REDBACK BIM

Developing a Browser-based Modeling Application Software Taxonomy

EMILY LEUNG\textsuperscript{1}, ANDREW BUTLER\textsuperscript{2}, ROB ASHER\textsuperscript{3}, NICOLE GARDNER\textsuperscript{4} and M. HANK HAEUSLER\textsuperscript{5}

\textsuperscript{1,2,3}Cox Architecture Sydney
\textsuperscript{1,2,3}\{emily.leung\|Andrew.Butler\|Rob.Asher\}@cox.com.au
\textsuperscript{4}UNSW
\textsuperscript{4}n.gardner@unsw.edu.au
\textsuperscript{5}UNSW & CAFA Visual Innovation Institute Beijing
\textsuperscript{5}m.haeusler@unsw.edu.au

Abstract. Browser-based platforms (Google Docs or Minecraft) have instigated the value of collaborative environments. Browser-based modelling point out a future for the AEC industry. Yet at present no literature review nor a taxonomy of browser-based modelling platforms exist. A key outcome of a unique taxonomy framework of existing BMA characteristics reveals that current BMAs do not take full advantage of the web’s unique capabilities such as centralising data across multiple tools within an ‘ecosystem’. Consequently, this taxonomy has productively guided the development of Redback BIM, a proof-of-concept BMA that enables the coordination of BIM data in a collaborative online context. Redback BIM further demonstrates how, through establishing a universal data-type, a diverse range of scripts can be consolidated together in an online platform to enable greater accessibility for a range of AEC professionals towards improved project communication and efficiency.

Keywords. Web 2.0; browser-based modelling; taxonomy; software development; standardisation of processes.

1. Research Motivation

With the introduction of web 2.0 in the mid 2000s, gaming and social media industries have developed new platforms and tools to enhance access to shared online communication and information spaces. Popular online platforms such as Google Docs and multiplayer games such as Minecraft have instigated the value of collaborative environments. They have also pointed to potentially new ways of working in the AEC industry that could improve design lead times and design quality through unified workspaces (French et al, 2016). Google’s G Suite, Microsoft’s Office 365, Trello and Slack are examples of platforms allow multiple users to access and collaboratively edit live documents which do not require software licensing or the hosting on private servers and devices, and for this reason
they are referred to here as ‘de-localised’ platforms (Leung et al, 2018). They typically work with 2 dimensional text and image-based documents, while AEC industry mainly engage with communicating information about 3 dimensional objects. While 3D modelling on the web is possible through browser-based modelling applications (BMAs), currently they do not offer an ecosystem which centralises data across multiple tools. Additionally, current BMAs lack the flexibility to address variegated ways of working as well as the wide range of users with diverse roles and levels of technical or computing skills. As built environment projects grow with complexity over time, collaborative ways of working becomes increasingly critical to the successful delivery of projects. This points to needing more integrated and de-localised software solutions. Conceptually, the challenges for architecture particularly, collaborative ways of working challenge the traditional idea of a single keeper and organiser of information, to one where authorship is shared and the result of teamwork oriented practice (Agullre Leon et al, 2009). Technically, challenges in high use of localised software platforms across the AEC industry can create a myriad of file-sharing and interoperability issues has served to further undermine the shift towards more collaborative ways of working. Yet, a key reason that localised platforms remain more popular is that the de-localised BMA alternatives do not currently possess the same functionalities or ease of use. Addressing its broad range of users, as well as enabling simultaneous multi-use is an issue that social media and gaming industries have attended to in their development of online collaborative environments. Minecraft is an example most powerful in multiplayer mode as users have the ability to converse with anyone in the same ‘world’ in real-time, as well as interact with the platform in ways to ‘build’ or modify that world. While multiplayer online games and social media industries have demonstrated success when multiple users engage and interact together on their platforms, the architectural software platform ‘usability’ in an architectural context carries a different meaning that focuses on how well it produces a digital artefact. In fact, software development has typically followed a model of continuous improvements as problems arise and in order to maintain relevance within the market (Holzer, 2016). Thus, while it is possible for current software to consolidate usability and format compatibility which may benefit end users in collaborative practice in the long term, this would carry cost implications for the companies who create them. While research has shown promise in improving data sharing through structured modes of collaboration such as integrating Grasshopper and Dynamo into existing software workflows, they fail to address varying degrees of technical proficiency. To resolve the similar challenges of understanding computer science programming concepts, Victor (2011) argued for the development of intuitive interfaces to improve methods of system thinking, where high-level effects can arise from low-level interactions. ShapeDiver or Project Fractal are existing BMA examples which allow users to interact with parameters in order to intuitively understand the outcomes without needing to comprehend its complexities. However, they are limited to creating only bespoke workflows that are yet able to better reflect the architectural process, which is a key objective that the research project aims to explore and address.
2. Research Aims, Objectives and Research Question

