

AS226079

Manage Revit Digital Content using Forge Design Automation API for Revit

Mustafa Salaheldin Ali
SNC-Lavalin Atkins

Marc Durand
SNC-Lavalin Atkins

Learning Objectives

- Through cloud-based content management, discover how to automate client requirements and standardize project templates.
- Understand how to generate native project templates through Revit I/O in the cloud.
- Learn how to link to Forge for real-time query and visualization.
- Learn how to manipulate and manage content in the cloud.

Description

In this class, you will discover how to create native Revit content as well as how to do project setup directly from a web portal. In this class, we will demonstrate how we used the Design Automation for Revit API to implement:

1. Cloud based content management platform with client requirements and standards.
2. Project template generation.
3. Real time query and visualization.
4. Native Revit file creation.

All directly from our web application!

Speaker(s)

Mustafa Salaheldin

A multi-disciplinary subject-matter expert and one of the few professionals in the Digital Transformation field who has a strong technical experience of BIM and is mastering the full-stack application development at the same time.

Mustafa joined Atkins Middle East and Africa, part of the SNC-Lavalin Group in Canada, as the Data Science Manager.

In the middle of the pace of technology there was a necessity for him to start looking at applying new techniques like Machine Learning(ML) and Artificial Intelligence(AI) to make use of the capacity of the machines to help in solving design issues and provide optimum designs.

Previously working for Engineering Consultants Group (ECG) in Egypt as the head of BIM R&D, he started up the BIM implementation and automation.

Besides he has a Bc.S. in Computer and systems engineering, he is a LEED GA certified, Autodesk Expert Elite, Microsoft certified applications developer and Autodesk Authorized Developer.

Marc Durand

Director for Digital Disruption for Atkins Middle East and Africa, his deep technical expertise, entrepreneurial skills and high-level strategic planning brings a new expertise and strength to capitalize on the technological growth opportunities that exists in our markets today. With over 15 years' experience in leading roles in technology/AEC firms, of which 12 years were offshore, he has led technology research and development, implementation and project delivery across several tech firms in France, Germany, ME, including Faust Consult, Burt Hill, 3D Kyvoss and most recently in UAE as a partner and co-founder with iTech a management consultancy firm and provider of Building Information Management (BIM) technology services.

His appointment to Atkins enables full implementation of the Atkins digital strategy across the region. His focus is on enabling creation of new revenue streams through new technologies.

Marc is originally from Boulogne Sur Mer, France, where he completed his Master's Degree in Industrial Data Processes at the University of Littoral, Cote d'Opale France.

Introduction

Before we go through the objectives, we just wanted to cast some rays on what Forge Design Automation for Revit (a.k.a. Revit.io) is. So far, Revit.io is a private beta, which means that few people have access to such technology for testing and evaluation purposes.

What is Forge Design Automation for Revit (Revit.io)?

Revit.io is part of the Forge Design Automation API for that lets you build web applications that create, read, and modify Revit models. It will allow developers to access the “business logic” of Revit on the cloud. Like Design Automation API for AutoCAD, developers can create client applications, which speak to the Revit engine via REST API.

Developers will be able to automate existing workflows or create new cloud workflows. Because Design Automation API for Revit is a Forge service, we can take advantage of other Forge services to easily connect our applications together.

What are some examples of what the service can do?

There are many options falling under three major workflows: Create, Extract, and Modify.

Create

- Convert from a third-party format to RVT/RFA
- Generate documentation from a specification
- Create families from a catalog

Extract

- Put data into a format Revit doesn't natively support
- Share model info with consultants who don't have Revit
- Create daily model reports

Modify

- Design option generation
- Automatically replace out-of-date content
- Fix common modeling mistakes

Revit.io Model

Like other Forge services, you will create a Forge client application, which will call the REST API endpoints. You can use any language, which can make REST calls. This client application will also do any “front-end” work you need – displaying UI to an end-user, for example. In addition to the client application, we will create one or more Revit add-ins to execute against the Revit engine. They will be uploaded to the Design Automation service and your client application will refer to them when invoking the service. They can be kept private or public depending on what you want.

