Design Tools and Workflows for Braided Structures

Petras Vestartas[1], Mary Katherine Heinrich[1], Mateusz Zwierzycki[1], David Andres Leon[1], Ashkan Cheheltan[1,2], Riccardo La Magna[3], and Phil Ayres[1]

[1] Centre for Information Technology and Architecture, The Royal Danish Academy of Fine Arts, School of Architecture, 1435 Copenhagen, Denmark
[2] Berlin University of the Arts, 10623 Berlin, Germany
[3] Konstruktives Entwerfen und Tragwerksplanung, Berlin University of the Arts, 10623 Berlin, Germany
petrasvestartas@gmail.com

Abstract. This paper presents design workflows for the representation, analysis and fabrication of braided structures. The workflows employ a braid pattern and simulation method which extends the state-of-the-art in the following ways: by supporting the braid design of both pre-determined target shapes and exploratory, generative, or evolved designs; by incorporating material and fabrication constraints generalised for both hand and machine; by providing a greater degree of design agency and supporting real-time modification of braid topologies.

The paper first introduces braid as a technique, stating the objectives and motivation for our exploration of braid within an architectural context and highlighting both the relevance of braid and current lack of suitable design modelling tools to support our approach. We briefly introduce the state-of-the-art in braid representation and present the characteristics and merits of our method, demonstrated though four example design and analysis workflows. The workflows frame specific aspects of enquiry for the ongoing research project *flora robotica*. These include modelling target geometries, automatically producing instructions for fabrication, conducting structural analysis, and supporting generative design. We then evaluate the performance and generalisability of the modelling against criteria of geometric similarity and simulation performance. To conclude the paper we discuss future developments of the work.

Keywords: Braided structures, Braid representation, Computational design.

1 Introduction - Context, Motivation and Objectives

Braid technique is based on a principle of oblique interlacing of three or more strands of yarn, filament or strapping [Rana & Fangueiro, 2015]. The technique is used to produce artifacts that are larger, stronger, more resilient and, often, more aesthetically charged than the original material. Braid offers tremendous versatility in terms of material, scale of artifact, and method of production (Fig. 1). As such, braid is applied across a broad range of industries, including medicine (stents), agriculture (industrial hoses) and leisure equipment (ropes). Braid is also used to produce preforms for advanced composite manufacture [Potluri et al., 2003]. The resulting composites are amongst the lightest yet strongest components currently produced and are utilised in high-performance arenas such as cycling, racing and aeronautics [Rana & Fangueiro, 2015]. Despite the versatility, resilience, and scalability demonstrated by braid across many industries and functional uses, it is not a commonly used or discussed technique within contemporary architecture. In parallel to the limited use of braid in architecture, there is also a lack of suitable design modelling tools for freely exploring complex braid topologies against architectural design, analysis, and fabrication considerations. In the work presented here, we concentrate on exploring hollow tubular braids (typically referred to as '2D braid' in industrial contexts [Ayranci & Carey, 2008]). Tubular braids offer a rich space of topological freedom, as well as structural potential when approaching architectural scale.

