

Time Enhanced Architectural Modelling (T.E.A.M.): Virtual reality project for the planning and visualization of kinetic architecture and dynamic design

V.Temporin*, J.Volpato*, P.L.Cocco*, A.D'Angelo*, M.Tieghi*

*Poplab (Performance Oriented Prototyping Fabrication Laboratory), academic spinoff of Iuav University of Venice - viale Porta Adige 45, 45100, ROVIGO, Italy.

E-mail: info@poplab-team.org

Abstract. T.E.A.M. is a research project that was created to facilitate the design of kinetic projects and components created through a computational design process. Time is the ingredient that allows dynamism. The following paper examines the core features of Platform One, an experimental digital application born within the project and aimed at supporting designers during the development, in a VE (Virtual Environment) of dynamic architectures and components, as well as the principles that inspired it. The application presents two key features: the first one is that everything modified and developed in the VE retains its geometric characteristics, allowing the user to reach an informed 3D model at the end of the process; the second one is the ease and enjoyment with which the user manipulates complex dynamic geometries in the three-dimensional environment through a natural interface design approach that focuses on direct manipulation of architectural objects and components. The simulator is designed to be used in a 6DOF virtual environment using a commercial VR headset. It has currently been loaded with several *archetypal test architectures* and soon it will be available to designers who want to test their work with it.

1. Introduction

The area of the research to be presented is the project workflow in architecture and design. In particular we refer to an established design environment where the operational tools are aimed at creating a parametric model of the project, a model able to maintain the accuracy of the data until the realization phase – and possibly for the entire life of the object or building. Below, in detail, is the workflow we are referring to:

- *Geometric form finding and performance optimization* through computational design tools with the use of a software such as McNeel Rhinoceros alongside a dedicated visual scripting programming environment such as McNeel Grasshopper and its add-ons.
- *Construction plans and manufacturability check* with a BIM software such as Graphisoft Archicad or a solid modeler such as Autodesk Fusion 360 or Dassault Systemes Solidworks.

These tools allow designers to create a data driven digital model (also called "maquette numerique") that allows them to manage increasingly popular geometrical complex structures and responsive components. As Mark Meagher states "The integration of responsive components in architecture offers (in fact) the potential to enhance the experience of the building by giving expression to fleeting,

changeable aspects of the environment. Responsive buildings enable a physical response to changes in the environment through specific building elements”[1].

The traditional design workflow described so far involves creating three-dimensional models within two-dimensional media. However, we believe that the increasing complexity of the shapes and structures and the need for an exact verification of the kinetic components will naturally lead towards design tools operating in an "immersive environment". Such tools would allow the designer to view the model directly in 1:1 scale - the same as the final user - from the inside and the outside, with the possibility to directly manage the parameters that govern it.

Currently, however, there is no software on the market that allows designers to create data driven elements and generate dynamic relationships between them while in a VE. Platform One is a first step in the quest to fill this gap with the intent of rethinking the technical process described above while providing a fresh way of approaching the creative design process in the VE.

2. Methodology

The aim is to bring new paradigms into the aforementioned areas, thus trying to overcome the limits that we see on both the traditional drawing board and the 3D digital modelers for desktop computers. We do this in order to exploit the computational possibilities – more precisely, the relationship between form and data – that can be activated in design and architecture in the VE. This is to encourage further integration between people and architecture, to facilitate the design of kinetic components and active buildings, to efficiently manage the generation of dynamic and responsive architecture, as well as to convey the possibility of creating future architectures that could exist in the intersection between the physical and the digital world.

To address this challenge and create our experimental platform, we divided our research and work into two areas of focus:

1. To find and develop the hardware and software tools suitable to maintain the link between data and form;
2. To rethink from scratch the ways in which we create and manipulate complex three-dimensional objects in space, while abandoning as much as possible the preconceptions of the mediated interfaces to which we have become accustomed to through the use of desktop computers; such as any traditional 2D interface elements like icons, pictograms, files, windows, menus etc. In other words the aim is to create a user interface without the use of a visual interface.