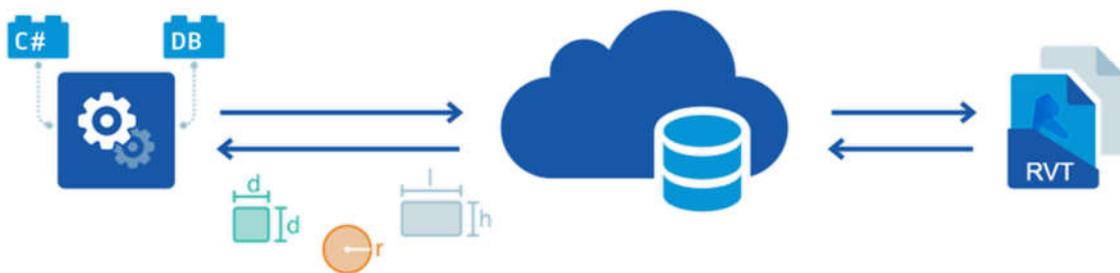


Fig.1 Architecture of Revit.io

Terminology:

There is some important terminology, which we should be aware of, in order to be able to develop the application.

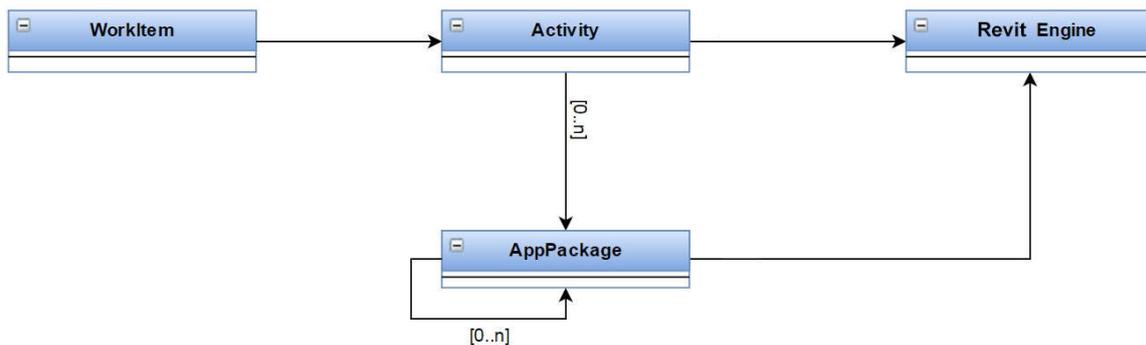


Fig.2 Revit.io components

In the following list I'll show you some of definitions and terms used:

- **Activity:** It is like a function definition, it specifies one or more apps, a set of inputs and outputs, and a product engine to run against.
- **WorkItem:** This is a specific invocation of an activity, indicating specific input files and an output location.
- **AppPackage:** Apps refers to one or more add-ins and can be used later by multiple applications.
- **Engine:** It is what executes a WorkItem job and processes the actions specified by an Activity.
- **NickName:** A nickname is a way to map a Forge App ClientId to a customized string. A nickname lets you create a friendly, easy-to-read string that can be used in place of the long Forge App ClientId.
- **Alias:** An alias is a label to a specific version of app or activity. Aliasing gives you more control over how you update appbundles and activities.

Design Automation, Design Automation for Revit, DA, DA4R and Revit.io will be used interchangeably to refer to the same thing.

Why Revit.io?

There is no difference between writing a code using the DA API and writing the code for a regular Revit add-in, they are almost equivalent except that in DA you don't have access to the UI components to the Revit interface and you only limited to address the db.

The real advantage is since the DA is a part of the Autodesk Forge platform that works on the cloud then you can make your add-in more powerful by giving it unlimited access to all other cloud service providers.

By connecting to other cloud services, you can use advanced features like Lambda functions from Amazon and even plug your add-in to the ML and AI provided from Microsoft or Google for example.

