

Collaborative Models for Design Computation and Form Finding—New Workflows in Versioning Design Processes

Jonas Runberger¹(✉)  and Julian Lienhard² 

¹ Chalmers Department of Architecture and Civil Engineering, Gothenburg, Sweden

jonas.runberger@white.se

² Structure GmbH, HCU Hamburg, Hamburg, Germany

Abstract. This paper explores the role of frameworks and conventions in design computation workflows for collaboration on the development of design, structure and detailed fabrication within a visual scripting environment. The frameworks were developed in the computational team Dsearch of the architectural practice White Arkitekter and the engineering practice of structure, and was used in the part-time 12-month design and build workshop Textile Hybrids at the HafenCity Universität Hamburg. During the workshop, the framework was further developed to facilitate iterative design/analysis studies between design models in a visual scripting environment and a FEM simulation environment through cloud-based data exchange protocols. The authors regard this continuous re-development of workflow frameworks during design development as emergent, and regard this as a valuable and potential mode of development also for architecture and engineering practice. This is shown in the individual practices of the authors, where additional layers have been added.

Keywords: Computational design workflows · Modelling of data · Collaborative design

Introduction

Generic models for design collaboration enabled by digital technology has been discussed as re-usable schema for interactions between designers and software (Oxman 2006), or as compartmentalized strategies for the capture of design intent and the rationalization of geometries (Hudson 2010). Both approaches seek to establish generic traits that can be classified and categorized. In this paper project-specific models for workflow are presented as emergent during the development itself, which in turn makes them specific to the conditions of the very particular material and structural set-up.

Given that visual scripting environments such as Grasshopper are node-based, with a functionality depending only on the association and topology of the contained definition elements, there is technically no need to use any other standards. In a collaborative environment however, it is already known that the legibility of visual scripts can be imperative in order to clarify intent and functionality (Davies et al. 2011). In this case, this was addressed through the use of frameworks for computational design

workflows developed within Dsearch at White Arkitekter. Initially set up as a pure graphic standard employed to cluster and encapsulate and colour code key parts of visual scripting elements (Runberger and Magnusson 2015), this standard has been expanded to include clustered functions in the form of User Objects (Fig. 5). The collected framework is based on several plugins developed elsewhere, but have contextualized and integrated into the conventions.¹

The next level of organization within the development at Dsearch workflow is the design system; the assemblage of linked methods and tools, but also physical and computational models and drawings; the human actors of the project and the groups they form—design team, R&D team, specialists, clients and external consultants. In a professional setting the design system is set up to handle all design issues, but also policies and contracts—the complete framework that facilitate and condition design development through computational means (Magnusson and Runberger 2017a) (Fig. 2).

Background and Experiment

The workshop assignment focused on hybrid structures, i.e. the combining of two load-bearing systems with different mechanics in order to achieve more efficient structures. The design focused specifically on textile hybrids, which are the result of the combination of form- and bending-active structures, where the two systems of mechanically pre-stressed textile membranes and bending-active beam elements become interdependent (Lienhard and Knippers 2015).

Based on this approach, a group of 20 architectural and structural engineering students developed designs from first concepts to design refinement and final fabrication/assembly. An initial design competition saw five competing teams develop site specific concepts for a foyer space at HafenCity Universität Hamburg, combining material experiments with computational design and form-finding. After a collective process of evaluation, one concept was selected (Fig. 1), and three teams were given this design concept, the set of computational design and analysis tools, and the collaborative workflow framework as starting points for further development. Guided by experienced tutors with many years of design, engineering and management experience from computationally developed projects, the students took on shifting roles of designers, structural engineers, project managers and fabricators. The particular challenge was found in the nature of a textile hybrid system as a closed force equilibrium, providing challenges that would not be present in pure bending-active structures (Schleicher et al. 2015). Therefore, it was not possible to split up the model into independent units for individual design teams, instead a distributed but connected design system needed to be set up, enabling evaluation at different stages of form finding development (Fig. 2).

¹ Key plugins used within the framework include Conduit, Elefront, Flux, Honeybee, Human, Human UI, Kangaroo 2, Ladybug, Lunchbox, MeshEdit, Metahopper, Python and Weaverbird.