The overarching aim of this research project which is to develop a workflow for a dynamic communications platform connecting multiple users within a unified or consolidated workspace to improve the interoperability of building information models (BIM) between scripts. More specifically, the dynamic workflow aims to interlink data and geometry of an architectural project and make it accessible on a browser-based modelling application (BMA). Connecting BMA capabilities with modularised scripts further aims to mitigate the drawbacks of utilising desktop-based software. And finally, by repositioning scripts and 3D models onto a unified web platform, it will enable users to interact, contribute, and organise their projects more efficiently, making an impact towards addressing the broader objective of further streamlining the architectural process.

1. In what ways can current digital tools within the AEC industry use the web to enhance the collaboration and the communication of dynamic information between various stakeholders who take part in an architectural project?
2. Why hasn’t the AEC industry adopted the Web as a central space for collaboration and communication?

3. Methodology

This research project adopts an Action Research methodology that aims both to understand a problem and provoke change and is deemed particularly suitable to addressing emerging issues in the design, delivery and research of information communication technology (ICT) systems and new media applications (Hearn & Foth, 2005). Action research involves stages of planning, acting, observing and reflecting which are cycled through iteratively towards delivering a solution to an initially identified problem. In this research project, this involves a reflective process that continuously creates knowledge from constructing and evaluating technological artifacts relevant to associated theories (Purao, 2008).

The following method has been applied in practice: (1) Develop a web application that stores the compatible data format in its database to display the geometry on its interface. (2) Create a script (which in this case, one which conducts a solar analysis) to work with the compatible data format. (3) Using Grasshopper, reference geometry built in Rhinoceros and send the converted geometry to the web server. (4) Fill in the necessary inputs required by the script in order for it to obtain its values which then returns the results back the user’s Grasshopper model. (5) The model within Rhinoceros, when changed, will send the data to the web server again and return new results back to the user’s Grasshopper model.

This is based on the hypothesis that: (1) It may unlock existing inefficiencies within architectural practice. (2) Removes the need for localised 3D CAD programs (if not similar features will be noted within the taxonomy). (3) BMAs can already be seamlessly integrated within the architectural workflow Scripts can be used through the web.
4. Background Research

