CAADence in Architecture - Budapest 2016

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.

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 personally selected or invented musical phrases, interspersed with previously played themes – in short, a free ground for virtuosic improvisation.”

Edited by Mihály Szoboszlay
CAADence in architecture
Back to command
Edited by Mihály Szoboszlai
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
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

MÜEGYETEM 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, Andráš 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
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

14  Keynote speakers

15  Keynote
15  Backcasting and a New Way of Command in Computational Design
Reinhard Koenig, Gerhard Schmitt

27  Half Cadence: Towards Integrative Design
Branko Kolarevic

33  Call from the industry leaders
33  Kajima’s BIM Theory & Methods
Kazumi Yajima

41  Section A1 - Shape grammar
41  Minka, Machiya, and Gassho-Zukuri
Procedural Generation of Japanese Traditional Houses
Shun Watanabe

49  3D Shape Grammar of Polyhedral Spires
László Strommer

55  Section A2 - Smart cities
55  Enhancing Housing Flexibility Through Collaboration
Sabine Ritter De Paris, Carlos Nuno Lacerda Lopes

61  Connecting Online-Configurators (Including 3D Representations) with CAD-Systems
Small Scale Solutions for SMEs in the Design-Product and Building Sector
Matthias Kulcke

67  BIM to GIS and GIS to BIM
Szabolcs Kari, László Lellei, Attila Gyulai, András Sik, Miklós Márton Riedel
<table>
<thead>
<tr>
<th>Page</th>
<th>Section</th>
<th>Title</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>73</td>
<td>A3</td>
<td>Modeling with scripting</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parametric Details of Membrane Constructions</td>
<td>Bálint Péter Füzes, Dezső Hegyi</td>
</tr>
<tr>
<td>79</td>
<td></td>
<td>De-Script-ion: Individuality / Uniformity</td>
<td>Helen Lam Wai-yin, Vito Bertin</td>
</tr>
<tr>
<td>87</td>
<td>B1</td>
<td>BIM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Forecasting Time between Problems of Building Components by Using BIM</td>
<td>Michio Matsubayashi, Shun Watanabe</td>
</tr>
<tr>
<td>93</td>
<td></td>
<td>Integration of Facility Management System and Building Information Modeling</td>
<td>Lei Xu</td>
</tr>
<tr>
<td>99</td>
<td></td>
<td>BIM as a Transformer of Processes</td>
<td>Ingolf Sundfør, Harald Selvær</td>
</tr>
<tr>
<td>105</td>
<td>B2</td>
<td>Smooth transition</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changing Tangent and Curvature Data of B-splines via Knot Manipulation</td>
<td>Szilvia B.-S. Béla, Márta Szilvási-Nagy</td>
</tr>
<tr>
<td>111</td>
<td></td>
<td>A General Theory for Finding the Lightest Manmade Structures Using Voronoi and Delaunay</td>
<td>Mohammed Mustafa Ezzat</td>
</tr>
<tr>
<td>119</td>
<td>B3</td>
<td>Media supported teaching</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Developing New Computational Methodologies for Data Integrated Design for Landscape Architecture</td>
<td>Pia Fricker</td>
</tr>
<tr>
<td>127</td>
<td></td>
<td>The Importance of Connectivism in Architectural Design Learning: Developing Creative Thinking</td>
<td>Verónica Paola Rossado Espinoza</td>
</tr>
<tr>
<td>133</td>
<td></td>
<td>Ambient PET(b)ar</td>
<td>Kateřina Nováková</td>
</tr>
<tr>
<td>141</td>
<td></td>
<td>Geometric Modelling and Reconstruction of Surfaces</td>
<td>Lidija Pletenac</td>
</tr>
</tbody>
</table>
Section C1 - Collaborative design + Simulation

Horizontal Load Resistance of Ruined Walls Case Study of a Hungarian Castle with the Aid of Laser Scanning Technology
Tamás Ther, István Sajtos