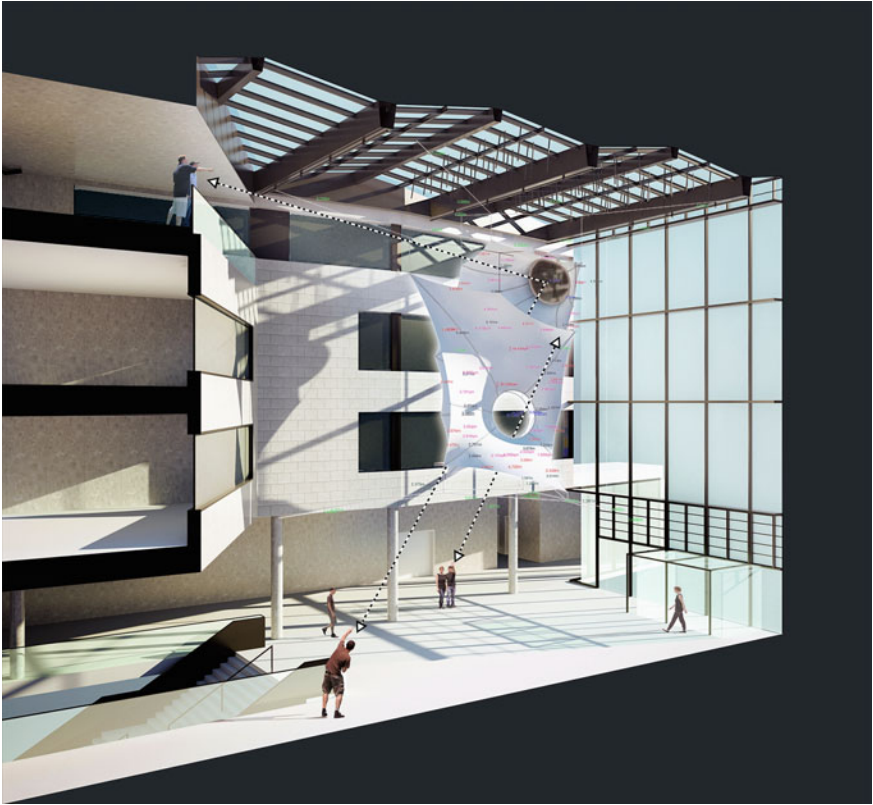


Fig. 1. Render of winning design concept from first stage of Textile Hybrids workshop at the HCU Hamburg

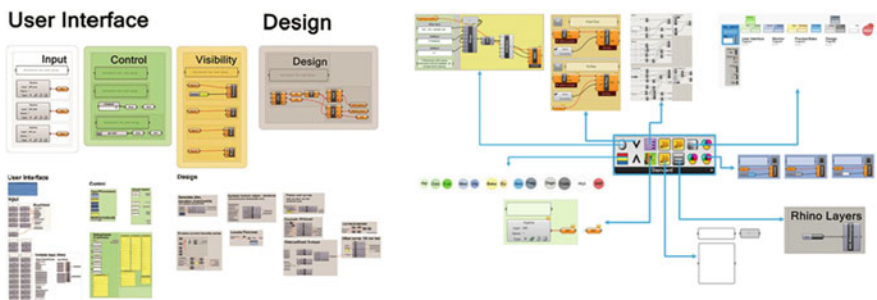


Fig. 2. Colour coding and clustering in the conventions developed by Dsearch. User Objects included in the given framework, including streamlined functions for Baking, use of Flux, accessing graphic standards etc

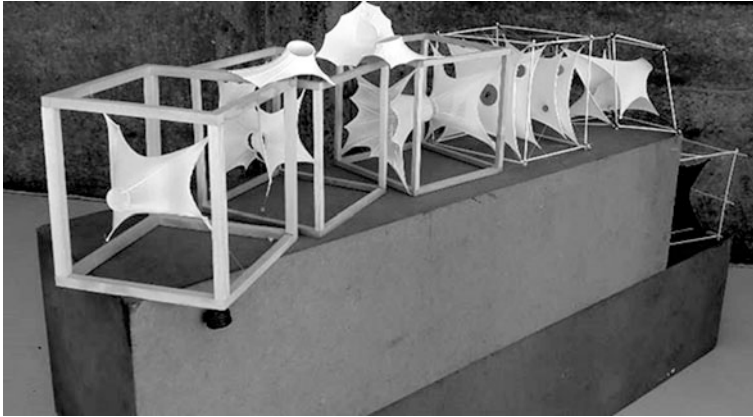


Fig. 3. Physical models in textile and glass-fibre reinforced plastic rods, within frames

Beyond the early conceptual physical models (Fig. 3), the collaborative work was primarily based on design iterations explored in digital design models and Finite Element analysis tools. A fully digital process is quite well established for the simulation and form-finding of membrane structures. Similar approaches for bending-active structures is less developed however, and the combination of the two in textile hybrids currently being developed by a few specialized research groups (Adriaenssens 2008; Ahlquist et al. 2013; Deleuran et al. 2016). The workshop employed Finite Element Analysis as a central part of the design process, handling both the form-finding and structural design phase as well as generating cutting patterns in the production phase. Thus, the engineering tools became an integral part of the design process rather than a mirrored instrument where FEM is used to check feasibility. In turn this required the setting up of bidirectional information flows that enabled fast iterations for design generation and evaluation. One important factor in this case was to include both an early more light-weight form-finding process through the integration of Rhino Membrane,² followed by a later more rigorous simulation in SOFiSTiK³ (Fig. 4).

Initial Framework and Workflow Conventions

The need for a distributed model and the integration of different simulation environments required an integration of different platform into a design system. The initial design system for the workshop was set up using Rhinoceros and Grasshopper as a main technical platform, with associated simulation software requiring external data exchanges as well as an online management and communications platform. This set up also defined the initial roles of the teams, including Object Planning (design and

² Rhino Membrane is a Rhino plugin for form-finding tensile structures, with an associated plugin for Grasshopper. <http://www.food4rhino.com/app/rhinomembrane-v20>.

³ SOFiSTiK is a software for finite element analysis. <http://www.sofistik.com>.