Regarding the first area of focus, we developed a strategy by which everything that is modified in the VE maintains the information regarding its mathematical characteristics throughout the modeling process by using NURBS technology – NURBS, Non-Uniform Rational B-Splines, are mathematical representations of 3D geometry that can accurately describe any shape from a simple 2D line, circle, arc, or curve to the most complex 3D organic free-form surface or solid –, thus allowing the user to end with an informed 3D model at the completion of the process. To enable this, we took advantage of a commercial 3D modeling software with which the generation of all the geometries and the dynamic relationships between them are processed in real-time outside the VR headset. This is a *special* version of the commercial modeler McNeel’s Rhinoceros, called Rhino Compute. It is a headless version of the application that runs on a server and can communicate bidirectionally with an external computer. In our case our application Platform One running on the VR headset connects to the server and we then visualize the model and the parameters that we previously decided to expose from Grasshopper (Grasshopper is a visual programming language and environment that runs within the Rhinoceros 3D application). Every time we change one of these parameters from the VE we are actually sending a request to the server that executes the computation and then returns the modified geometry, sending a set of vertices that we visualize as a set of triangular meshes within our VR application (Figure 1). Our application is developed with Unity 3D, a cross-platform game engine by Unity Technologies.

This use of Rhino Compute, in addition to providing a solid and effective ready-made environment to perform mathematical operations on geometries, allows us to load *standard* Grasshopper definitions, making them immediately viewable as parametric models that can be then manipulated in the VE.

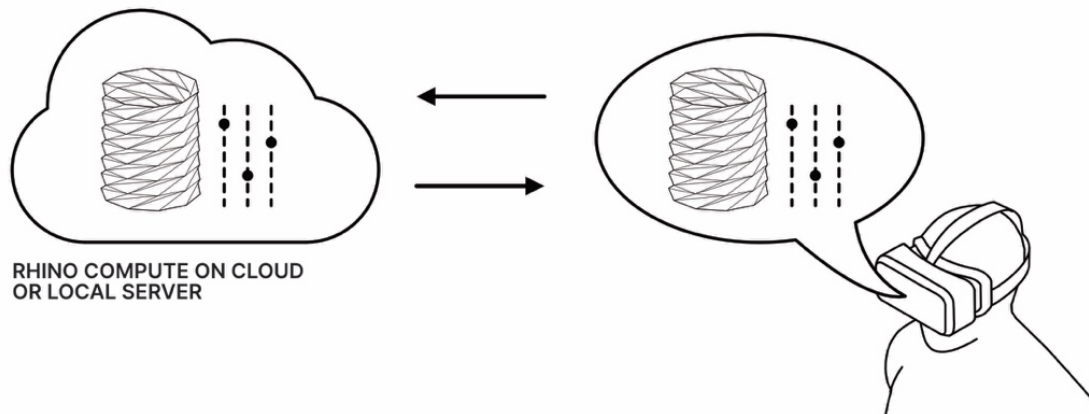


Figure 2. System operation logic: all variations made on parametric models within the app running on the headset are computed by an external server in real time.

Regarding the second area of focus, the user interface, we wanted to rethink from scratch the ways in which we create and manipulate complex three-dimensional objects in space. We started by considering that, most often in western culture, we have developed a method of creating “things” by dividing the representation of our projects into two specific acts: first designing on paper, followed by the building of scaled-down models of the final design. We need the “technical” drawing in order to understand and to explain to other people all the parts and details of the project idea and, the tridimensional model – or a perspective representation of it –, to effectively figure out its shape and to see and show to others how it could be perceived in “real space”. In fact, in this traditional process, the designer has the perception of the architectural object in its actual 1:1 scale – with its spatial attributes – only at the moment in which the architecture has been built. The VE gives us the possibility to visualize the objects in their real 1:1 scale, with their spatiality, without them ever having been realized. Nevertheless, the reduced scale model of an architecture and the technical drawings are fundamental because they allow us to dominate the dimensional and proportional characteristics of the architecture while it’s still being designed.

Following the principles of *direct manipulation interfaces* we have connected these two fundamental practices, creating a system that allows the user to work on a scaled down model while the modifications that they make are brought back on an identical model in 1:1 scale.

“Direct manipulation interfaces” is a term originally coined by Shneiderman (1974, 1982, 1983) [2] to refer to systems having the following properties:

1. Continuous representation of the object of interest.
2. Physical actions or labeled button presses instead of complex syntax.
3. Rapid incremental reversible operations whose impact on the object of interest is immediately visible. (Shneiderman, 1982, p.251) [3] [4].

