[BES468450]

[Solve Clashes Automatically with Forge, BIM 360 & Revit Design Automation]

[Xiaodong Liang] [Don Whittle] [Mikako Harada]

[Autodesk]

Learning Objectives

- Learn the capabilities of the BIM 360 Model Coordination API.
- Discover the capabilities of the Design Automation API for Revit.
- Learn how to integrate various Forge components to automate workflow.
- Explore possible ways to solve clash automatically.

Description

Availability of BIM 360 Model Coordination software made the clash detection portion completely automatic; i.e., uploading files to a given space or a folder in BIM 360 Docs software automatically triggers cloud-based, Navisworks-like, clash-detection service. However, how to interpret those result and resolve the issues are currently completely left up to the users. In this session, we will explore possibilities to further automate some portion of analyzing clashes and suggesting possible resolution options. We approach this goal by making use of all the available Forge capabilities, combining the BIM 360 Model Coordination API to detect clashes, keeping track of issues and resolution with the Issues API to accumulate as a knowledge base, and suggesting/making possible changes to the model using Design Automations for Revit software. We will share how much we accomplished with our dream project, and how we did it.

Speaker(s)

Xiadong Liang
Principal Advocate. joined Autodesk in 2007 as a developer advocate. Starting with desktop products' APIs, Xiaodong is now engaging in Autodesk Forge and AEC solutions, including BIM360 and Revit etc. He is one of the contributors of Forge blog and ADN DevBlog, researching integrations by Forge and web & cloud technology . Xiaodong is based in China, yet mingling with the global community of Autodesk and programming. He is also the ambassador of Intel IoT technology

Don Whittle

Software Architect. Joined Autodesk in Feb 2017 as the Software Architect for BIM 360 Model Coordination, based in Sheffield, UK and working with teams across BIM 360. Have over 20 years experience of building distributed client server applications in hosted environments and have worked on applications and services in many different verticals for both large enterprises and small start-ups. Since 2011 I have been exclusively working with teams developing native cross-platform mobile and browser-based web applications running in the cloud on Azure and AWS.

Mikako Harada

Senior Manager AEC. Senior manager for Forge/Developer Technical Services team at Autodesk. She provides API technical support for AEC products, such as Revit and BIM 360. Her interest is in the areas of interactive techniques, optimization and layout synthesis. More at her blog: https://fieldofviewblog.wordpress.com/

Disclaimer

This presentation includes the demonstration of sample application and discusses about the potential use case for future. **They are experimental, using public APIs available today.** They are not meant to represent the product roadmap. Rather, they are meant to help us imagine what is possible and learn what's needed to work toward our goal.

Agenda:

We organize this talk as follows:

- We start with background and motivation to our experiment.
- We then talk about ideas through demonstrations of sample applications for analyzing and solving clashes.
- The sample applications you see in this sessions are all built using publicly available API. We will give you an overview of developer tools used behind the scenes, with focus on two main components: Model Coordination and Design Automation.
- We conclude with the discussion about what we have learned and future work.

Background and Motivation

This talk is all about Coordination or Clashes.
Navisworks is a popular tool for clash detection, which is a desktop application.BIM 360 Glue brought clash detection functionality to the cloud. BIM 360 Model Coordination brought Naviswork-like engine to cloud and further brought it to the next level, by integrating to BIM 360 Docs or common data environment.

Currently available coordination tools can detect clashes automatically. However, the user is presented with thousands of clashes. What the user do next?
Typically the user would export the result to Excel file to analyze. or more recently to a popular tool like PowerBI. Then need to come up with the resolution. However, the currently available tool provides very little help beyond detecting clashes.

So the goal with this experiment in this class is to go beyond clash detection. We would like to explore how much we can go beyond clash detection.

Another area of motivation is technologies.
Cloud in general, and Forge is a set of web services that Autodesk uses and provides to 3rd parties as a “building blocks” for design data.

Model Coordination and Design Automation API are recently addition to Forge API.

- Model Coordination API is a powerful API. It is built with API first philosophy. It gives you much more flexibility to query data beyond what you see in UI. It gives you total freedom to customize and present the resulting data.
- With availability of Design Automation, we can now modify a model data in the cloud without requiring the users to go to a desktop product.

Although they are not the technologies we used in our experiments, Generative Design and Machine Learning are also the areas which we need to prepare for future; i.e., How to accumulate data to apply these technologies.

Analyzing Clashes

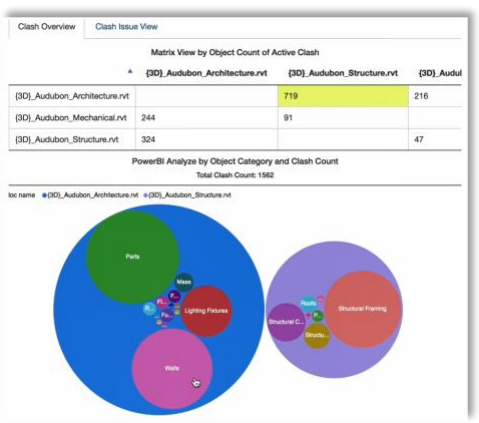
The most important to analyze is the clash can be visualized. The participants in the project will need to see the clashed elements interactively, which can help them to make the best decision. The project models may have huge number of clashes. In reality, it does not happen people will check all clashes in one view. Instead, a view with specific context will be more useful. e.g. by types, by levels, by rooms, by functionalities.

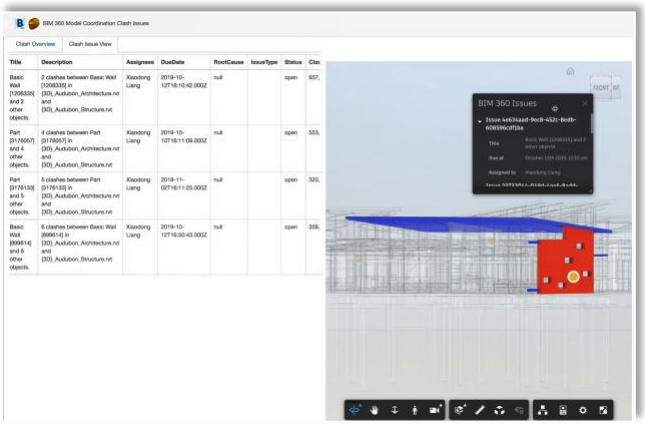
We can take advantage of many visual tools to save the energy to check many clashes. Bubble diagram, matrix table with PowerBI , or room/level checking with forge viewer.

