Collaborative Design and Live Interaction with Parametric Models using UbiMash

Flora Dilys Salim, Hugo Mulder, Przemyslaw Jaworski, James Willems Ransom and Pierre Cutellic, Takehiko Iseki, Stefan Di Leo and Yelta Köm
Collaborative Design and Live Interaction with Parametric Models using UbiMash
Flora Dilys Salim, Hugo Mulder, Przemyslaw Jaworski, James Willems Ransom and Pierre Cutellic, Takehiko Iseki, Stefan Di Leo and Yelta Köm

Abstract

Due to the complexity of parametric modeling, it has been the task of only a handful of experts in the design team to develop, build and modify parametric models. The complexity of parametric models increases exponentially with the number of design aspects it incorporates. The ambiguity of parametric models towards the modeled design intent explains that modifying the model is often isolated as an individual exercise. Developments in physical and ubiquitous computing, however, allow for collaboration and interaction with parametric models in other ways. Communication, coordination, and interaction between parametric models and the physical and social environment are the context of this paper. The paper describes some of the projects that were outcomes of the SmartGeometry 2010 workshops. These projects are dealing with mass collaboration using Twitter, tangible interfaces, parametric design and construction coordination and geometrical interpretation of datasets.

Keywords: parametric design • interactive parametric model • design collaboration • physical computing • ambient computing • form fostering • UbiMash • SmartGeometry
I. Introduction

The rise of Computer-Aided Design (CAD) has led to widespread adoption of digital modeling. CAD has eventually transitioned from drafting tools to designing tools only in the past decade. The demand for flexible modeling approaches that allow rapid adaptations to design variations to be produced has been accommodated by employing parametric design, either through customization of tools or programming of the model using parameterization, a term introduced by Kalay in 1981 [1]. Pioneering work in computer-aided parametric design includes [2-4]. Parametric modeling was first introduced as a means for design reuse [5]. Parametric modeling uses parameterized relationships between components in the design to define forms [6]. A parametric model comprises variable attributes, which are called parameters and fixed attributes, which are called constraints [7]. Parametric modeling is powerful since design options can be rapidly generated. A parametric model responds to changes of parameter value or definition without the needs for erasure of parts of the model or starting the design model from scratch. Designers alter the value of the parameters to explore design options generated from the same model [7]. Proprietary parametric design software, such as Digital Project (DP) by Gehry Technologies, Bentley GenerativeComponents (GC), and Rhino Grasshopper, has just been introduced in the recent years and it has become a niche within CAD.

However, the uptake of parametric modeling has been arduous given its complexity [8]. As the number of design aspects and behaviors to be captured in the model increases, the complexity of the parametric model escalates exponentially [8]. Therefore, parametric modeling is accessible only to handful experts in the design team. Since parametric models are ambiguous due to the multiple ways of discovering design intent in a single model [8], modifying a parametric model is often isolated as an individual exercise. Round table discussions involving a design team often retreat from the use of digital parametric models. Instead, physical and analogue models are used to engage a collaborative design discussion. The intricacies of parametric modeling require new methods to interface and interact with the objects, associations, and parametric behaviors in the model.

The advances of ubiquitous computing and the proliferation of affordable hardware (sensors, actuators and microcontrollers) has promoted the research and development in ambient intelligence and context-aware computing, where computer and mobile applications associated with personal and wearable devices are adaptive and adaptable to the users and the environment. A vast amount of environmental data is now accessible through the sensors in our world. The mounting availability of Web2.0 Application Programming Interfaces (API), such as Google Maps API, YouTube API, Yahoo API, Twitter API, Flickr API, opens up online data sources and platforms for delivering rich-content user interaction and collaboration.
The current state-of-the-art physical and ubiquitous computing technologies provide new opportunities in extending parametric modeling for design collaboration.

Undoubtedly, there are many facets of collaborative design, such as Building Information Modeling (BIM), model interoperability, and many more. However, the focus of this paper is in leveraging communication, coordination, and interaction between parametric models and the ambient environment and physical devices in the context of collaborative design.

