Exploring Interfaces and Interactions for Graph-based Architectural Modelling in VR

Mohamed-Anis Gallas[1] and Adrien Coppens[2] (✉)

[1]Architectural Design Unit
numediart Research Institute
University of Mons, Belgium
mohamed-anis.gallas@umons.ac.be

[2]Software Engineering Lab
numediart Research Institute
University of Mons, Belgium
adrien.coppens@umons.ac.be

15th February 2019

Abstract The current integration of Augmented Reality and Virtual Reality in architectural design tasks is mostly limited to visualisation purposes. The designer typically has to go back and forth between the immersive application and the traditional desktop software. We believe that the resulting workflow is too complex to be used in practice, and greatly limits the adoption of those technologies. We started working on providing more efficient VR design environments through a parametric modelling bridge linking existing software to the HTC Vive, but further work is needed on the subject. In particular, in the context of this workshop on multimodal interfaces, we will work on finding appropriate interaction with graph-based parametric models in these immersive 3D environments.

Keywords: Computer-aided Design · Virtual Reality · Parametric modelling · Architectural modelling · Human-Computer Interaction

1 Project goals

The overall goal of this eNTERFACE'19 project proposal is to explore new VR-based multimodal interfaces for *virtual* architectural design and modelling.

A popular desktop environment for parametric and generative architectural modelling is Grasshopper[1]. It provides a visual dataflow-based language to edit and generate architectural designs, and is tightly integrated with the Rhino 3D modeller[2]. The corresponding parametric models can also be seen as (directed acyclic) graphs that define some kind of algorithm.

We already implemented a tool allowing to stream the geometry (mesh) of architectural models, created with Grasshopper, to a VR environment[2]. Our tool therefore allows to visualise the changes made to the graph in real-time.

More recently, our work has been focused on converting Grasshopper files to a more general and abstract graph, easily editable from an application. Using this tooling, we can now envision a VR (or even AR) editing environment for parametric architectural modelling. This paradigm switch from a 2D screen to an immersive 3D context greatly widens the possibilities for interaction with the generated geometries. Human-Computer Interaction (HCI) in a 3D context is an existing field of research boosted by a significant decrease in cost for VR/AR devices with high-quality tracking in the last decade. The standards for 3D User Interfaces (UIs) and interaction techniques are yet to be defined and we believe that having researchers with different minds and different backgrounds collaborate on the topic is highly beneficial to that kind of exploration. The goal of the project is therefore to explore and propose appropriate interfaces and interaction techniques allowing to edit and modify such designs in a 3D VR/AR environment.

[1] https://www.grasshopper3d.com
[2] https://www.rhino3d.com

Since this year's special topics are deep learning and reinforcement learning we could, depending on the participants' interests, explore and integrate such techniques to provide behaviour-based suggestions to the user. Because design processes very often involve multiple collaborators, we could also implement and validate conflict-handling mechanisms for concurrent modifications of such a graph.

2 Related work

2.1 Immersive environments for architectural modelling

There have been several attempts at providing annotation and sketching tools in a VR context, some of which were targeted at architectural design activities (e.g. [3] that relies on a tablet to control a 3D cursor used for sketching). As an example, IrisVR Prospect[3] provides immersive visualisation capabilities for many popular modelling tools.

We previously cited our paper [2] that describes a proof-of-concept VR application (see Figure 1), enabling parameter sharing and geometry streaming, and compares it to traditional desktop-based ones. As far as we know, this was the first contribution that brought parametric modelling editing (limited to parameter sharing) into VR. There is a clear need to expand the tool's capabilities with graph editing.