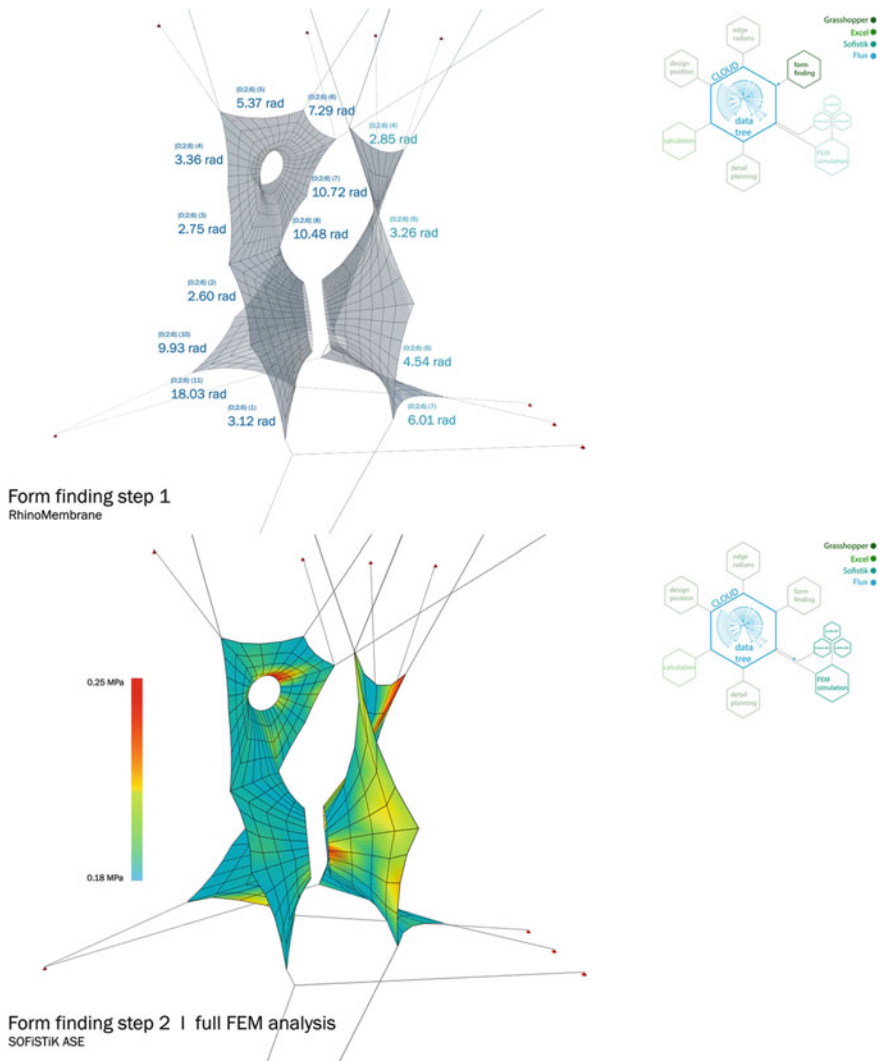


Fig. 4. Initial form-finding in Rhino Membrane and subsequent FEM analysis in SOFiSTiK

detailing), Project Management (material purchase and general time management) and Structure Planning (form-finding and structural simulations) (Fig. 5). With an expectation that data exchange between platforms would be required, this framework also

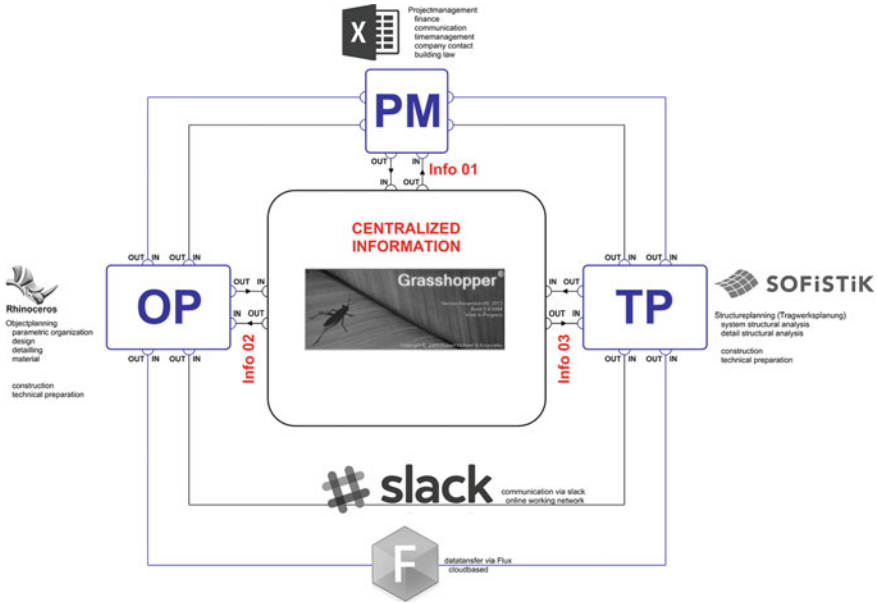


Fig. 5. The initially assumed setup of software and data flows that was used for the outset of the design system, also including the organization of the (human) design team resources

integrated the cloud based data exchange platform Flux⁴ and the project communication platform Slack.⁵

Process Development

The academic setting and the clear objective in terms of structural and material applications allowed the development of a comprehensive model for the general workflow, as well as the establishment of a design system integrating several different design and simulation environments. Given these conditions, the student teams could, over time, establish a workflow and a design system that enabled several design iterations, and which also could facilitate late changes to parts of the proposal. The controlled process also gave the opportunity operate efficiently in the parallel teams with different roles. The general workflow (Fig. 6) allowed an initial design phase in which early form finding could be conducted within Rhino and Grasshopper using analytical equations to calculate force directions, and within Rhino Membrane for form finding iterations, with simplified models using straight non-bending members. In this first stage the cable forces were optimized for their resultant force directions to point in

⁴ Flux is a cloud based platform for data exchange between different Grasshopper files as well as between different platforms such as Rhino, Grasshopper, Dynamo, Revit and Excel. <http://Flux.io>.

