visualization tool for the user to understand the current environment.

From Control Logic to Physical Structure

When interpreting the VMC into a material structure, the literature focuses on direct relationships between the physical and the encoding, where each edge and vertex of the VMC graph has a one-to-one relationship with a physical element in the structure (see Figure 1). This interpretation style restricts the possible material structures to tree-shapes, shown in the literature for robot modules [38] and for tubular braid [9].

In our *Integrated Growth Projection*, we extend to flexibility in physical structure by decoupling the control of material aggregation from the control logic that discovers environmentally advantageous locations for growth. A second layer of control for material aggregation allows for the distribution of heterogeneous tasks, and therefore the creation of hybrid control. Such hybrid control can provide the flexibility typically sought in early phase architectural design.

[11] In the software setup available for download (**https://github.com/florarobotica/IGP-for-Grasshopper/releases**), this portion is implemented in the Anemone plug-in (**http://www.food4rhino.com/app/anemone**), for user accessibility without programming.

When being hybridized with an existing self-organizing controller (in this case, the VMC) control for material aggregation can itself also be self-organizing, or can instead be simply rule-based. Here we illustrate with rule-based control for placement of basic building elements: bricks or walls (see Figure 4). In this scenario the VMC can be used to sense and avoid the environmental obstacle *in situ*, thereby solving the specified task reliably through features of its behavior, while the second layer of control—for material aggregation—allows the architect to have flexibility in the design of an artifact, rather than being restricted to tree-shaped structures.

After applying hybridized control, the final resulting material aggregations can be viewed and evolved in the IE setup (see Figure 5), making use of the previously described features of simultaneous results and environment visualization.

2.3 User Tests

The software setup was provided to architectural designers and engineers, during a one-week workshop including tutorials and group project work. Each group used the software setup to design a controller for a distinct design task. After the workshop's end, the participants received a survey about their experiences with the software setup and relevant concepts.[12]

User-made Design Projects

After tutorials on concepts and software, each group of participants chose a scope for their project. They chose either a building component to shape (e.g., doorway, column) or an environmental condition to respond to (e.g., thermal, occupant circulation). Within the selected scope, each group defined a design task and worked to create a controller that could reliably solve that task in simulation, in at least three different environments.

Survey of User Experience

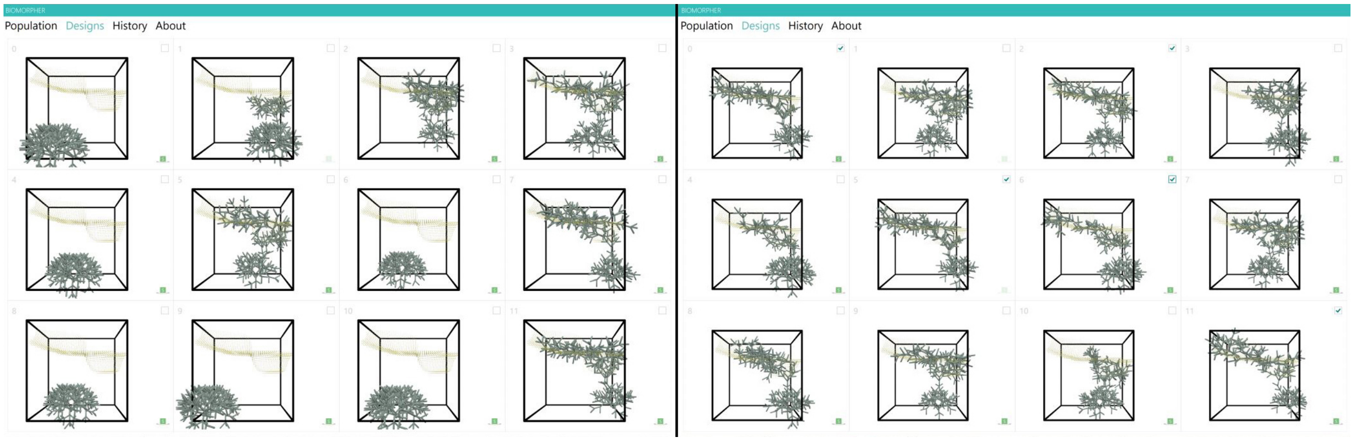
Participants were asked about 1) their prior experience, 2) their understanding of topics before and after the workshop, 3) the helpfulness of specific software aspects for their understanding and their project work, and 4) their likelihood to use specific software aspects in the future. Survey questions, responses, analysis, and plots can be viewed in the supplemental data set [14].

