Parametric Details of Membrane Constructions

Bálint Péter Füzes[1], Dezső Hegyi[2]

[1,2]Faculty of Architecture, Department of Mechanics, Materials and Structures
Budapest University of Technology and Economics, Hungary
e-mail: balintfuzes@gmail.com, hegyi.dezso@szt.bme.hu

Abstract: *The present paper represents a novel method of creating algorithms to accelerate and optimize the complex modeling of membrane details for designers. The Rhino® Grasshopper® software is used for the work. A set of parametric membrane models were developed with robust functionality. According to the typical test cases, the new method is a proper tools to simplify the manual modeling of complex membrane geometries. The goal of the present paper is to validate its consistency with the actual processes of engineering design and to clarify the potential for extensive industrial usage. The present research is based on the former Membrane Detail Project (http://www.membranedetail.com) of the authors. An extensive collection of tensile membrane details had been analyzed, presented and published on the website in 2012.*

Keywords: *tensile membrane structure, parametric detail, constructional algorithm*

DOI: *10.3311/CAADence.1647*

INTRODUCTION

The present paper introduces an extension of a previous work of the authors which focuses on the structural detailing of tensile membranes: typical connections, categorization of details, analysis of forces, CAD representation [7]. The recent research extends the collection of the details to practical usage by parametric algorithms developed to build 3D models for complex membrane structures.

Membrane structures are spatial load-bearing elements. Structural tents, public and stadium covers, temporary or permanent shading systems, even facades are built out of textile membranes. These structural solutions are gaining more and more investment due to developments in the fabric industry, especially in case of buildings with higher quality level. In accordance with the mechanical behavior of membranes the architectural shape is determined by mathematical requirements [8,9]. As membranes can support no compression or bending moment, a tensioned structure must be designed - it must form a double curved surface with hyperbolic shape. Membranes have an almost negligible dead-load (<~1 kg/m^2), which makes them uniquely practical for covering larger spans [3, 4, 5].

The first stage of the Membrane Detail Project was a pioneer work that summarized and analyzed the typical joint details in a free and transparent web library (http://www.membranedetail.com). Such a collection is useful as a knowledge basis for engineering work and as a platform for education. A virtual compilation of cca. 90 CAD-models was made in Autodesk® AutoCAD Architecture (Figure 1) [7].

Based on the Membrane Detail Project a novel tool kit is developed using up-to-date CAD methods.