⁵ Slack is an online communications platform for teams, that allows both open threads for all members to exchange information and direct person to person chat functions. <http://Slack.com>.

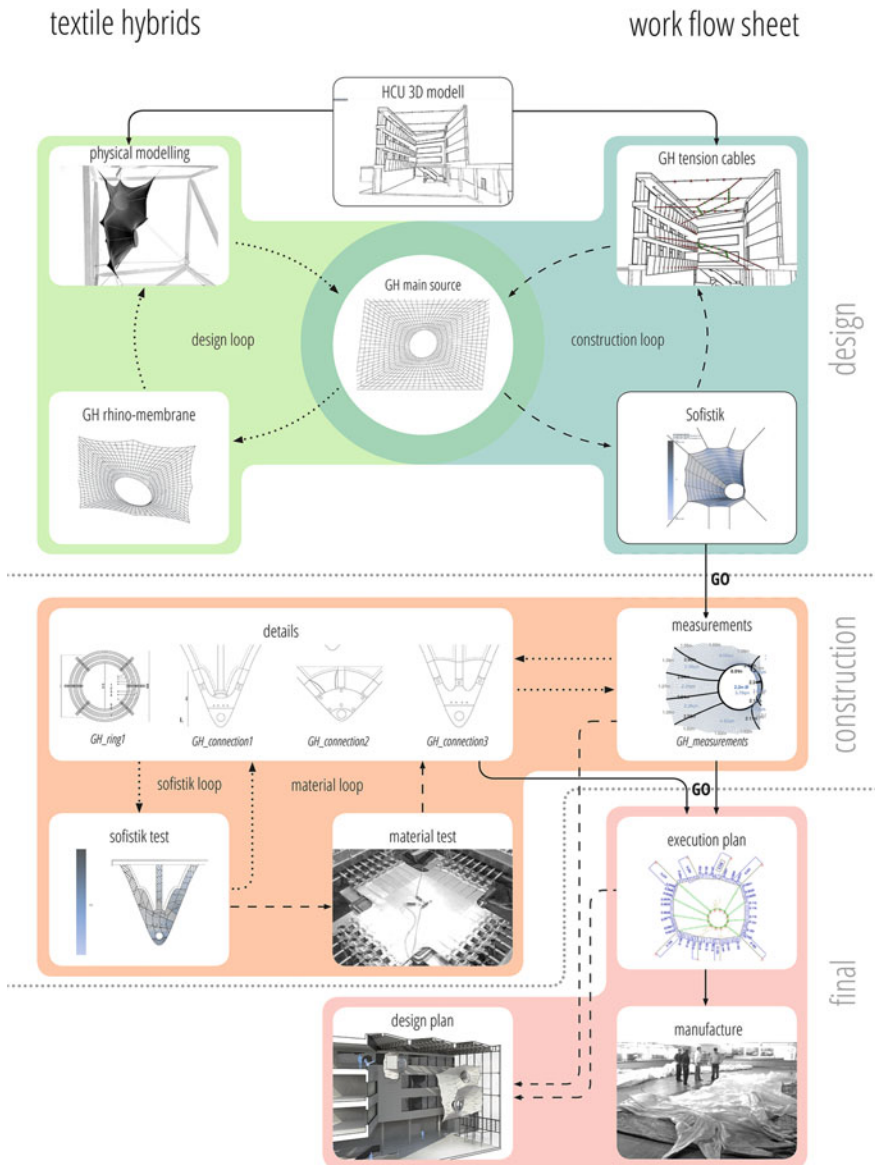


Fig. 6. Overview over general workflow indicating iterations of design and key thresholds between design, construction and final manufacture and assembly, conceptualized in the beginning of the design

the directions of supports and GFRP beams. At a second stage key elements (nodes, cables, beams and quad elements) were extracted from the model using the Rhino STiKbug plugin,⁶ to be imported into Sofistik for static FEM calculations. During the development process, there were primarily four emerging aspects of workflow and conventions that also affected the process and the project outcome: a further developed design system organizing information exchange and workflow, the specific use of cloud-based data transfers in Flux, the use of graphic conventions in the Grasshopper definitions, and the overall management of the process using Slack.

Design System

The developed and refined design system can be seen as a more detailed map of all constituents of the design process, that also includes initial trials for information exchange and the dual form-finding processes. Covering all different technical platforms as well as dedicated grasshopper definitions facilitating the data exchange, it also informally depicts the relation between different teams for parallel development. The central information model is linked to six different environments, some more closely associated with direct Grasshopper access (such as `rhMembrane.gh`), others requires dedicated Grasshopper files for exchange (such as `GHtoSofi.gh` and `SofiToGH.gh` providing links to SOFiSTiK). The exchanges of data were generally facilitated through Flux, with a central Grasshopper definition as a main information model, and a set if individual definitions dedicated to information exchange. All other documented communication was conducted in Slack, enabling distant tutors to participate as well as inter-team collaboration not depending on synchronizing time (during periods the workshop was running in parallel to other student activities, differing between participants).