2D-Hygrothermal Simulation of Historical Solid Walls
Michela Pascucci, Elena Lucchi

Responsive Interaction in Dynamic Envelopes with Mesh Tessellation
Sambit Datta, Smolik Andrei, Tengwen Chang

Identification of Required Processes and Data for Facilitating the Assessment of Resources Management Efficiency During Buildings Life Cycle
Moamen M. Seddik, Rabee M. Reffat, Shawkat L. Elkady

Section C2 - Generative Design -1

Stereotomic Models In Architecture A Generative Design Method to Integrate Spatial and Structural Parameters Through the Application of Subtractive Operations
Juan José Castellón González, Pierluigi D’Acunto

Visual Structuring for Generative Design Search Spaces
Günsu Merin Abbas, İpek Gürsel Dino

Section D2 - Generative Design - 2

Solar Envelope Optimization Method for Complex Urban Environments
Francesco De Luca

Time-based Matter: Suggesting New Formal Variables for Space Design
Delia Dumitrescu

Performance-oriented Design Assisted by a Parametric Toolkit - Case study
Bálint Botzheim, Kitti Gidófalvy, Patricia Emy Kikunaga, András Szollár, András Reith

Classification of Parametric Design Techniques
Types of Surface Patterns
Réka Sárközi, Péter Iványi, Attila Béla Széll
227  Section D1 - Visualization and communication

227  Issues of Control and Command in Digital Design and Architectural Computation
Andre Chaszar

235  Integrating Point Clouds to Support Architectural Visualization and Communication
Dóra Surina, Gábor Bödő, Konsztantinosz Hadzijanesz, Réka Lovas, Beatrix Szabó, Barnabás Vári, András Fehér

243  Towards the Measurement of Perceived Architectural Qualities
Benjamin Heinrich, Gabriel Wurzer

249  Complexity across scales in the work of Le Corbusier
Using box-counting as a method for analysing facades
Wolfgang E. Lorenz

256  Author’s index
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.
Backcasting and a New Way of Command in Computational Design

Reinhard Koenig, *1,2,3  Gerhard Schmitt1,3

*1 Corresponding Author: Reinhard Koenig, ETH Zurich, Department Architecture, Chair of Information Architecture
1 email: {reinhard.koenig|schmitt}@arch.ethz.ch, web: http://www.ia.arch.ethz.ch/koenig/
1 ETH Zurich, Future Cities Laboratory, Singapore-ETH Centre, Singapore
2 Junior-Professorship for Computational Architecture, Bauhaus-University Weimar, Germany
3 ETH Zurich, Department Architecture, Chair of Information Architecture, Switzerland

Abstract: It’s not uncommon that analysis and simulation methods are used mainly to evaluate finished designs and to proof their quality. Whereas the potential of such methods is to lead or control a design process from the beginning on. Therefore, we introduce a design method that move away from a “what-if” forecasting philosophy and increase the focus on backcasting approaches. We use the power of computation by combining sophisticated methods to generate design with analysis methods to close the gap between analysis and synthesis of designs.

For the development of a future-oriented computational design support we need to be aware of the human designer’s role. A productive combination of the excellence of human cognition with the power of modern computing technology is needed. We call this approach “cognitive design computing”. The computational part aim to mimic the way a designer’s brain works by combining state-of-the-art optimization and machine learning approaches with available simulation methods. The cognition part respects the complex nature of design problems by the provision of models for human-computation interaction. This means that a design problem is distributed between computer and designer.

In the context of the conference slogan “back to command”, we ask how we may imagine the command over a cognitive design computing system. We expect that designers will need to let go control of some parts of the design process to machines, but in exchange they will get a new powerful command on complex computing processes. This means that designers have to explore the potentials of their role as commanders of partially automated design processes.
In this contribution we describe an approach for the development of a future cognitive design computing system with the focus on urban design issues. The aim of this system is to enable an urban planner to treat a planning problem as a backcasting problem by defining what performance a design solution should achieve and to automatically query or generate a set of best possible solutions. This kind of computational planning process offers proof that the designer meets the original explicitly defined design requirements.

