pCOLAD: online sharing of parameters for collaborative architectural design

Hans J.C. Hubers[1], Michela Turrin[2], Irem Erbas[3], Ioannis Chatzikonstantinou[4]
[1,3]Faculty of Architecture, Delft University of Technology, PO Box 5043, 2600 GA, Delft, The Netherlands
[2]Faculty of Architecture, Delft University of Technology & Faculty of Architecture at Yaşar University [4]Faculty of Architecture at Yaşar University, University Street, No:35-37, Ağaçlı Yol Bornova, İzmir, Turkey

[1,2,3]{j.c.hubers|M.Turrin|I.Erbas}@tudelft.nl [4]i.chatzikonstantinou@yasar.edu.tr

Simultaneous interdisciplinary architectural design from the very start of a project faces challenges in properly sharing information across disciplines. This research developed a method and related digital tool to improve collaborative design and aimed at making selected information to be shared faster and more transparently. The method consists of developing alternative parametric solutions for different parts of the design in such a way that crucial parameters form a link between these parts. The digital tool has been developed for Grasshopper and permits synchronic (real-time over the Internet) and a-synchronic sharing of these parameters. The design alternatives are evaluated with specific criteria, pros and cons in an Internet Forum and discussed via a video-conferencing tool. Decisions are then taken in a collaborative manner through voting. The paper describes the method based on a case study.

Keywords: *Parametric, collaborative, design, plug-in, stadium*

INTRODUCTION

Collaborative architectural design is simultaneous interdisciplinary design from the very start of a project. It is also called co-design or concurrent engineering. There are several reasons for collaborative design (Hubers, 2009). The two most important are:

- The big influence of decisions in the beginning of the design process on the cost/quality ratio of the final product.
- The potential contribution through knowledge and experience of all stakeholders at this beginning.

Despite these important reasons, still there are no generally accepted methods to support collaborative architectural design teams at the very start of a project. Current BIM principles are mostly intended for advanced phases of design; the early stage of design is mostly based on geometric modellers that do not support interdisciplinary collaboration. This research aimed at developing a collaborative method for early design phases. Considering the large diffusion of parametric design and its potentials in forcing the formulation of design goals and abstract design models (Turrin, 2014), this research intended to couple concurrent engineering and parametric design.

The objective of this work is to find out if a collaborative architectural design team would work faster and more transparently if they could share crucial parameters, both synchronically (real-time over the Internet) and a-synchronically.

The research questions were:

1. How can a collaborative design process be facilitated in a parametric design setting?
2. What is the best format for the parameters and their values?
3. Where should the parameters be stored?
4. Which system to use for multiuser real-time updating?
5. Which system to use for discussion and voting?
6. Use existing Grasshopper components or plug-ins or make our own in VB.net?

The methods used to answer the research questions are literature study, software prototyping and a case study. The literature study was used to answer question 1, especially to find the state of art in BIM, collaborative and parametric design.

BIM

Recently Building Information Modelling (BIM) was introduced. The BIM handbook of Eastman et al. gives a good overview of the advantages of BIM (Eastman et al., 2008). Benefit 9 in Table 1 'Earlier collaboration of multiple design disciplines' is in the Construction execution/coordination phase and thus not collaborative design as defined in this research. The challenge is to develop methods and tools for collaborative architectural design in the concept design phase.

Feasibility study	
1	Support for project scoping and cost estimation
Concept design	
2	Scenario planning
3	Early and accurate visualizations
4	Optimize energy efficiency and sustainability
Integrated design/construction	
5	Automatic maintenance of consistency in design
6	Enhanced building performance and quality
7	Checks against design intent
8	Accurate and consistent drawing sets
Construction execution/coordination	
9	Earlier collaboration of multiple design disciplines
10	Synchronize design and construction planning
11	Discover errors before construction (clash detection)
12	Drive fabrication and greater use of prefabricated components
13	Support lean construction techniques
14	Coordinate/synchronize procurement
Facility operation	
15	Lifecycle benefits regarding operation costs
16	Lifecycle benefits regarding maintenance

Table 1
Advantages of BIM (Adapted from Eastman et al., 2008, p. 321)

Earlier PhD research developed a method and prototype for collaborative design in virtual reality with multi-player game development software, where team members worked at different versions of the building concept (Hubers, 2008). They could participate in, comment and evaluate each other's versions. That research had to conclude that the tools worked fine, but that the proposed method didn't work, mainly because the advisors lacked knowledge about conceptual design and 3D software. It was proposed to develop interconnected parametric software and use video conferencing and an internet forum. Team members should use their own familiar software. It has been demonstrated that full parametric design has many advantages for collaborative design (Hubers, 2010). An important one is that changes can be processed until the end of the design process because the design can be regenerated within minutes.