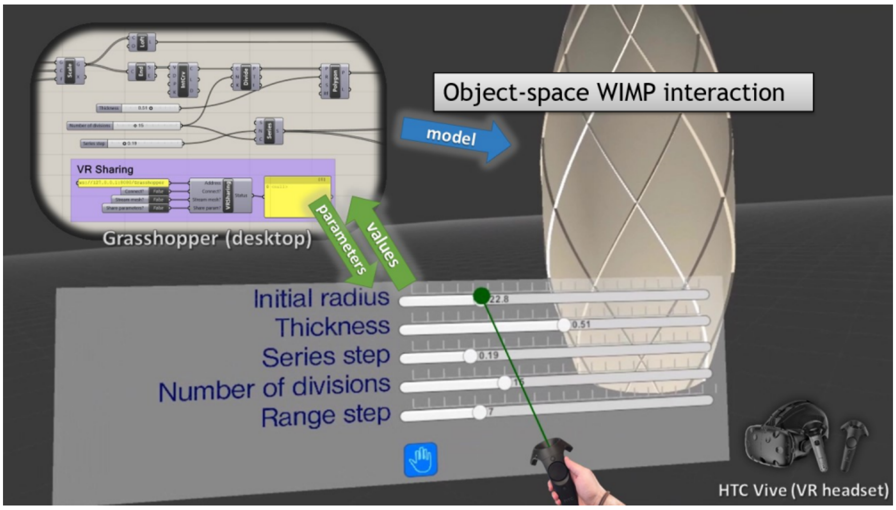

Figure 1. Our proof-of-concept application that enables parameter sharing and mesh streaming in VR, with the HTC Vive headset.

2.2 3D UIs and interaction techniques

In order to go beyond desktop-like interfaces (WIMP interfaces for Windows, Icons, Menus and Pointing device) i.e. to develop post-WIMP [6] interfaces, many approaches and techniques can be used. [4] is a thorough (and recent, 2017 for the 2nd version) book on the subject, describing the different aspects and disciplines involved in achieving and evaluating efficient HCI in a 3D context.

[3] https://irisvr.com/prospect/

2.3 Conflict handling

Our paper [1] describes different challenges faced when developing a collaborative application for parametric modelling, including conflicts resulting from users working on a model simultaneously. Since Grasshopper graphs can also be considered as algorithms, it may be interesting to experiment with software merging techniques [5] to resolve concurrent modification issues.

3 Technical description

3.1 System overview

The previous system shown on Figure 2 relies on a plugin for Grasshopper that is meant to be connected to a WebSockets server. That server is used as a relay to communicate with the clients (e.g. a VR application). Parameters are shared between Grasshopper and the clients, and each modification on the values of those parameters triggers a re-calculation of the generated geometry, which is then updated on the client side.

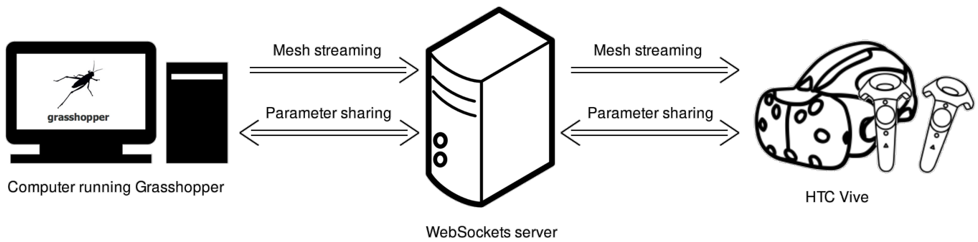

Figure 2. Overview of the general architecture for our previous tool.

Before the start of this workshop, we intend to make our general architecture more modular (see Figure 3). The previously mentioned work on converting Grasshopper files to graphs will allow us to add, remove or edit components (vertices and arcs) directly. In addition to that we will set up a REST API that will generate geometries from those graphs. This combination of changes will allow us to work with a model without having to run an actual desktop instance of Grasshopper.

We usually work with Unity to create AR/VR applications and write our code in C# since it is the preferred choice for that platform. It should be rather straightforward to transition to it from another OOP language, especially Java. Please note that Unity is available on Windows, Linux and Mac, and that multiple C# code editors with Unity plugins are available on all 3 platforms.

3.2 Necessary resources

Participants will need a computer with a recent version of Unity (the free version will be sufficient) and git installed. As for IDEs/code editors, we recommend Visual Studio (or Visual Studio code) or Rider. We will provide (or will make arrangements with the host institution for) the VR, AR, and interaction equipment (including at least one VR-ready computer for testing). Of course, participants are free to come with their own equipment, should they own similar devices.