A taxonomic framework was created to classify BMA’s most applicable to the AEC industry [URL WITH HOLD] as a taxonomy sets out to understand, analyse and make visible, relationships between complex domains (Nickerson et al, 2010). ‘AEC Projects’ refers to software implemented within the AEC industry to document projects (similar to CAD and BIM). ‘Non-AEC Projects’ targets any user on the internet and is able to produce small scale projects focused towards 3D printing applications and rendering software. ‘Programming’ refers to software that require programming knowledge in order to design parametric models. They are limited to text-based and visual programming interfaces. Finally, software which do not require any pre-requisite knowledge to model in 3D exist under the heading ‘Non-Programming’. They also mimic gestural modelling tools, equivalent to sculpting. Contemporary architectural practice reflects increased demands for overseas collaboration, highly compressed time frames, and generating data for simulating building performance and environmental outcomes (Chok, 2011). With the growth of co-located work practices in architecture and the desire to work synchronously on BIM, this suggests much could be learned from industries that have successfully developed environments that feature synchronous communication and dynamic information sharing. Minecraft is an example which constitutes collaborative environments that echo desired ways of working in architecture to achieve design lead times and design quality from unified workspaces (French et al, 2016). Consequently, there are clear implications for great potential to emerge when exploring applications for Information Communication Technologies (ICT). More specifically, exploration into web-based technology for designing and planning processes within the AEC industry is supported by many scholars in the fields of architecture and computer science. As opposed to current CAD software and BIM methodologies, scripts embrace the notion of making possible the optimisation of complex and repetitive tasks. This not only encourages systematic thinking, but also streamlines processes when design changes occur. Although the adoption of computational design techniques has streamlined methods of existing AEC practice, the organisation of information across stakeholders of a project is still a common problem when working collaboratively, a number of flaws which fail to address the skill level of current AEC professionals. In fact, although many architects have the ability to program, they unfortunately do not have the same programming abilities to that of a software engineer. Evidently, Woodbury (2010) describes designers as ‘amateur programmers’ who have enough knowledge to get by accomplishing the tasks at hand, but still lack the time or inclination to do it strategically. As such, computational design practice is not widely deployed. Howe (2016) sees issues of interoperability and conservatism to be negative contributors for implementation within architecture firms. Additionally, a survey of approximately 2000 bespoke visual scripts created using Grasshopper were analysed, indicating that designers in fact do not organize and manage their scripts into modules than programmers do to organise their programs (Davis et al, 2011).

To address the communication between multidisciplinary practices, standardisation is argued to make life easier (Scheurer, 2012), and is reflected
upon the tools used. Strategies from computer science provide examples of interdisciplinary approaches that have similarly addressed these problems through implementing methods of modularisation and standardisation. But as well, research has shown potential in addressing problems of communication through the integration of the most understood medium which everyone is familiar with - the web page interface (Chok, 2011). In fact, existing scholars of computer science have explored its potential through the development of ‘React-Planner’ (Marino et al, 2017) and ‘Luna Moth’ (Alfaiate et al, 2017) as parametric tools that enable real-time feedback and traceability features respectively. However, these software target users for such applications focus on professionals of sound programming experiences and therefore limit its potential to be applied across a number of projects. To address this problem, similar challenges of understanding programming concepts in computer science have led Victor (2011) to argue for intuitive tools that enable improved methods of system thinking. Thus, developers of recent web-based software platforms targeting non-technical users also understand the benefits of combining simplified interfaces with complex computation. Precedents such as ShapeDiver and Project Fractal are current examples that run the computational design tools Grasshopper & Dynamo respectively on cloud servers, where the results are presented simply on an interactive BMA user interface. The power of these tools allow designers and clients to engage with their designs and ultimately iterate through options much faster. However, the programs developed are evidently bespoke to each project and specifically target its use towards early design stages, limiting its potential to be reapplied across other stages. In fact, these BMA precedents highlight reasons for inconsistency in the organisation of computational design tools and plugins that additionally do not reflect the architectural process. In order to improve the relationships of scripts, it would be advantageous to look deeper into questioning the way software is integrated within the architectural process. This area of research reflects Marble’s (2012) category: Designing Industry, which focuses on the organisational issues that drive multidisciplinary integration of information that is collected, modelled and efficiently managed. In its simplest form, architectural design is a process of developing, describing and communicating ideas and of generating, transforming and exchanging information over different media and between the numerous parties involved. This process is an interconnected one, which can be visualised as an infinite loop that cycles through the birth of a building to the death of a building. It is the result of an overall design task which is broken down into smaller units of work which are fragmented and spread across multiple stakeholders, making the reconnection an important task on its own. However, the potential for improving this workflow is possible through the connection of the computer to the internet, which has formed the most powerful platform for information-interchange ever (Scheurer, 2012). Implications of ubiquitous connectivity and standardisation of data exchange allow various stakeholders to think and act at scale within an organisation filled with varying roles of high and low level decision makers (Chok, 2011).
5. Case Study (REDBACK BIM)