For some advanced analysis, we may probably connect with Design Automation , design engine to check in depth.

Bubble Diagram, Issues

PowerBI is widely used in the industry to provides rich types of visuals: Power BI core engine plays a role to bridge the visuals interactively and intelligently. e.g. when a cluster of types in Pie chart is selected, the corresponding rows of table will be also summarized. It can support various data source such as file, database, streaming data, on-demand custom data etc. It also allows extending the basic visuals abilities. The 3rd party can produce their own visuals to present data in unique ways. Since it provides web services and embedded container, the integrations with visuals and data connecting will be becomes pretty more efficient.

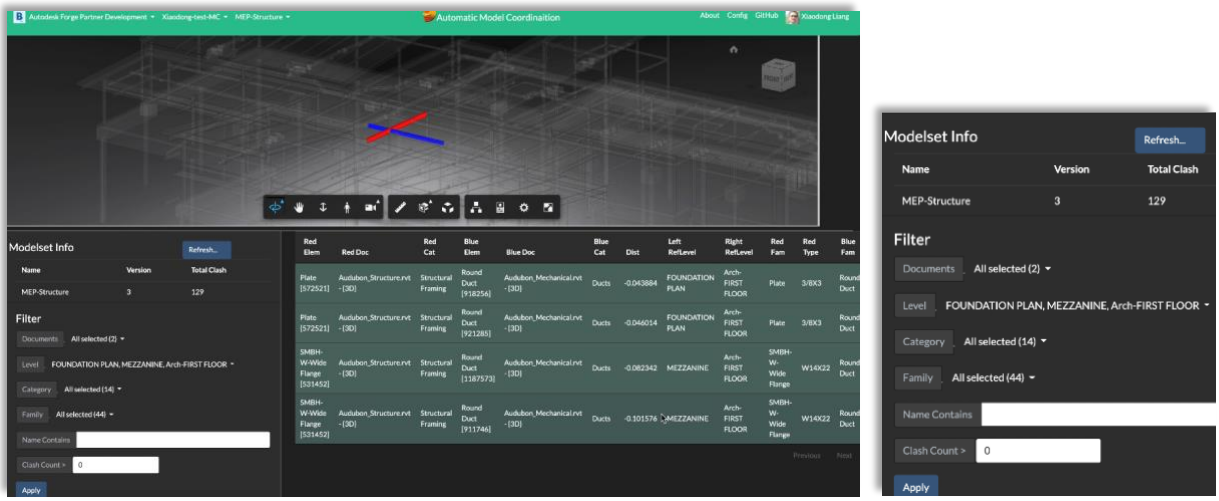




This sample checks clashes of pair of documents. The clash data are from model coordination of BIM360, , they are clustered by elements type in bubble diagram in Power BI report intelligently. Table view presents clashes of individual element. The two views are linked. They also associate with Forge Viewer for user to inspect clashes interactively . If helped to narrow down inspection of clashes and focus on the specific types and documents

Filtering by Properties

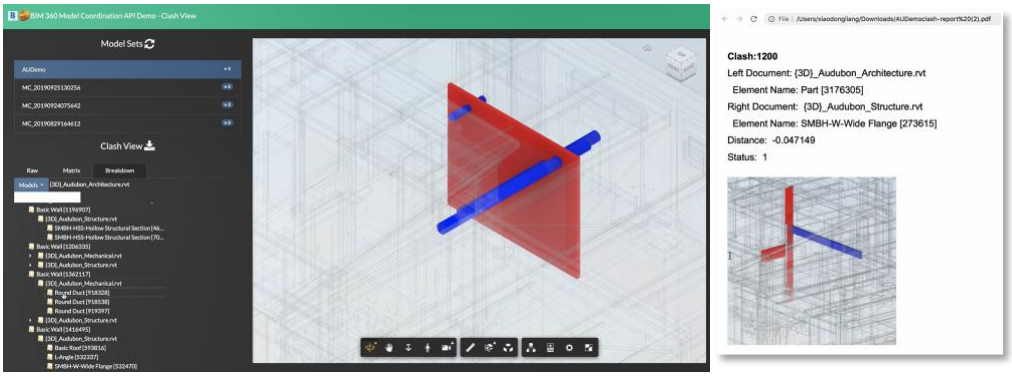
By Model coordination of BIM 360, we can not only get clash data, but also the elements properties, by these information, this sample provides the ways to check clashes of elements in specific levels , elements with specific categories, families, elements in specific documents, or even clashes of elements with specific names. It can help the users to focus on the most important clashes quickly.



This is the demo video, as you can see , at the beginning, all clashes are listed. After applying with filter, only a limit number of clashes will be displayed.

Customized Viewer and Report Generation

The next sample gets clash data, and allow the user to inspect clashes of model coordination space in breakdown view. In the another word: which elements in other documents clashed with specific elements in one document. It is similar to the breakdown view of BIM360 product, but we can extend with more abilities such as, search specific elements by keyword, export PDF with rich contents for team to review.



Solving Clashes

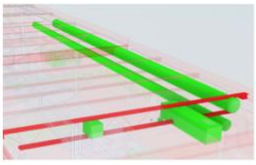
Based on analyzing , we can move to the next step, solve clash. And our motivation is to resolve automatically without manually open design tool (such as Revit) to edit.

When we talk about resolve clashes something we will need to think about, typically, the models will need to be modified. but, The question is: what will be the options to resolve? Move elements, change family, size, or make MEP system with U shape to avoid obstruction ?

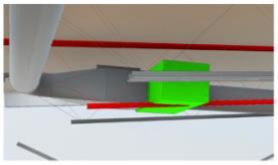
In addition, before concrete modification, we may want to preview the modification, By Forge viewer, which is well known people can see the models interactively in the browser., we can do something, like previewing movement. We can transform the objects. It does not change the model itself, but may still be useful to explore the possibility of movement. Even you could transform partial meshes of the elements (such as a wall of one room). In another word, transient modification is possible with viewer.

Another type of understanding is about a characteristics of clash. Some clashes type would be challenge. e.g..