To enhance direct manipulation of objects we have discarded the use of the hand controllers provided with the headset so that the designer uses only their hands for interactions within the VE. This is possible thanks to the hand tracking technology built into the Oculus Quest headset – the device that we are using

as a reference for the development of the application. In the Platform One application every interaction with the models takes place through a standard set of gestures that are performed with user's bare hands, while every control operation or activation of features not directly related to the manipulation of objects is done through the use of voice commands. As a secondary option the vocal control functionalities are also duplicated by a simple graphical interface HUD that the user can activate by performing a specific gesture with the left hand.

The work methodology for the team developing the Platform One application follows the principles of the framework called Scrum. Scrum is a framework utilizing an agile mindset for developing, delivering and sustaining complex products. The work is therefore organized in weekly *sprints* – goals that can be completed within time-boxed iterations – at the end of which we have implemented a number of features that, in order to be considered complete (*done* in Scrum terminology), must be fully functional and tested during the week. This way at the end of each week we end up with a potentially releasable iteration of the application.

For this reason, all the features described in this document are to be considered complete, operational and usable as of the date of publication of this paper.

The following is a description of the main features that we already built into our experimental system.

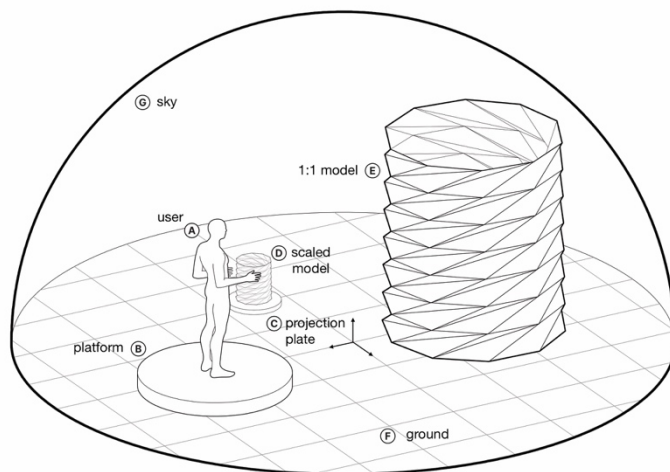


Figure 2. Main elements of the Platform One. Note the scaled model on the projection platform and its 1:1 scale counterpart.

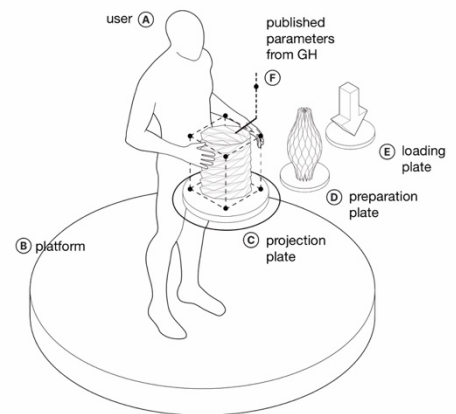


Figure 3. Main elements of the “dock”, the user interface of the Platform One.

3. Elements and features of Platform One

3.1. The Platform

The heart of the whole system in the relationship between the user and the virtual environment is the circular platform (Figure 2-b). The purpose of the platform is that the physical space in which the user is located must necessarily correspond with the virtual space. The platform has a size of 150 cm (equivalent to the area determined by the "stationary boundary" of the Oculus headset). At the start of the experience the user is placed at the center of the platform which in turn is placed in the center of the scene. With a voice command, the dock system and its 3 main plates can be displayed (Figure 3). The

platform can also move across the X and Y axes from the origin to any point of the scene by means of the "navigator" tool, which is a directional joystick that can be activated by a specific voice command.

3.2. The Dock

- *The dock* (Figure 2-c, Figure 3) is the element that allows the manipulation and management of models and geometries in the scene, it is composed of three main elements:
- *The loading plate* (Figure 3-e): this allows users to load the models they intend to place inside the scene. By rotating the plate, it is possible to visualize the different pre-loaded models or load new models. The user can upload both dynamic models in the *.gh* format (Grasshopper definitions) and static models in the *.fbx* format.
- *The preparation plate* (Figure 3-d): this allows users to temporarily place on it the models that they intend to use in the scene – there can be more plates, in fact it can be set by the user to have from 1 to 6 plates.
- *The projection plate* (Figure 3-c) is the most important feature of the user interface, because it allows for the "projection" of the model that is placed on it in 1:1 scale inside the scene/environment. The moment the scaled down model is placed on the projection plate the model visually enters into editable mode (Figure 3-f), while at the same time a copy of it in 1:1 scale is displayed in the scene (Figure 2-e). While on the projection plate a semi-transparent perimeter is displayed around the scaled down model providing a set of handles that allow the operations of scaling, rotation and translation for the 1:1 model in the scene. The parameters that the user previously decided to "publish" from Grasshopper are controlled with one or more sliders (Figure 3-f) that are shown as line segments connected to the scaled down model. Once the characteristics of the model have been defined, the user can "freeze" the full-scale model in the environment with a dedicated voice command and begin working on a new geometry with the same modalities. The process is reversible. If the user wants to return to work on the model, the model can be easily unlocked from a distance using a special hand gesture.