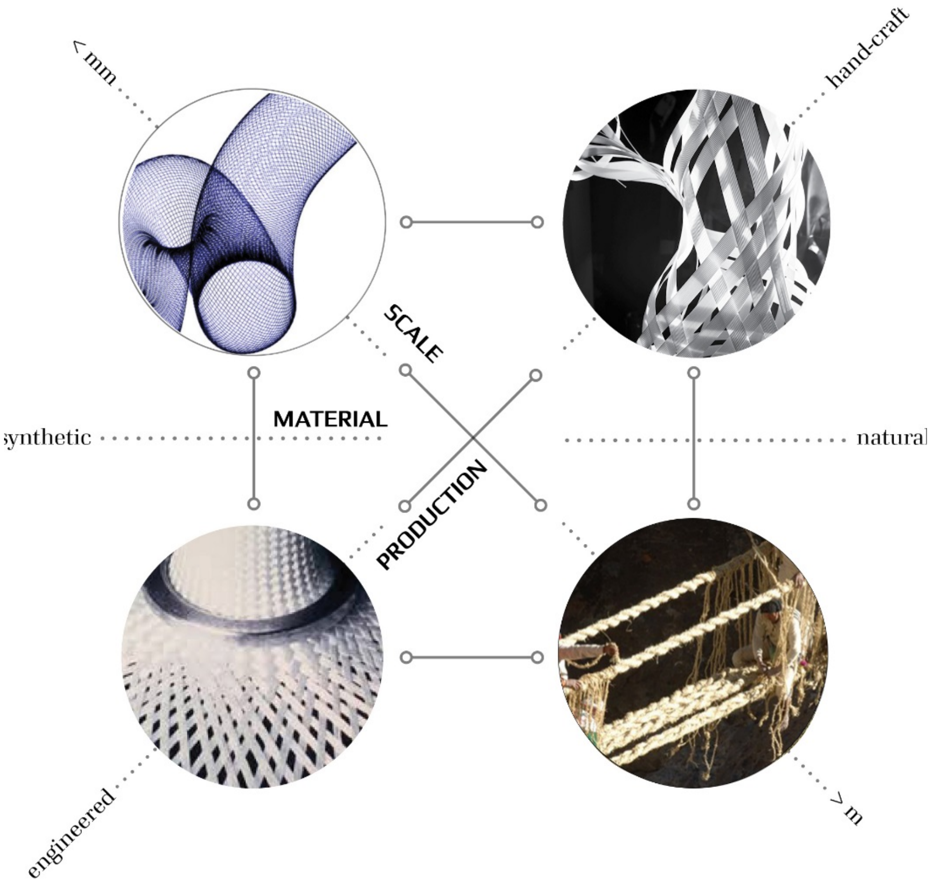

Fig. 1. The versatility of braid technique spans across scales, materials and production processes.

Our motivation for exploring braid derives from the on-going *flora robotica* project [Hamann et al., 2015], focusing on symbiotic relationships between biological plants and distributed robotics, for the purpose of constructing architectural artifacts and spaces. The project pursues braided structures as adaptive mechanical scaffolds, on which biological (plant) and artificial (robot) materials can grow and act (Fig. 2, left). Spatial and topological organisation of the bio-hybrid system is steered over time by a combination of high-level design objectives, distributed robot controllers, and user interaction. Achieving this broader goal requires that the braided scaffolds have the ability to reorganise continuously in situ, through a distributed construction process. The scaffolds may be collaboratively fabricated by stationary centrally-controlled braid machines [Richardson, 1993], swarms of distributed mobile braiding robots [Heinrich et al., 2016], or manual braiding by users (Fig. 2, right). To achieve these aims, it has been necessary to develop tools and workflows that can support design speculation and specification, and provide an adequate representation of braid in the context of diverse and often independent design and evaluation tasks. Within *flora robotica*, these include the modelling of complex braids produced by hand, the generation of fabrication instruction for both hand and automated methods, the analysis of mechanical performance of braids, and the adaptive generation of braid morphology.

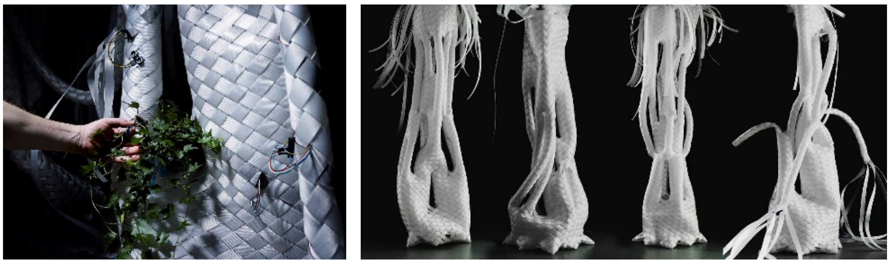

Fig. 2. (left) Braided scaffolds supporting plants and robotic elements in the *flora robotica* project. (right) Hand braided examples of self-supporting complex braid topologies.

2 Braid Representation Method

This section briefly reviews our developed braid representation method [Heinrich et al, 2017]. In general, the method uses a precomputed set of tiles, which, in their underlying logic, combine different approaches seen in the literature [e.g., Mercat, 2001; Kaplan & Cohen, 2003; Akleman et al., 2009]. The method works directly on polygon meshes that can be modified in real-time.