A key way in which digital tools can support designers is by generating design proposals. Evolutionary multi-criteria optimization methods allow us to explore a multi-dimensional design space and provide a basis for the designer to evaluate contradicting requirements: a task urban planners are faced with frequently. The vision for a cognitive design computing system is to enable an urban planner to treat a planning problem as a backcasting problem by defining what performance a design solution should achieve and to automatically query or synthesize a set of best possible solutions.

In another part we reflect why designers will give more and more control to machines. We investigate first approaches learn how designers use computational design support systems in combination with manual design strategies to deal with urban design problems by employing machine learning methods. By observing how designers work, it is possible to derive more complex artificial solution strategies that can help computers make better suggestions in the future.

**Keywords:** Cognitive design computing, backcasting, machine learning, evolutionary optimization, design synthesis

**DOI:** 10.3311/CAADence.1692

### 1. INTRODUCTION

New types of computing, such as cognitive computing, are extending the application of IT into new areas. A general definition of cognitive computing is “the simulation of human thought processes in a computerized model” (Rouse, 2014). IBM’s Watson computing initiative and the associated programs, represent an important development in this direction. A computational device as an opponent in a game of chess or a computer that triumphs on Jeopardy is just the beginning, and many other applications will follow (Kelly & Hamm, 2013). The latest success of a machine was DeepMind’s program AlphaGo, which beat a human professional player in the ancient game of Go (Gibney, 2016). The approach we present in this contribution focuses on the application of cognitive computing to the domain of urban design. In our context, urban design means the arrangement and proportioning of spatial elements such as streets, open spaces and buildings taking into consideration functional aspects like accessibility, visual qualities, or solar radiation. Our ultimate objective for this cognitive design computing system, as we call it, is to develop a program that is able to make urban designs...
that are comparable to or even better than human designs. For this, the cognitive design computing system needs to be able to learn from existing designs as well as from human design strategies. The way this system interacts with designers is therefore a crucial aspect. To be able to evaluate the performance of a design, we need various evaluation methods. Known urban analysis and simulations methods are one option, systematically collected ratings by humans another.

Architecture and urban design have always been excellent application areas for artificial intelligence and cognitive computing, but the small relative and absolute numbers of researchers in architecture made the advances appear less significant than they actually were. Design applications of artificial intelligence methods and techniques were introduced into education as early as the 1970s and 1980s in the United States (Mitchell, 1977) and later in Europe. Architecture is an interesting application area, because it involves a combination of structured input that can be produced with rule-based systems and the appraisal of past experiences and expectations of the future. This mix of requirements corresponds almost exactly to the computational tools already used: structured input and constraints, e.g. as defined by city authorities; historical data and information, which can serve as the basis for future design decisions; and user requirements that come in very different shapes and sizes and representations.

Urban design is an even more interesting application area of cognitive computing, as the amount of structured information and rules is relatively small compared to architecture, but the amount of decisions that can be derived from input from citizens, transportation needs, and external requirements is much higher than similar information for individual buildings. To achieve this, we need better ways of collecting and mining opinions, proposals, and requests that can be represented as data.

Cognitive design computing can be understood as a combination of the above: from architectural design, it draws on the very efficient abstraction methods and deep knowledge of materials, climates, and people’s use of habitats that go back thousands of years. From urban design, it draws on the necessity to provide for large numbers of people that do not necessarily live in the urban system, but which rely on its infrastructure and central functions. The advent of big data is of relevance for both cases as mining big data for patterns and individual preferences has the potential to make urban design computing systems more and more powerful.