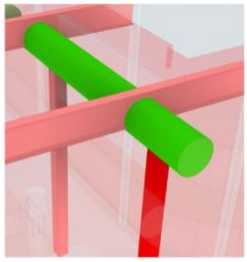
- Hard to decide which direction to move
- Solve clash may cause new clash
- Change element would be fail if associating with other elements, the design tool does not not allow you edit like that way
- Not suitable to make angle with primary pipe or duct



Transfer works only a specific direction



Solving a clash may cause new clash

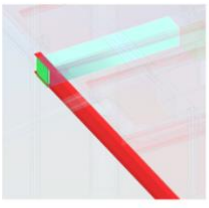


Change of elements will cause connected elements

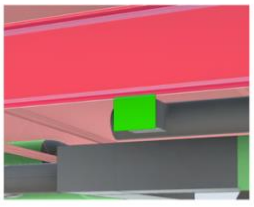
U-shape is not suitable

While resolve options for some types of clashes could be easily decided e.g.

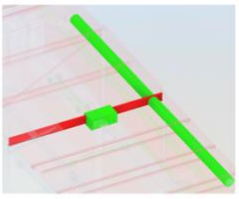
- obviously, it can be resolved by move a little bit of one element , up/down,
- Or we could make a U shape on Pipe/duct which are not primary elements, to avoid the obstruction.



Transfer arbitrary direction



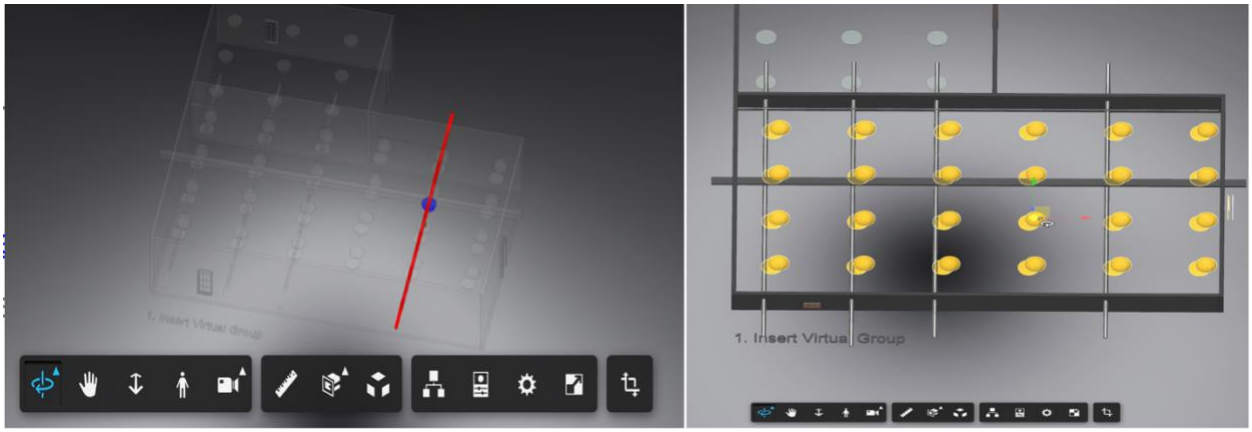
Change family type/size



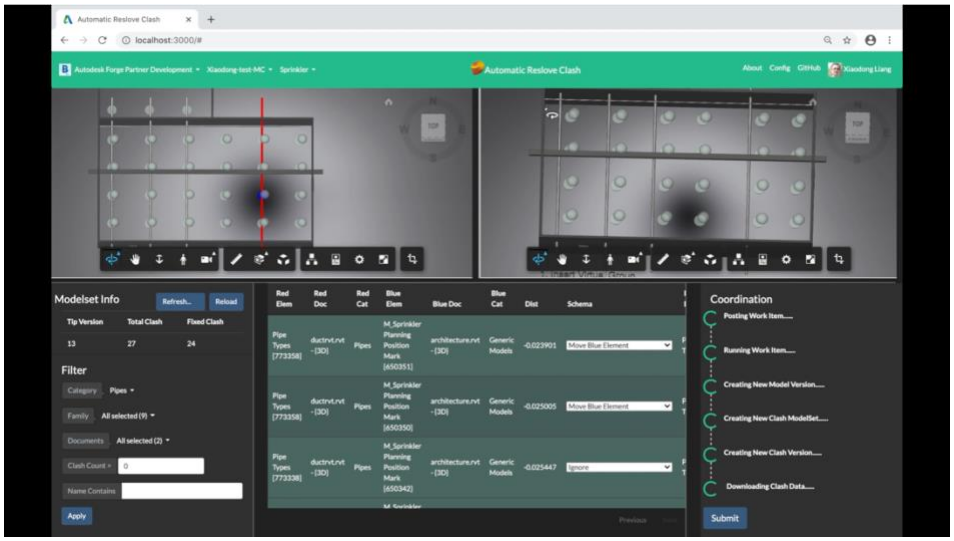
MEP elements (duct/pipe) that can go around (U-shape)

Demo: Solve Clash of Sprinklers with Pipes

With that, we tried with some experiments. The first is to move groups of clashes of sprinklers and pipes. It firstly allows the user to preview the transformation inside forge viewer, and send the parameters to design automation of forge. Design automation will modify the Revit model on the cloud, and finally update with new version of refit model. When model coordination starts the next checking, it will find the clashes will disappear.



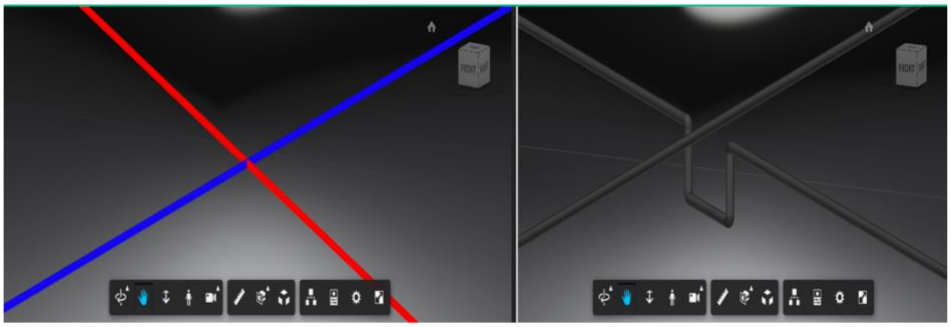
So, as you can see, two models, one is architecture with some sprinklers, the other is a pipe system. With filter, we can narrow down to the clashes of pipes only. Next. We try to move groups of sprinklers inside forge viewer to see the effect, decide the direction and distance. Then send to design automation. The job will start , after new model is available, it will send to bim360, and trigger new model coordination. Finally, we can get the updated model and the clashes are resolved. You can also verify inside BIM360. The users do not need to modify in Revit back and forth.