The first step in the method is tiling. Following the approach defined by Mercat [2001], a predefined tile dictionary provides a complete description of possible braid strip organizations (Fig. 3). In the tile dictionary utilised in this paper, there are three possible relationships between neighboring tiles: no connection (green colour in Fig. 3); connection with two separated strips (yellow); connection with two crossing strips (blue). To be able to simulate the physical characteristics of the braiding pattern, it is necessary to translate the tile notation into geometry. The method uses a point grid, with each strip declared as a series of grid-based coordinates on the mesh.

The next step is relaxation. In evaluating the physical properties of the digital model, the braid cannot be approximated to a grid-shell due to torsion occurring in the flat strips. The mesh topology is therefore constructed from triangle meshes with varying density. The constraint-based geometry solver Kangaroo 2 (update to [Piker, 2013]) is used to perform relaxation, with objectives to equalise mesh edges, detect collisions for zero length mesh manifolds, and add shell-like behaviour. The problem of 'overshooting' may occur when mesh faces are undesirably stretched or compressed by large percentages, so a dynamic constraint incrementally increases initial edge lengths to reach the target. This results in longer calculation time, but achieves tight braids with strips that, if unrolled, approximate the straightness of physical strips.

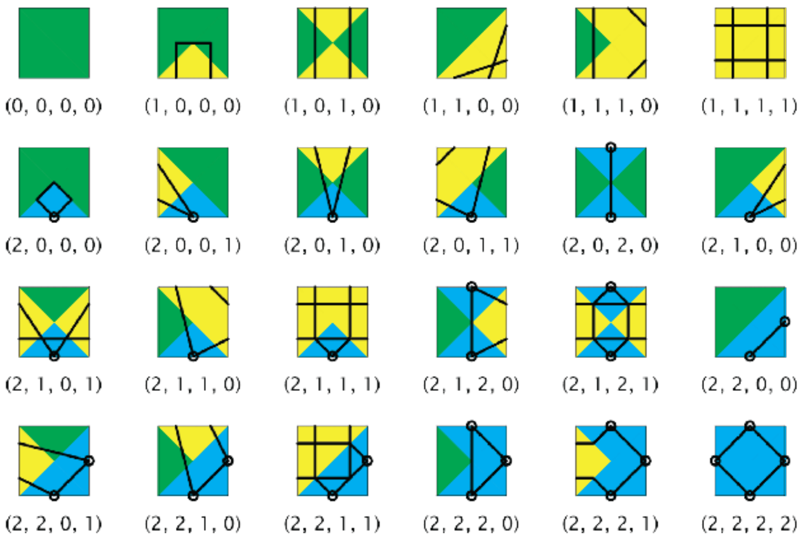

Fig. 3. An example of a complete set of tiles for 3 colours and 4-sided polygons. 24 tiles guarantee all the combinations of colours.

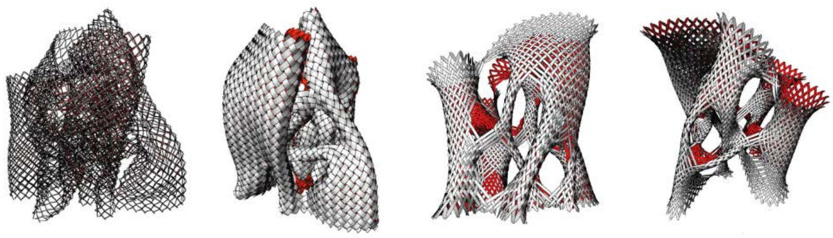

Fig. 4. Initial geometry (left) and three example relaxation results when changing target mesh edge length.

3 Design Workflows

In this section, we describe four examples that integrate the braid representation method with possible design workflows (Fig. 5). As outlined in the previous section, these workflows support design efforts of the *flora robotica* project.

In the first workflow (modelling target geometries), we compare our method to existing state-of-the-art approaches by tiling the input of a manually modelled mesh. In the second (generating fabrication instructions), we interpret a model to provide the output of textual instructions for hand braiding. Third (analysing mechanical properties), we address calibrated simulations for the output of assessing structural performance. Fourth (generative design), we generate braids from the input of an environmentally responsive robotic controller. In combination, these workflows show that our braid representation is sufficiently generalised that it is able to both receive multiple inputs (workflow 1 and 4) and be interpreted for multiple outputs (workflow 2 and 3).