In the following sections we introduce the framework of our cognitive design computing system and its four main parts, as shown in Figure 1: data analysis, user interaction, learning, and geometry. These we then examine in detail in the sections that follow. Data analysis focuses on the indexing of geometries or of spatial configurations in general. A prerequisite for indexing is that we can distinguish geometries. For this purpose, we introduce a method of individually ‘fingerprinting’ spatial configurations. The section on user interaction and learning describes how the cognitive skills of the designer are involved and how the computer system can learn problem solving strategies by observing the user’s actions. The geometry section introduces methods for synthesizing spatial configurations, which are in turn used as input for the data analysis. We focus on an automatic synthesis procedure to show first examples for the generation of pareto-optimal design solutions. We demonstrate the automatic synthesis procedure using the example of a concrete planning project. Finally, we conclude with a reflection of the developed system and outline next steps for its further development in the outlook section.

2. COGNITIVE DESIGN COMPUTING FRAMEWORK

The central objective of the cognitive design computing system is to serve as a planning support tool that can lead the designer towards better solutions or suggests new, useful alternative solutions. The technical realization can be either as a plug-in for existing systems or separate tool. In both cases a separate user interface is needed to allow the designer to design interactively using the system. The main elements of the cognitive design computing framework are illustrated in and explained in the following.
Unlike approaches that employ urban data to exclusively analyze existing situations, the intention of cognitive design computing is to transcend the retrospective view by integrating data via a model container into the urban design and planning process. The main technique for doing this is the model container, which is shown in the data analysis domain in Figure 1. The model container can hold all models that describe relationships between the built environment and any kind of data. Rather than undertaking a systematic analysis of which models or data would be necessary for a comprehensive, rational planning process, we take an opportunistic approach and adapt a concept described by Maruani and Amit-Cohen (2007, p. 5), using models for which data is available or that are promising for certain planning problems.

### 3. DATA ANALYSIS

The data analysis part of the cognitive design computing framework aims first to distinguish designs by analyzing their geometry, and second to add as many indexes to the geometries. The corresponding problems are to find a generalizable way to create individual fingerprints for any kind of geometry, and to be able to aggregate results from various kinds of analysis (from the models container) to indexes.

We start with the issue of how to compare different designs based on their geometric representation. Existing methods can be divided into two main groups according to the particular kinds of compared data: either they compare shape determining rules instead of a shape itself (Stiny & Mitchell, 1978) or they compare characteristic values computed as shape features of a design (Derix & Jagannath, 2014; Dillenburger, 2010) and topology informed labels (Langenhan, Weber, Petzold, & Dengel, 2011). A more general method for characterizing or labelling designs based on their purely geometric representation was introduced by Standfest (2014). He proposed a Deep Learning method for unlabeled 3D polygon meshes. The resulting characterization of a design can be understood as its fingerprint. This method can be considered as algorithmic modeling and is part of an observable trend towards minimizing the amount of semantic information needed for state-of-the-art data analysis. Because the method is domain independent, it can be applied at various scales, e.g. to evaluate apartment plans, buildings...
Figure 2: Result of the experiment conducted on clustering 48 randomly chosen building blocks according to the latent semantics of the unlabeled mesh geometry [Standfest, 2014]. The building blocks are preprocessed with a Delaunay triangulation for each plane. Bottom row, left: Domain maps of level 1 (small mesh face neighborhoods consisting of 4 triangles), middle: level 2 (slightly bigger mesh face neighborhoods, consisting of 9 triangles) and right: level 3 (biggest mesh face neighborhoods consisting of 22 triangles).

(Figure 2), facades, streets or whole neighborhoods. For a shallow learning approach, other spatial entities can be used instead of polygon meshes. Such similar sized feature vectors, also referred to as fingerprints, are important for data analysis, especially in the context of cognitive design computing. Firstly, they can be used to distinguish geometries to ensure that only significantly different ones are added to the search space. And secondly, it makes it possible to correlate geometry with empirical observations, sensor data, or computed measures from stochastic models.