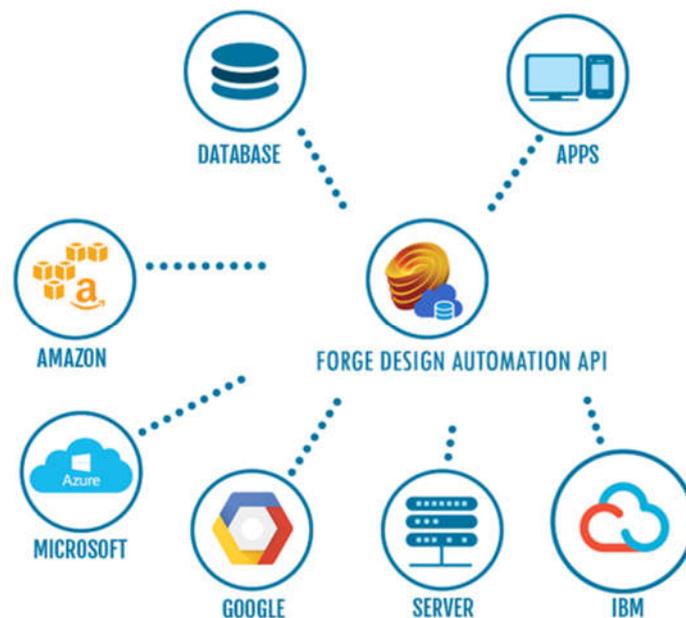


Fig.3 Revit.io integration with cloud-platforms

The harmonization between a digital operating model, predictive analytics, and a microservices architecture allows digital organizations to quickly take customer desires and feedback and turn them into actionable, revenue-generating business solutions. As Artificial Intelligence or AI becomes more prevalent in business applications, these organizations use rudimentary machine learning to improve decision making in big data analysis.

Cloud based content management with automated client requirements and standardization

Using DA4R we were able to build a Revit integrated cloud-based Revit content management platform that helps us to produce healthier, higher quality models faster.

Our solution helps Digital Managers to organize and control the quality of their Revit family libraries.

How it works

The first step is to upload the Revit model to portal. Once the upload process is done the Web App will send a copy of the Revit file to the Revit.io engine and trigger the Families extraction add-in. As soon as DA4R finishes the extraction, it will save the Families into folders named after their Category names. As the DA4R only will allow us to get the result of our processing in the form of single file, then we need to zip all the folders into a single archive.

The archived categorized families will then be uploaded to an AWS bucket where we will make benefit from the Amazon S3 services and the Lambda functions. After the upload is complete, we will use Lambda function to trigger an extraction process, which will unzip the contents into another bucket. Whenever the extraction is finished, another Lambda function will be triggered to upload the extracted families into their related buckets in the Forge associated buckets. Applying the corporate standard to the families like file name convention, validating the families or testing the parameters can take place at any stage after the extraction, either by triggering another Lambda function or at your Web App after the Families are uploaded to the Forge buckets.

