Rationale BIM-Programmierung mit Forge und Vergleich mit der Revit API

Jeremy Tammik
The Building Coder, Autodesk
Key learning objectives

- Envision the most efficient, productive, optimal BIM workflow
- Split software portfolio between desktop add-ins and cloud apps
- Optimal use of Revit, Forge and other programming environments
- Understand pros and cons of commercial, open source and DIY
- Foster ubiquitous connectivity between components
- Enable informed software architecture analysis and design
- Overview of existing samples
Agenda

- Desktop versus Cloud-Based BIM Programming
- Autodesk Forge and Design Automation for Revit
- Using Web Technologies for BIM Programming
- Connecting Desktop BIM and Cloud
- Technologies and Implementation Details
- Resources
Shorter Still...

- Generic BIM programming using web technologies versus Revit-specific add-ins
- BIM connections bridging desktop and cloud to make best use of both worlds
Desktop versus Cloud-Based BIM Programming
Takeaway

- Split wisely between Revit API and cloud based programming
- Revit is for BIM design
- Many BIM related workflows are read-only
- Many BIM related programming tasks do not require Revit
- Design Automation API for Revit is coming soon
Optimal BIM Workflow

- What is your task?
- Who needs what?
- Where, when, and how?
- Is your user involved in hard-core BIM design work?
- Or, more commonly: view, annotate, adjust, construct, mobile, field work?
BIM Collaboration Roles

- Participant counts grow by orders of magnitude
- Building design, construction, maintenance, use
  - design - architect, engineer - Revit
  - visualise - client, everybody - Viewer
  - collaborate - management – Glue, Plan
  - make - construction – Field, Layout
  - use - inhabit, maintain, FM
Trends, Tools and Technologies

- Revit and BIM Design
  - Expensive
  - Shrinking numbers of desktop computers
- Forge and BIM Use
  - Partially free and open source
  - Growing number of mobile devices
- Glue code, connected custom components
  - Open source libraries
  - JavaScript, HTML5, SVG, WebGL
What Cloud? How Secure?

- Locally installed, totally private
  - 2D room editor uses private CouchDB web server
- Global, open source, with standard security measures
  - FireRatingCloud uses node.js web server on Heroku and MongoDB database on mongolab
- Forge OAuth2 user access authorisation
  - Roomedit3dv3
Keep It Simple!

- Simplify your data
  - Customise, optimal workflow, minimal complexity
  - Based on 'need to know'
- Use existing components
  - Minimise add-ins, custom components, glue code
  - Open source
  - Forge
Quotes on Three Fundamental Aspects

- **Perfection**
  - Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away – Antoine de Saint-Exupéry

- **Lazy**
  - ... develop the three great virtues of a programmer: laziness, impatience, and hubris – Larry Wall

- **Simple**
  - Simplicity is the ultimate sophistication – Leonardo da Vinci
  - There is no greatness where there is no simplicity – Leo Tolstoy

- **KISS**
Autodesk Forge and Design Automation for Revit
Forge is a platform to empower developers
They can in turn empower their users

- design
- visualise
- collaborate
- make
- use
Forge is a Platform

- Autodesk uses Forge to build its products
- Your application has access to the same APIs
Forge APIs

https://autodesk-forge.github.io
Create or Manipulate BIM in the Web

- Do not misuse Revit as a server
- It is an end user product
- If you have such a requirement, talk with Autodesk
  
  http://thebuildingcoder.typepad.com/blog/about-the-author.html#5.28b
Forge APIs in the Past

- 3 years ago

<table>
<thead>
<tr>
<th>Share</th>
<th>Collaborate</th>
<th>Create</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation API</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Viewer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Forge APIs at Present

- Today, 'Forge 2.0'

<table>
<thead>
<tr>
<th>Share</th>
<th>Collaborate</th>
<th>Create</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Derivative API</td>
<td>BIM 360 HQ API</td>
<td>Design Automation - AutoCAD</td>
</tr>
<tr>
<td>Viewer</td>
<td>Data Management API</td>
<td></td>
</tr>
</tbody>
</table>
### Forge APIs Near Future