We applied the method by Standfest (2014) to create the fingerprints of 48 building volumes randomly chosen from the district of Zürich Altstetten and provided by the city of Zurich. After labelling the buildings, they are clustered using a Self-Organizing-Map (Figure 2). Despite the limited volume of the data set, the resulting maps of different abstraction levels show significant clustering and topologically correct alignment of the evaluated building blocks. Since the approach is strictly data driven, the characterization of design alternatives may differ from those of a designer. The example application shown in Figure 2 illustrates how different unlabeled polygon meshes can be aligned according to latent semantics.

4. USER INTERACTION AND LEARNING

In this section we combine user interaction and learning methods. First we aim to collect empirical data that can be used for indexing geometry, and second we observe how a human designer solves a design task and learn from it to derive an artificial design strategy. The long-term objective is to combine the strengths of human observation, cognition, experience and local knowledge into our system to improve the planning, design, management and transformation of buildings and cities.

Based on the models container described in section 3, we have various ways to measure the qualities of an existing or a new urban design, depending on social, cultural, and functional contexts. For instance, one could calculate the level of street noise, air pollution, or solar exposure. With this
in mind, we assume the designer always pursues a number of goals in the form of criteria and constraints when developing a design. If a machine could know the formal descriptions of the criteria and their importance weighting, it could also optimize a design accordingly. The quality of the solution then depends on the quantity and quality (sensitiveness) of such design criteria, as well as on an estimation of the user’s goals for these criteria. The challenge in implementing our learning mechanism is to develop an algorithm that estimates the user’s preferences with regard to the various design performance measures.

To provide an adequate user interface for human-computer interaction, we developed an initial prototype that make use of current web-based technologies to render urban designs and simulation results via a web browser (Figure 3). It facilitates visualizing, editing, creating, and evaluating spatial configurations at various scales. Various urban simulation and analysis tools can be run via a webservice using LUCI (Treyer, Klein, König, & Meixner, 2015) as middleware and the results can be visualized on the website. In addition, the web user interface can be used to present a design problem and observe the strategy a designer applies to find a spatial solution. These can then be used as input for a learning mechanism with the aim of applying it independently to new similar design tasks.

Through the learning domain we aim to implement a design routine that, on the one hand, proposes design alternatives to a planner and, on the other, obtains feedback in the form of the selection of a design variant to proceed with, thereby helping the system learn and adapt to the user’s needs.

5. GEOMETRY

Beside using existing urban designs or manually creating them, another crucial part of the cognitive design computing framework is to automatically synthesize geometry. According to Weber, Müller, Wonka, and Gross (2009), the synthesis of urban structures consists of a sequence of several processes: the creation of a road network, the definition of land use and parcelling, and building placement. Systems have been developed for the procedural creation of road networks based on L-systems (Parish & Müller, 2001). In particular, the system CityEngine by ESRI facilitates the three-dimensional, rule-based modelling of cities and urban structures to the level of building details (Gool et al., 2006; Weber et al., 2009). In all these examples, the rules for the creation of an urban design solution have to be specified a priori in detail. The rules of generative or procedural algorithms are also very technical, abstract and not related to a planning problem. More importantly, they are not combined seamlessly with evaluation models and optimization methods. With these methods we therefore “have a model that can generate designs but has no means of establishing whether those designs are any good” (Radford & Gero, 1988, p. 20).

To achieve more advanced and more meaningful geometry synthesis, we therefore need to find a representation that is able to create realistic geometry for a design and can incorporate a lot of performance measures (objective functions) that can be defined by a designer. This information should make it possible for the synthesis system to generate a correspondingly large amount of possible design solutions without needing to then
analyze if objectives contradict or not. To this end, we introduce a primary method for synthesizing geometry using Evolutionary Multi-criteria Optimization (EMO) and show how this method is applied in a synthesis case study.

5.1. Evolutionary Multi-criteria Optimization

The basic technique we use for synthesizing geometry is evolutionary algorithms (EA) due to their flexibility with regard to problem representation as well as their robustness. This allows us to flexibly experiment with how we technically encode a design problem in the knowledge that the EA still work in an acceptable way even if we have a poor technical implementation. EA can be applied on various scales for layout design (Koenig & Knecht, 2014), building volume arrangement (Koenig, 2015b), urban district planning (Knecht & Koenig, 2012), or network development (Koenig, Treyer, & Schmitt, 2013; Schaffranek & Vasku, 2013). The EA may be supplemented by a number of local search strategies in order to optimize its calculation speed (Koenig & Schneider, 2012).