Cloud-Based Data Transfers

The Flux data keys are pre-defined containers of data that allows cloud-based data exchanges between different applications.⁷ Each link in the design system has its dedicated data keys, while the final exchange with SOFiSTiK was conducted through text files. Each key can handle multiple types of data, which in this case allowed a simple yet precise transfer of information that also included important meta-data such as a central key providing an overview of all other keys. This feature allowed the different teams to work independently over periods, and still be aware of any changes in the structure of data transferred through Flux. It also contained an overview of what keys and what data was to be exchanged to and from all different files, such as the Central Information Model definition, the `rhMembrane` definition and the definitions providing connections to and from Sofistik (Fig. 7). The online interface could also

⁶ STiKbug is a plugin at development stage, and not yet released, courtesy of R. La Magna.

⁷ <http://flux.io>.

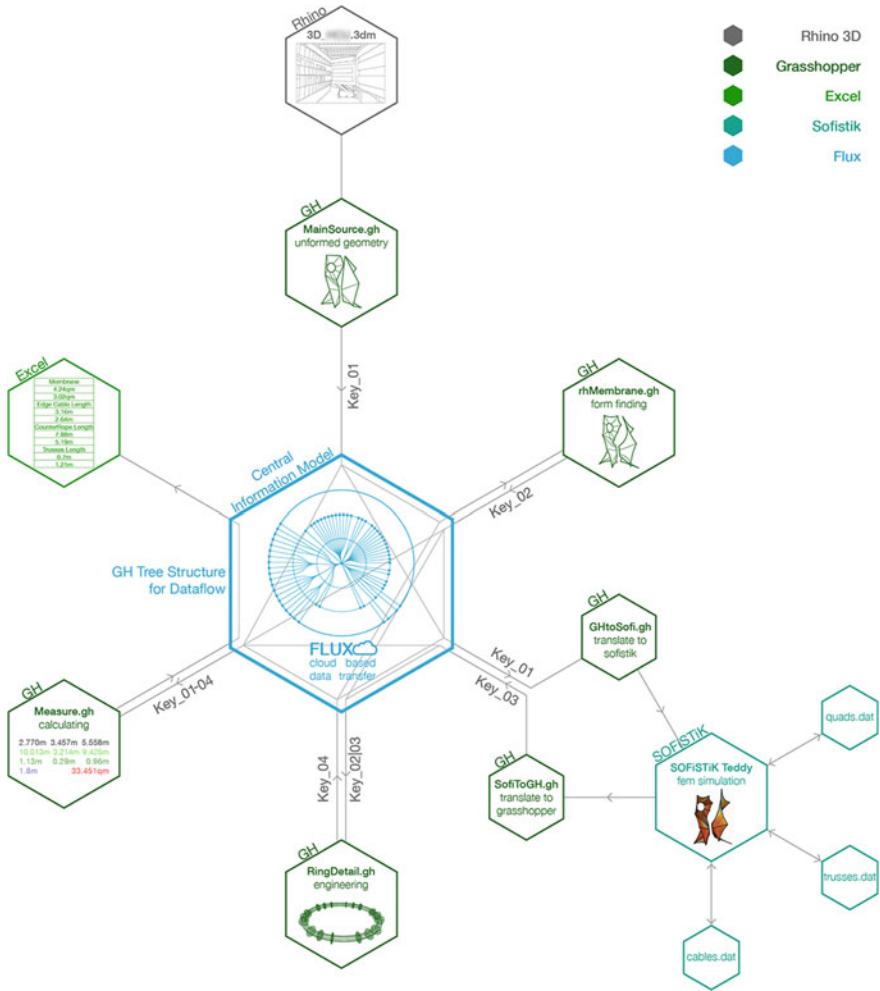


Fig. 7. The refined setup of software and data flows that constituted the design system

during the process provide visual cues to the status of geometrical data (Fig. 8). The inherent capacity for Flux to retain uploaded data furthermore allowed parallel work independent in time.

Graphic Conventions

The introduction of graphic conventions initially provided an improved legibility of all graphic definitions used. Further development adapted the conventions to include project specific aspects such as clear identification of geometry control mechanisms within the central file, specific modules for streaming data to different locations within the files, the monitoring of data key content within the Grasshopper definition, and