- Next year, 'Forge 3.0'

<table>
<thead>
<tr>
<th>Share</th>
<th>Collaborate</th>
<th>Create</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Derivative API</td>
<td>Data Management API</td>
<td>Design Automation - AutoCAD</td>
</tr>
<tr>
<td>Viewer</td>
<td>Webhooks API</td>
<td>Design Automation - Revit</td>
</tr>
<tr>
<td>BIM APIs (HQ, RFIs, Issues, Checklists)</td>
<td></td>
<td>Design Automation – Inventor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Photo-to-3D API</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Forge IDX</td>
</tr>
</tbody>
</table>
Design Automation for Revit

- What is the value?

- **For Partners**
  - Create and deploy services and apps that leverage partner's core competencies to *enhance customer workflows*

- **For Customers**
  - Quickly extend Revit-based workflows to *solve specific business problems*

- **For Autodesk**
  - Facilitate the creation of *new cloud-based solutions* to common customer problems
What Can and Can't It Do?

**What can it do?**
- Full access to Revit DB API outside Revit via cloud services
- Execute .NET add-ins and Dynamo scripts
- Access custom functionality while operating on a Revit model
- Read data from anywhere, save data anywhere

**What can’t it do?**
- No “sticky sessions” – only batch operations
- Access live Revit data in a RESTful way
- No Revit UI API access – there is no UI!
Workflows

Partner/Customer Application

Update

Create

Revit

Extract
Workflow 1 – Create

Use Cases

- Create families (RFAs) from a catalog
- Create RVTs from layout app
- Stair layout generator
- Convert from third party format to RVT/RFA
- Generate full documentation from specification
Workflow 2 – Extract

Use Cases

- Custom extraction (Excel, private database, unsupported formats, etc)
- Automated extraction (e.g. nightly PDF or DWG)
- Model QA/QC & code checks
- Clash detection
- Share model information with consultants without need for Revit
Workflow 3 – Update

Use Cases
- Execute nightly model tasks on C4R models
- Fix common modeling and documentation mistakes
- Batch upgrades & updates
- Automatically replace out-of-date content
- Scheduled Dynamo scripts
- Design option generation
- And many more!

Hypothetical Example
How Does It Work?

Users → Website or Service

Submit a work item. Track the progress.

Download Data or RVT. Upload results.

Revit
Using Web Technologies for BIM Programming
Web-Based BIM Programming Opportunities

- Revit API is BIM and end user product specific
- Many BIM related tasks can be solved more generically
Revit API Characteristics and Limitations

- Revit API is BIM specific
- Very powerful in certain areas
- Leverages Revit product functionality

Therefore:
- Making use of it requires Revit product understanding
- It works only within a running Revit session
- It is completely event driven, hard to 'drive from outside'
- It is limited to the UI oriented Revit end user product
- Programming it requires special training
- Using a Revit add-in requires a Revit installation
Ex: Calculate and Display Signal Attenuation
RvtFader

- C# .NET Revit add-in
- Calculate and display signal attenuation

https://github.com/jeremytammik/RvtFader

- The Analysis Visualisation Framework AVF
- Ray tracing using ReferenceIntersector
ForgeFader

- Forge viewer extension app
  
  https://github.com/jeremytammik/forgefader

- Demo
  
  https://forge-rcdb.autodesk.io/configurator > Meta Properties
  https://forge-rcdb.autodesk.io/configurator?id=59041f250007f5c0eef482f2

- JavaScript
- Three.js
- Custom shader
Connecting Desktop BIM and Cloud
Sample Overview

- Simplified 2D BIM room editor, SVG graphics
- FireRating in the Cloud, the simplest sample
- Roomedit3d, Forge based 2D + 3D simplified round-trip BIM editor
- Forge meta property editor + RvtMetaProp
2D Room Editor

Room Editor

Start: Home
Model: roomedit_rooms.rvt
Level: Level 0
Room: Room 2

1 furniture and equipment item in room Room 2 on level Level 0 in model roomedit_rooms.rvt.