The existing research on tangible interaction with CAD, such as [9-10] and the recent DeskCube by Autodesk Research [11] demonstrated tangible interfaces which are limited merely to perform navigation and modify viewpoints in CAD. The Luminous Planning table by MIT Media Lab and School of Architecture and Planning was developed for urban simulation [12]. The integration between the physical building models and the digital drawing, plans and models allows real-time simulation of sun, wind, and shade being projected on the physical models located on the table. The building models can be moved or rotated to generate real-time feedback of the impact of the variations. However, the physical-digital integration is not feeding into parametric design software, and therefore the modification of the model is limited to changing the location and orientation of the buildings. To our knowledge, the use of tangible interaction, physical sensing, and Web2.0 as part of the tool and medium for parametric design has not been previously addressed elsewhere.

The common parametric design approaches associate digital parameters with virtual data sources or simulated data. The presence of parametric design is limited to the virtual world and the parameters have not been exposed to the physical world. In order to support collaborative design, a parametric model can be extended with ambient computing, physical parameterization and interaction. Form fostering [13], which refers to the activities of form-making and form-finding of the ambient and interactive parametric models, aims to support collaborative design.

2. Form Fostering Approaches to Parametric Design

Form fostering challenges the traditional approach of informing a virtual model purely with virtual knowledge. It seeks to integrate and analyze data from the physical world and capture relations and interactions that exist in the world as physical parameters which constitute to a larger model that is not only virtual. The architectural design models therefore become mixed: partly virtual and partly physical.

Figure 1 depicts a scenario of associations between parameters in a model following the form fostering approach. Figure 1 (1) depicts the traditional parametric design. A parametric model is set up in a digital software environment with a number of virtual parameters. A parameter can be associated with another virtual parameter and informed by virtual events.
Figure 1 (2) illustrates that a parameter can also be linked with the physical environment and receives input from sensor or haptic devices. Therefore the parametric model becomes a vehicle for ambient architectural design. This is useful even in early stage of design, given that real-time ambient information from the site can be streamed as parameters that drive changes to the parametric model. Figure 1 (3) illustrates that a parameter can also be static or constrained if it is informed for example by a static object in the built environment. This can be useful if an existing building or an already built part of the building is to be included in the model. This allows for design to take place in the mixed reality. Also, because the model will exists in various stages of the mixed reality, both the early stages of design (mainly virtual) and the final stages of construction (mainly physical) can be captured in the same model. In responsive architecture the relations between the building and the environment remain dynamic, even when the construction process has finished and the parametric model has materialized.

Form fostering requires interfaces for the mixed reality, both to interact with users as with the environment. The dynamic feedback loop is facilitated through informing the virtual models with changes in the physical world as well as informing the physical objects with changes in the virtual world. Real-time updates in the model can be managed in form fostering since designers can now use sensors, which stream data from the physical environment, as input to drive the parametric variations in the model.

Early experiments of form fostering include the Wii and the Eye, from SmartGeometry 2009 [13]. In these experiments, a Nintendo Wiimote was configured to manipulate parametric models in GC and a motion tracking...
webcam with 180 degrees of freedom around two axes was interacting with parametric models in GC. The early experiments led to the development of UbiMash, given the need for a generic platform that can be utilized for bridging various physical and sensor devices and parametric 3D software.

3. UbiMash

UbiMash software platform [14] is a generic and open interoperability tool for designers to connect any physical and hardware device with CAD software. UbiMash is brief for Ubiquitous Mashups. It facilitates physical devices such as the Wiimote, web cameras, Arduino microcontroller boards, sensors, or systems like reacTIVision tangible interfaces [15] and Web 2.0 APIs (such as for Twitter) to be connected to software, such as Grasshopper, GC and Processing (see Figure 2).

UbiMash is built on the concept of publish-subscribe software architecture (Figure 3). It utilizes the User Datagram Protocol (UDP) and a central server. The server publishes information from various sources, such as sensors, the web, Arduino, or other physical interfaces. Parametric software as well as physical devices can subscribe to the server to obtain specific real-time updates. The UDP connection can be programmed as a custom feature, plug-in or script in association with software or hardware interfaces. The software architecture is extensible, flexible and allows integration with any platform that supports the UDP network protocol.

Publishers are applications on the same machine or connected on the same network that publish data using the UDP protocol to UbiMash central server. Examples of publishers are Processing, Arduino, GC, tangible tables, Web 2.0 clients, sensor networks, and physical models that are attached with sensors. Subscribers are applications that subscribe to receive updates.
from UbiMash central server about a particular message. Examples of
subscribers are GC, Rhino, Grasshopper, Arduino, physical models that are
attached with actuators. Note that a system in the UbiMash networked
environment can either be a publisher, subscriber, or both. When an
application or device wants to publish a variable with other programs via
UbiMash central server, it sends a query string such as the following:
“Publish, myVar, 0.005” using the UDP protocol. The central server
broadcasts the variable and its value to all subscribers of myVar variable. In
order to subscribe, an application needs to send a query string such as the
following “Subscribe|myVar” to the UbiMash central server. Applications that
are connected to UbiMash and subscribed to myVar variables will receive
updates whenever the publisher of myVar sends the new value.