PARAMETRIC DESIGN SOFTWARE

Parametric design software is CAD software in which objects and their attributes remain variable. E.g. the width of an object is stored in a variable "W" and not in a constant "10". When the value of "W" changes, the object and it's connections update. This can be very efficient in collaborative design, where often changes in the design have to be modelled in order to be evaluated with the software of other team members. There are several parametric software applications already: Digital Project/Catia (Gehry Technologies), Generative Components/Microstation (Berkley), Grasshopper/Rhino (McNeel), and Autodesk is developing Dynamo for Revit. For this project Grasshopper was chosen because it is widely used in universities.

Three principles of parametric modelling are recalled here following, since relevant for the work described in the paper. The first one is the need of readability of the model, including its parametric definition and geometric associations, to be achieved by means of well-structured and ordered components. The second one is the need to avoid redundancy. This means that the same parameters should not be (re)defined on different places, because if the values of those parameters have to be changed, it is easy to overlook some occurrences, leading to discrepancy in the solution. The third one is the need of properly choosing the input parameters and the parametric logic. This is because the input parameters have large impact on the solution space of the model and therefore on the design alternatives that can be actually generated. Different geometric associations may lead to the same output geometry, but some may be more appropriate for the design process than others. Preliminary understanding which ones are more appropriate is important since revising the parametric logic may imply rebuilding the model (with consequent loss of time).

THE PROPOSED APPROACH

An extensive study of the work of Foqué, Shön, Akin, de Jong and van der Voort, Moughtin et al., Stellingwerff and Hamel, to name but a few exponents, leads to the conclusion that the two most relevant processes in collaborative architectural design are the iterative development and evaluation of alternatives in accordance with criteria (Hubers, 2008). This is confirmed by the standard work of Lawson (2006), provided that creation is seen as the combination of analysis and synthesis (Figure 1).

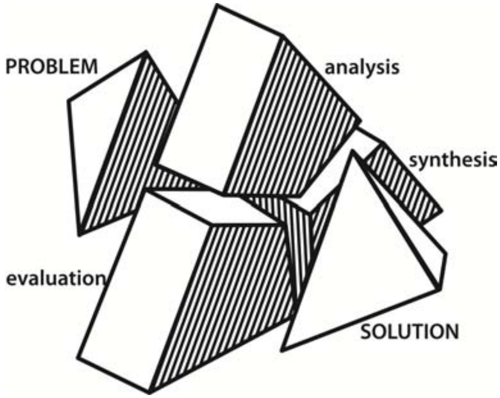

Figure 1
Design process (redrawn from Lawson (2006))

Both sub processes, creation and evaluation, should become better and faster, because a collaborative design team is expensive. As explained above, this could be the case if linked parametric solutions can be developed for complex buildings that are to be designed several times with anticipated differences. The project developed a method and a software tool to support this.

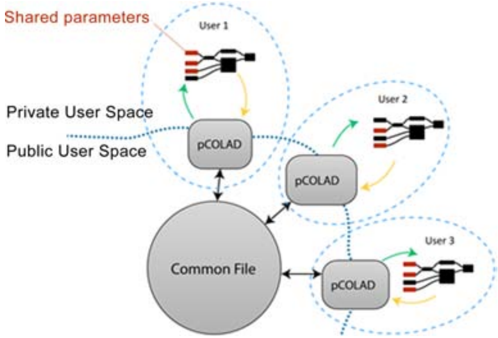

Figure 2
Overview of the pCOLAD method

THE METHOD

The method addresses the workflow across various disciplines during the design process. In order to overcome issues related to limited software-interoperability, it proposes to share information via exchange of parameters next to occasionally exchange of geometry. In order to avoid overflow of interdisciplinary information, it tries to structure priorities based on which information can be selected before being shared (Figure 2).