3 WORKSHOP RESULTS

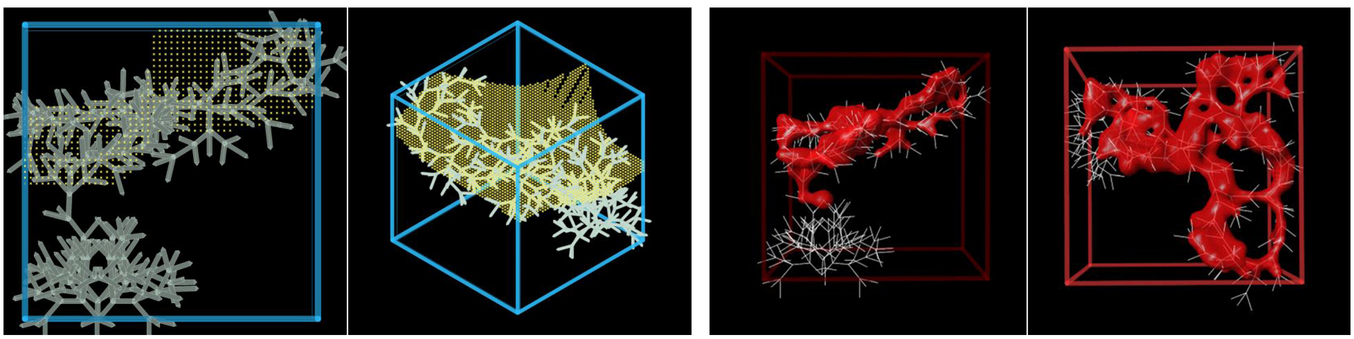
Two groups chose building components to shape—a bridge, a staircase—and three groups chose an environmental condition to respond to—sunlight, wind, rainfall.

Four out of five groups were able to successfully use the *Integrated Growth Projection* setup to design a non-deterministic self-organizing behavior to reliably construct artifacts that solved their chosen design task (see Figures 6, 7). All four of those groups used the full set of software aspects provided, including interactive evolution and the addition of their own layer of control for material aggregation, hybridizing with the VMC. As well, the second layer of control designed by each

[12] The workshop participants gave survey responses anonymously. Authors of this paper who were workshop participants had no contact with the process of survey preparation, analysis or plotting of its results, or writing of relevant descriptions.



(a) Screenshots of the interactive evolution process, using the VMC and *Integrated Growth Projection* setup. Left, an early, random generation. Right, a generation where the behavior results shown seem to indicate convergence on an area of parameter states that reliably solve the design task at hand.



(b) Designers' visualizations, output from the final controllers designed in the *Integrated Growth Projection* setup. Left, the VMC locating the area of light conditions where the group's chosen design task requires a the placement of a barrier. Right, the structure resulting from the group's hybridization of the VMC with a second self-organizing controller for design-driven amorphous material aggregation.

Figure 6. Project results from the group focused on response to sunlight.

group incorporated some aspect of self-organizing behavior in the decision-making process for material aggregation. The sunlight group (see Figure 6) was furthermore able to evolve the parameters of the 3D VMC to reliably find a specific desired feature in multiple example environments, then using the second layer of their own control to aggregate material in a design-driven way. The bridge group (Figure 7, left) used decentralized decision-making to find topology features in the VMC graph structure that were advantageous as initiation points for their material aggregation. The wind group (Figure 7, center) responded to feedback from the sensed environment not only with the VMC's behavior, but in the behavior of their own swarm agent controller for material aggregation. The water collection group (Figure 7, right) used their material aggregation control to find and build off of useful shape features in the volume of the VMC's growth, rather than simply refilling the zone defined by the VMC with a different type of structure. This group was able to usefully distinguish between tasks for the VMC and their own control rules, such that both are necessary for their hybridized controller to be successful at its task.