When we extend EA to include more sophisticated selection mechanisms that are able to consider more than one objective function for the evaluation of design solutions, we speak of Evolutionary Multi-criteria Optimization (Deb, 2001). For our design synthesis prototype we developed an individual evolutionary strategy in combination with a selection mechanism using the HypE algorithm (Bader & Zitzler, 2011) from the PISA framework (Zitzler & Thiele, 1999). This allows us to filter the non-dominated solutions out of all generated solutions, especially if we have to deal with a variable set of contradicting and non-contradicting criteria. During the computer-supported design process, planners obtain immediate feedback in the form of a set of design solutions that fulfill the formulated design requirements as well as possible. The presented system for synthesizing designs offers the possibility to experiment with various restrictions and objectives for a design project. This is an important feature since the definition of a design problem can be considered as a main step towards its solution (Rittel & Webber, 1973).

5.2. Synthesis Case Study

We assessed the applicability of the EMO method for geometry synthesis using an example scenario in Singapore. To demonstrate how our approach works in an existing urban context, we chose a defined area and assumed it needed to be completely re-planned (Figure 5, a). The choice of this example in Asia reflects an urgent need for fast and comprehensive planning systems. Necessary data on the street network was taken from Open Street Map, and information about neighboring built structures in 3D was available from the Future Cities Laboratory of the Singapore ETH Centre.

We apply the design synthesis methods for creating road networks with defined centrality characteristics, such as integration or choice of specific locations. We used these to define a location with high centrality for a new central business district, and a separate location with quite low traffic for a new residential area. Both requirements cannot be fully fulfilled, since they contradict each other where the locations adjoin. Here we need the ability of the EMO to find pareto-optimal solutions for contradicting problems. A set of these best compromise street networks is shown in (Figure 4c). Inside the blocks of the road networks we generate building layouts with defined densities, taking into account specific properties of the open space qualities measured by Isovist fields. Again these criteria may contradict each other. A set of generated pareto-optimal building layouts is shown in (Figure 4b). We illustrate how a user can interact with the developed prototype system (Figure 4) and how it can be used to help develop an urban planning proposal in a step-by-step approach (Figure 5).

The planning process starts with the empty planning area defining the border for placing new street segments, and the starting street segments from which the street network is grown. The starting segments are taken from the existing network where it intersects with the planning area (Figure 5, a). Initially the user has to execute the EMO first for the street layouts and later for the building placements by specifying the respective properties on the right-hand section of the software prototype window shown in Figure 4d.
The user can, for example, select the size of the population, the number of generations to calculate optimal layouts, and the size of the archive to store the solutions. The user interface shown in Figure 4 is structured in three main areas for visualizing the generated spatial configurations (6a-c). Figure 4b and 6c show the archives of best variants for the building layouts (6b) and street networks (6c) generated so far, and (6a) presents a 3D view that shows the configurations selected by a user out of the archives.

The centrality analysis can be run for the new network connected to the existing network in a user-defined radius around the planning site. They are combined with each other and the environment’s geometry. Based on our representation of a design by the chromosome structure of the EA, our software prototype makes it possible to move, rotate and scale individual objects (street segments and building volumes) during the planning and optimization process. This is made possible by a specially-developed mechanism that sends information on the changed geometry to its numeric representation in the chromosome. This is a very important feature of the system, since it allows a designer to modify selected urban design solutions according to their individual needs during the optimization process.

Corresponding view control functions for zooming, panning and rotating the view are available for each of the views of the software prototype (Figure 4). After several iteration steps, street graphs and building layouts appropriate to the objective values are found. Figure 5 shows the results of our prototype for a proof of concept.