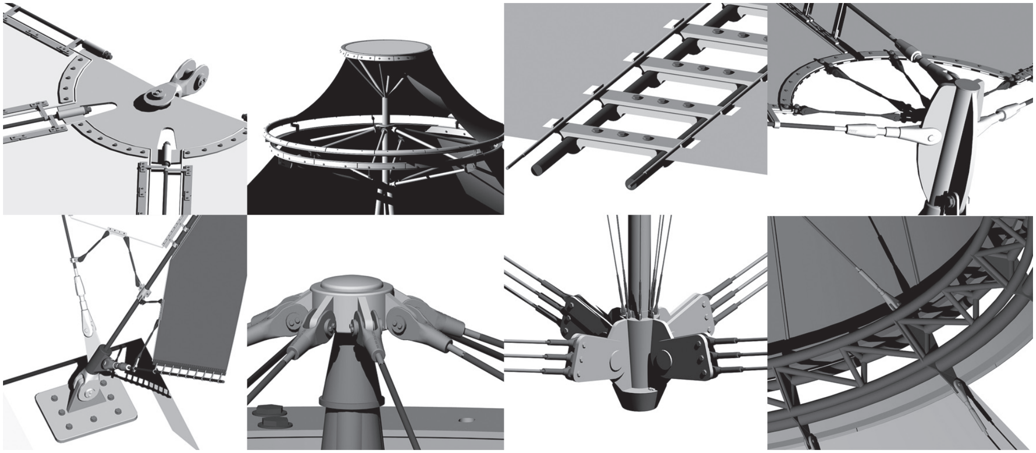


Figure 1:
Detail rendering examples from the Membrane Detail website

With parametric CAD algorithms a set of detailed joint models were formed. They can be integrated into real projects in a flexible way. One of the advantages of parametric design is the possibility of saving on the time of manual work for engineering companies.
Parametric modeling means creating algorithms representing the internal relations of different geometries. Codes can generate all possible versions of forms between pre-defined limits and parameters. In parallel with traditional modeling this coding is useful in case of complex geometries, where repeated elements (such as details of tensioned membranes) and/or precise positioning are on the table. In the case of membrane projects engineers model lots of complex forms: curves, surfaces, meshes, bolts, connecting elements, etc. Because of time and budget constraints, especially in the case of final execution drawings, it is reasonable to reduce the amount of manual work as much as possible. Furthermore the design of 3D models directly builds the basis even for steel components' manufacturing.
A group of algorithms was created with a user friendly functionality for engineers inexperienced of coding. For example the code is able to generate a complex corner detail with standard steel elements by expanding a manual model including three curves with a common intersection created by the user by traditional 3D modeling. After the user repositions the intersection node of the axis curves, the detail realigns itself. The algorithmic elements are simplified models of steel items and are free to change, scale and calibrate. Models of real products can be inserted flexibly into the system.
There are a some commercial membrane software on the market with a focus on form-finding: Easy (http://technet-gmbh.de), formfinder (http://www.formfinder.at), IX-Cube (http://www.ixray-ltd.com). They already have steel details featured but are relatively expensive for firms that do not have an exclusive focus on membranes. Since CAD systems (such as Rhino) are usually used for the modeling of entire buildings, it is reasonable to exploit the features of their plug-ins to avoid the export-import between different software. Consequently, all of the models and codes are made in the Grasshopper® plug-in of the Rhino 5 McNeel® program.

PARAMETRIC MEMBRANE ALGORITHMS

Parametric algorithms express the internal coherence of geometries through logical, mathematical and geometrical steps. The efficiency of the algorithms can be better than that of traditional modeling, if the geometry is reasonably automatable. For example situations with strong repetition or dependency with the criteria of strong accuracy.
Membrane details are problematic to visualize because of their complex 3D geometry. In the case of

bigger tents repetitive situations occur with small differences in large number of elements, regarding the length, angles, planes, etc. Using generated details can make a faster and more efficient modeling process.

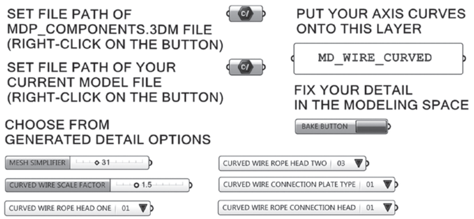


Figure 2:
An example for the parameter list and for the simple steps to follow to activate the algorithm

The background of the MD algorithms is based on the simultaneous functionality of two files. The first is the *tensioning element components file* (.3dm extension): it contains the 3D models of tensioning items: cable heads, bolts, edge plates, etc. They are arranged to a special inner coordinate and vector system bound for the functionality of the algorithm. The other file is the Grasshopper algorithm file (.ghp file extension) with its own complex internal connections - it completes the detail for the user. These two, synchronized files can be easily integrated into active projects and can be disseminated through the website.

The file of the components contains typical models of real tensioning products: wire rope heads, steel bolts, tensioning plates, etc. It also contains strongly simplified elements to save memory and computational time. This file of components can be extended with models of products of companies - so the algorithms can be a proper tool for companies to trade their own elements.

These are the simple steps to generate the model (Figure 2.):

- browse the file path of the components file on the computer
- browse the file path of the actual model file on the computer
- put the axis curves on a layer with a pre-defined name
- choose from the given options to refine the detail
- freeze the model detail in the model space when it fits the demands for further rendering, visualization and work.

The entire usage is simplified for the user. The internal functionality of the code is hidden in a system of folders containing the compressed codes called 'Clusters', therefore for the user only the upper commands are visible in a very neat, ordered form.

There is a group of pre-defined parameters to optionally calibrate the details through number sliders, value lists, control knobs and other visual buttons. In this Parameter list one can choose for example the type of the head of the cable, the number of bolts on a tensioning plate, the scaled diameter of a cable, the number of the cable connections, etc. The user can also define the density of "meshing". Meshes are substitutions of surfaces to simplify the geometry for saving on computational time.