Please pick and drag furniture and equipment around to select and move it, then click the buttons to rotate clockwise, counter-clockwise, refresh, and save.

Properties Rotate Ccw Refresh Save

Subscribe
Subscribe to or unsubscribe from updates.
FireRating in the Cloud Commands

- Create the shared 'Fire Rating' parameter
- Export fire rating values for all doors
- Import the modified values back into BIM
- Store data for multiple projects
  - Cloud database, Revit UniqueId
- Subscribe to changes
FireRating in the Cloud Architecture
Forge Based BIM Editor
Advantages of a Forge Based App

- Realistic model rendering in both 2D and 3D, optionally linked
- Complete access to all BIM data, geometry, structure, properties
- Not bound to any specific model
- Secure authenticated access
- Embedded in a full ecosystem of mature CAD related web services
- Minimal amount of coding based on boilerplate sample code
Forge Meta Property Editor

- [https://forge-rcdb.autodesk.io/configurator](https://forge-rcdb.autodesk.io/configurator)
- Meta Properties
- Use Model: Office

https://forge-rcdb.autodesk.io/configurator?id=5904729b0007f5ead5b1196d

- Add and edit custom properties on elements in Forge viewer
- Download custom property data in CSV or JSON
RvtMetaProp

https://github.com/jeremytammik/rvtmetaprop

- Read custom property data in CSV or JSON
- Associated with individual building elements
- Modify existing element parameter data
- Generate shared parameters for new
- Requires suitable category sets
NoSQL

- “Not only SQL”
  - Next generation database paradigm
- Some characteristics
  - Non-relational, distributed, open-source, scalable, huge data
- Frequent other characteristics
  - Schema-free, easy replication support, simple API, eventually consistent, i.e., BASE, not ACID
    - [http://nosql-database.org](http://nosql-database.org)
    - [https://en.wikipedia.org/wiki/NoSQL](https://en.wikipedia.org/wiki/NoSQL)
    - [http://www.mongodb.com/nosql-explained](http://www.mongodb.com/nosql-explained)
CAP and ACID versus BASE

- **ACID**
  - Atomicity, Consistency, Isolation and Durability guarantee that database transactions are processed reliably

- **CAP Theorem**
  - The ACID paradigm cannot simultaneously guarantee consistency, availability and partition tolerance (distributed system)

- **BASE**
  - Basic Availability, Soft-state, Eventual consistency

Data Source, Repository and Consumer Client

- BIM – Building Information Model
- Cloud-based data repository
- 2D rendering on mobile device
Real-time Edit Triggers Database and BIM Update

- Graphical room editor on mobile device
- Update cloud database
- Reflect real-time changes in BIM
CouchDB Database Implementation

- Everything is a document
- All documents are JSON
- Every document has built-in id and revision
- The database design is also a document
- The design defines views and attachments
BIM Model

- Model – a RVT project file
- Level
- Room
- FamilyInstance – furniture or equipment
- FamilySymbol – geometry
BIM Object Graphics

- Room has boundary loops and can contain holes
- FamilySymbol has a single boundary loop
- FamilyInstance has a 2D placement
  - Translation
  - Rotation
BIM Object Relationships

- CouchDB ids are Revit unique ids
- Family instance → room → level → model
- Family instance → symbol
NoSQL Database Structure

- DbObj base class
- DbModel
- DbLevel
- DbRoom
- DbFurniture
- DbSymbol
Database Object Relationships

- DbFurniture.symbolId → DbSymbol
- DbFurniture.roomId → DbRoom
- DbRoom.levelId → DbLevel
- DbLevel.modelId → DbModel
Database Object Graphics and Placement