This is illustrated by the following code (written in Processing):

```java
void publishVar(String name, int variableValue)
{
  String message = "Publish,";
  message += name + "," + nf(variableValue, 0, 0);
  // send the message
  udps.send( message, ip, port );
}

publishVar(“myVar”, 0.005);
```

In GC for example, a GC graph variable can be associated to a variable
published by the server. UbiMash publisher and subscriber plug-ins are used
to load up script functions for communication with the UbiMash central
server. If a GC feature is associated to a graph variable and if the graph
variable is subscribed to a variable published via UbiMash central server,
whenever the publishers send new values of the variables, the feature which
have been subscribed to receive updates from UbiMash server will be
automatically updated. In this example scenario, a b-spline curve feature,
namely BSP, is defined by points, which are associated to a graph variable,
namely ptList. Using the UbiMash subscriber plug-in, ptList is subscribed to
myVar and whenever a new value of myVar is published, b-spline curve BSP
in GC is updated.

UbiMash was introduced prior to the SmartGeometry 2010 workshop
where a number of form fostering experiments with UbiMash took place.
The software was released as an open source in order to facilitate a
growing community of users, contributors, and practitioners who seek to
design physical interactions with parametric models.
Table 1. Experiment Matrix

<table>
<thead>
<tr>
<th>Category</th>
<th>Project Name</th>
<th>Main Collaborator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Mass collaborative form finding using social networking</td>
<td>TweetForm</td>
<td>Author 4</td>
</tr>
<tr>
<td>2. Tangible interfaces for parametric design collaboration</td>
<td>Ur-moeba Table</td>
<td>Author 5</td>
</tr>
<tr>
<td>3. Parametric design and construction coordination</td>
<td>Rapid design coordination</td>
<td>Author 6</td>
</tr>
<tr>
<td>4. Sensor-based form fostering</td>
<td>Subjective Space Scanner</td>
<td>Author 7</td>
</tr>
<tr>
<td></td>
<td>Occupant Motion Tracker</td>
<td>Author 8</td>
</tr>
</tbody>
</table>

4. Interactive Experiments with Parametric Design

The experiments (Table 1) were designed and developed in a four-day intensive workshop about Parametrics and Physical Interaction, led by the first three authors of the paper. The workshop was part of SmartGeometry 2010.
4.1. Mass collaborative form finding using social networking

TweetForm was first conceived and developed during the workshop and is an ongoing experiment that examines the potentials of newly accessible social networks such as Twitter to influence and inform parametric design methods. It utilizes the power of Twitter as a social networking platform to inform a parametric model in Rhino Grasshopper. This enables **crowdsourcing**\(^1\) for collaborative form finding. Therefore, form finding activity is no longer restricted to individuals. Instead, a much larger community can access the form, suggest modifications online, and preview the variations to occur in the form as suggested.

TweetForm advances traditional parametric design by expanding the range of parameters traditionally considered within architectural parametric design methods. Specifically, it attempts to draw linguistic parallels to conditional organizations by opening a wide range of new communications for collaborative design. Additionally it opens new access for ubiquitous and mobile devices to engage in crowdsourcing parametric design.

Collaborative design of the parametric model is made simpler in TweetForm by integrating a natural language processor. TweetForm acts as the negotiator between defined inverse relationships. These relationships are defined in natural language. TweetForm uses the Twitter API through an open Java library (Twitter4J) to perform keyword searches. An initial search for the specific keyword ‘PPIWorkshop’, which refers to a specific Twitter account and hash tag, is performed and all tweets containing the keyword are pulled. A second search is then conducted within the retrieved tweets for natural language keywords which are associated with specific parametric behaviors to influence the form. For example, ‘taller’ and ‘shorter’ play against each other. A string of commands to make the model ‘taller’ will counteract the same number of commands to make the model ‘shorter’ in a linguistic tug o’ war. A set of these inverse keywords can be thought of as analogous to a number slider. In this way TweetForm acts as a threshold or a series of thresholds between formal extremes. The parametric behavior descriptor engine of TweetForm is an integrated process of data analysis from the Twitter engine, Processing 3D script, and natural language processor, to a Rhino Grasshopper parametric model (Figure 4).

