

# De-blackboxing BIM

Why and how should architects get BIM out of proprietary software and closed file formats

## 1. Abstract

This dissertation talks about the use of computer coding in the field of architectural representation, and it analyses the different ways in which coding can impact architecture.

Since the 1970s, when computers became available to architects, this new human-computer relationship became a challenge for architecture in multiple fundamental aspects, such as representation, education, and practice. Half a century later, in the 2020s computers are not only available, but in most cases, they are indispensable.

New digital processes of representation, such as Building Information Modeling (BIM), have become not only a possibility but sometimes mandatory by governments in many countries. For this reason, I argue that it's imperative to understand both its potential and limitations.

Chuck Eastman, one of the fathers of BIM explains in his BIM Handbook that "BIM is not a thing or a type of software but a socio-technical system that ultimately involves broad process changes in design, construction, and facility management." Digital objects are coded to describe and represent real-life building components. This facilitates a dialog with the model that was previously not possible. BIM could be defined as a communication and collaboration tool. Originally created to connect people, processes, and data. However, that is not exactly the way the Architecture, Engineering, and Construction (AEC) industry is using BIM. The status quo is to work with proprietary solutions and closed file formats. There are a few vendors that control the industry usually, promoting their own proprietary file formats. The issue is that when working with closed formats, the whole process becomes a black box.

A black box is a system that can be understood only in terms of its inputs and outputs, but which process is not accessible or even visible to users. To "de-blackbox" is to understand the process, parts, and connection of the system.

Two conditions must be met before architects can fully de-blackbox the digital toolbox. Firstly, architects must have a working knowledge of computer programming. Secondly, the source code—the list of human-readable instructions that define a computer program—of the software that they use must be accessible for modification by the user.

This research warns about the obstacles that architects must overcome in order to benefit from a fruitful relationship between programming and architecture, putting the goal of designing better buildings for people at the core.

Coding has the potential to further expand the current limits of our imagination within BIM. Computers are a tremendous contribution to our architectural exploration if we continue to explore innovative research that includes human input with computer logic and processing power to arrive at collaborative solutions.

## 2. Table of Content

---

|      |                                                            |    |
|------|------------------------------------------------------------|----|
| 1.   | Abstract                                                   | 1  |
| 2.   | Table of Content                                           | 3  |
| 3.   | Introduction                                               | 4  |
| 3.1. | Computers processes, programs, and software for architects | 4  |
| 3.2. | Building Information Modeling (BIM): The new paradigm (?)  | 6  |
| 3.3. | Open-source: One way out                                   | 9  |
| 3.4. | Digital education for architects                           | 11 |
| 4.   | Literature Review                                          | 12 |
| 4.1. | Coding as a language                                       | 12 |
| 4.2. | Coding as a tool                                           | 15 |
| 4.3. | Coding as an apparatus                                     | 19 |
| 4.4. | Conclusions                                                | 21 |
| 5.   | Epistemic Object                                           | 22 |
| 5.1. | Trapped in the black box                                   | 22 |
| 5.2. | New tools                                                  | 24 |
| 5.3. | Bespoke                                                    | 24 |
| 5.4. | Web-based                                                  | 24 |
| 5.5. | Open-source                                                | 25 |

## CHAPTER 5

### 5.1. Trapped in the black box

As described in the previous chapters the main BIM authoring tools for the AEC industry are mostly captured by a few vendors. As of 2022 the most competitive solutions are proprietary. Which means that such a toolbox is normally the first option for practitioners in the AEC industry. The first decades of BIM have been dominated by proprietary software. AEC Companies have invested big money in training, hardware, adapting workflows, etc. Additionally, vendor lock-in makes AEC companies and users dependent on a particular vendor.

Technically speaking, the most significant outcome of BIM as a process is the BIM model or models. When BIM is used the way it was designed for, the whole process is based on a paperless delivery method. Which means that the whole process depends on digital workflows and the entire exchange of information is done through file formats. File formats are standards to store and exchange data in a computer file.

For an industry as complex as the AEC, standards are crucial. They allow the different actors and stakeholders of a project to communicate with each other, reducing ambiguities in the language that can lead to misunderstanding. A standard is “a formula that describes the best way of doing something”.<sup>[i]</sup> The International Organization for Standardization (ISO) is an independent, non-governmental international organization that develops consensus-based, market relevant international standards for multiple industries<sup>[ii]</sup>.