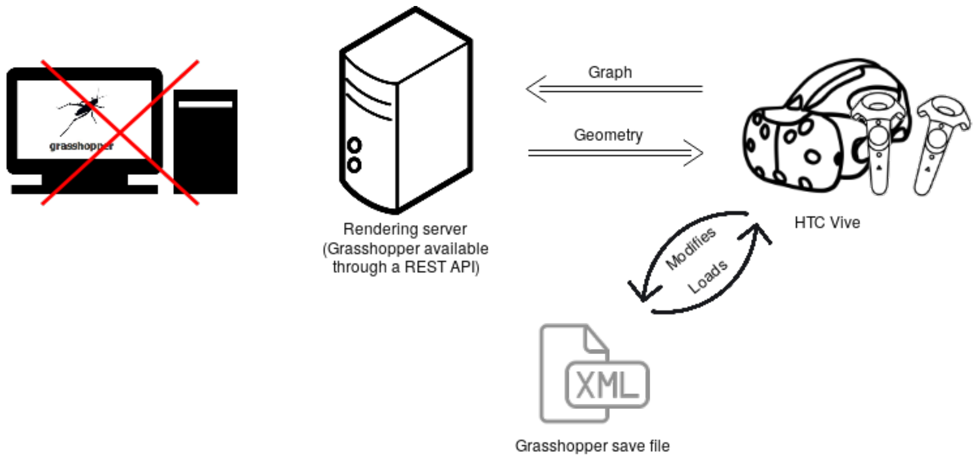

Figure 3. Tentative architecture for the tool developed as part of this workshop.

4 Work plan

Tentative Work packages are as follows:

WP1: Introduction & Tooling
WP2: The table metaphor → Deliverable 1: basic application
WP3: 3D graph → Deliverable 2: exploratory application (wrt visualisation)
WP4: Exploring gestural interaction
WP5: Exploring tangible user interfaces and/or speech recognition
→ Deliverable 3: more advanced application (wrt interaction)
WP6: Behaviour-based component type/link/names suggestions
WP7: (Remote) user testing and paper writing → Deliverable 4: paper

They result in the following schedule:

	Week 1	Week 2	Week 3	Week 4
WP1	■			
WP2	■			
WP3		■		
WP4		■		
WP5		■	■	
WP6		■	■	
WP7				■

Table 1. Schedule for this project

5 Benefits of the research

At the end of the workshop, we intend on having a broad view on a range of interaction techniques and interfaces adapted to manipulating graphs in a 3D environment. The outcomes (and deliverables) will be tightly linked to parametric architectural modelling but that context should only be seen as an adapted use case. We in fact believe our results will be translatable to other domains and sufficiently generic so as to be beneficial to the general HCI community, that still has to define the standards of 3D interaction.

Each software deliverable will be open-sourced and advertised, and could be the starting point to a toolkit or framework for that kind of interaction. Should participants be willing to extend collaboration to future research, subsequent papers could be considered.

6 Team

6.1 Leader

-Dr. Mohamed-Anis Gallas
Architect, holder of a PhD in Architecture Science and Associated Professor at the Faculty of Architecture and Urban Planning of the University of Mons. His research interests include the integration of parametric methods into the architectural design process as well as advanced interaction methods with digital models.

6.2 Base staff

-Adrien Coppens
PhD student in Computer Science at the University of Mons. Strong interest in AR/VR and in general HCI technology and usage. His PhD topic covers the use of those technologies for parametric architectural modelling, as well as design space exploration.

6.3 Other researchers

Ideally, we would like to be joined on this project by:

- 2-3 other AR/VR and (multimodal) HCI enthusiasts with a computer science/engineering background
- 1-2 machine learning practitioners
- 1-2 psychologists, ergonomists or researchers with some experience in validating usability and analysing user experiences

References

1. A. Coppens and T. Mens. Towards collaborative immersive environments for parametric modelling. In *International Conference on Cooperative Design, Visualization and Engineering*. Springer, 2018.
2. A. Coppens, T. Mens, and M.-A. Gallas. Parametric modelling within immersive environments: Building a bridge between existing tools and virtual reality headsets. *36th eCAADe conference*, 2018.
3. T. Dorta, G. Kinayoglu, and M. Hoffmann. Hyve-3d and the 3d cursor: Architectural co-design with freedom in virtual reality. *International Journal of Architectural Computing*, 14(2):87–102, 2016.
4. J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. Bowman, and I. P. Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.
5. T. Mens. A state-of-the-art survey on software merging. *IEEE transactions on software engineering*, 28(5):449–462, 2002.
6. A. Van Dam. Post-wimp user interfaces. *Communications of the ACM*, 40(2):63–67, 1997.

Additional information for evaluators

Rhino is not free to use (although a trial version is available). If this is an issue for evaluators, the workshop's topic can easily be further abstracted into general graph manipulation. The results would then be "connected" to the Grasshopper context later.

As mentioned in section 3.2, this workshop needs access to a set of devices that are rather uneasy to carry around for travelling. It would be much easier if the organisation could provide some of them. Ideally, we would like to have access to a "high-end" VR headset (e.g. HTC Vive or Oculus Rift) with the associated VR-ready computer, an AR headset (e.g. Microsoft Hololens) and any potentially interesting interaction device (e.g. Kinect, Leap motion, any eye tracker).