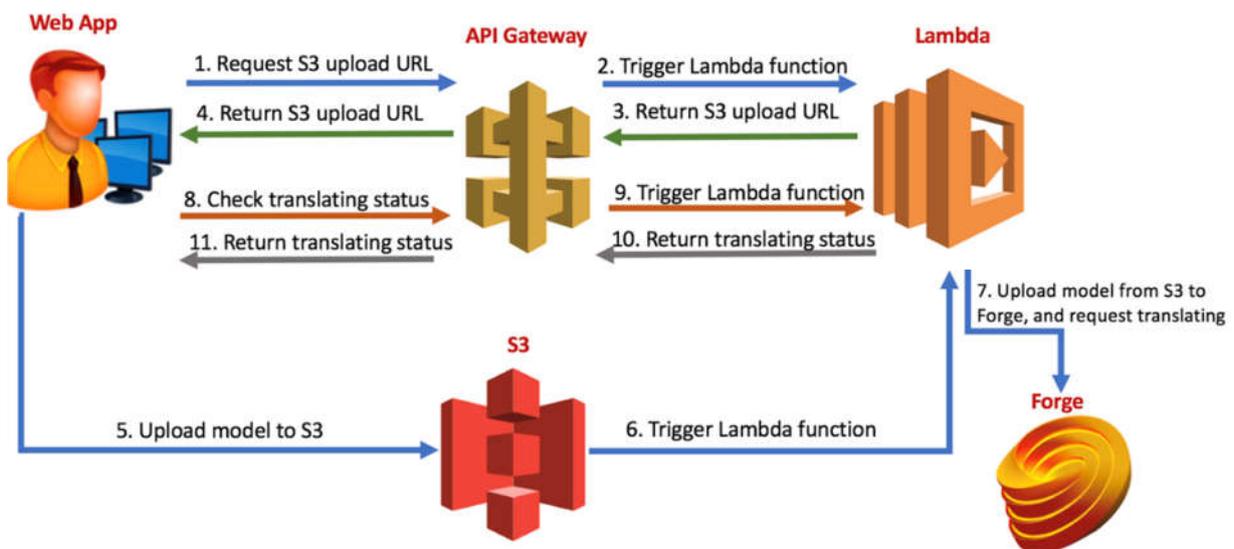


Fig.4 Revit.io integration with cloud-platforms

Advantages

By connecting to Cloud-Services from different vendors, you can improve your web application by integrating many Machine Learning and AI algorithms with it.

This will help you quickly build projects by getting suggestions from the ML about the Families you need to use or optimum design layout and building orientation for instance.

As you are keeping all your content in one place, you have a Single point of truth where everything is up-to-date and sanitized. In addition, that will make you sure of Templates consistency.

Generating native project templates through Revit I/O on the cloud

The most powerful benefits from using Revit.io is the ability of generating Revit digital content on the cloud. Centralization will lead to a single point of truth consistency and easy maintainability, tracking the Project history and automatically update the status.

Creating Templates

Creating a robust collection of content is critical to any architectural design firm. In this section, we will consider the issue of content as it pertains to your project templates. Instead of using content that is preloaded in the templates, we can provide it externally and load when needed. You have the choice of which families should be included.

Embedding the Loadable families

Using DA4R we were able to create templates that contains the corporate's Families from a standard Families Catalogue. This has been achieved by gathering the families from each project uploaded to the portal, validated and making sure it is following the standard.

Note that many Families are customized and modified through the project lifetime. By using a simple ML algorithm you can create a Suggestion System which is able to recognize the Families of the same geometry but different in parameters. That way the system is able to do versioning and ask you which version you want to keep or you need to keep both or merge them into one.

The ML algorithm can recognize the type of the project from the EIR and then suggest you the right Families to include in your project.

In our portal user can drag and drop or simply select the required Families for the next project into a Forge viewer; User can also reposition and change the orientation of the each Family.

Once you finish the laying out, you can execute the DA Workitem that will create a Revit template file with all the positioned Families at the same location.