Regarding the internal functionality of the algorithms, an initial geometry is provided for every model, which is a set of axes to generate the joint onto. The user only works with these lines/curves/points, etc. These geometries are in every case evident for the detail: such as the axes of cables for rope details or the intersecting axes of the cables for corners.

Connected to this simple, reference geometry the code operates with a file path and an "import gate". These coding parts ensure the connectivity and synchrony of the two files. With their help the elements are imported into the active model file. The coordinate vectors and lines are also set, they define the lengths and the angles of the elements.

The complex geometrical processes and the parameter lists are different in every case with a strong repetition. The process moves the imported elements to their final coordinate system.

The collection currently contains 6 algorithms for typical membrane situations:

Curved and linear cables: the most typical details in the membrane industry. The code generates a cable structure onto a curve or a line. The cable aligns its own heads in the plane where the actual rope would rotate. For linear cables there is an additional parameter to rotate the cable heads. One can choose separate heads for both ends of the curve. In case the user chooses to split the cable into more segments, the number and position of connection plates get listed as well (Figure 3).

Compressed beam: a beam structure, generated onto a line. The possibility for the rotation is the same as for the linear cable. There are different support solutions for both ends of the beams, depending on the mechanical behavior, the degree of freedom and the rigidity. A complex cable structure is generated around the main axis if required as a supporting system against the buckling of the beam. The cable heads and diameters, and the radial and linear count of the extended cable structures are all on the parameter list amongst other options (Figure 3).

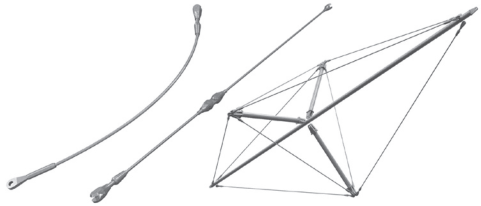


Figure 3:
Curved and linear cables and a compressed beam

Simple corner detail: corner details are modeled very often. This detail is generated from two intersecting curves. The number of the connecting bolts, the dimensions of the hole in the steel plate, the radius for the filleting of the plate, etc. are listed parameters (Figure 4).

Flexible corner detail: a detail with a very complex geometrical process in the background, which generates a huge amount of elements along its edge curves. The detail is generated onto three intersecting curves. This detail is the best representation of how complex the modeling can be in case of membranes. It can be practical for a stadium roof with a high number of such details, especially when they have different angles and dimensions (Figure 4).

High-point mast detail: this is a good example of how powerful the parametric design for complex membrane details can be and it shows that even with longer codes it is still robust. The mast is generated onto two simple circles in the modeling space and has plenty of parameters to choose from: number of segmentation for the surface with cables, number of secondary cables, height ratio of the secondary membrane surface, etc. (Figure 5).

In the background of every algorithm there is a large amount of volatile data streaming through the code: value parameters, coordinates, referenced surfaces, imported curves, length data, etc. Part of these data is referenced simultaneously to the traditional modeling space. In order to maintain an ordered information flow and a valid result the programmer has to focus on the data trees at every step. Data trees represent the grouping of the data. This is the most delicate issue of the parametric coding. There is no clean solution for the extended partitioning of the numeric information. By grouping the data into ordered Tree branches or lists, the coder has to know the logical struc-