When developing a BMA that systematically coordinates information across the architectural process, the main aim is the optimisation of translating data across each task and user whilst being able to automate repetitive tasks through modular scripts. In relation to Redback BIM, there are two types of users within the process. Specifically, the backend user (such as a Computational designer) is able to develop and upload their modular script; and the frontend user (such as an architect) is able to access and update data from a web-based interface and implement scripts necessary to complete the task. The outcome of the script is then returned to the frontend user for reviewing and decision making before passing the information to the next user and progressing to the next task. Given that there are several steps in this process, the paper looks specifically into constructing a scenario to run scripts using a common data format so that it returns new information in the same consistent format, but is presented in a way that will benefit users to make better informed design decisions.

Enable communication between web, data and modular scripts Current BMAs available on the web are built differently. Some of which require additional proprietary software or the need to upload specific file formats, and others are capable of generating geometry through writing lines of code within the UI. A secondary study established key features necessary to addressing the aims outlined previously, but as well identified Flux.io as the basis for developing Redback BIM. Analysing the source code of Flux provided a basis for understanding how Redback BIM will and should work. This knowledge was used to derive a workflow diagram encapsulating the inner workings of how it resolves the research question and aims methodically. In particular, within this workflow, 3d architectural models (1) and scripts (2b) enter through their own path, and make their way into
a central web server (3) which stores the data that creates the models in a database (3a), but also the scripts which automate processes within a virtual machine (3b).

**Easily Translate Models and Data** JSON is the most effective data-interchange format between desktop-based and web-based software (Scheurer, 2012). Additionally, the 3D JavaScript Library, Three.js, which is used to build Redback BIM, also has the ability to use JSON data to describe its models. This therefore highlights that it is possible for there to be one format shared amongst various digital tools within the AEC industry. Therefore custom Grasshopper components were created to allow geometry to be converted into the JSON format. Standardised data when sent between different software make it possible and easy to translate information back and forth, or even converting between other software.

![Figure 2. Workflow for Redback BIM User Interface.](image)

**For any level of technical knowledge** Once the Three.js JSON was created from the Grasshopper script, a full stack web application was built using Django’s backend Rest Framework to host the server and database. The frontend user interface was developed using the React JS framework with the Three.js module. The JSON created from Grasshopper was then stored within the backend database where a viewport was developed using Three.js to display the geometry. Additionally, we can also display additional information (just like BIM) where metadata can be embedded within the object. Thus this step follows the aim of developing an interface so that users of any technical knowledge can easily access and understand 3D model information and data.

**Modularisation of Scripts** Presenting information about a model on the web now allows the ability for information to be shared across multiple users. The last aim addresses the ability to create a script that can obtain the model data and automate a repetitive task completely on the web. There are three things that we...
need to first determine: (1) A series of inputs, (2) A script to take in the inputs, (3) An output (or series of outputs), here using LadyBug, a script was created to address a solar analysis example. In particular, this script was specifically built as it has the potential to be used across any design project containing building masses.

**Convert the Inputs into JSON** The name of the inputs of the script were combined as keys of a JSON object, where the values of each input was stored within the keys of the JSON object. The outputs of the script is another JSON object which is returned to the user. The process can be easily done through a web interface. Given that a user is able to store geometry as JSON keys within a database, the same values can be input into a form. User then chooses dropdown menu, the keys representative of the building and which is the area to test.

**Make a POST Request** Presented within another workflow diagram, the input values determined by the user set the starting point of the process. Once the inputs are defined, a user can click on the submit button of the form which will send a POST request to the web server, by sending data to a specific URL. Through clicking the submit button, the user’s input variables are collated together into one JSON object - the data that is sent to the URL through the POST request.