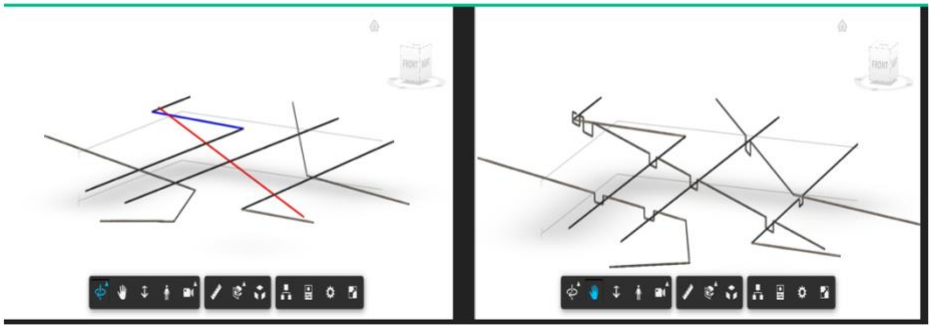


P.s. the sprinklers sample Revit models are from Fire eHelp of AGACAD. Thanks AGACAD!

Demo: Solve Multiple Pipes Obstruction

The next experiment is to solve obstruction of pipes. As mentioned, in some cases, we can make U shape of pipe /duct ,in order to resolve the clash with other elements. So this sample does the job to split the pipe, make U shape, but keeping the original connectors with other sections.



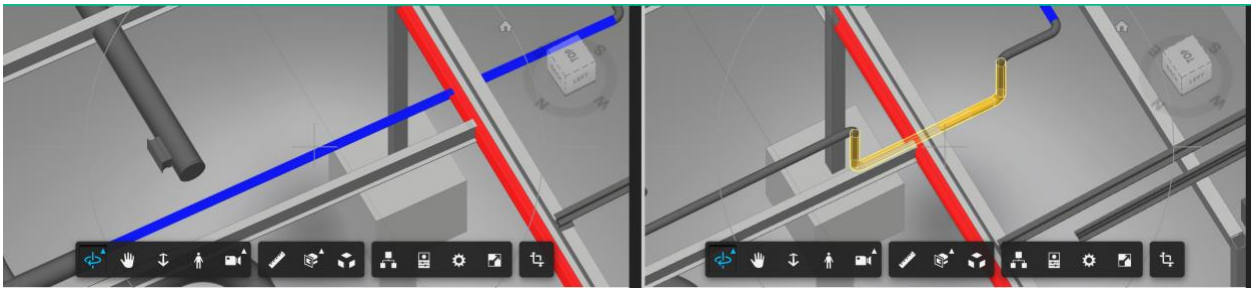


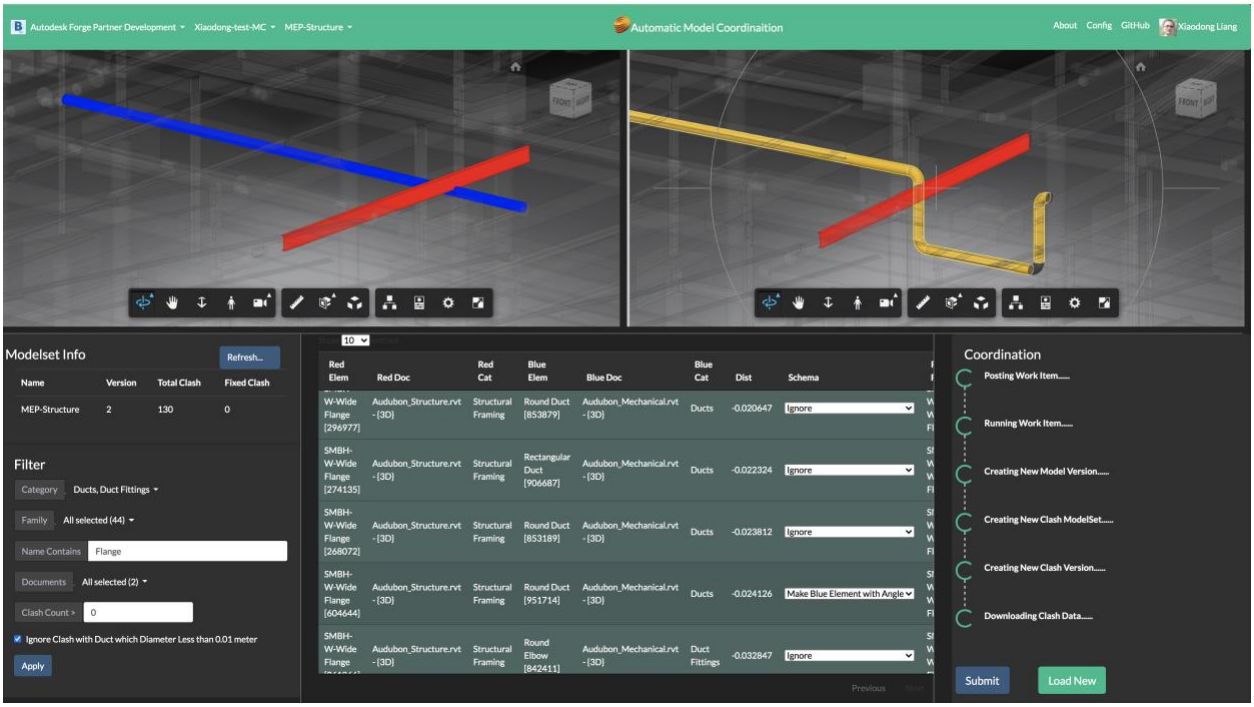
Thee are simple test models. Two groups of pipes. They have clashed. The users check them, and decide the option to make one pipe system with U shape. The app will get the clash center and build the segments array. The parameters will be sent to design automation of Revit. Revit engine will modify the source model and generate a new version. In the new version, all clash segments are now U shape, the clashes are solved.