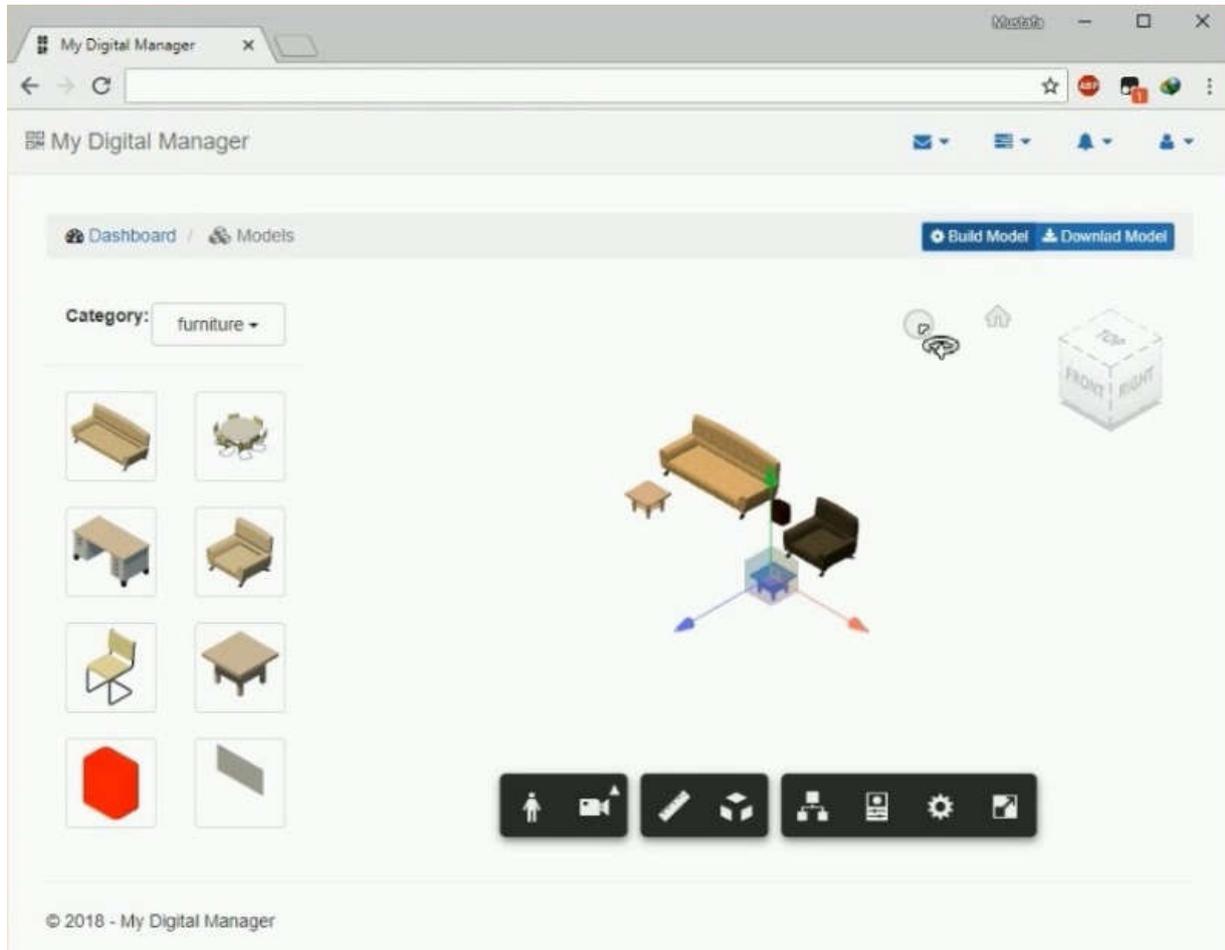


Fig.5 Family Catalogue created by Revit.io

Embedding the System families

It is easy now to create all your System Families now online on the fly. Keeping your Families centralized will assure the consistency of your model and guarantee easy maintaining and upgrade.

In our portal, we are able to create different type of System Families like Walls, Floors, Ceilings and Roofs. Once the Family is created, it is easy to embed it into the template.