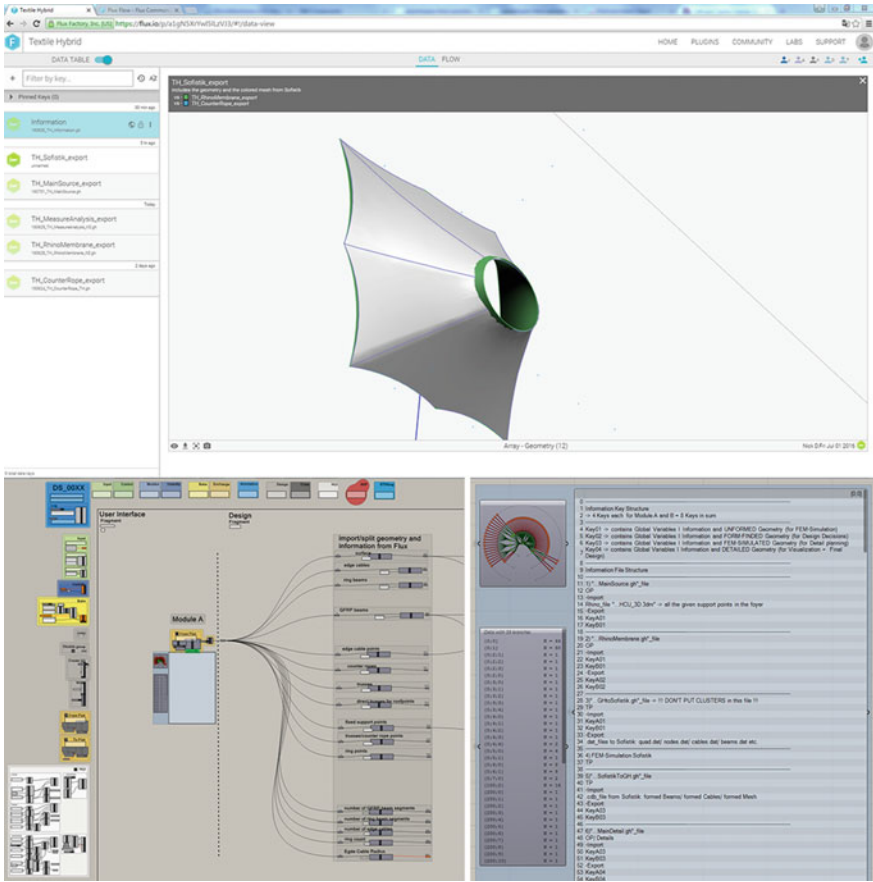


Fig. 8. Flux on-line preview of geometry uploaded from Grasshopper, extract of Grasshopper definition dedicated to receiving data from Flux for further processing, and a multi data key containing data structure, annotation and description of all data keys used

principles for the dedicated exchange definitions (Fig. 9). The use of clear naming conventions for data and different modules were an additional asset in the collaborative work. The overall initial conventions also included principles for dividing the Grasshopper canvas into different fields such as Control, Design and Export were further refined given that each definition included a complex setup of export functions to Flux.

Process Management

Slack was introduced as a general platform for communication, allowing tutors and supervisor to be engaged in the development from a distance.⁸ Over time it also became

⁸ <http://Slack.com>.

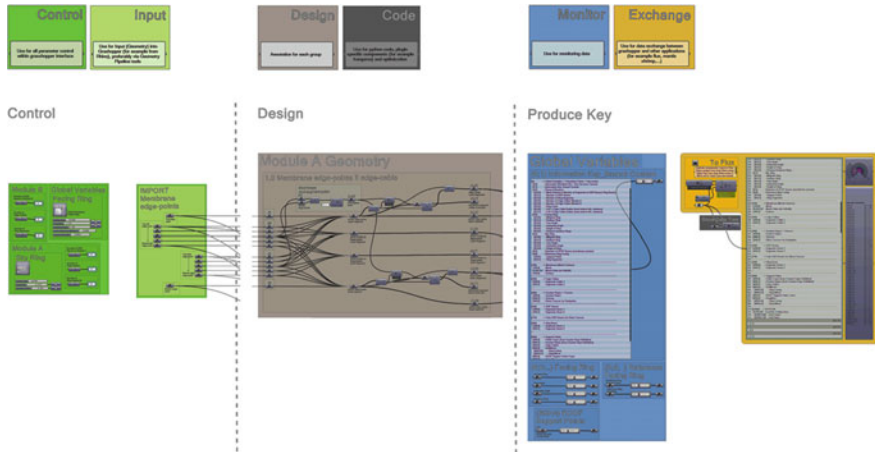


Fig. 9. Adapted graphical conventions

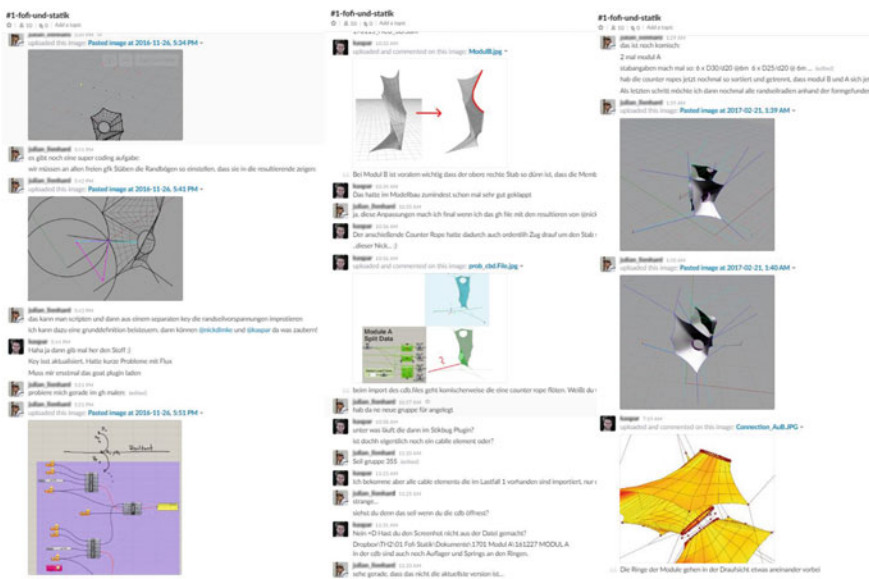


Fig. 10. Examples from the slack management in the thread specific to form-finding and statics

the platform for all exchange within the team, which in turn made the project development narrative documented in real time, including key decisions as well as general conversations and even arguments and disagreements. The platform includes the possibility to post images and files, further allowing direct discussions and evaluations over specific annotated model files or images. At times the communication also addressed specific items in the data streams in Flux, allowing direct feedback from

tutors or team members, which in effect integrated the different platforms seamlessly (Fig. 10).