For BIM, one of the most relevant standards is the ISO 19650 also called: Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM)— Information management using building information modelling—<sup>[iii]</sup> This standard has a whole section that talks about exchange of information. The standard states that “Throughout a design and construction project, information will pass through multiple software solutions. During these exchanges it is the information, not the software used, that provides value. The software is merely a tool.” In this exchange of information, the standard distinguishes between proprietary data and open data. “Proprietary data is restricted to specific software solutions”. It is stored in proprietary file formats and its content is not accessible, available, or even visible to users.

Often for the AEC industry, proprietary software produces proprietary files. The issue is that when we use proprietary formats, we are without doubt dealing with a black box. The data is opaque, and we can hardly control it. All software needs to store and communicate data to one another. If the data format is closed and proprietary, it cannot evolve to meet the needs of the public, or be inspected by the public that it is truly correct. It also means that you can only inspect it using a limited set of tools which are not under your control. It is likely to be inefficient, store unwanted data, and the user has to trust that their data is recorded correctly, with no way to validate it.<sup>[iv]</sup>

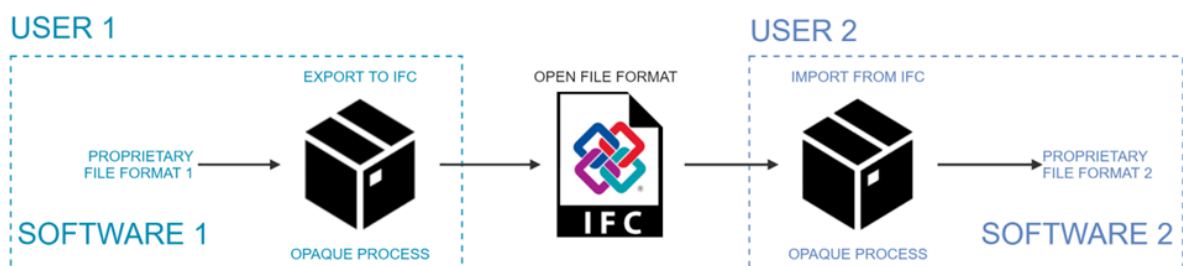
BuildingSmart International is an open, neutral not-for-profit organization committed to creating and developing open digital ways of working for BIM. They call themselves “the worldwide authority driving the digital transformation of the built asset environment, through creation and adoption of open, international standards for infrastructure and buildings.”[v] They promote the concept of *openBIM* which they define as a process that extends the benefits of BIM by improving its accessibility, usability, and sustainability of digital data in the building industry. It is a vendor-neutral collaborative process that facilitates interoperability to benefit projects throughout their entire lifecycle. “OpenBIM processes can be defined as sharable project information that supports seamless collaboration for all project participants.” OpenBIM ensures: Openness of assets and processes, collaboration between stakeholders, flexibility of choice of technology, and sustainability, safeguarded by long-term interoperable data standards.[vi]

OpenBIM is necessary to tackle some of the issues of BIM, however, what is ironic is that they had to come out with the concept of openBIM in the first place. As mentioned earlier, BIM was originally intended as an open, transparent, interoperable, a collaborative approach to lifecycle management. Indeed, the openBIM definition describes very well the original purposed of BIM. It was basically what BIM was designed for. That other thing that we currently refer to as “BIM” is what we should call “ClosedBIM”, which is BIM developed in proprietary software and exchanged through closed file formats.

Nowadays, closedBIM is the most common practice in the AEC industry. There are several highly sophisticated software specialized in it. If the user stays within a specific vendor, or a package of interconnected solutions, the data normally flows very smoothly. Also, the software company usually offers support in case there is any issue with their products. The whole process is seen as secure and reliable and that is why, most of the large companies in the industry use proprietary solutions for the majority of their tasks.

However, I will argue that closedBIM, leads to multiple problems. Just to mention a few of them:

1. First, if stakeholders do not use products from the same vendor throughout the project, the flow of data is obstructed. For the data to be shared and accepted by a different software, it needs to be exported to a vendor-neutral file format. If the stakeholder that receives the data is also using a closedBIM software, the process of sending and receiving the data will require two conversions.



2. The process of converting is also a black box. When using proprietary software conversion tools to go from one software/file format to another, the process is opaque, and it is difficult to control the output. One may lose data without being able to monitor it.