- All graphics represented by SVG path element data

```xml
<svg width="4cm" height="4cm" viewBox="0 0 400 400"
  xmlns="http://www.w3.org/2000/svg" version="1.1">
  <rect x="1" y="1" width="398" height="398"
    fill="none" stroke="blue" />
  <path d="M 100 100 L 300 100 L 200 300 z"
    fill="red" stroke="blue" stroke-width="3" />
</svg>
```
JSON Symbol Database Document

- Family symbol
- Define geometry

```json
{
  "_id": "11cc6e52-519e-49b2-9813-c9561b59a1fd-0005f5fc",
  "_rev": "1-d575ca095533db4ccbed9f7ab2607a12",
  "loop": "M-191 922 L190 922 216 862 190 859 -191 859 -216 862 -216 919Z",
  "type": "symbol",
  "description": "FamilySymbol Furniture <390652 Table ronde a chaises>",
  "name": "Table ronde avec chaises - 01"
}
```
Furniture doc represents family instance and defines relationship to room and family symbol. Placement = transform = translation + rotation.
CouchDB Views

- A view defines a map and optional reduce function
- The map produces key-value pairs
- Reduce produces an accumulation

```javascript
exports.models = {
  map: function (doc) {
    if ('model' == doc.type) {
      emit(doc, null);
    }
  },
  reduce: function (key, values, rereduce) {
    return sum(values);
  }
};
```
Room Editor Views

- models
- levels
- rooms
- furniture
- symbols
- map_room_to_furniture
- map_level_to_room
- map_model_to_level

```javascript
rooms = {
  map: function (doc) {
    if ('room' == doc.type) {
      emit(doc, null);
    }
  }
};

map_level_to_room = {
  map: function (doc) {
    if ('room' == doc.type) {
      emit(doc.levelId, doc);
    }
  }
};
```
Minimal Predefined HTML Scaffolding

- `<h1>Room Editor</h1>`
- `<div id="content"></div>`
- `<ul id="navigatorlist"></ul>`
- `<div id="editor"></div>`
- `<p id="current_furniture"></p>`
- `<script type="text/javascript" src="modules.js"></script>`
- `<script type="text/javascript" src="raphael-min-jt.js"></script>`

- Populated entirely using JavaScript adding HTML and SVG nodes using jquery, raphael and db for CouchDB queries
Resources
Full Documentation

Please refer to the class handout documentation at

Old Sample Repositories

- **2D RoomEditorApp and roomeditdb**
  [https://github.com/jeremytammik/RoomEditorApp](https://github.com/jeremytammik/RoomEditorApp)
  [https://github.com/jeremytammik/roomedit](https://github.com/jeremytammik/roomedit)

- **FireRatingCloud and fireratingdb**
  [https://github.com/jeremytammik/FireRatingCloud](https://github.com/jeremytammik/FireRatingCloud)
  [https://github.com/jeremytammik/firerating](https://github.com/jeremytammik/firerating)

- **Roomedit3dApp, roomedit3d and roomedit3dv3**
  [https://github.com/jeremytammik/Roomedit3dApp](https://github.com/jeremytammik/Roomedit3dApp)
  [https://github.com/jeremytammik/roomedit3d](https://github.com/jeremytammik/roomedit3d)
  [https://github.com/Autodesk-Forge/forge-boilers.nodejs/tree/roomedit3d](https://github.com/Autodesk-Forge/forge-boilers.nodejs/tree/roomedit3d)
New Sample Repositories and Demos

- **RvtFader**
  
  [https://github.com/jeremytammik/RvtFader](https://github.com/jeremytammik/RvtFader)

- **ForgeFader**
  
  [https://github.com/jeremytammik/forgefader](https://github.com/jeremytammik/forgefader)
  
  
  [https://forge-rcdb.autodesk.io/configurator?id=59041f250007f5c0eef482f2](https://forge-rcdb.autodesk.io/configurator?id=59041f250007f5c0eef482f2)

- **RvtMetaProp**
  
  [https://github.com/jeremytammik/rvtmetaprop](https://github.com/jeremytammik/rvtmetaprop)