According to the proposed workflow, each discipline works on an individual parametric model, in what can be named "private space" of the workflow; while a core model and a set of selected information are shared across all disciplines, in what can be called "public space". The private space consist of individual models including individual geometric models and individual parameters. Individual models refer to the core model as main reference, but can be used to freely explore a number of parametric design alternatives, without necessarily sharing these explorations (geometry and parameters) with other disciplines. A user may in fact wish to have some totally private space in which can be experimented with new parameters without bothering the other team members. The public space consists of a core model including shared geometry and shared parameters. While the geometry of the core model is updated only once in a while, shared parameters are meant to be the main stream of interdisciplinary communication, according to the logic described here following. Information that is actually relevant for other disciplines should be shared in the public space, by means of parameters. Specifically, during individual explorations, as soon as the parameters of a user become a more consistent choice, then the team members may need to be "informed" about this. The public shared space mainly consist of parameters and their values. The latter are to be interpreted at the private user space, after of course appropriate "binding" in the private space, which is a responsibility of each user. In this respect, there is not a single state in the public space. There are multiple co-existing states and each user is free to (partially) apply them to their models, contribute new ones and comment. On the other hand, for reference, a master state should be maintained, including all the shared parameters of the project and their values. The master state in the public space should be chosen based on common interdisciplinary agreement, through explicit evaluations and consequent expression of preference from the point of view of each discipline. This leads to a decision making process based on voting.

THE TOOL

Based on the method described in the previous section, a tool is proposed. The tool is structured based on the following parts:

- A text file stores all the records of the shared parameters in the public space.
- A plug-in named pCOLAD for the parametric modeller which manages the records by writing/retrieving and editing records, associates each record with the user that contributed it and collects the voting when required.
- An on-line application is used for synchronization.
- A forum is used for supporting interdisciplinary communication and debates.

Rhino and Grasshopper were chosen as parametric modeller. Dropbox was chosen as a real time file update platform for synchronization, because it is widely used for multiuser real-time collaboration. After some successful tests it was decided to use Blackboard Collaborate for discussion and voting. The reasons for this choice were that up to six video connections can be displayed simultaneously, desktop and application sharing works good and it is available on the LAN of the university with good support. Blackboard Discussion Forum is used for listing the criteria and pros and cons of the alternatives.

Before the tool could be programmed several questions had to be answered. What is the best format for the parameters and their values? Where should the parameters be stored? Which system to use for multiuser real-time updating? Given that Rhino and Grasshopper were chosen to implement the tool, should existing Grasshopper components or plug-ins be used or should new ones be made? The *.csv (Comma Separated Values) format was considered, because it is easily readable in a spread sheet and in Grasshopper. But also the *.json (Javascript Object Notation) format, because it can easily be linked to a parameter class and allows for multiple levels of data nesting. Later it was found that Revit also uses a text file for shared parameters, however with even less information about the parameters than pCOLAD. Still it could be interesting for further developments.

Because an overview of the parameters was needed with several interaction possibilities it was decided to develop two alternative approaches with MS Visual Studio. One in vb.net and one in C#. The first one used a ListView that displayed the information in the shared *.csv file. The second used a GridView that displayed user-generated parameter sets, each containing a user defined set of parameters in the Grasshopper solution. In another GridView the shared parameters of the selected set were displayed. The sets were stored in *.json format also on DropBox. Extra GridViews were added for voting and commenting on the parameter sets.

After a first demo and hands-on exercise the version in vb.net was chosen. The main reasons were that this prototype had more functionalities and was more flexible in the connection to the parameters in the solutions.

FINAL RESULT

The developed prototype consists of three custom components in the Grasshopper environment. pCOLLECT collects information about a parameter in the private Grasshopper solution and outputs it to pSHARE in the *.csv format. pSHARE combines all the pCOLLECTs output with in the *.csv file and provides editing functions. The pPARAM component is used to get a shared parameter from the output of pSHARE (Figure 3). Some of the important functions of pSHARE are:

- Warning if somebody changed the shared *.csv file while pSHARE is active. And forcing to start over again. This is necessary in order to avoid overwriting information that is not yet shared.