Demo: Solve Clashes of Duct with Flange

One more demo is tested with real design model. It shows the clash scenario of structure and mechanical. These two models. One is structure the other is mechanical. There are a lot of clashes. To quick demo, we narrow down to one clash only. It is a pipe clashing with a flange. well. Obviously, the best option to solve would be make pipe with U shape to avoid the clash, while keeping its original start point and end point and connectors with other pipes.

After design automation processed, in the new model, the U type is available, and the previous clash is resolved. To make it clearer, I hided the top floor and roof, going inside in the structure. We can see the difference before and after.

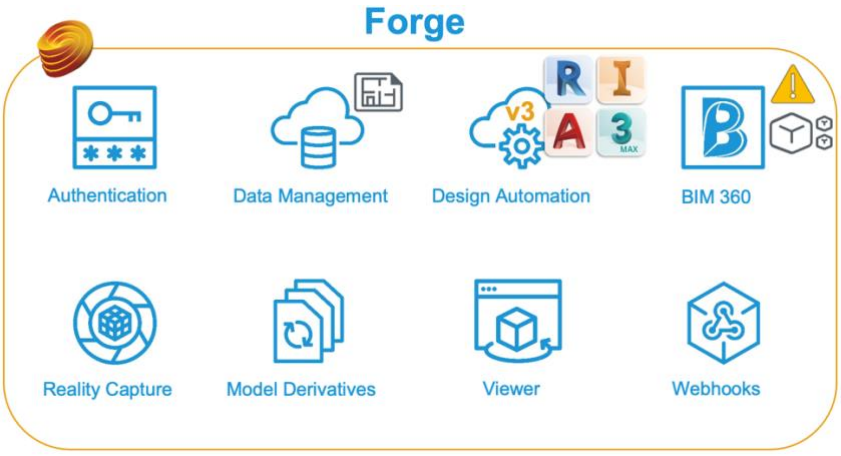




These are just experiments. It has not yet designed for much more complicated scenarios. But it shows the possibilities for some scenarios and it proves it is feasible.

Developer Tools Behind

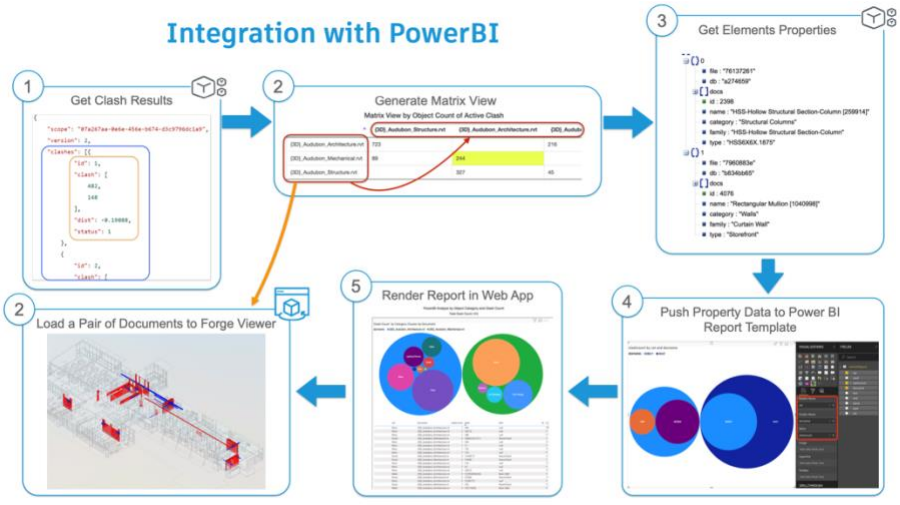
All samples above are integrated with Autodesk Forge technologies. As you have known Forge is a cloud collection with web services and APIs.Totally speaking, these samples are combinations of Forge web services and APIs. e.g. data management to manage the file and version in BIM360, design automation is used to update Revit models with the parameters, forge viewer loads models and also helps the users to check clash interactively. At BIM360 side, the model coordination API play the role to provide the clash data, elements properties for analysis.



Workflow of Integration Power BI with Clash Data from Model Coordination

This is a comprehensive sample.

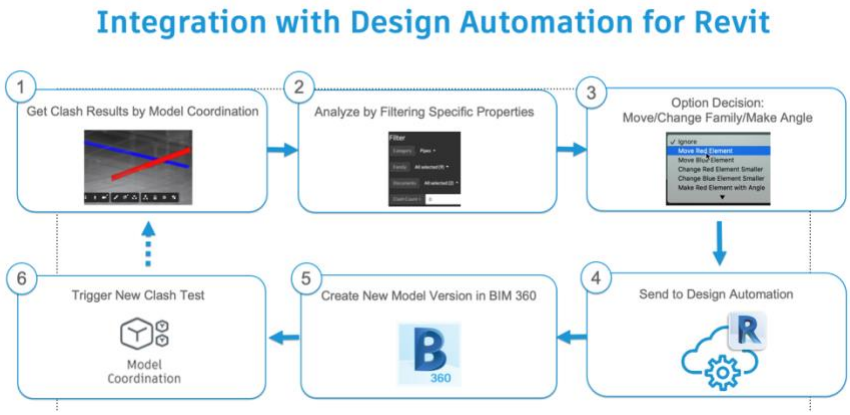
1. It calls BIM 360 API to get model coordination collection of one project. As we know, model coordination checks the aggregated models to detect the clash of the designs.
2. this sample analyzes the clash data, and build matrix of documents clashes in the coordination space, when the user clicks a pair of documents, the corresponding documents will be loaded into Forge viewer.
3. in the same time, the code will dump the properties of the clash elements and send to Power BI report template.
4. once the data is refreshed, the report will be update. The report presents a bubble chart and a table view. The bubble chart indicates how many elements are clashed in one documents, clustered by the element types. e.g. windows, doors, ducts pipes, walls, framing, air terminals etc.
5. the updated report is reloaded in the web app. The bubble chart and table view are associated, once the one bubble is selected, the summarized table rows will be highlighted. The corresponding elements will also be highlighted in Forge Viewer.



Integration with Design Automation for Revit

About the samples of resolving clash automatically, their basic workflows are similar.

1. get clash data by model coordination. It will tell clashed elements and their properties
2. the user will check the clashes by some specific properties. By this way, he can check the most important clash more efficiently.
3. the user will decide the option to resolve the clash, maybe move element, maybe make U type for pipe/duct.
4. once the options are decided, the related parameters will be sent to design automation of Forge. Design Automation will download the old version model. Update with the parameters, and generate the new model
5. the new model will be sent to bim 360 folder to create a new version.
6. Once a new version is available, model coordination core engine will start to check the the models and generate new clash data. The code will get the data and render with new documents. The user can compare the old version and new version. Ensure the clash has been resolved.