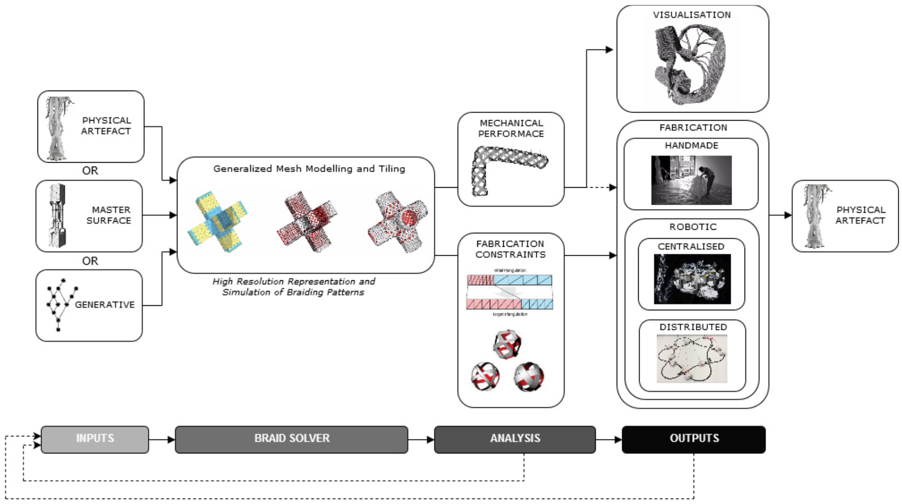

Fig. 5. Overview of the workflows supported by the generalised modelling approach for braided artifacts. The generalised modelling method can receive one of the following inputs: target physical artifact (Fig. 2), target master surface (workflow 1), or generative input (workflow 4). The mesh resulting from the generalised modelling approach can be interpreted in multiple complementary ways, such that it can be analysed (workflow 3), visualized (Fig. 4 and 6), and used to generate instructions for hand braiding or braiding robots (workflow 2).

3.1 Workflow 1: modelling target geometries

In this workflow we demonstrate modelling to pre-defined design targets, such as those shown in Figure 2. In these cases, material geometry is a hard constraint that must be considered to ensure an adequate representational approximation. In addition, conforming to local braid conditions such as asymmetric bifurcations (in terms of strip numbers), inversion and 'cornering' displayed by these physical prototypes present further modelling challenges. This workflow has five stages. First, a low-poly quad mesh model is created of the physical prototype. Next, quad mesh edges are equalised by using a custom Kangaroo 2 solver with spherical collision and equal length line constraints in a parallel thread for faster calculation time. Thirdly, all braid conditions are specified by the application of tile colours to the quad mesh. The braid tiles are then applied from the existing tile dictionary (Fig. 3). Finally, strip-strip simulation is used to achieve a tight and realistic braid model. Within *flora robotica* the making of physical prototypes is an essential mode of exploration, as is the ability to accurately represent these complex morphologies once produced.

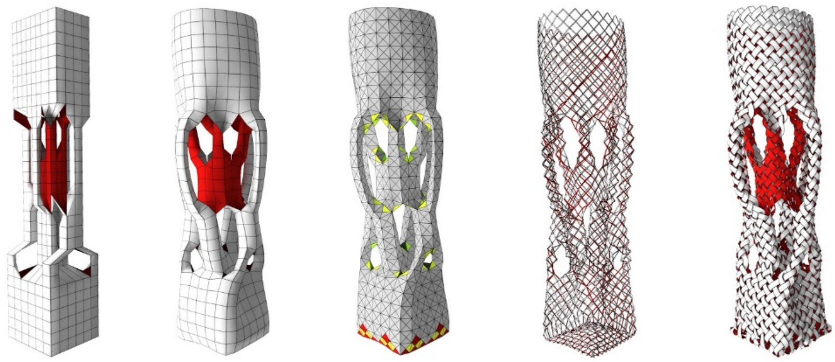

Fig. 6. 3D modelling braid typology informed by the hand-braided structures. Modelling steps from left to right are: a) 3D mesh modelling; b) Equalise edge lengths; c) Assign colours to mesh edges; d) Apply tiling; e) Strip-Strip relaxation.