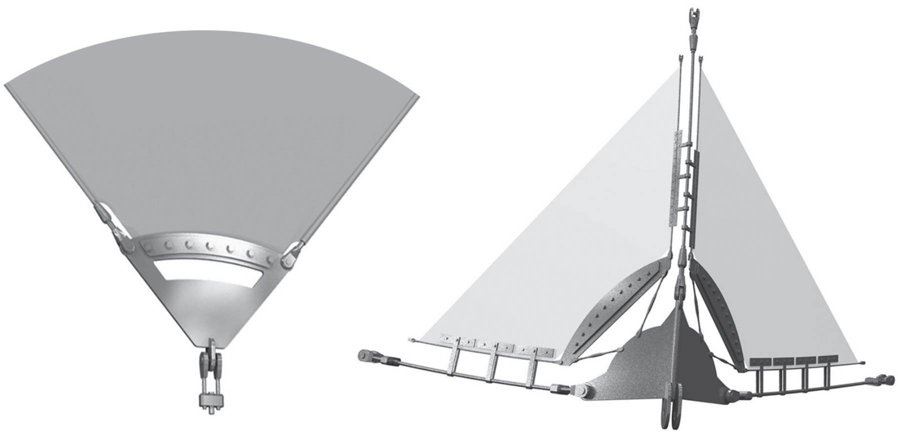
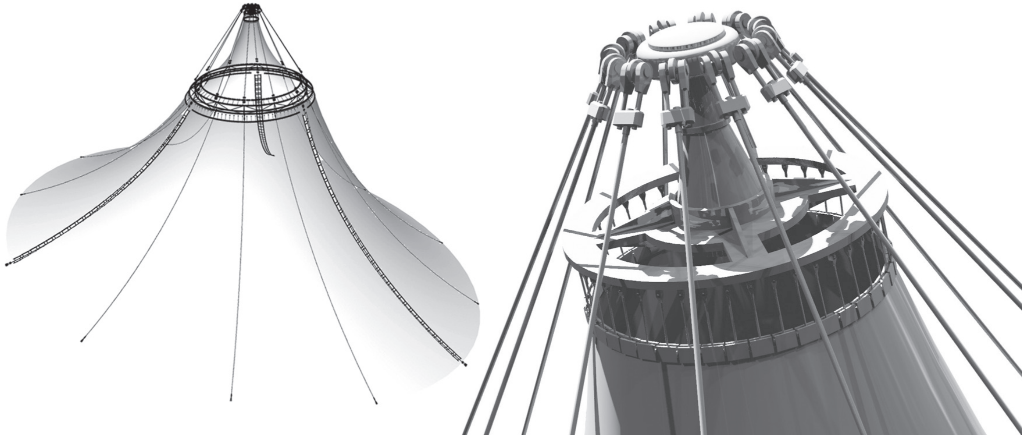


Figure 4:
Generated versions of a simple corner (left) and a complex corner detail with flexible edge elements (right)

Figure 5:
High-point mast detail



ture of each component. There is a certain group of commands to regulate the information flow. Component examples: *Graft Tree:* every list item will become a separate group. *Flatten Tree:* the partitioning dissolves, one collective group remains. *Trim Tree:* the depth of the grouping "folder" structure is reduced by x levels. *Simplify Tree:* common branches disappear. In other cases the coder uses logical-mathematical distributors and conditional statement commands to administer the aimed pattern. (Figure 6) [1, 2]

FURTHER DEVELOPMENT

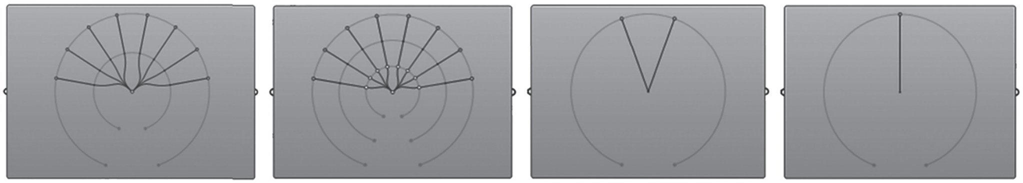
The presented project has a wide range of possibilities for its usability, future content and business model. In the near future, the new parametric details are going to be intagrated into the website.

Another plan is to combine the codes with the Kangoroo plug-in in Grasshopper. Kangoroo is a code-system to put loads onto structures and affect their geometry. In Kangoroo it is simple to tension a curved cable or a surface. In case of the Membrane Detail joints the details would realign themselves in response to the load parameters. The details would be real-time extensions to active, live structures. [6]

There is also a possibility to code entire parametric structures: complex masts or matrixes of parametric shading systems, etc. These could be representative models for the sales of real industrial products (tents, wire rope systems, profiles, etc).