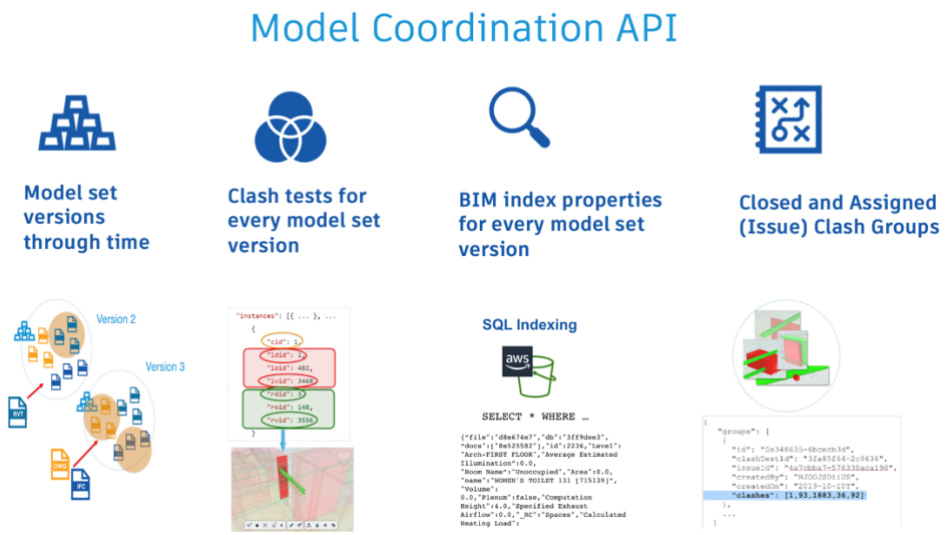
So, you can see the cool things with these samples are not only Forge Viewer, behind the scene, model coordination api and design automation api are working.

Model Coordination API

Model Coordination API provides 4 categories APIs so far.

- Modelset: managing aggregated models in the model coordination space. We can tigger a new clash test with the models in this space. A model set is one snapshot of aggregated models. Model coordination will start a new clash test with one snapshot. After it is completes, it will check if there is new update, and trigger next clash test.
- Clash tests: it provides the clash results of the coordination. By the data, we can know the clashed elements, which documents they belong to. And build the map with other clash information
- Index Property API: it provides you to query elements properties by SQL schema on the cloud.
- Clash issue API: it allows the developer to manage clash issues. Create or read. Each clash issue is based on general issue. So you can use Issue API to get its general

information. While the clash issue also stores clash ids, which are specific with model coordination.



The demo samples heavily adopted Clash API and Index Property API: Clash API tells the clash instance with the elements id of the clashed object, the elements id are dbid we used with Forge Viewer API. The API can also map with the manifest of the documents, so we can know which documents the elements belong to. And load the corresponding documents and objects, the clash can be visual in Forge Viewer.

The other is index property API: It is an interesting API. You can use SQL schema to query the elements properties on the cloud. It follows AWS S3 Selection schema. Each property has a unique key in the index. Whe query the elements with the keys, the properties will be returned. By this way, we can analyze clash with properties on the cloud.

Available and Unavailable Data by Model Coordination API

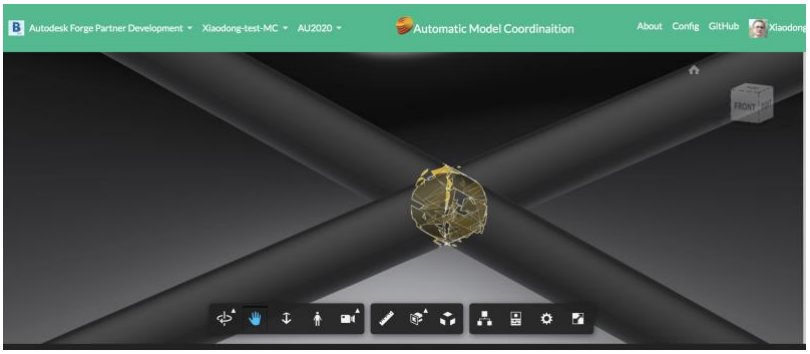
Model Coordination API provided plentiful information of clash, element id, document id etc.. especially, we can get elements properties by SQL query on server side. It is very efficient to us to build the analyzed data.
However, it does not tell the clash box, clash center. So when we want to make U shape of pipe/duct, it is difficult to calculate the center. It does not either tell which direction, the clash can be resolved with shortest distance. So the option to move or up / down will depend on manual interference.

Last it does not provide wekhook when new model clash test is done. So it is not efficient to get status by polling

We have sent these feedbacks to engineer team. Hopefully something would be available in the future.

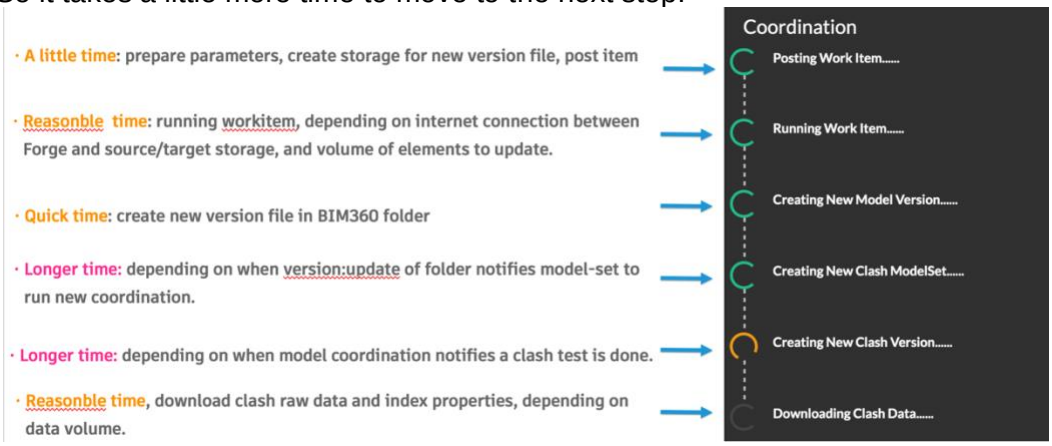
Calculation of Clash Center and Range

In the sample of making U shape for pipe. Once we know the clashed elements, we get mesh data by Forge Viewer API, and use CSG library to build the body with the mesh, calculate the intersection mesh, from which the body sphere center and radius will be used as clash center and range.