6. CONCLUSION

The presented system for cognitive design computing incorporates methods for integrating various kinds of urban analysis and simulations, based either on stochastic or algorithmic modeling. They are combined in a models container, so that they can be used for the automatic labeling of geometries that are taken either from existing designs or from design synthesis processes. A crucial aspect of the system is the ability to integrate human cognition as a means of enriching and directing the computational design process.

The capabilities of the cognitive design computing system enable an urban planner to treat a planning problem as backcasting problem by defining what a solution should achieve and to automatically query or generate a set of the best possible
solutions. This kind of computational planning process we can call evidence-based planning. It offers proof that the designer meets the original explicitly defined design requirements. This way of thinking offers a new approach for taking command on a computational design process.

Our cognitive design computing system is developed for the specific requirements of the urban planning context, in which planning goals and considered influences change often during the process. In other words, the problem is defined during the planning process. To achieve this, a computational planning support system needs to enable a user to interact with the geometrical elements, change restrictions and objective functions and produce understandable visualizations.
during the iterative search process. Because of the close collaboration between computer and designer, we call this approach cognitive design computing. The result is an urban design support system that guides urban planners efficiently through an ever-changing search space, thereby assisting them in finding good compromise solutions for complex planning problems.

For the technical realization of an understandable map of design solutions, we introduced a method based on Self-Organizing-Maps for clustering design variants. The obvious advantage of this arrangement is that it allows us to find similar variants close to each other and helps to clearly identify the number of more distinctly different design solutions since they form separate clusters. The clusters can be also understood as representation of design strategies that can be explored in more detail.

ACKNOWLEDGEMENT

The section on data analysis is based on the PhD of Matthias Standfest and the one on user interaction and learning is based on the PhD of Artem Chirkin. The research presented in this paper was partially conducted at the Future Cities Laboratory at the Singapore-ETH Centre, which was established collaboratively between ETH Zurich and Singapore’s National Research Foundation (FI 370074016) under its Campus for Research Excellence and Technological Enterprise program. The research was also funded by the Swiss National Science Foundation (100013L_149552). Thanks to Julian Reisenberger for the language editing.