The algorithms could be extended to the Autodesk® Revit Dynamo parametric plug-in (which operates with Families, pre-defined models). Nevertheless, Graphisoft ArchiCAD has been recently expanded with the GSM-LCF Exporter plug-in to ensure the synchronized connectivity between the algorithms and the traditional architectural CAD models which opens up new possibilities.

Figure 6:
Tree branches: representations of data grouping variations



SUMMARY

The present paper represents the parametric modeling tool of the Membrane Detail Project. The aim of the parametric modeling is to reduce the large amount of working time and manual work of 3D CAD modeling of complex structures on company level. Typical membrane structure details are represented to illustrate the efficiency of parametric modeling. Accordingly, complex structural elements can be formed and modified with simple and user-friendly tools.

The current state of the project has to be presented for industrial users and realistic tests are required for the verification of the efficiency in real situations. The system has plenty of development and extension possibilities.

REFERENCES

[1] Zubin Khabazi, *Generative algorithms (using Grasshopper), morphogenesism* (2012), 31-35.

[2] Andrew Payne and Rajaa Issa, *The Grasshopper Primer*, Second Edition - for version 0.6.0007©, morphogenesism (2009), 36-52.

[3] Lajos Kollár, *Ponyvaszerkezetek, (Membrane structures)*, Műszaki könyvkiadó (1987)

[4] Michael Seidel, *Tensile Surface Structures: A practical guide to cable and membrane constructions,* Ernst & Sohn Verlag für Architektur und technische Wissenschaften GmbH & Co. HG Berlin (2009)

[5] http://www.tensinet.com/

[6] http://www.grasshopper3d.com/group/kangaroo

[7] Bálint Péter Füzes, *Feszített membránszerkezetek részletmegoldásai - virtuális csomópontgyűjtemény*, BME ÉPK Szilárdságtani és Tartószerkezeti Tanszék, [2012] - Consultant: Dezső Hegyi, PhD

[8] Gáspár O, Sajtos I, *Power of form. Form finding methods*, ÉPÍTÉS-ÉPÍTÉSZETTUDOMÁNY (ISSN: 0013-9661) (eISSN: 1588-2764) 41: (1-2) pp. 67-113. (2013)

[9] Dezső Hegyi, PhD, Ponyvaszerkezetek alakmeghatározási módszerei, ÉPÍTÉS-ÉPÍTÉSZETTUDOMÁNY XXXI (3-4.) pp. 233-251. (2003)

20 16 CAADence in architecture

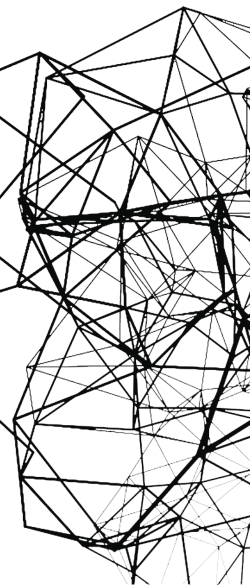
Back to command

Edited by Mihály Szoboszlai

2016 CAADence in architecture

Back to command

Edited by Mihály Szoboszlai



Editor

Mihály Szoboszlai
Faculty of Architecture
Budapest University of Technology and Economics

2nd edition, July 2016

CAADence in Architecture – Proceedings of the International Conference on Computer Aided Architectural Design, Budapest, Hungary, 16th-17th June 2016. Edited by Mihály Szoboszlai, Department of Architectural Representation, Faculty of Architecture, Budapest University of Technology and Economics

Cover page: Faraway Design Kft.

Layout, typography: based on proceedings series of eCAADe conferences

DTP: Tamás Rumi

ISBN: 978-963-313-225-8
ISBN: 978-963-313-237-1 (online version)

CAADence in Architecture. Back to command
Budapesti Műszaki és Gazdaságtudományi Egyetem

Publisher: Faculty of Architecture, Budapest University of Technology and Economics

CAADence in Architecture

Back to command

Proceedings of the International Conference on
Computer Aided Architectural Design

16-17 June 2016
Budapest, Hungary
Faculty of Architecture
Budapest University of Technology and Economics