Process Time

And one thing with the whole workflow is: in some steps, the clash test completes, but not notify timely. So it takes a little more time to move to the next step.

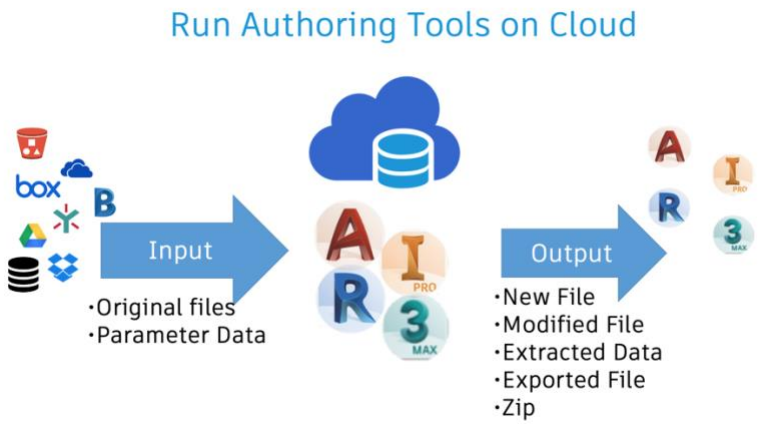


Design Automation API

Although we can preview some model update inside Forge Viewer, Forge Viewer is not a tool to modify the source model. While our motivation is to resolve the clash without depending on manual editing in the design software. So we took advantage of design automation API of Forge.

Basically speaking, Design Automation provides design engines on the cloud. It can handle the original model files with the parameters, and update the model, it create new model. Or extract

customized data. Or even output a couple of formats of data (e.g. you may export elements data to csv, while you also want to export model to ifc, stl, obj..). Finally make a zip.
Currently, Forge has 4 engines: Revit: AutoCAD, Inventor and 3DsMax.

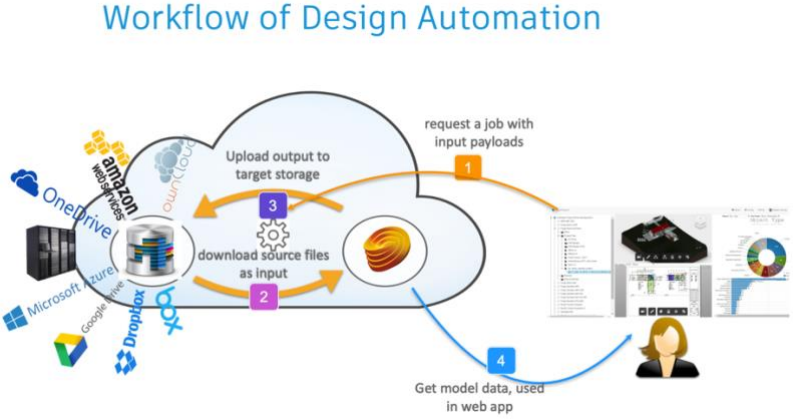


The workflow of design automation is, firstly, the user will submit a request: either creating a new model with template, or asking for editing model with input parameters, or asking for export some data from the model. app will request a job of Forge Design Automation with input payloads

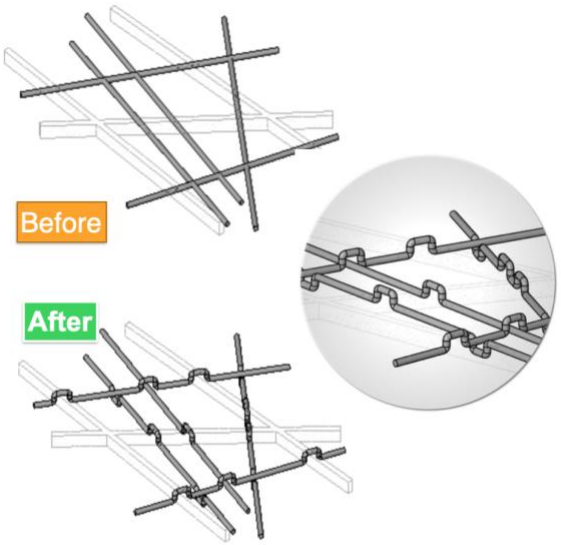
When design automation starts to work, it will download the source file model. The download link is specified by the workitem. After the model is ready and the parameters data is ready, design automation will start to follow the plugin function to handle the model or extract the data. The plugin is defined by you developer.

When the job is done, the output file will be sent to the target storage of cloud. The target url is also defined in your requested job.

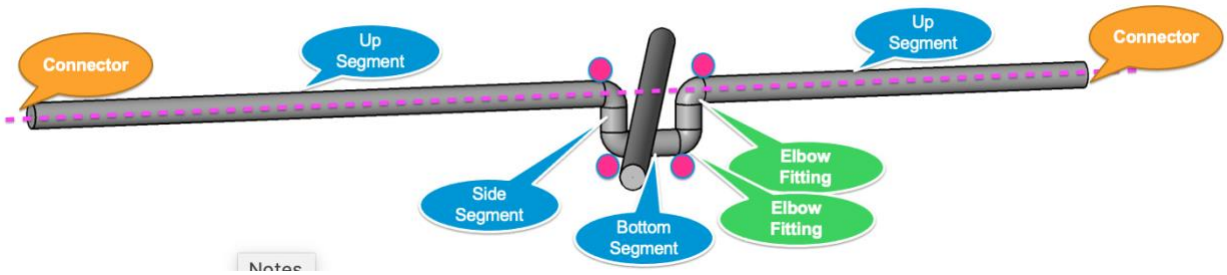
When the new file is ready, your ap can proceed further, e.g. send the updated model to Forge translator, in order to display it in your app. Now, your end user will be able to see the updated models in your app.



So, the key thing is the plugin you defined. Design automation can do almost anything like you work on local plugin.
Our experiments to make U shape is one example. We got idea from Revit SDK sample. It can detect all pipes/ducts interferences in one Revit model, get the clash position, and make the pipe/ducts with U shape at that position. In the snapshot, before is clashed status, after is when the clashes are resolved. The pipes bypass the other pipes at that point.