Findings and Applications in Practice

While based on the engineering expertise of structure and the workflows for computational design development established by Dsearch at White Arkitekter, the specific conditions of the workshop required additional development and adaptation conducted by the participating students. The computational design framework was in this way adapted during the process to facilitate new challenges as the process commenced, such as the need for version history management and archiving during design iterations. The process faced critical conditions at several points in time. Several issues such as fire hazard led to the use of a pure glass-fibre fabric with a stretch factor of less than 0.2%, something that posed a major challenge to the manufacturing process which was entirely carried out by students and became visible in the final outcome in terms of wrinkling (Fig. 13). At a late stage an initially included mirror had to be removed since an appropriate product could not be found. This first led to a heated and documented debate on Slack, but once the decision was made, the design system and the frameworks set up allowed for design adjustments and the generation of a new equilibrium system, detail solution and associated production data in the timespan of 1 week, thus proving the power of the comprehensive design system.

The team of students here acted not only as participants in the design process, but also as focus groups for the assessment of the employed framework. In analogy, the cloud based data exchange protocol not only operated as a vessel for data transfer, but also as an archive during iterative design development. This was complemented by the Slack on-line forum for managing the process, where the development over time could be monitored by the authors. The mode of investigating principles for workflow by the engagement of student design team has precedence (Davis et al. 2011), but the experiences of the particular set-up for design and engineering collaboration is regarded to be a new contribution to the field of computational design. The workflow framework applied in the workshop is under constant development, as exemplified by the practices of the authors (Fig. 11).

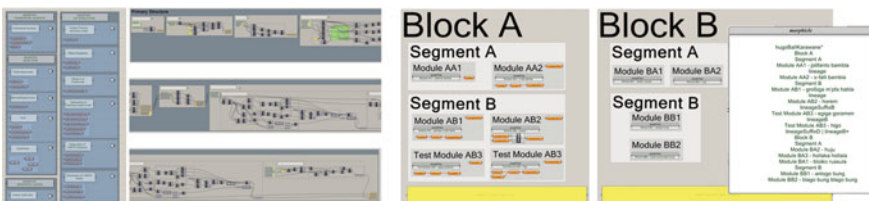


Fig. 11. Monitoring devices providing controls of previews of each definition module, and the assembly of morphological notes in an autolog, as part of the Dsearch adaptation of the framework after the workshop

Dsearch has in parallel further developed the framework in project applications. In several respects the lessons already learned from the Textile Hybrids workshop has informed the progress of development of standards and conventions, such as different trials for better monitoring of the step-by-step functionality of individual definitions. This is also partly based on additional sources where several layers of definition representations are proposed for enhanced legibility as well as preliminary specifications of definition functionality similar to pseudo-code (Zboinska 2015). First steps are also taken towards automatic documentation within definitions in terms of definition development as well as the morphological functions of definitions (the process of generating architectural form) in an autolog, which in turn could be linked to documentation and communication platforms such as Slack (Magnusson et al. 2017).

Within the practice of structure the workflows have also been adopted. So far parametric models were mostly used in the development of structures or parts of projects that can be distinctively and independently defined such as the ETFE roof in Imst, Austria (Fig. 12). In this case, the roof consists of pneumatically pre-stressed membranes, which are combined with a cable net to control the shape and maximum membrane forces in the ETFE film. For the design of the roof and the inner beam structure, several form-finding routines were integrated into the design system. The parametric model was used throughout the early design phases to define the shape of the roof and perform structural analysis with a direct linked to the FEM environment. The developments made during the Textile Hybrids workshop highlighted possibilities in working on a central and cloud based information model with separate expert groups by storing topology information in a versioned data tree. This approach now renders the possibility to introduce parametric workflows also to more complex types of structures with involvement of separate groups of experts (Fig. 13).

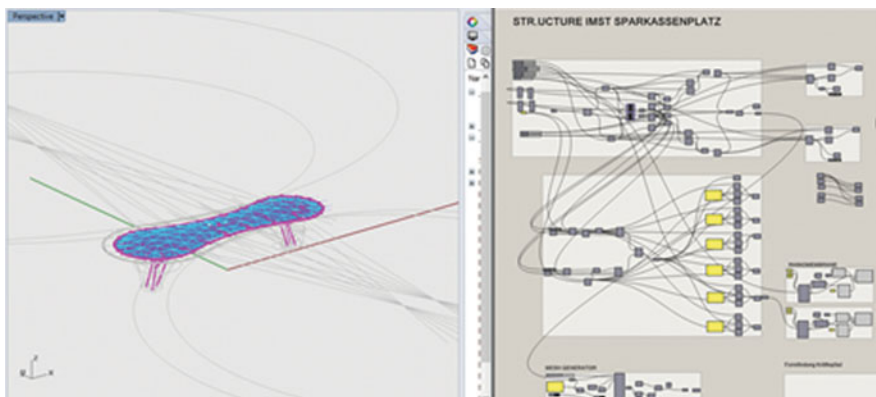


Fig. 12. The ETFE roof structure of Imst, employing aspects of the workflow framework



Fig. 13. Photograph of the final prototype installed at HCU Hamburg

Concluding Remarks

The ambition for this paper is to further open what we consider to be a very important topic—the management of all processes around the actual flow of information within computational design in architecture, preferably in an integrated and project specific way. The design outcome of the presented workshop project shows a factual value of this, and as indicated this has already affected similar processes in architecture and engineering practice. The paper maps out emerging workflow frameworks as they are continuously formed and reformed during computational design development in visual scripting environments. In terms of the experiment conducted with students this includes principles that allow iterative development for initial and final form finding, detail design and fabrication planning, as well as material testing across different computational design platforms and general project communication specific to computational design processes.