Edited by
Mihály Szoboszlai

Theme

CAADence in Architecture
Back to command

The aim of these workshops and conference is to help transfer and spread newly appearing design technologies, educational methods and digital modelling supported by information technology in architecture. By organizing a workshop with a conference, we would like to close the distance between practice and theory.
Architects who keep up with the new design demanded by the building industry will remain at the forefront of the design process in our IT-based world. Being familiar with the tools available for simulations and early phase models will enable architects to lead the process. We can get “back to command”.
Our slogan “Back to Command” contains another message. In the expanding world of IT applications, one must be able to change preliminary models readily by using different parameters and scripts. These approaches bring back the feeling of command-oriented systems, although with much greater effectiveness.

Why CAADence in architecture?
“The cadence is perhaps one of the most unusual elements of classical music, an indispensable addition to an orchestra-accompanied concerto that, though ubiquitous, can take a wide variety of forms. By definition, a cadence is a solo that precedes a closing formula, in which the soloist plays a series of personally selected or invented musical phrases, interspersed with previously played themes – in short, a free ground for virtuosic improvisation.”
Nowadays sophisticated CAAD (Computer Aided Architectural Design) applications might operate in the hand of architects like instruments in the hand of musicians. We have used the word association cadence/caadence as a sort of word play to make this event even more memorable.

Mihály Szoboszlai
Chair of the Organizing Committee

Sponsors

GRAPHISOFT.
ARCHICAD

AUTODESK.

STUDIOIN-EX
ARCHITECTS&ENGINEERS

MŰEGYETEM 1782

Építészeti Ábrázolás Tanszék
Department of Architectural Representation

Acknowledgement

We would like to express our sincere thanks to all of the authors, reviewers, session chairs, and plenary speakers. We also wish say thank you to the workshop organizers, who brought practice to theory closer together.
This conference was supported by our sponsors: GRAPHISOFT, AUTODESK, and STUDIO IN-EX. Additionally, the Faculty of Architecture at Budapest University of Technology and Economics provided support through its "Future Fund" (Jövő Alap), helping to bring internationally recognized speakers to this conference.
Members of our local organizing team have supported this event with their special contribution – namely, their hard work in preparing and managing this conference.

Mihály Szoboszlai
Chair of the Organizing Committee

Local conference staff
Ádám Tamás Kovács, Bodó Bánáti, Imre Batta, Bálint Csabay, Benedek Gászpor, Alexandra Gööz, Péter Kaknics, András Zsolt Kovács, Erzsébet Kőnigné Tóth, Bence Krajnyák, Levente Lajtos, Pál Ledneczki, Mark Searle, Béla Marsal, Albert Máté, Boldizsár Medvey, Johanna Pék, Gábor Rátonyi, László Strommer, Zsanett Takács, Péter Zsigmond

Workshop tutors

Algorithmic Design through BIM

Erik Havadi

Laura Baróthy

Working with BIM Analyses

Balázs Molnár

Máté Csócsics

Zsolt Oláh

OPEN BIM

Ákos Rechtorisz

Tamás Erős

GDL in Daily Work

Gergely Fehér

Dominika Bobály

Gergely Hári

James Badcock

List of Reviewers

Abdelmohsen, Sherif - Egypt
Achten, Henri - Czech Republic
Agkathidis, Asterios - United Kingdom
Asanowicz, Aleksander - Poland
Bhatt, Anand - India
Braumann, Johannes - Austria
Celani, Gabriela - Brazil
Cerovsek, Tomo - Slovenia
Chaszar, Andre - Netherlands
Chronis, Angelos - Spain
Dokonal, Wolfgang - Austria
Estévez, Alberto T. - Spain
Fricker, Pia - Switzerland
Herr, Christiane M. - China
Hoffmann, Miklós - Hungary
Juhász, Imre - Hungary
Jutraz, Anja - Slovenia
Kieferle, Joachim B. - Germany
Klinc, Robert - Slovenia
Koch, Volker - Germany
Kolarevic, Branko - Canada
König, Reinhard - Switzerland
Krakhofer, Stefan - Hong Kong
van Leeuwen, Jos - Netherlands
Lomker, Thorsten - United Arab Emirates
Lorenz, Wolfgang - Austria
Loveridge, Russell - Switzerland
Mark, Earl - United States
Molnár, Emil - Hungary
Mueller, Volker - United States
Németh, László - Hungary
Nourian, Pirouz - Netherlands
Oxman, Rivka - Israel
Parlac, Vera - Canada
Quintus, Alex - United Arab Emirates
Searle, Mark - Hungary
Szoboszlai, Mihály - Hungary
Tuncer, Bige - Singapore
Verbeke, Johan - Belgium
Vermillion, Joshua - United States
Watanabe, Shun - Japan
Wojtowicz, Jerzy - Poland
Wurzer, Gabriel - Austria
Yamu, Claudia - Netherlands