The demo sample borrows some core logic of the code, and customized for our samples for automatic clash resolving. Similarly, my plugin will get center of clash, get the direction of the base pipe, calculates the offset points, create the pipe section, and finally makes elbows with these sections.



Lessons Learned & Future Work

Our thought on lessons learned and future work. The goal of our initial experiment was to see how much we can do to go beyond clash detection within the coordination workflow.

- although it is not practical, we believe we demonstrated ideas and potential possibilities.
- We proof that it is possible to modify a model within the coordination workflow with currently available technologies.
- automating some portion of manual process might be still useful.
- Transient or temporary modification in Viewer may be useful
- From technical side, Model Coordination's SQL query on server side is very efficient and powerful to integrate.

Limitations

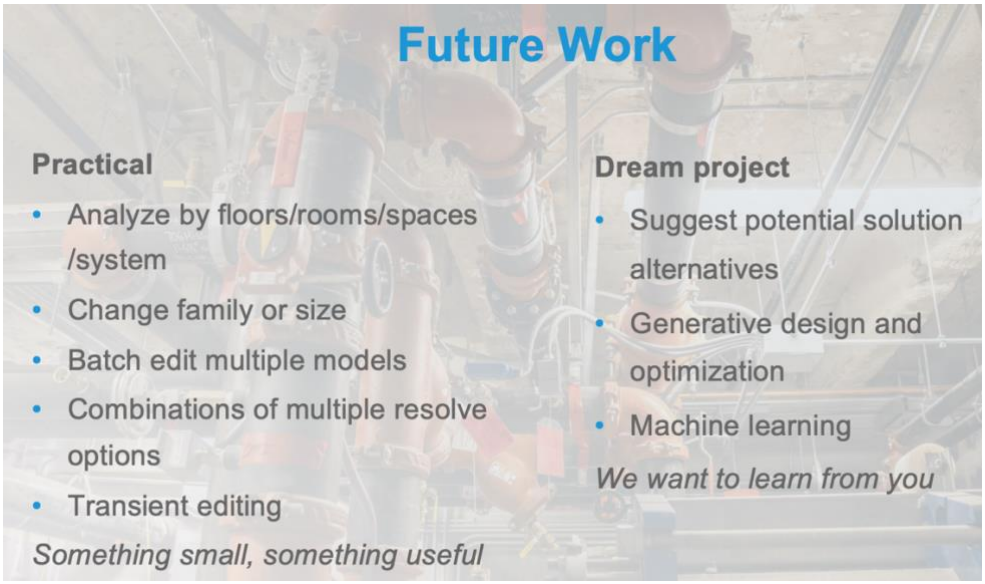
What we showed is only a proof of concept. We have many limitation.

- First of all, Model Coordination only work for RVT, DWG and IFC.
- Design Automation engine only supports Revit, AutoCAD, Inventor, and 3ds Max. (while Forge Viewer supports a wider range of file format.)
- Performance. While we found that the process of detecting clashes and Design automation happens within reasonable time, automatic clash detection is not always triggered timely manner. We need notification mechanism.
- data about how clashes are solved is not available.

Future work

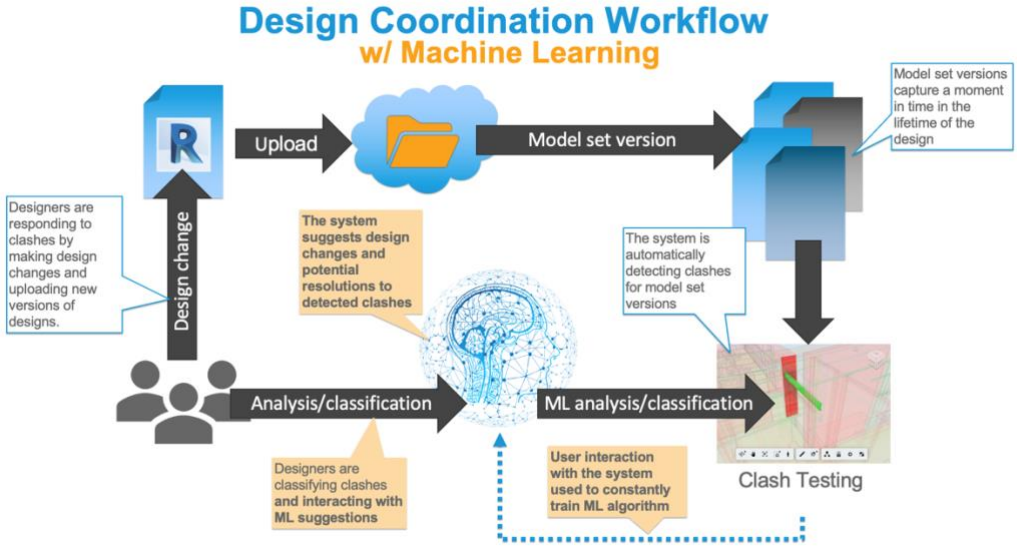
From practical side, we would like to see if adding more flexibility to view and filter clashes, such as by floors/rooms/spaces and MEP systems beyond between documents. Transient editing in viewer is another. Thinking something small, but something useful.Of course, we would like to pursue our dream project as well, such as applying generative design and machine learning.

We want to learn from you!



For example, using Machine Learning, we may be able to draw additional insightful information. Currently, it is the designers who are analyzing and classifying the clashes.

With Machine Learning, user interaction with the system is used to constantly train Machine Learning algorithm. The system suggests design changes and potential resolutions to detected clashes. (Designers are classifying clashes and interacting with Machine learning suggestions.)



Materials

- AU 2019 Class BIM 360 API Update (with Model Coordination API introduction)
- Model Coordination API lighting talk
- Model Coordination API tutorial
- Practice tricks and tips with PowerBI & Model Coordination API
- Basic Clash View by Model Coordination API in Node.js
- Walkthrough sample of Model Coordination API in NetCore
- Postman Collections for Model Coordination API
- Getting Started on Design Automation for Revit on Forge
- Automation Workflows with the Forge Design Automation API for Revit
- Design Automation API Tutorials
- Revit Windows Family Creator
- Extract Quantity for Cost Analysis
- Upgrade Revit file Automatically