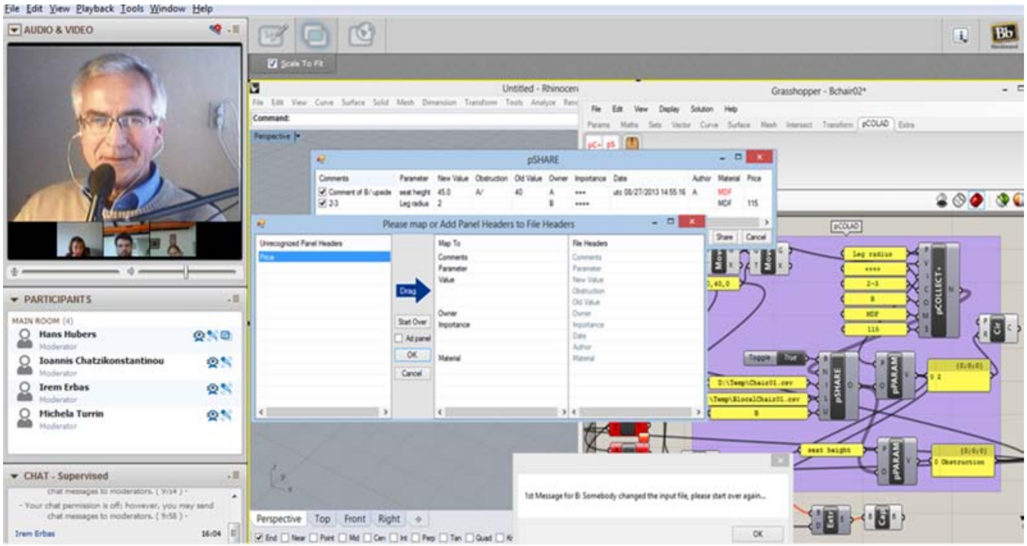

Figure 3
Some functions of pCOLAD in a still from the Blackboard Collaborate recording

- Blocking of parameters in the output if team members don't agree with a parameter value.
- Display in red the changes since last time the shared *.csv file was loaded. This was realised by copying that file from the DropBox to a local path and comparing it with the last copy. This also solved the problems when several team members worked at the same time with that file.
- The standard attributes of parameters are: Comments, Parameter (name), New Value, Obstruction, Old Value, Owner, Importance, Date, Author. The latter two were used to generate a history of sharing.
- Later also adding or removing attributes to parameters through pCOLLECT+ was made possible. The consequence was that mapping became necessary if those attributes were not recognized as existing attributes.
- Of course some fool proofing is needed. E.g. to check if a parameter name is already used, because the name is used by pPARAM to find the attributes of the parameter in the pSHARE output.

A CASE STUDY

The prototype of pCOLAD was evaluated with the collaborative design of a hypothetical Ice Stadium, used as a test case. For the test, the design focused on a limited amount of interdisciplinary design requirements. An architect focused on design of the tribunes, a structural designer on the roof's structure, an envelope designer on the roof's cladding and a climate designer on the optimisation of PV panels versus daylight openings. Two weeks before the test design a demo of 45 minutes was given of the whole system and a hands-on training with the design of a tent for also 45 minutes. A very short brief was mailed some days before the first meeting on Blackboard Collaborate stating that a consortium of building companies is designing an Ice Stadium in the middle of the Netherlands, in the polder. It is part of a competition where the brief says that the main ice rink should have 15.000 spectators. The focus is on the tribune and the roof. During the test some changes in the brief were simulated: e.g. because of information about the concurrent designs the number of seats should be augmented to 20.000. It was asked to imagine that the team members belong to different commercial companies. They don't want to give away their parametric models into which much time and knowledge is invested. In this research it was assumed that they only want to show (intermediate) results during Blackboard Collaborate meetings and share some crucial parameter values. Mostly, parameters referred to independent variables of the parametric models (type 1); however, also output values (type 2) and geometric entities (type 3) were used as parameters shared via pCOLAD. In order to test limits and potentials, participants were asked to not exchange geometric models and not load geometry in the geometric core model until the end of the test.

The architect started with a concept for the tribunes. Since it is an important demand to have seats with good view on the ice rink, the seats are represented as points on curves which are projected on the tribune. This makes it possible to have a visual control on the number of seats. The tribune is made by a sweep using the circumference of the ice rink as a rail and a section line made of threads of 1m and augmenting rises starting with 45 cm as a profile.