Contents

Keynote speakers

REINHARD KÖNIG

Reinhard König studied architecture and urban planning. He completed his PhD thesis in 2009 at the University of Karlsruhe. Dr. König has worked as a research assistant and appointed Interim Professor of the Chair for Computer Science in Architecture at Bauhaus-University Weimar. He heads research projects on the complexity of urban systems and societies, the understanding of cities by means of agent based models and cellular automata as well as the development of evolutionary design methods. From 2013 Reinhard König works at the Chair of Information Architecture, ETH Zurich. In 2014 Dr. König was guest professor at the Technical University Munich. His current research interests are applicability of multi-criteria optimisation techniques for design problems and the development of computational analysis methods for spatial configurations. Results from these research activities are transferred into planning software of the company DecodingSpaces. From 2015 Dr. König heads the Junior-Professorship for Computational Architecture at Bauhaus-University Weimar, and acts as Co-PI at the Future Cities Lab in Singapore, where he focus on Cognitive Design Computing.
Main research project: Planning Synthesis & Computational Planning Group see also the project description: Computational Planning Synthesis and his external research web site: Computational Planning Science

BRANKO KOLAREVIC

Branko Kolarevic is a Professor of Architecture at the University of Calgary Faculty of Environmental Design, where he also holds the Chair in Integrated Design and co-directs the Laboratory for Integrative Design (LID). He has taught architecture at several universities in North America and Asia and has lectured worldwide on the use of digital technologies in design and production. He has authored, edited or co-edited several books, including "Building Dynamics: Exploring Architecture of Change" (with Vera Parlac), "Manufacturing Material Effects" (with Kevin Klinger), "Performative Architecture" (with Ali Malkawi) and "Architecture in the Digital Age." He is a past president of the Association for Computer Aided Design in Architecture (ACADIA), past president of the Canadian Architectural Certification Board (CACB), and was recently elected future president of the Association of Collegiate Schools of Architecture (ACSA). He is a recipient of the ACADIA Award for Innovative Research in 2007 and ACADIA Society Award of Excellence in 2015. He holds doctoral and master's degrees in design from Harvard University and a diploma engineer in architecture degree from the University of Belgrade.

Author’s index

The aim of these workshops and conference is to help transfer and spread newly appearing design technologies, educational methods and digital modelling supported by information technology in architecture. By organizing a workshop with a conference, we would like to close the distance between practice and theory.

Architects who keep up with the new designs demanded by the building industry will remain at the forefront of the design process in our information-technology based world. Being familiar with the tools available for simulations and early phase models will enable architects to lead the process.
We can get “back to command”.
The other message of our slogan is <Back to command>.
In the expanding world of IT applications there is a need for the ready change of preliminary models by using parameters and scripts. These approaches retrieve the feeling of command-oriented systems, although, with much greater effectiveness.

Why CAADence in architecture?

"The cadence is perhaps one of the most unusual elements of classical music, an indispensable addition to an orchestra-accompanied concerto that, though ubiquitous, can take a wide variety of forms. By definition, a cadence is a solo that precedes a closing formula, in which the soloist plays a series of personally selected or invented musical phrases, interspersed with previously played themes – in short, a free ground for virtuosic improvisation."

CAADence in Architecture
Back to command
International workshop and conference
16-17 June 2016
Budapest University of Technology and Economics
www.caadence.bme.hu

ISBN 978-963-313-225-8
9 789633 132258