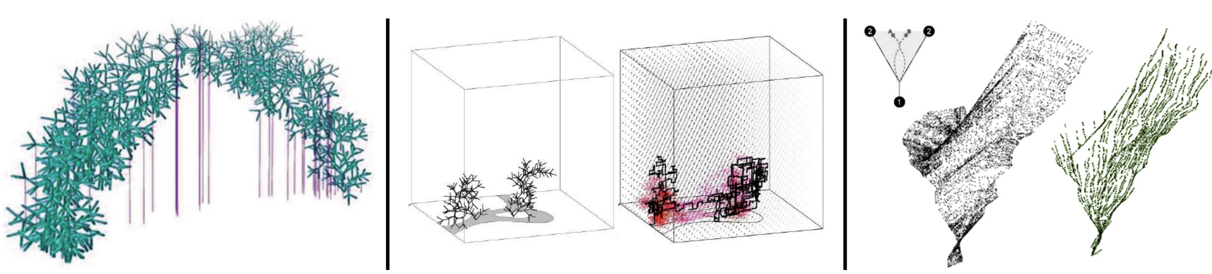


Figure 7. Designers' visualizations of their final hybrid controller outputs for (left) a bridge, (center) shielding a sidewalk from adverse wind, and (right) forming a basin to collect rainwater at an advantageous position.

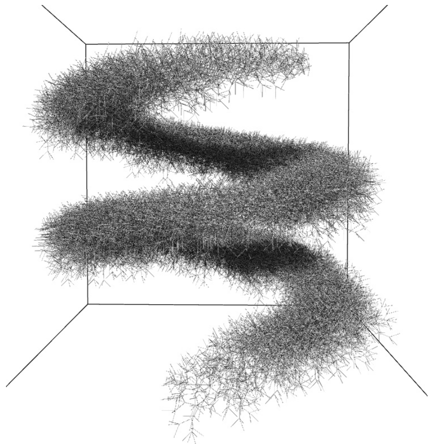


Figure 8. Designers' visualization of their staircase, output from their extended VMC implementation.

The staircase group did not incorporate separate control for material aggregation, but after the workshop instead took the approach of extending the VMC's behavior to solve their design task. In this group's approach, once an instance of the VMC finishes its growth—according to its parameters and the values sensed in the environment—their extension globally selects a leaf of the resultant graph structure to become the root location for a new VMC instance. The leaf that becomes a new root is selected according to globally assessed features of overall height and proximity to desired staircase shape. Although the group completes this process of new root selection through global control, a decentralized decision-making process could potentially be implemented to achieve a similar result, thereby substantially broadening the types of structures able to be grown with VMC.

The survey responses (see supplemental data set [14]) indicate that no workshop participant had previously designed a self-organizing controller using interactive evolution, so the project results and survey results (see [14]) give strong evidence that the *Integrated Growth Projection* setup helped the participants to understand this design approach enough to use it to solve their chosen design tasks.

4 DISCUSSION

Though the design projects resulting from the workshop are strong in their use of the VMC to find advantageous zones in the environment, their development of a second controller for material aggregation is very limited. Though the *Integrated Growth Projection* method is open-ended enough to be highly receptive to plausible construction elements (see walls and bricks in Figures 4 and 5), it was a difficult task for the workshop participants to develop controllers to place such elements, instead tending towards amorphous material (see red structure in Figure 6(b)). In order for such controllers to be useful for construction in reality, the control of material aggregation would need to be much more developed, especially in terms of the process by which materials are moved into place. This itself is a significant open challenge in self-organized construction, with the state-of-the-art notably being set by the TERMES project [33].