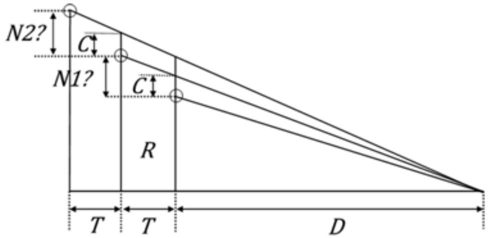

Figure 4
Calculation of the C value

The augmentation of every rise is such that the formula of good view at the top row (C=D(N+R)/(D+T)-R) has a value of at least 9cm (Figure 4), which is the space between the line from the eyes of two spectators to the nearest border of the skating rink at the eyes of one in front.

Later it was changed into calculation with a visual basic script of the rise of every row, such that the C-value of every row is 9cm. This resulted in a much lower tribune. Paths and entrances are subtracted from the seats. The remainders are divided in equal parts of at least 60cm per seat. Then as a check the solution counts these parts. If there are not 15.000 or a little more, then the number of rows can be adapted with a slider. Paths are planned such that no more than 25 seats are on a row. The gross number of seats and the net value (so paths, entrances and facilities subtracted) are shown in the model as text (Figure 5).

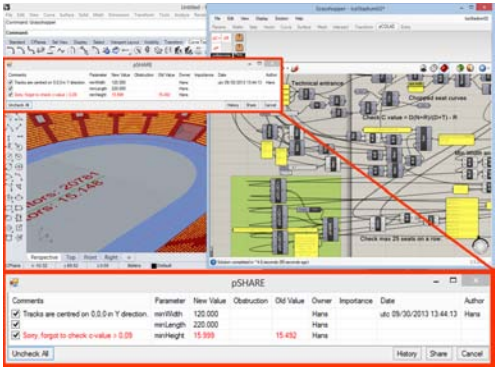

Figure 5
First sharing of parameters by the architect

All this makes a changing height, width and length of the tribune, which of course is important for the roof structure. So as a start these output values are the crucial parameters that were shared.

This exemplifies type 2 (output parameters). The number of rows and the dimensions of the rink are examples of independent input parameters (type 1).

For the climate design the main targets were identified as maximizing indirect daylight, because direct light would have a negative effect on the ice. It also reduces electricity use for lighting. At the same time a maximum of PV panels should be placed without casting shadow on each other, with optimal inclination and orientation to produce electricity with a maximum yield. Due to the limited time, daylight effect could not be quantified. The focus was put on the PV panels: the module area, module orientation, module inclination, type of PV and yearly sum of global horizontal radiation of the location were the parameters to calculate annual yield in Excel. The first three of these parameters were the crucial parameters to be collected and shared with the other stakeholders. A data link was made in Excel that updates every minute to the shared parameters in the *.csv file and warns through conditional formatting in red when values of the selected parameters change. For the roof envelope of course the structural system is important (Figure 8). The architect imagined trusses in symmetrical diagonal standard steel tubes with a maximum length of 12m, then designed by the structural designer. Instead of interfacing the trusses and the envelope based on numeric parameters, only the top curves were shared via pCOLAD and the *.csv file. But how to put a curve into a *.csv file? The architect suggested to give the curves a name and a value consisting of x,y,z of as little points as possible that could be used to regenerate the curves. By separating the x,y,z values with forward slashes (/) a text string is made that represents a curve. In Grasshopper a Split String component can then be used to get the points again as input for an Interpolation Curve component to regenerate the curves. In this way, the curve could serve as a shared parameter. This exemplifies type 3 (geometric parameters).

During the first meeting in the Blackboard Discussion forum also the starting criteria were set: Cost, Ease of construction, Good sight from everywhere, Optimum balance of electricity use versus daylighting, Sustainability in general; specific aspects of indoor climate control were not included for the test. The second meeting the consequences of augmenting to 20.000 seats and a presumed maximum span of the trusses of 125m were discussed (Figure 8).

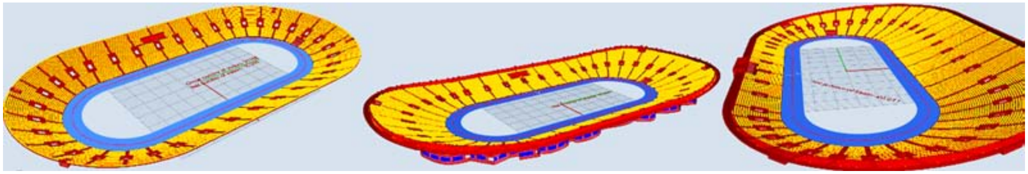

Figure 6
Alternatives A, B and C.