3.3. The voice and graphic control system

All non-direct interactions with objects and models in the scene, as well as all possible settings and configurations of the system, are either done through voice commands or through a graphical panel activated with a specific gesture to be operated with the left hand which is displayed through a dedicated interactive HUD. This interface allows for the same functionalities that can be controlled by voice.

3.4. The two modes for the environment

The system has two operating modes:

- **World off:** the user operates in a neutral non-illuminated space. Light is not reflected from surfaces, volumes do not cast shadows and all objects created or manipulated are characterized by a semi-transparent self-illuminating material.
- **World on:** a natural environment is simulated with a dynamic skybox, light reflections on surfaces are respected, volumes cast shadows and real materials are simulated for objects.

3.5. The sky

The scene is composed of a hemispherical skybox (Figure 2-g) that can be completely black ("world off" mode) or can dynamically simulate the behavior of the sun and light in the night/day cycle ("world on" mode). The lighting simulation reproduces the behavior of the light on the architectural volumes, the variation of the color of the light in the different moments of the day and the generation of dynamically cast shadows. If the user wants to verify the interaction between natural light and 1:1 geometries, they can activate the "world on" mode in order to perceive the flow of the day and,

consequently, the dynamics of the casted shadows of the 1:1 models. By default, the motion of the sun follows the sun's natural path and it is synchronized with the system clock of the device. The user can modify the speed of the sun's motion by interacting with a dedicated interface called the "clock" that can be activated with a voice command. Otherwise the user can choose to operate in a completely "abstract" mode, done by keeping the world off.

3.6. The Ground

The ground (Figure 2-f) acts as a two-dimensional canvas for positioning the geometries in the scene. In the "world off" mode the ground is transparent; in the "world on" mode the base material represented is a sheet of water that extends endlessly to the horizon. On the ground a grid (100x100 cm) can be superimposed for the precise placement of models. The terrain has potentially infinite size – the horizon expands as the user moves towards it.

4. Conclusions

All the features described in this paper are fully integrated and tested within the application Platform One. With this experimental system it is now possible to:

- manipulate large architectures directly from small scale models, giving the user great freedom and control;
- visualize and modify dynamic models, maintaining the relationship with the data that has generated them (from Grasshopper via Rhino Compute). This feature is fully operational when the headset is connected to a Windows PC (using a build of the application for Windows). We are working on enabling this feature also in standalone mode (using a build of the application for Android OS).

Platform one is a tool that designers can use both in the concept phase of their projects and in the study phase of their dynamic models. In its current incarnation, however, it is not suitable for use as an output tool for the final design and construction phases.

As of this document's publication, the research has entered a more collaborative phase. Having the need to test the platform with professionals in the design field, it was decided to involve about 100 Italian designers, with different experiences and professions, to get user feedback that will allow us to optimize the tool. For this phase, the testing is done with the manipulation and modification of elementary architectures that we designed and uploaded into the Platform One experience.

Acknowledgments

This publication has emanated from research supported by a research grant from the Veneto Region POR – ERDF 2014-2020 (Action 1.1.1 "Support to research projects for companies that involve the employment of researchers in the companies themselves").

References

- [1] Mark Meagher *Designing for change: The poetic potential of responsive architecture* (University of Sheffield School of Architecture, Arts Tower, Western Bank, Sheffield S10 2TN, United Kingdom)
- [2] Shneiderman B 1982 *The future of interactive systems and the emergence of direct manipulation. Behavior and Information Technology I*, p 237-256
- [3] Metzinger, T. 2017 *The problem of mental action, in Philosophy and Predictive Processing* (eds T. K. Metzinger and W. Wiese) p 1–26
- [4] Edwin L. Hutchins, James D. Hollan, and Donald A. *Human Computer Interaction, Volume 1, T. Direct Manipulation Interfaces* (Norman University of California, San Diego) p 311-338