*Exporting to an IFC? Will your roofs become IfcRoof, or will Revit decide your roof is a slab instead? Will your windows become IfcWindow? Who knows. Revit does what it wants. Will your building element turn up at all? Will it even export? Will it import? Ah, the mysteries of the universe.[vii]*

3. Closed systems difficult interoperability. BIM software are normally AEC focussed with a high learning curve which makes it hard to collaborate with non-AEC professionals. And even within the AEC industry, we spend a lot of time and energy converting files from one format to another.
4. Licenses are expensive and can only be afforded by big companies. The most popular BIM authoring software licences varies between \$2500- \$3500 per year per user[viii]. Student can get them for free at reduced price, but this practice is also questionable. Rushkoff states that:

*Most schools with computer literacy curriculums teach programs. ... These basic skills may make them more employable for the entry level cubicle jobs of today, but they will not help them adapt to the technologies of tomorrow... their entire orientation to computing will be from the perspective of users.[ix]*

With free or reduced licences student become dependent of the software and when they become professionals, they find it hard to practice without the proprietary tools. And because at entry level it is almost impossible to afford the licences their only option is to become employers of a big firm.

5. Also, when a company has invested in a specific tool, it is expensive to change. Incorporating a new software involves not only the cost of the licence, but there is also training, adaptation, and hardware associated cost.
6. Proprietary tools can be very complex: The software we use gets larger and heavier with each release. Often, more tools are added, whether you want them or not.
7. They are not very customizable. Proprietary software usually come as-is and the user cannot improve them. Moreover, they do not always meet the user's current needs and there is nothing to do about it. The user must wait until the next release and hope it includes the desired improvements.

8. Finally, for long term projects, lots of time is wasted upgrading models to the latest software version. Upgrading processes are also black boxes, so there is data loss that the user cannot control.

These are just a few of the problems associated with the dependency of proprietary software and closed file formats. Black boxes are useful to hide unnecessary processes for the user point of view, but in the case of BIM they could prevent the data to easily flow. BIM cannot be BIM when it cannot be freely shared and controlled by the users/stakeholders. If we go back to the definition of openBIM, most of its fundamental elements are impossible with proprietary solutions and closed file format. In conclusion, if the user thinks openBIM adds value to the building process, it is crucial to explore alternatives that support it. But does the alternative even exist?

This epistemic object explores a new toolbox, one that is centered in open formats, transparency, and flexibility.

## **5.2. New tools**

Before 2020 it was challenging to imagine an alternative to proprietary software for the AEC industry. The sector was too used to their “stable” proprietary tools and despite some of the issues mentioned earlier, everyone was willing to keep with the status quo. The high investment involved into changing technology, well-known practices, and workflows, the uncertainty and risk that innovation can bring, the complexity of the processes and tools required, and the generalized idea that coding and software development are areas that must only be accessed computer scientists, lead to a blind dependency into the main vendors’ solutions. Naturally, these software companies capitalized on this with impressive craftiness and marketing techniques.

There have been a few attempts to produce an open-source solutions for BIM such as FreeCAD[x]—FLOSS 3D parametric modeler released in 2002 as a CAD software primarily aimed at engineers. FreeCAD started supporting IFC files in 2011—but it did not get the real impact of becoming a plausible alternative to the main vendors. Up until the end of 2019, anything different was difficult to conceive. However, the decade of the 2020’ started with a few disruptive milestones that begun to challenge the existing perception about the access to software for the AEC:

In October 2019, Dion Moulton—architect and software developer who has worked for BuildingSmart International developing the Industry Foundation Classes (IFC) schema—released the first version of BlenderBIM. BlenderBIM is an Add-on for the popular FLOSS 3D engine Blender—“Blender is libre and open-source 3D computer graphics software used for creating animated films, visual effects, art, 3D printed models, motion graphics, interactive 3D applications, virtual reality, and computer games”[xi]. Blender was publicly released as a FLOSS back in 2002 and regardless of being free, it has become one of the most powerful and popular 3D engines in the world. Since 2005 the software is downloaded downloaded in average 1 million times per month. In 2019 it was downloaded over 10 million

times (“almost doubled compared to 2018 in any metric”<sup>[i]</sup>) and in 2020 over 14 million times. Up until then, it did not have any specialized BIM features. With BlenderBIM’s release at the end of 2019, the door to AECO professional to use FLOSS was finally wide open. BlenderBIM uses IFC as its base and unique schema and file exchange format.

A few months after that historic release, in February of 2020 the OSArch community is created. OSArch, is an online community that promotes that the built environment can be designed, constructed, operated, and recycled with free/libre and open-source software. With the creation of the community, the awareness of an alternative to the status quo of software started to permeate through the AEC industry.