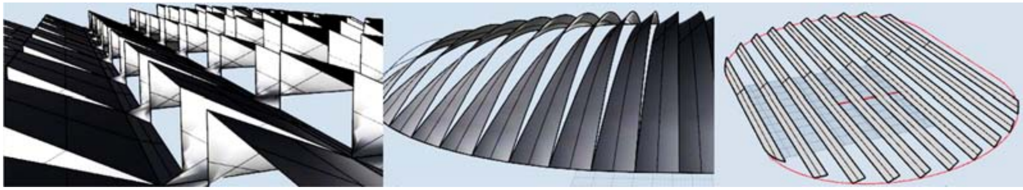

Figure 7
Alternatives 1, 2 and 3.

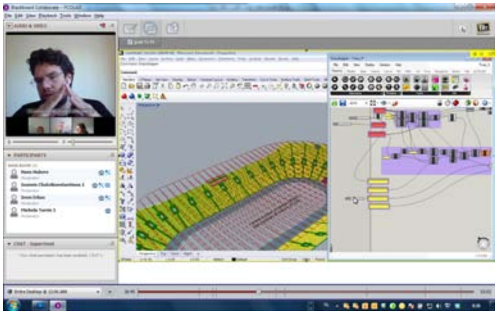

Figure 8
Combination of Trusses and cut-off tribune

A solution would be to chop off the sides of the tribune and add more rows to the ends. This leads to a kind of bathtub form. The architect was not satisfied with the kink in the top border and later used a rounded form for the Boolean difference operation that was used for the chopping off.

Based on the inputs regarding tribune and trusses, the design of the envelope tested the use of pCOLAD in case of (simplified) design alternatives. For the PV panels on the roof it was found that an angle of 36 degree and orientation 5 degrees to the south would be best with windows facing north with louvers in between. After several discussion threads in the Blackboard Discussion Forum and some e-mails there was a choice to be made between alternatives: A) Regular; all rows same total height, B) vertically chopped; in order to keep within 125m width, C) inclined chopped; in order to accommodate 36 degree PV panels on the roof (Figure 6). To be combined with 1) PV/Glass units, 2) PV panels in width direction 3) PV panels in length direction (Figure 7). And also a decision had to be made if the roof should be flat. After some calculations it was obvious that C3 would be best for PV, but not for the indoor climate (a-symmetric conditions for skaters) and also it would not be possible to use standard 12m tubes for the trusses. After investigating all pros and cons related to the criteria, that were set at the start, B3 with a flat roof was chosen through voting and worked out in the next week. B3 was then parameterized based on additional independent parameters, some of which were shared (type 1) and discussed during the third meeting.

During the fourth and last meeting, all geometric parts were loaded into the core model, from the individual models of each discipline; and the overall result was assessed by all stakeholders.

DISCUSSION OF THE CASE STUDY

Despite average satisfaction of the participants regarding the workflow and the advantages of linked parametric solutions through shared pCOLAD, the state of the design at the fourth meeting highlighted a number of problems. A first major problem consisted in the parts not properly fitting together due to wrong reciprocal position in the Cartesian space (Figure 9). A second major problem consisted in a number of basic design criteria not being met. This regarded especially the criteria not explicitly prioritized into design priorities as focus of the test. This could also be due to the fact that three team members could only spend four half days on the case study. A questionnaire was filled in by the team members just after the first demo and hands-on exercise and also within some days after the last design meeting. It contained 45 questions. Most important were: Is getting started easy? Was it easy to start a discussion about a parameter? Will the method and prototype lead to faster decisions? Will the method and prototype lead to more transparency? Were too many or unnecessary actions avoided? Is synchronic (real-time) and a-synchronic collaboration working well? The score after the test design was much lower on these questions than before, even mostly negative. The comments show that this is due to the misunderstandings about the meaning of the parameters.

The meaning of parameters regards the understanding of the geometric properties that are parameterized within the design. This meaning should be univocal for all team members. In the test case, this aspect emerged as problematic. The use of sketches and email helped, but not enough. Improving the support in agreeing on the meaning of parameters was identified as necessary in the method and tool.