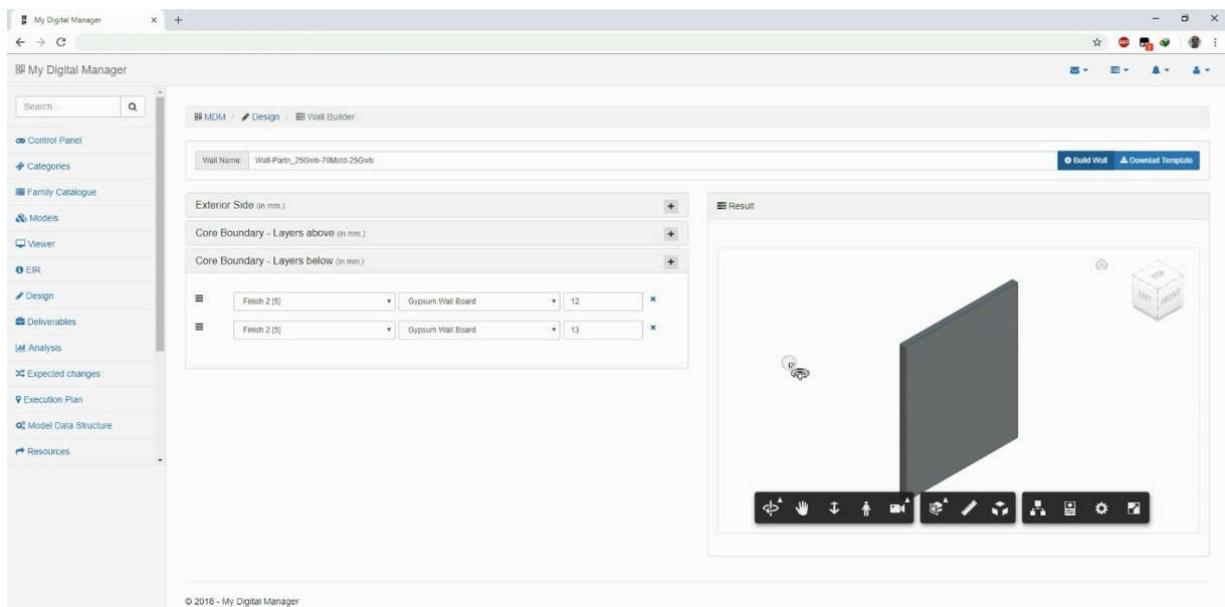


Fig.6 Creating Revit system Families using Revit.io

Learn how to link to Forge for Real-Time query and visualization

As DA4R is a part of the Forge platform, it became so easy to integrate it with different components of Forge. Making advantage of this privilege, we managed to push the model geometry and data back and forth between Forge Viewer and Revit.io. This allows us to create models in the Forge Viewer then send it to Revit.io for further processing and file creation. On the other hand, we managed to extract information from the model directly using DA4R to build dashboards, insights and reports.

Real-Time Query:

In the following example, we will show how we used Revit.io to make enquiries from Revit Models using DA4R. To check the model health there are many indicators, one of them is the Model Warnings. By checking the status of the Model Warnings, we can solve issues that can enhance the performance of the model and minimize the errors.

In our portal, we can enquiry the warnings directly from the model and present them in different types of graphs.