Experimental design workshops involving students always include unknown factors beyond design issues, such as skill level, ambition, engagement and general scheduling. In the case of the Hybrid Structures workshop, several teams achieved very advanced results, and not only carried through workflow-related conventions, but also contributed to new findings. The dynamic conditions of practice provide other challenges, such as changing project conditions, lessons learned from prototypes, or new fabrication limitations post tendering. Practice conditions rarely allow the establishment of comprehensive design systems, instead computational tools are often used to automate parts of the overall design process or fabrication deliveries for more complex situations of projects, merged with manually produced CAD information.

To establish analogous emergent workflow models in practice is challenging, but shows potential since it ensures correct processing and delivery of data, proper documentation of process in order to learn from past mistakes, systematizing re-occurring functions into re-usable scripts, as well as creating empirical data for future research. The employed workflow frameworks and the integration not only of design and simulation tools, but also communication platforms, are targeting these issues. Not only does this add another layer to best practice within computational design, it can also challenge conventional design practice and standardized quality systems.

Acknowledgements. Student Team: Marcelo Acevedo Pardo, Fabrizio M. Amoruso, Vasileios Angelakoudis, Nick Dimke, Kaspar Ehrhardt, Luisa Höltig, Timo van der Horst, Tobias Hövermann, Michel Kokorus, Dennis Mönkemeyer, Anies Ruhani-Shishevan, Gert Salzer, Anton Samoruko, Dino Tomasello, Zeliha Atabek, Björn Bahnsen, Niklas Dürr, Maria Fritzenschaft, Matthis Gericke, Mahmoud Ghazala Einieh, Can-Peter Grothmann, Moritz Seifert, Timo Volkmann

Surveying: Erik Jensen, Dominik Trau]

Supervision: Lehrstuhl Tragwerksentwurf Prof. Michael Staffa, Wiebke Brahms

Membrane Manufacturing workshop: Jakob Frick

HCU Labs: Thomas Kniephoff, Jens Ohlendieck, Kai Schramme

Fire Simulation: Ingenieurgesellschaft Stürzl mbH

Sponsors: MAX HOFFMANN, SOFISTIK, Serge Ferrari, Textilbau GmbH

References

- Adriaenssens, S.: Feasibility study of medium span spliced spline stressed membranes. *Int. J. Space Struct.* **23**(4), 243–251 (2008)
- Ahlquist, S., Lienhard, J., Knippers, J., Menges, A.: Physical and numerical prototyping for integrated bending and form-active textile hybrid structures. In: Gengnagel, C., Kilian, A., Nembrini, J., Scheurer, F. (eds.) *Rethinking Prototyping: Proceeding of the Design Modelling Symposium*, Berlin, pp. 1–14 (2013)
- Davis, D., Burry, J., Burry, M.: Understanding visual scripts: improving collaboration through modular programming. *Int. J. Archit. Comput.* **9**(4), 361–375 (2011)
- Deleuran, A.H., Pauly, M., Tamke, M., Tinning, I.F., Thomsen, M.R.: Exploratory topology modelling of form-active hybrid structures. *Procedia Eng.* **155**, 71–80 (2016)
- Hudson, R.: Strategies for parametric design in architecture: an application of practice led research. Ph.D. Thesis, University of Bath (2010)

- Lienhard, J., Knippers, J.: Bending-active textile hybrids. *J. Int. Assoc. Shell Spat. Struct.* **56**(1), 37 (2015)
- Magnusson, F., Runberger, J.: Design system assemblages: the continuous curation of design computation processes in architectural practice. In: Hay, R., Samuel, F. (eds.) *Professional Practices in the Built Environment*, pp. 197–204. University of Reading, Reading (2017)
- Magnusson, F., Runberger, J., Zboinska, M., Ondejcik, V.: Morphology & development—knowledge management in architectural design computation practice. (*eCAADe 35*, forthcoming) (2017)
- Oxman, R.: Theory and design in the first digital age. *Des. Stud.* **27**(3), 229–265 (2006)
- Runberger, J., Magnusson, F.: Harnessing the informal processes around the computational design model. In: Thomsen, M.R., Tamke, M., Gengnagel, C., Faircloth, B., Scheurer, F. (eds.) *Modelling Behaviour: Design Modelling Symposium 2015*, pp. 329–339. Springer, Berlin (2015)
- Schleicher, S., La Magna, R.: Bending-active plates: form-finding and form-conversion. In: Thomsen, M.R., Tamke, M., Gengnagel, C., Faircloth, B., Scheurer, F. (eds.) *Modelling Behaviour: Design Modelling Symposium 2015*, pp. 53–64. Springer, Berlin (2015)
- Zboinska, M.: Boosting the efficiency of architectural visual scripts: a visual script structuring strategy based on a set of paradigms from programming and software engineering. In: Thomsen, M.R., Tamke, M., Gengnagel, C., Faircloth, B., Scheurer, F. (eds.) *Modelling Behaviour: Design Modelling Symposium 2015*, pp. 479–490. Springer, Berlin (2015)