5 CONCLUSION

A method—the *Integrated Growth Projection*—has been introduced, to facilitate the use of interactive evolution as a tool in the open challenge of designing architecture built by non-deterministic self-organized construction. The method has been implemented in a software pipeline accessible to architects. The software has been tested by an initial user group in a workshop setting. The workshop results provide strong evidence for the usefulness of the method and pipeline in helping architects to both understand and design behaviors for self-organizing (swarm) construction.

ACKNOWLEDGMENTS

Author MKH led the paper writing and associated workshop, and developed the method and software. PZ made substantial contributions to the method, paper and workshop, and contributed to repackaging the VMC software. JH contributed to the paper and workshop, and implemented the updates to the Biomorpher software. PN and PA supervised the paper and workshop. All other authors are listed alphabetically; they were participants in the workshop, and contributed the design projects described in the paper.

The authors would like to thank Sebastian Gatz for his teaching assistance during the workshop, and Hannah Arkell, Emil Ballegaard, Harry Clover, Kalle Ladekjær Jacobsen and Joanna Maria Lesna for their participation in the workshop.

Project '*flora robotica*' has received funding from the European Union's Horizon 2020 research and innovation program under the FET grant agreement, no. 640959.

REFERENCES

1. Ardiny, H., Witwicki, S., and Mondada, F. Are autonomous mobile robots able to take over construction? a review. *Int. Journal of Robotics, Theory and Applications 4*, 3 (2015), 10–21.
2. Bonham, C. R., and Parmee, I. C. An investigation of exploration and exploitation within cluster oriented genetic algorithms (COGAs). In *Proc. of the 1st GECCO Conf.* (1999).
3. Camilo Cruz, Michael Kirley, J. K. An interactive approach for evolving pareto optimal architectural form. In *Proc. of SimAUD* (2017), 35–44.
4. Chaszar, A., von Buelow, P., and Turrin, M. Multivariate interactive visualization of data in generative design. In *Proc. of SimAUD* (2016).
5. Cheney, N., MacCurdy, R., Clune, J., and Lipson, H. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. In *Proc. of the 15th GECCO Conf.*, ACM (2013), 167–174.
6. Clune, J., and Lipson, H. Evolving 3d objects with a generative encoding inspired by developmental biology. *ACM SIGEVOlution 5*, 4 (2011), 2–12.
7. Edenhofer, S., Rädler, S., Hoß, M., and Von Mammen, S. Self-organised construction with revit. In *FAS*, IEEE Int. Workshops on* (2016), 160–161.

8. Evins, R. A review of computational optimisation methods applied to sustainable building design. *Renewable and Sustainable Energy Reviews 22* (2013), 230–245.

9. Hamann, H., Soorati, M. D., Heinrich, M. K., Hofstadler, D. N., Kuksin, I., Veenstra, F., Wahby, M., Nielsen, S. A., Risi, S., Skrzypczak, T., et al. Flora robotica–an architectural system combining living natural plants and distributed robots. *arXiv preprint arXiv:1709.04291* (2017).

10. Harding, J. Evolving parametric models using genetic programming with artificial selection. In *Proc. of the 34th eCAADe Conf.* (2016), 423–432.

11. Harding, J. E., and Shepherd, P. Meta-parametric design. *Design Studies* (2016).

12. Heinrich, M. K., and Ayres, P. Using the phase space to design complexity. In *Complexity & Simplicity* (2016).

13. Heinrich, M. K., Wahby, M., Soorati, M. D., Hofstadler, D. N., Zahadat, P., Ayres, P., Støy, K., and Hamann, H. Self-organized construction with continuous building material: Higher flexibility based on braided structures. In *FAS*, IEEE Int. Workshops on* (2016), 154–159.

14. Heinrich, M. K., Zahadat, P., Harding, J., Nicholas, P., and Ayres, P. Survey of participant experience in workshop for testing IGP software setup: supplemental dataset for SimAUD 2018 (Version v1). Zenodo, 2018. [Data set]. `http://doi.org/10.5281/zenodo.1196027`.

15. Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE trans. on pattern analysis and machine intelligence 24*, 7 (2002), 881–892.

16. Kassabian, P., Gumpertz, S., Menges, H. A., and Werfel, J. Towards force-aware robot collectives for on-site construction. In *Proc. of the 37th ACADIA Conf.* (2017).

17. Kazadi, S. T. *Swarm engineering*. PhD thesis, California Institute of Technology, 2000.

18. Koza, J. R. *Genetic programming: on the programming of computers by means of natural selection*, vol. 1. MIT press, 1992.

19. Lehman, J., and Stanley, K. O. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation 19*, 2 (2011), 189–223.

20. Pantazis, E., and Gerber, D. Emergent order through swarm fluctuations. In *Proc. of the 35th eCAADe Conf.* (2017).

21. Parmee, I. C. Cluster oriented genetic algorithms (cogas) for the identification of high performance regions of design spaces. In *First Int. Conf. EvCA*, vol. 96 (1996), 66–75.

22. Saidi, K. S., Bock, T., and Georgoulas, C. Robotics in construction. In *Springer handbook of robotics*. Springer, 2016, 1493–1520.

23. Secretan, J., Beato, N., D Ambrosio, D. B., Rodriguez, A., Campbell, A., and Stanley, K. O. Picbreeder: evolving pictures collaboratively online. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, ACM (2008), 1759–1768.

24. Singh, V., and Gu, N. Towards an integrated generative design framework. *Design Studies 33*, 2 (2012), 185–207.

25. Stanley, K. O., D'Ambrosio, D. B., and Gauci, J. A hypercube-based encoding for evolving large-scale neural networks. *Artificial life 15*, 2 (2009), 185–212.

26. Takagi, H. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proc. of the IEEE 89*, 9 (2001), 1275–1296.

27. Taron, J. M. Speculative structures: Reanimating latent structural intelligence in agent-based continuum structures. In *Proc. of the 30th eCAADe Conf.* (2012).

28. Tsiliakos, M. Swarm materiality: a multi-agent approach to stress driven material organization. In *Proc. of the 30th eCAADe conference* (2012).

29. Vierlinger, R. Towards ai drawing agents. In *Modelling Behaviour*. Springer, 2015, 357–369.

30. von Mammen, S., and Jacob, C. Evolutionary swarm design of architectural idea models. In *Proc. of the 10th GECCO Conf.*, ACM (2008), 143–150.

31. Wang, Z., and Schwager, M. Force-amplifying n-robot transport system (force-ants) for cooperative planar manipulation without communication. *The Int. Journal of Robotics Research 35*, 13 (2016), 1564–1586.

32. Wareham, T., and Vardy, A. Putting it together: the computational complexity of designing robot controllers and environments for distributed construction. *Swarm Intelligence* (2017), 1–18.

33. Werfel, J., Petersen, K., and Nagpal, R. Designing collective behavior in a termite-inspired robot construction team. *Science 343*, 6172 (2014), 754–758.

34. Wortmann, T., and Nannicini, G. Black-box optimization for architectural design. In *Proc. of CAADRIA* (2016), 177–186.

35. Yang, G.-Z. Digital architecture and robotic construction. *Science Robotics 2*, 5 (2017).

36. Yeom, K. Morphological approach for autonomous and adaptive system: The construction of three-dimensional artificial model based on self-reconfigurable modular agents. *Neurocomputing 148* (2015), 100–111.

37. Zahadat, P., Hofstadler, D. N., and Schmickl, T. Development of morphology based on resource distribution: Finding the shortest path in a maze by vascular morphogenesis controller. In *Proc. of the 14th ECAL Conf.* (2017), 428–429.

38. Zahadat, P., Hofstadler, D. N., and Schmickl, T. Vascular morphogenesis controller: a generative model for developing morphology of artificial structures. In *Proc. of GECCO Conf.*, ACM (2017), 163–170.