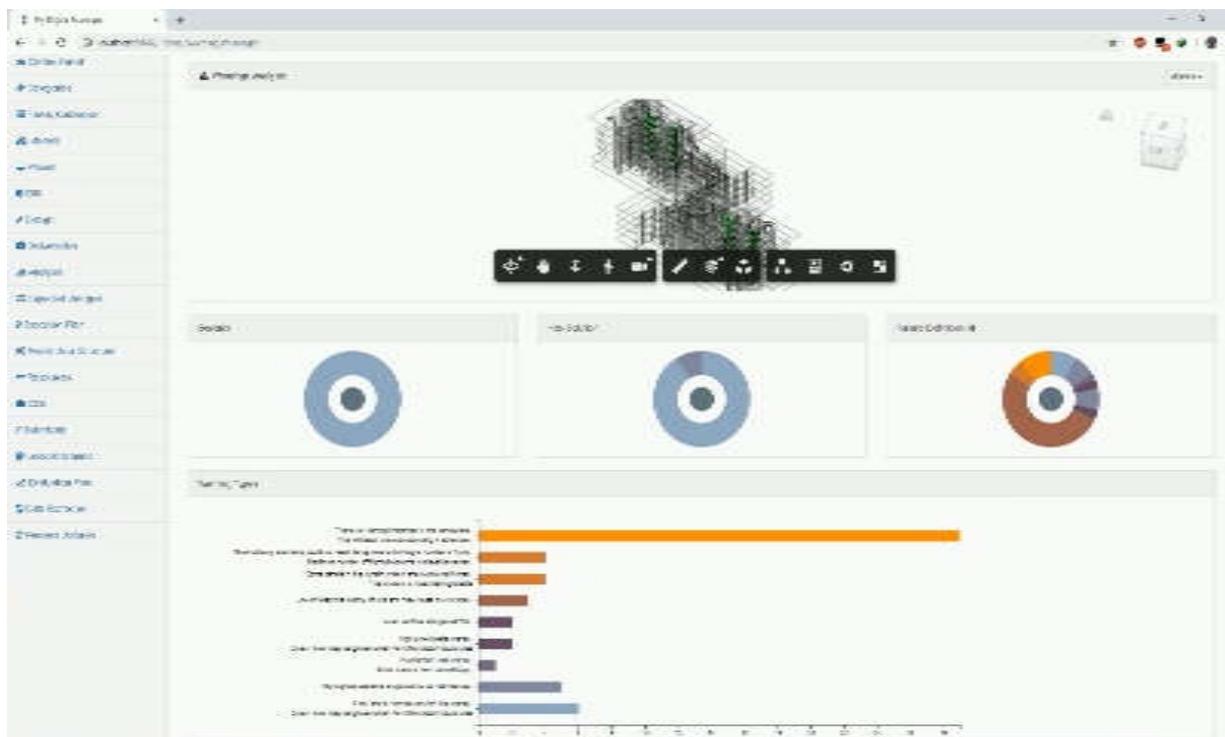


Fig.7 Creating Revit Model Queries on Real-Time using Revit.io

Real-Time Visualization:

In this example, we will demonstrate how using Revit.io can be very helpful to build a dashboard for Revit Model. One of our projects was a MasterPlan project, where the client needed to visually get analytics of the GIS information of the plots to help him making some decisions. Using Revit.io, we were able to extract the data directly from the model, where all the GIS data were stored, and then recalled them online directly from the model to build the insights dashboard.

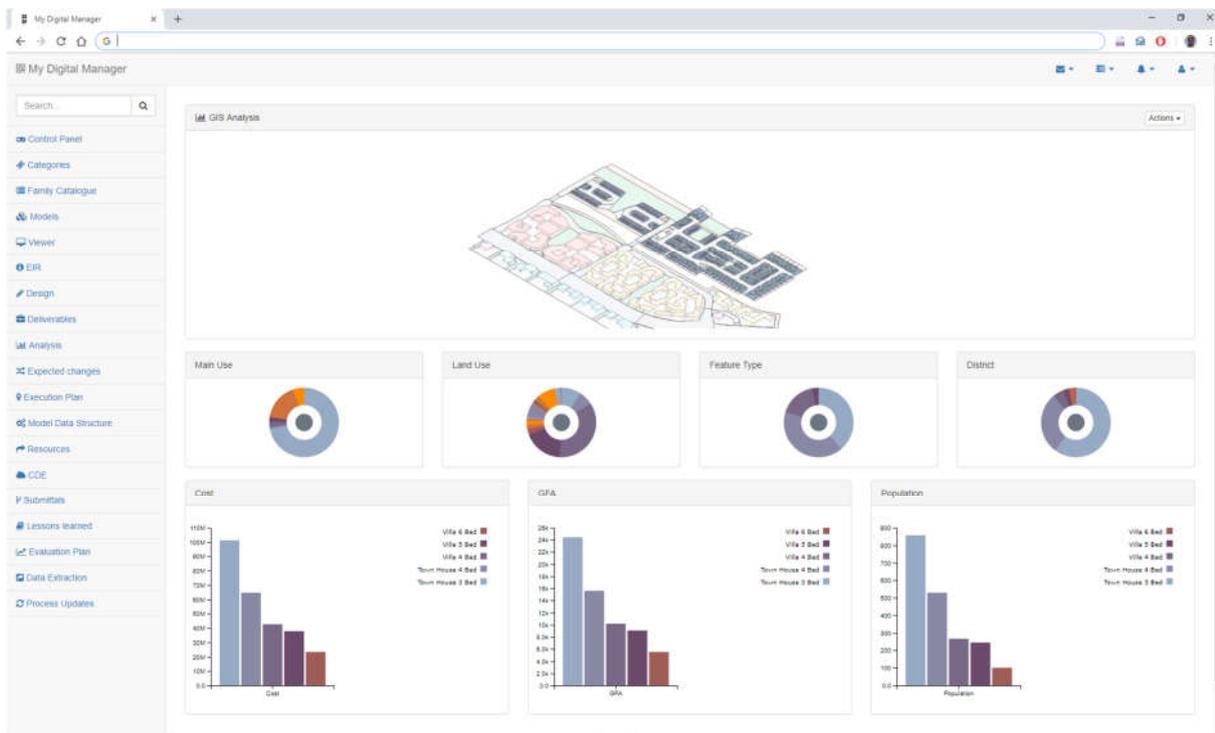


Fig.8 Getting Revit Model Analytics using Revit.io

Learn how to manipulate and manage content in the cloud

How many times did you need to sanitize your content after delivering a project?