**Pulling apart data in Grasshopper** On the web server, the Grasshopper software that is running on the virtual machine detects that a request has been made. Then the Grasshopper script obtains the JSON sent from the user interface and deconstructs the object at the start of the script by the following keys that were predetermined. Evidently this clearly emphasises how important it is to standardise data exchange between users and platforms. The following values are run through the script, which then returns a JSON with the specified outputs. In this case, if we assume that this was done completely through Redback BIM’s user interface, the output would be the Three.js readable JSON values illustrating how much shade is cast by a building and for how long over a period of time.

**Web Server Response** The following outputs from the solar script is returned to the same URL, which is known as a response. With that response, the Web application picks up on the change and does something with that information. So in this case, the outcome of using Redback BIM will therefore produce a visual representation of the solar analysis, allowing users of any technical knowledge to understand and make better streamlined and informed design decisions. But as well, the integration of Redback BIM’s simplified interface will allow the implementation of the exact same solar analysis script across any project, by anyone of any technical skill. Ultimately, this can open up more opportunities to address other numerous types of repetitive tasks within the architectural process throughout the lifecycle of a project.

### 6. Significance of Research

This paper concludes and argues that the ability to have access to 3D models, data and computational tools would benefit all stakeholders of a project. Based on the research the development and application of a unique taxonomic framework has contributed to defining a systematic workflow for successfully removing incompatibilities within the multitude of software. This was done through
REDBACK BIM

accessing, updating and converting data which translate 3D models between platforms that are able to run scripts. By converting geometry and data into a single unified format, access to JSON data through web-based interfaces has the potential to easily share information across various stakeholders without installation of proprietary software. The creation of Redback BIM as a working proof of concept, has shown great potential to succeed as a step forward as a tool and workflow to enable synchronous collaborative workspaces in the AEC industry. Specifically, this research has demonstrated that it is possible to allow non-computational design users to use computational design tools and a translation of information between stakeholders can be improved to make streamlined informed design decisions.

7. Evaluation and Conclusion

This study focused heavily on creating new knowledge through exploring interdisciplinary fields. Overall, success was seen in creating a workflow that enables multiple stakeholders to engage with a centralised web server, capable of integrating both data that describe 3D architectural models, but also allow users to streamline repetitive design tasks with computational design tools through the browser. Whilst the case study provides many benefits, there are many things to consider which will need to be highlighted. (1) Given the wide range of software used across the multitude of AEC disciplines, it suggests varying ways in which software understand and implement data inputs. I.e. an engineer may require centreline curves to represent beams within their simulations, but the same curve may not mean anything to an interior designer. This leads to creating variations of the same model across users and thus, language barriers between different disciplines can present problems in standardising information exchange. (2) Concerns for liability and the loss of confidential business information are two of the main underlying factors for hesitation to utilise cloud-based services (NASEM, 2016) and thus BMAs in practice. (3) While the implementation of computational design tools within architectural practice enable streamlined workflows of repetitive tasks, there are limitations to how far these tools can replace the emotive aspects of human design. Specifically, as with all digital tools, these software cannot decide which design is the most beneficial for the client, nor articulate strategies for improving the design of a building, but require the user to establish parameters and decide which justifies the most appropriate design option. Research in the fields of architecture, computer science and computational design has emphasised the need to implement emerging technologies to streamline communication between multiple stakeholders of a project, as demands of design tasks today cannot be completed by humans alone. The slow adoption of web-based platforms has led to the development and application of a unique taxonomic framework, making connections between BMAs most applicable to the AEC industry. With the introduction of Redback BIM as a proof of concept, the outcomes of this research made possible the ability to streamline methods of communication between multiple stakeholders and automate repetitive tasks through modularised computational design tools. Ultimately this research is a step closer to removing inefficiencies in practice as it demonstrates benefits of web-based platforms capable of streamlining existing practices. Thus hindering
the negative connotations of being challenged by the technicality of any complex project, and instead re-establish creativity as the forefront of design.

References