The request of transformation is not only coming from open-source communities. In July 2020 several leading UK and international AEC firms wrote an open letter to Autodesk CEO, Andrew Anagnost, highlighting a range of deep concerns about their software related to costs, licensing & business practices. The letter had a clear impact for the software company because at the end of the same month, Amy Bunszel, the Executive Vice president of the company, replied to the customers who wrote the letter. This letter was followed one month after by Anagnost who published a second reply, this time trying to address each of the topics described on the open letter.

These apparently linked events were building the impression that the big software companies were understanding that the disconnection with an important part of their customer base could damage their business model. However, the optimism that the improvements to the industry’s old and questioned practices were going to come from a top-down approach started to quickly fade. Two years after not seeing progress with the demands, in September 2022, the Danish, Norwegian, Finnish and Icelandic Architectural Associations, which represent 14,000 architects wrote a follow up Open Letter to Autodesk stating that nothing had changed since the first one. The former letter was fully endorsed by the Architects’ council of Europe (ACE). What is more interesting is that while some people were writing letters to the main players demanding changes, others were taking actions with their own hands.

In December 2020, the IFC.js library is released. IFC.js<sup>[i]</sup> is the first JavaScript library fully dedicated to parse IFC files so they can be displayed and manipulated in any web browser. Their mission is to provide AEC professionals with easy and free methods to build their own BIM tools. Besides the IFC parser and geometry generator (written in C++ and compiled via WebAssembly), it uses Three.js—the popular application programming interface (API) used to create and display 3D computer graphics in a web browser using WebGL. Additionally to the 3D geometry, it enables users to access all the BIM metadata directly from the browser for reports and scheduling. IFC.js provides a viewer with examples of how to create your own BIM application: Scene navigation, material changes, element selection by clicking on them, section plans, etc. Finally, they develop affordable courses to educate the community of how to use the library and how to create your own features.

This is a new reality for architects, that makes BIM tools more accessible. These tools are not only for professional coders anymore, anyone can start making them right away specifically for

the own workflows. IFC.js is just one free open-source online package that exists, but there are multiple for various purposes. The user can combine different packages to start making the application that better fit her own needs.

## Conclusion

---

[i] ISO <https://www.iso.org/home.html>

[ii] <https://www.iso.org/about-us.html>

[iii] ISO 19650

[iv] <https://thinkmoult.com/five-fundamental-facets-of-ethical-software.html>

[v] <https://www.buildingsmart.org/about/>

[vi] <https://www.buildingsmart.org/about/openbim/>

[vii] Dion Moulton published in his personal blog in December of 2018 a controversial post called “Why Revit is shit” <https://thinkmoult.com/why-revit-is-shit.html>. This is the first time that the author heard of the term “Black-box BIM” that inspired the title of this dissertation. In this article, Moulton describes all the issues that he experienced using the 2018 version of the popular software Revit at his architecture practice. The article is a sort of manifesto and theory behind BlenderBIM, software that he releases one year after.

[viii] Autodesk Revit (2023) <https://www.autodesk.ca/en/products/revit/overview> \$2,675 per year per user. Graphisoft Archicad (2022): <https://www.scan2cad.com/blog/cad/archicad-pricing/> \$3971/year per user. Price refers to the commercial version.

[ix] Rushkoff, 129-130.

[x] <https://www.freecadweb.org/>

[xi] <https://wiki.osarch.org/index.php?title=Blender>

[xii] <https://www.blender.org/press/blender-by-the-numbers-2019>

[xiii] <https://aecmag.com/bim/letter-to-autodesk-aec-customers-demand-better-value/>

[xiv] <https://osarch.org/2020/12/21/a-new-browser-ifc-viewer-is-released-in-ifc-js/>  
IFC.js: <https://ifcjs.github.io/info/>

[xv] <https://www.the-nordic-letter.com/>



[xvi]

[https://www.ace-cae.eu/services/news/?tx\\_ttnews%5BbackPid%5D=88&tx\\_ttnews%5Btt\\_news%5D=2386&cHash=2017df07b74dd3e2dc6a9f5b6b126767](https://www.ace-cae.eu/services/news/?tx_ttnews%5BbackPid%5D=88&tx_ttnews%5Btt_news%5D=2386&cHash=2017df07b74dd3e2dc6a9f5b6b126767)

[xvii] Frascari, *Eleven Exercises*, 34.