Maintaining the Models and keeping them healthy are very important to ensure the consistency of the project and its robustness. This is why Revit.io is playing a great role in that, as we can manipulate the Revit project contents to verify that we met the corporate standards to keep the model healthy as possible.

In our example, we created a set of criteria as a checklist that can be applied to the Revit project to ensure its cleanliness. Such list can contain some regular tasks like opening the file with Audit, deleting all the unused Sections, removing all unplaced Rooms and Groups, resaving the file with Compact option selected. That way, as we manipulated the Revit file content, we can get better Model performance and minimize the storage space.

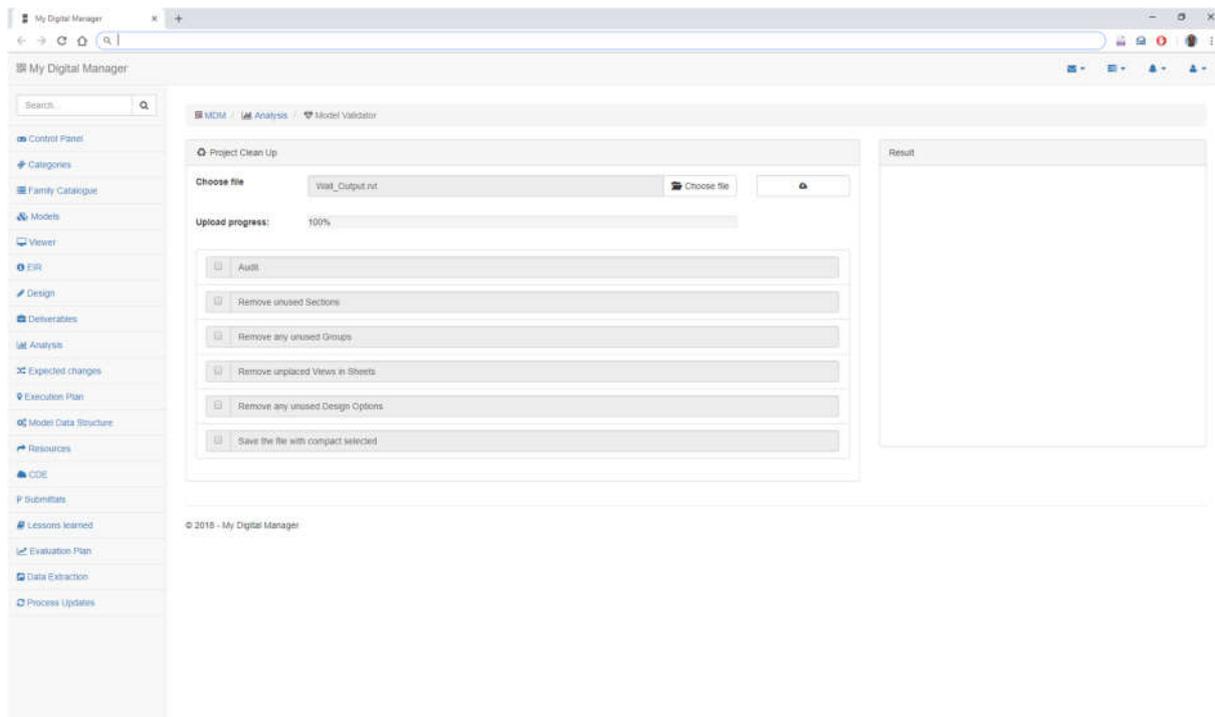


Fig.9 Manipulating Revit Model content for sanitizing

Conclusion

With new technologies constantly emerging, companies need to be at the forefront in order to continue providing competitive and relevant offerings to their clients. The evolution of the AEC industry dictates a shift in which companies need to adapt to such changes. The recent development and deployment of Building Information Modelling (BIM), for example, emphasizes the importance of such endeavors.

By finding a way to glue the new technologies with the BIM standards, we can say that Revit.io is the best technology to do so. As this technology is still evolving we expect more and more in the future to come.