The exchange of geometry was the most discussed aspect already during the test. While exchange of parametric models across different disciplines was unanimously considered to be avoided, often updating the geometric core model with geometry (in this case, baked from Grasshopper) was considered necessary by part of the team. In this scenario, the core model would serve also as support for interdisciplinary design exploration and brainstorming. The team also believes that pCOLAD forces the team to think carefully about what the crucial parameters of the design are. It is impossible to measure the effect of that, but it is to be expected that it also has a positive effect on the quality of the design.

CONCLUSIONS

The goals of the pCOLAD project were: Develop a solution for synchronic (real-time) and a-synchronic collaborative architectural design over the Internet with shared parameters in Grasshopper/Rhino and other software.

The research questions were:

1. How can a collaborative design process be facilitated in a parametric design setting?
2. What is the best format for the parameters and their values?
3. Where should the parameters be stored?
4. Which system to use for multiuser real-time updating?
5. Which system to use for discussion and voting?

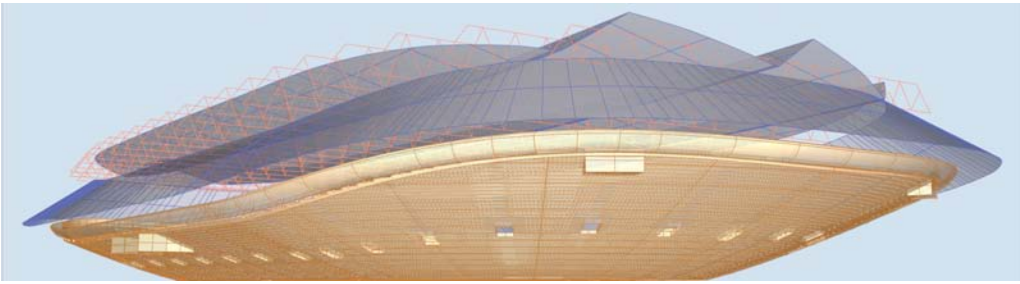

Figure 9
The parts don't fit.

6. Use existing Grasshoppercomponents or plug-ins or make our own in VB.net?

And the answers are:

1. A method was developed with sharing parameters, but it should be improved. Sharing of geometry and better explanation of what a parameter means was mentioned.

2. The pCOLAD *.csv format that was used was found adequate for the required tasks. It included fields for comments, parameter name, new value, obstruction, old value, owner, importance, date, author. Of these, the fields of obstruction and importance were not used, and new attributes were not added.

3. We needed a simple file storage system to store and update the csv file. We made use of Dropbox, which functioned well. In principle, a system based on pCOLAD could be setup in most real-time file sharing platforms.

4. pCOLAD sends warnings when the *.csv file is updated by somebody else and forces to start over again to avoid conflicting information. That worked well. Also changes since last update were coloured in red.

5. Blackboard Discussion board was used. And that worked rather well, though at the end under time pressure much of the communication was done by mail. Printed the forum contained 15 pages A4.

6. Developing a solution using standard GH components was found very constraining. That is why a prototype was developed in C# and in vb.net. Finally custom components were developed in vb.net.

The goals were met, the tools of pCOLAD worked as expected, but the method should be improved. The test of pCOLAD with the design of an Ice Stadium showed too many misunderstandings, causing misfits on the crucial moment when the models were put together. A solution could be found in developing an ontology and/or the possibility to attach hand sketches or other geometric representations with annotations to the pCOLAD form where the parameters and their attributes are displayed.

REFERENCES

EASTMAN, C., TEICHHOLZ, P, SACKS, R and LISTON, K 2008, *BIM Handbook*, Hoboken, Wiley, New Jersey, U.S.A

HUBERS, JC 2008, *Collaborative architectural design in virtual reality.*, Ph.D. Thesis, Delft University of Technology, The Netherlands

HUBERS, JC 2009 'Collaborative design in Protospace 3.0', *WAMELINK, H., PRINS, M. & GERAEDTS, R. (eds.) Changing roles; new roles, new challenges,. Delft: TU Delft Faculty of Architecture Real Estate & Housing*

HUBERS, JC 2010 'Collaborative parametric BIM', *BENNADJI, A., SIDAWI, B. & REFFAT, R. (eds.) ASCAAD conference 2010. Robert Gordon University, Scotland*

LAWSON, BR 2006, *How designers think*, Architectural Press/Elsevier, Oxford

TURRIN, M 2014, *Performance Assesment Strategies, a computational framework for conceptual design of large roofs*, Ph.D. Thesis, Delft University of Technology, The Netherlands.