3.2 Workflow 2: generating fabrication instructions

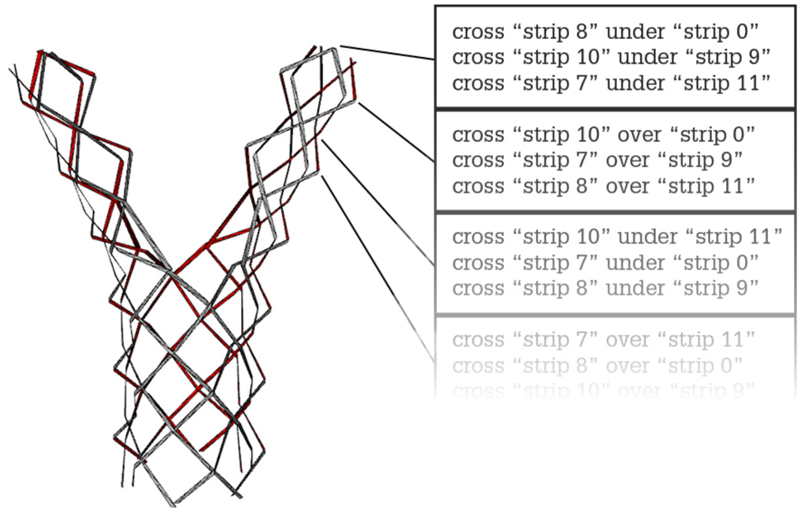

Fig. 7. Example instructions for hand braiding, generated using a low-level graph representation interpreted through an instruction dictionary.

In this workflow, a braid result from the solver is used to automatically generate fabrication instructions for distributed robots, centralised robots, or hand fabrication. In this example, we look at hand fabrication of a simple 12-strip bifurcating braid. We developed a method, firstly, to generate a low-level graph representation of the braid’s organisation and, secondly, to interpret that graph into instructions using a string replacement dictionary. The low-level graph represents the braid as a series of connections, in which each intersection of braid strips receives a unique ID, and the under or over condition of a strip is indicated by a sign attached to the intersection ID. In order to generate this low-level graph, an agent starting at the beginning of each strip walks it until it finds a new strip intersection point. Once it has found an intersection, it waits there until it sees that its neighbors have also found intersections. At this time, the agents each log their newly found connection, and resume walking along their assigned strips. This process is continued until the agents all find the end of their assigned strip. Conceptually, this method could be extended to be used in real-time with a continuous fabrication process. After the low-level graph is produced, their connections are interpreted through a dictionary into instructions for hand braiding (Fig. 7). These instructions could potentially be implemented into a user interface and combined with a braid visualisation to interactively show the intended result of each instruction step.

3.3 Workflow 3: analysing mechanical properties

Fig. 8. Strip-strip interaction for testing physical properties of braid (tension).

Explicit and dynamic formulations of structural elements are particularly attractive for the simulation of large deformations and nonlinear phenomena. Especially in the context of form-finding, new and alternative approaches have recently emerged which are well-suited for these types of problems. The results shown in this paper clearly demonstrate the capacity and versatility of such formulations. Modern nonlinear Finite Element packages, the de facto standard in engineering simulation, are completely equipped to perform simulations of complex mechanical systems and accurately describe their behaviour, but it still requires a certain effort to organise entire simulation routines for large design explorations in Finite Elements environments. For this reason, analysis and evaluation of mechanical performance of braided systems could be divided into the following two steps:

1. Strip-strip interaction (form-finding and system generation, as developed so far)
2. Shell-like behavior (subsequent Finite Element Analysis for evaluation of system stiffness and buckling behavior, through further developments)

As in most cases, pre-stressing effects emerging from the deformation of thin and slender elements can be safely disregarded. The geometry emerging from the form-finding step of this paper could therefore represent the direct input for FEM analysis. This two-step workflow aims to explore and analyse the characteristics of mechanical performance, focusing in particular on the assessment of axial, bending and torsional stiffness, along with potential buckling behaviour of the braided systems (Fig. 8).

3.4 Workflow 4: generative design

In this workflow, a generative input is given to the braid solver. As an example of such an input, we use a low-level robotic controller from the literature — the Vascular Morphogenesis Controller (VMC) [Zahadat et al., 2017] — to supply a macro scale graph that grows over time based on environmental conditions. In a case where a braided artifact is manufactured robotically in situ and sensors are embedded in the physical braid, a controller like the VMC could be used to guide the shape of the braid in a way that is adaptive to dynamics of the environment (and provides behavior diversity [Zahadat and Schmickl, 2014]). Therefore, in this workflow, we use a simulated VMC graph output to generate a mesh topology for the braid solver (Fig. 9).