We have chosen keywords that are not technical or specific to the operations of the parametric model (Figure 5). For example, instead of saying ‘increase polygon segments’ we chose ‘rounder’. In actuality nothing is getting more round but the effect of increasing the number of polygon segments within the model gives the effect that the form is getting rounder. The language was chosen to describe characteristics of form and not how the model was constructed. The exact parametric relationship between

---

\(^1\) Crowdsourcing (a compound word of ‘crowd’ and ‘outsourcing’) was first coined by Jeff Howe [16], who defines it as an act of outsourcing the tasks of developing new technology or application to the crowd through an open call.
language and geometry is currently defined explicitly by the authors of the parametric model.

Twitter itself draws a certain demographic. The geo-tagging feature of Twitter as well as the ability to scan Twitter streams of time-zones and currently six different languages may become instrumental to this project in the future for exploring demographic trends related to form. TweetForm also raises questions as a democratic design environment. As the number of votes/tweets grows the amount of influence of each vote grows less significant. After pooled with thousands of commands the user is less able
to understand the influence of their participation and the logic of the overall form. It is worth further exploration into more sophisticated linguistic relationships. Instead of ‘taller’ the semantic expression ‘much taller’ or ‘a little taller’ begins to imply magnitudes of change and different weights to the vote contribution.

TweetForm models were streamed online during the system testing, and projected in one of the public spaces in the workshop venue in order to display the live updates from the Twitter stream (Figure 6). At present, TweetForm models are viewable through an online applet and visual in nature (Figure 7). This visual feedback is limited in comparison to the mobile accessibility of Twitter and raises the need for a more sophisticated visual representation platform for mobile devices.
4.2. Tangible Interfaces for parametric design collaboration

The Ur-moeba table top design interface demonstrated an application for tangible interfaces with urban planning. It aims to enhance the versatility of conventional CAD software as a design tool by connecting it to a more interactive and dynamic environment and enhance it for a specific simulation scenario. In this context, it demonstrated a generative urban simulation.

In the set up of the Ur-moeba table, Processing is connected to GC to add time scale to a generative urban simulation. The reacTIVision TUIO [15] plugin is connected to the platform to perform visual recognition of fiducial markers and monitor their positions and rotations on the table. Participants touch and move the markers on the table to drive a generative process of growth, where city cells behave differently depending on the position of attractors. In this set-up there are three types of attractors, *city*, *village* or *industry*, each with their own growth pattern. These patterns can be adjusted.
by rotating dedicated markers to change the properties, such as speed of growth and sensitivity towards neighbors. As a result, a responsive design platform is created, where despite of complex chains of information exchange, users can access the interface easily and manipulate the urban cells in an intuitive way, without any technical knowledge (Figure 8). The Processing application uses cellular automata to organize the various types of cells. This network of relationships is updated continuously. The information is published to the UbiMash publish-subscribe server (Figure 9). A parametric city model in GC is subscribed to the server and delivers a three dimensional snapshot of the growing city (Figure 10).

This prototype enhances the parametric design experience in a number of ways. Firstly, the tabletop design platform provides parallel and simultaneous access to the design process. Non-expert access to the parametric model is enabled via the tangible interface. Since the interactive interface does not require contact with a mouse device, design discussions around the model can occur more intuitively. Secondly, mapping between 2D graphics on the table and a 3D urban model in parametric CAD allows concurrent updates of digital parametric models as physical object markers are modified. This leverages communication and design exchange. It also provides the opportunity to develop the ideas testing it directly on the table. Finally, the customized software platform, which is an integration of Processing, UbiMash, and GC custom plug-ins, enhances processing speed, particularly in dealing with complex interactions of multiple agents. The graphic representations are flexible and accessible to non-technical users.
Real-time updates are accommodated. Therefore, it reduces the need for interoperability on file sharing level.

4.3. Parametric design and construction coordination

This project delivers a proof of concept of a design communication interface, which is aimed to synchronize various phases of building designs. Traditionally, the development of an architectural project is conceived in a linear and unidirectional fashion practically, contractually, and organizationally. From Schematic Design (SD), to Development Design (DD), to Construction Design (CD), and eventually, the physical realization of the project, the scope for design interventions and flexibility are reduced by the scale of complexity. The design process has been systematically arranged in this very linear process and every step is adjusted to fit a set of local constraints that form parts of the global digital ecosystem of the building industry (Figure 11).