REFERENCES


# Author's index

<table>
<thead>
<tr>
<th>Author</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbas, Günsu Merin</td>
<td>185</td>
</tr>
<tr>
<td>Balla-S. Béla, Szilvia</td>
<td>105</td>
</tr>
<tr>
<td>Bertin, Vito</td>
<td>79</td>
</tr>
<tr>
<td>Botzheim, Bálint</td>
<td>213</td>
</tr>
<tr>
<td>Bödő, Gábor</td>
<td>235</td>
</tr>
<tr>
<td>Castellon Gonzalez, Juan José</td>
<td>177</td>
</tr>
<tr>
<td>Chang, Tengwen</td>
<td>163</td>
</tr>
<tr>
<td>Chaszar, Andre</td>
<td>227</td>
</tr>
<tr>
<td>D'Acunto, Pierluigi</td>
<td>177</td>
</tr>
<tr>
<td>Datta, Sambit</td>
<td>163</td>
</tr>
<tr>
<td>De Luca, Francesco</td>
<td>195</td>
</tr>
<tr>
<td>De Paris, Sabine</td>
<td>55</td>
</tr>
<tr>
<td>Dino, Ipek Gürsel</td>
<td>185</td>
</tr>
<tr>
<td>Dumitrescu, Delia</td>
<td>203</td>
</tr>
<tr>
<td>Elkady, Shawkat L.</td>
<td>169</td>
</tr>
<tr>
<td>Ezzat, Mohammed</td>
<td>111</td>
</tr>
<tr>
<td>Fehér, András</td>
<td>235</td>
</tr>
<tr>
<td>Fricker, Pia</td>
<td>119</td>
</tr>
<tr>
<td>Füzes, Bálint Péter</td>
<td>73</td>
</tr>
<tr>
<td>Gidófalvy, Kitti</td>
<td>213</td>
</tr>
<tr>
<td>Gyulai, Attila</td>
<td>67</td>
</tr>
<tr>
<td>Hadzijanisz, Konsztantinosz</td>
<td>235</td>
</tr>
<tr>
<td>Hegyi, Dezső</td>
<td>73</td>
</tr>
<tr>
<td>Heinrich, Benjamin</td>
<td>243</td>
</tr>
<tr>
<td>Iványi, Péter</td>
<td>221</td>
</tr>
<tr>
<td>Kari, Szabolcs</td>
<td>67</td>
</tr>
<tr>
<td>Kikunaga, Patricia Emy</td>
<td>213</td>
</tr>
<tr>
<td>Koenig, Reinhard</td>
<td>15</td>
</tr>
<tr>
<td>Kolarevic, Branko</td>
<td>27</td>
</tr>
<tr>
<td>Kulcke, Matthias</td>
<td>61</td>
</tr>
<tr>
<td>Lam, Wai Yin</td>
<td>79</td>
</tr>
<tr>
<td>Lellei, László</td>
<td>67</td>
</tr>
<tr>
<td>Lorenzo, Wolfgang E.</td>
<td>249</td>
</tr>
<tr>
<td>Lovas, Réka</td>
<td>235</td>
</tr>
<tr>
<td>Lucchi, Elena</td>
<td>155</td>
</tr>
<tr>
<td>Matsubayashi, Michio</td>
<td>87</td>
</tr>
<tr>
<td>Nováková, Kateřina</td>
<td>133</td>
</tr>
<tr>
<td>Nuno Lacerda Lopes, Carlos</td>
<td>55</td>
</tr>
<tr>
<td>Pascucci, Michela</td>
<td>155</td>
</tr>
<tr>
<td>Pletenac, Lidija</td>
<td>141</td>
</tr>
<tr>
<td>Reffat M., Rabee</td>
<td>169</td>
</tr>
<tr>
<td>Reith, András</td>
<td>213</td>
</tr>
<tr>
<td>Riedel, Miklós Márton</td>
<td>67</td>
</tr>
<tr>
<td>Rossado Espinoza, Verónica Paola</td>
<td>127</td>
</tr>
<tr>
<td>Sajtos, István</td>
<td>149</td>
</tr>
<tr>
<td>Sárközi, Réka</td>
<td>221</td>
</tr>
<tr>
<td>Schmitt, Gerhard</td>
<td>15</td>
</tr>
<tr>
<td>Seddik, Moamen M.</td>
<td>169</td>
</tr>
<tr>
<td>Selvær, Harald</td>
<td>99</td>
</tr>
<tr>
<td>Sik, András</td>
<td>67</td>
</tr>
<tr>
<td>Smolik, Andrei</td>
<td>163</td>
</tr>
<tr>
<td>Strommer, László</td>
<td>49</td>
</tr>
<tr>
<td>Sundfør, Ingolf</td>
<td>99</td>
</tr>
<tr>
<td>Surina, Dóra</td>
<td>235</td>
</tr>
<tr>
<td>Szabó, Beatrix</td>
<td>235</td>
</tr>
<tr>
<td>Széll, Attila Béla</td>
<td>221</td>
</tr>
<tr>
<td>Szilvási-Nagy, Márta</td>
<td>105</td>
</tr>
<tr>
<td>Szollár, András</td>
<td>213</td>
</tr>
<tr>
<td>Ther, Tamás</td>
<td>149</td>
</tr>
<tr>
<td>Vári, Barnabás</td>
<td>235</td>
</tr>
<tr>
<td>Watanabe, Shun</td>
<td>41, 87</td>
</tr>
<tr>
<td>Wurzer, Gabriel</td>
<td>243</td>
</tr>
<tr>
<td>Xu, Lei</td>
<td>93</td>
</tr>
<tr>
<td>Yajima, Kazumi</td>
<td>33</td>
</tr>
</tbody>
</table>
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."