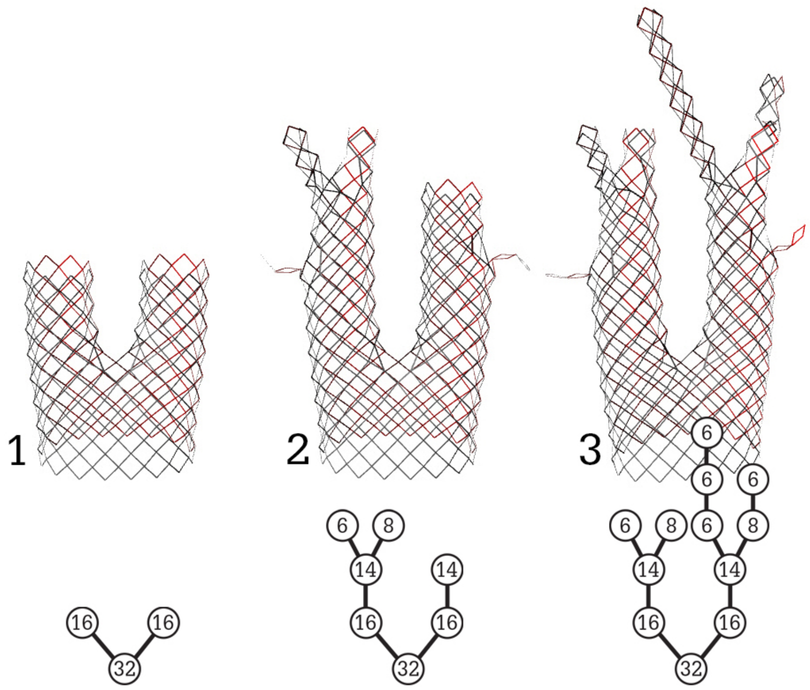

Fig. 9. Three example time-steps from the simulated growth of a VMC graph, and the braided artifacts resulting from those graphs.

Unlike the inputs of a physical target or a master surface, which can be manually defined, a generative input necessitates a solver to automate the integration of fabrication constraints. To guarantee that a generated braiding pattern can be fabricated by hand or with a braiding robot, the underlying mesh has to have all the faces marked with a direction pointing in the fabrication direction. A macro scale directed graph approach was developed to generate meshes following this constraint (Fig. 10). The graph serves as a scaffold for the mesh tiling, and has to comply with several constraints to satisfy the fabrication method and mesh generation routine.

Fig. 10. Example mesh solution (center) from a weighted macro scale directed graph topology (left) and the eventual result from the braid solver (right).

4 Evaluation

We evaluate the performance and generalisability of the method against geometric similarity and simulation performance. To assess simulation performance, we compare approaches to collision detection within the braid relaxation phase of the method to extrapolate the limits on complexity of currently achievable models. To assess geometric similarity we compare geometries of simulated results against physical examples.

Simulation performance is a limiting factor to the complexity that can be represented, with the relaxation phase being the most computationally demanding due to monitoring line-to-line collision detection in the mesh. We tested several methods to evaluate collision detection performance. Using an input mesh with 11105 edges we obtained the following results:

- using a Spatial-Grid method (Teschner et al. 2003) without multi-threading results in a running speed of 60 - 100 ms per frame;
- using an R-Tree search method results in 80 - 200 ms per frame;
- using a conventional line-to-line constraint when calculating collision between all possible pairs runs at 6.1 - 7.5 sec per frame and runs out of memory for larger models.

The geometric similarity of relaxed braid meshes is visually assessed. In Fig. 11, two features are physically prototyped, and then modelled for comparison. In both cases, the macro geometry of the model conforms closely to the physical prototype, whilst yarn-to-yarn relations show some geometric deviation. The modelled braid appears looser around regions of large geometric transition (Fig. 11, right).

Fig. 11. Evaluation of the method by visual comparison to physical prototypes.

5 Conclusion & Further Work

This paper has reported on a method and its integration within design workflows that address a gap in suitable tools for braid pattern generation and physics-based simulation. We have described and demonstrated workflows that target the exploration of generative or target based geometry, the production of fabrication instructions, and the conducting of structural analysis. These workflows support current research efforts in the *flora robotica* project, but are sufficiently generalised for the exploration of braid patterns beyond this project.