Parametric design provides the answer to the needs for variations and for rapid adaptation to constantly evolving constraints during project development. The state-of-the-art Building Information Modeling (BIM) technology provides an interoperability platform to share a digital model for an Integrated Project Delivery (IPD). However, there are still gaps between various stages of designs in which changes in between need to be coordinated and synchronized.

The Rapid Design Coordination (RDC) system was proposed to promote an integrated and bidirectional live update between physically built components and the associated parametric models to efficiently answer the needs for synchronizing variations between fabricated design and digital design that are yet to be built. The system allows real-time feedback from a construction site (physical model) to a parametric architectural (virtual) model of the building. Physical models can be compared with the virtual model. Variations which are part of the design environment can be synchronized accordingly.

This novel system provides a continuous loop of production, bringing light to new potentials to a design practice that can be perceived as a single process from conception to construction (Figure 12). RDC catalyses several software and hardware dedicated to applications not related to the building industry. An initial system was first realized during the workshop by
integrating efficient algorithms, ambient environmental parameters, physical computing interfaces, effective communication protocols and parametric BIM.

The prototype consisted of a virtual model in GC and a physical representation of that model equipped with Tangible User Interface Object (TUIO) markers (Figure 13). First, a set of tangible markers was generated and indexed on a table interface. These markers, which can be attached to physical models, were to be linked with virtual objects later in the process. The locations of these markers were captured by a camera. The video was filtered to match the marker detections with the corresponding index using the TUIO platform. The filtered data was analyzed in Processing and published via the UbiMash central server. A 3D parametric model that subscribed to receive updates of variables was established. Whenever the corresponding data was updated on the table, the parametric model was updated simultaneously.

In the first experiment, using the reacTIVision library [15], the position and orientation of the markers were monitored in Processing. In order to
feed the data directly to the parametric model in GC, UbiMash was used. As the physical wireframe model was modified (Figure 14), the variations of the tangible markers were detected, and the virtual model in GC was updated to reflect the changes in the physical model. The process is relatively straightforward and adaptable for design coordination in various architectural design, engineering, and construction scenarios.

This experiment has proven a real potential in this system given its portability, accessibility and flexibility. Further development and refinement of RDC is currently underway, with aims to transform design process and construction projects by providing real-time feedbacks from construction sites. This will eventually accelerate quantity surveying and refine the project process from conception to construction within a single loop.

4.4. Sensor-based form fostering

Two projects in the workshop collected sensor data to be used for form generation. The Subjective Space Scanner used proximity sensors, a digital compass and an Arduino microcontroller board to collect spatial data. The Occupant Motion Tracker used an Eye Toy camera and motion tracking software to collect pedestrian traffic data.

Subjective Space Scanner is a device that senses and records the space, the context, and the boundaries of our personal environment (Figure 15). Coupled with a timer, spatio-temporal data that can be associated with daily patterns can be captured. As shown in Figure 16, it was used to record personal space in a student housing complex and associated with the hours spent for various activities in different spaces. The 15 Westminster building was an office building until the office topology changed and was later renovated to student housing. The spatio-temporal mapping device has revealed the latent potentials for understanding occupancy schedules for the purpose of retrofitting. Based on the data collection over a week, 43 hours were spent in the living room (for studying, relaxing, exercising, and...
miscellaneous activities), 12 hours were spent in the kitchen, 52 hours spent in the bedroom (49 hours for sleeping and 3 hours for miscellaneous activities), 2:30 hours spent in the shower, and 1 hour was spent in the washroom (Figure 16). Other recorded data includes movement along the corridor, elevator and the porch of the apartment. Upon integration with a parametric model, the data can be further analyzed to generate optimal conditions for configuring accommodation plans for specific occupancy types and schedules.

The device consists of sonar sensors and a digital compass connected to an Arduino microcontroller board (Figure 17). The scanner can be held in a small backpack to take samples of personal space and movement over a period of time (Figure 18 bottom). The data, which comprises of positions,
maneuvers and boundaries, was read into GC using UbiMash and GC script. An algorithm was developed to derive the geometry based on the data recordings. The 3D geometry was also rendered in GC (Figure 18 top).

Occupant motion tracker used a camera right above the workshop area to record the motions generated by the occupants of the space. The aim of the project was to depict people movements and their relationships as a measure of their personal space and use the spatial relationships to inform a geometric parametric model. The geometry would therefore result that changes with time, adapting to new situations or changing requirements. The environment that reflected the ‘now’ informed a digital form that was fostered with physical data. And an orchestration of space through emerging
patterns was formed unconsciously by many individuals. The use of motion detection to inform a parametric model supports the design and configuration of space with ambient data.

In this project, the motions, not the videos, were recorded using frame-differencing and blob-detection in Processing. Larger clusters of motions were grouped together, in order to distinguish movements of individuals. This data was fed to GC using UbiMash (Figure 19). In GC, a triangulation algorithm was used to produce a 3D geometry that visualizes the sampling of the data that is harvested every 30 minutes (Figure 20).

With motion capture, blob detection and gesture recognition techniques can be incorporated as data that feeds to an interactive parametric model or a built façade.

5. Conclusion

This paper has demonstrated that parametric modeling can be further enhanced for design collaboration by integrating social networks, ambient intelligence and tangible interaction in the design process. Digital prototyping is not only limited to manufacturing of physical static models, but can be expanded to include dynamic interactive and adaptive architectural installations, multimedia and various networks of sensors and communication devices. This opens up a new path to intelligent architecture, which is sensitive to changes in environment, energy use and social information.

By using the UbiMash software platform, a number of software and hardware platforms in their projects can be easily integrated. Its early versions clearly served a purpose and took away many of the problems encountered by designers as they start to work with code. Development of the software will focus on ease of use and documentation, on other
transferable datasets for the UDP server and the development of plug-ins ready to use in other software.

The projects presented in this paper show a number of practical examples that demonstrate the use of this technology for practical design applications. These prototypes can be developed for further use in the building industry at various stages during design and construction.

The TweetForm project shows the use of social networking for form finding. In a very practical application, this could be used to influence strategic decisions about grand projects like the routing of a high-speed rail line, or the expansion of an airport.

The two projects in this paper that employ tangible interaction demonstrate its usefulness as a means of collaboration in round-table discussion settings. Ur-moeba table shows its potential as an interface that facilitates direct feedback on a complex simulation process. Rapid Design Coordination demonstrates that coordination and change management in various stages of building design to construction can be dealt with in a far more intuitive manner.

The data collection examples demonstrate that ambient data can be incorporated in the design process in order to inform a responsive parametric model. Sensing, motion capture and pattern recognition can provide contextual knowledge about the occupancy and the environment of the sites we are designing for. Potential future directions include experimenting with sensor data of existing Building Management Systems.

Most of the current work presented in this paper assumes that a parametric model exists. Future work will focus how objects and relationships in a parametric model can be defined in the first place using ubiquitous computing technology.

**Acknowledgements**

The authors would like to acknowledge the SmartGeometry group, SmartGeometry Conference 2010 organizers, Bentley Systems, IAAC staff and students for the amazing support for the Parametrics and Physical Interactions (PPI) cluster in SmartGeometry Workshop 2010. We want to express our gratitude to our collaborators and other participants of the PPI cluster, Rafael Urquiza, Konstanze Grammatikos, Filipo Ferraris, and Hans-Georg Bauer who have also produced other interesting responsive architectural experiments, as shown in UbiMash.com. We also acknowledge the support from our academic and practice offices, SIAL (RMIT University), Arup, Foster+Partners, Gehry Technologies, University of Buffalo, Rhode Island School of Design, and Yildiz Technical University.
References


Flora Dilys Salim  
Spatial Information Architecture Laboratory (SIAL), RMIT University  
Australia  
Email: flora.salim@rmit.edu.au

Hugo Mulder  
Arup Advanced Technology + Research  
United Kingdom  
Email: hugo.mulder@arup.com

Przemyslaw Jaworski  
Jawor Design Studio  
Poland  
studio@jawordesign.com

James Willems Ransom  
University of Buffalo  
United States  
Email: willransom@gmail.com

Pierre Cutellic  
Gehry Technologies  
France  
Email: pierre@gt-eu.com

Takehiko Iseki  
Foster + Partners  
United Kingdom  
Email: tiseki@fosterandpartners.com

Stefan Di Leo  
The Rhode Island School of Design  
United States  
Email: stefan.dileo@gmail.com

Yeltaş Köm  
Yildiz Technical University  
Turkey  
Email: yeltakom@gmail.com