Further work aims to develop calibrated modelling of mechanical performance of modelled braids, and to fully test fabrication workflows using braiding machines and distributed robotic methods currently under development within the *flora robotica* project.

References

1. Akleman, E., Chen, J., Xing, Q., Gross, J. L.: "Cyclic plain-weaving on polygonal mesh surfaces with graph rotation systems." in *ACM Transactions on Graphics 28*, 3, 1 (2009).
2. Ayranci, C., Carey, J.: "2D braided composites: a review for stiffness critical applications." *Composite Structures 85*, 1, pp.43-58 (2008).
3. Hamann, H., Wahby, M., Schmickl, T., Zahadat, P., Hofstadler, D., Stoy, K., Risi, S., Faina, A., Veenstra, F., Kernbach, S., Kuksin, I., Kernback, O., Ayres, P., Wojtaszek, P.: "flora robotica – Mixed societies of symbiotic robot-plant bio-hybrids." in *IEEE Symposium Series on Computational Intelligence*, Cape Town, pp.1102-1109 (2015).
4. Heinrich, M. K., Zwierzycki, M., Vestartas, P., Zahadat, P., Leon, D. A., Cheheltan, A., Evers, H. L., Hansson, P. K., Wahby, M., Hofstadler, D., Soorati, M., Hamann, H., Schmickl, T., Ayres, P.: "Representations and design rules – Deliverable D3.1 – EU-H2020 FET project 'flora robotica'" available at https://zenodo.org/record/816270/export/hx#.WV8-g4SGNJE [verified 07 July 2017]
5. Heinrich, M. K., Wahby, M., Soorati, M., Hofstadler, D., Zahadat, P., Ayres, P., Stoy, K., Hamann, H.: "Self-organized construction with continuous building material: Higher flexibility based on braided structures" in *IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, Augsburg, pp.154-159 (2016).
6. Kaplan, M., Cohen, E.: "Computer Generated Celtic Design." in *Proceedings of the 14th Eurographics Workshop on Rendering*, 44, pp.9–19 (2003).

7. Mallos, J.: “How to Weave a Basket of Arbitrary Shape.” in *Proceedings of International Society of the Arts, Mathematics and Architecture*, E. Akleman and N. Friedman, Eds., pp.13–19 (2009).
8. Mercat, C.: “Les entrelacs des enluminure celtes.” in *Dossier Pour La Science 15* (2001).
9. Piker, D.: “Kangaroo: form finding with computational physics.” *Architectural Design* 83, 2, pp.136–137 (2013).
   Updated to Kangaroo 2, URL: http://www.food4rhino.com/app/kangaroo-physics.
10. Polya, G., Read, R. C.: Combinatorial enumeration of groups, graphs, and chemical compounds. Springer-Verlag (1987).
11. Potluri, P., Rawal, A., Rivaldi, M., Porat, I.: “Geometrical modeling and control of a triaxial braiding machine for producing 3D preforms.” in *Composites Part A: Applied Science and Manufacturing*, Vol.34, 6, pp.481-492 (2003).
12. Rana, S., Fangueiro, R.: Braided Structures and Composites: Production, Properties, Mechanics, and Technical Applications. CRC Press (2015).
13. Richardson, D.: *U.S. Patent No. 5,257,571*. Washington, DC: U.S. Patent and Trademark Office (1993).
14. Teschner, M., Heidelberger, B., Müller, M., Pomerantes, D., Gross, M. H.: “Optimized spatial hashing for collision detection of deformable objects.” in *Vmv* 3, pp.47–54 (2003).
15. Yuksel, C., Kaldor, J. M., James, D. L., Marschner, S.: “Stitch meshes for modeling knitted clothing with yarn level detail.” in *ACM Transactions on Graphics 31*, 4, pp.1–12 (2012).
16. Zahadat, P., Hofstadler, D.N., Schmickl, T.: “Vascular Morphogenesis Controller: A Generative Model For Developing Morphology of Artificial Structures.” in *Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2017)*, Berlin (2017).
17. Zahadat, P., Schmickl, T.: "Generation of Diversity in a Reaction-Diffusion Based Controller." in *Artificial Life 20*, 3, pp. 319–342 (2014).

Acknowledgements

We would like to thank Payam Zahadat for feedback on the generative section and for providing us with example VMC graphs, from simulations tailored to braided structures.

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