Automating Large Rail Project Workflows

Justin Racelis, justin.racelis@stantec.com
Civil Engineer, Digital Practice

Steven Costa, steven.costa@stantec.com
CAD/BIM Specialist, Digital Practice
Introduction
A Changing Paradigm

• Contemporary rail projects are vastly complex due to an intersection of several key factors. The deliverables and support that agencies and contractors ask for are shifting.

• On two large Design Build rail projects, tailored workflows and custom automation tools were developed in order to meet the project requirements.
Problem

• Contractor and owner requirements scaling with new technology
  – No current project execution standards

• Project requirements
  – ProjectWise – main collaboration environment
    • BIM 360 integration did not exist yet for Civil 3d, models lived in PW
    • CDs produced from Revit
  – AM and Visualization deliverables
  – Amorphous model deliverable requirements
Solution

• Workflow Design
• Parametric Wall Design with Custom Subassemblies
• Parametric Bridge Modeling with Inventor
• Data Replication (ProjectWise <> BIM 360 <> Azure)
• Dynamo Civil 3D for Utility Model Optimization
• Geolocated dwg sheet automated export from Revit
• Asset Management
Projects 1 & 2

- Design-Build
- ProjectWise for Documentation
- BIM 360 for Federation
- Asset Management
Project 1 Data Workflow
Project 2 Data Workflow
Project 1 & 2 Model Workflow
Project Execution
Subassembly Composer
Sample from Project 1

- **Understanding the Engineering Problem**
  - Foundation design is rule based with several design cases
  - Design cases must properly be represented in cross sections
  - Foundations must be used in clash detection

- **Defining SAC Parameters**
  - Define variables
  - Set SAC geometry
  - Build design logic
HOW MANY RETAINING WALLS DO WE NEED TO DESIGN MANUALLY?

WHAT!? 9000?!

OVER 9000!!
Parametric Wall Design

Design case iteration for retaining walls and subsurface conditions
Parametric Design with Custom Subassemblies
Inventor Implementation
• Inventor solids brought into AutoCAD for reference.
• Abutments are modeled in AutoCAD.
  – Bridge references and existing information are easily accessible in AutoCAD.
  – Abutment shapes are irregular. Developing parametric criteria for all abutments in Inventor is not cost effective in this case.
ProjectWise/BIM 360/Azure Replication
• Database Relationship

• Versioning Logic
Dynamo for C3D
- PLS-CADD has limited interoperability with non-PLS authoring platforms
- DXF export is the only option
Revit Sheet Geolocation and Conversion
REVIT COORDINATE SYSTEMS

Revit has two coordinate systems, (1) a project coordinate system, which utilizes an internal origin point that provides the basis for positioning all elements in the model and never moves and (2), a survey coordinate system that provides a real-world context for the building model and is intended to describe locations on the surface of the earth.
CIVIL 3D COORDINATE SYSTEMS

Civil 3D utilizes an internal origin of 0,0 and one can establish either a user-based coordinate system or in the case of the two projects we are discussing, a known coordinate system such as the State Plane Coordinate System (SPCS).
SHARED REFERENCE POINT UTILITY

To simplify the exchange of data between Revit and Civil 3D, Autodesk developed the Shared Reference Point Utility which uses a handful of clicks and an XML file to establish a common point in both platforms.
THE CHALLENGE

How do you automate the process of getting Revit sheet views exported so that any plan views that are exported with their respective sheets are accurately geolocated with respect to all the other drawings.
Many subject matter experts argued that Revit already exports geolocated plan views, so if you need to perform project coordination, the plan views should be used, not the sheets.

1. We were contractually obligated to deliver fully coordinated DWGs
2. Many trades/disciplines are not using Revit
3. The process of exporting sheets in its current state is incomplete

EXPORT SHEETS TO DWG USING SHARED COORDINATES
PROBLEM RE-EXAMINED
ISSUE: NAMING CONVENTION

REVIT
A-445 - OVERPASS BRIDGE - PLANS & RCP
- Floor Plan OVP BRIDGE - PLAN
- Floor Plan OVP BRIDGE - ROOF PLAN
- Reflected Ceiling Plan OVP BRIDGE - RCP
- Reflected Ceiling Plan OVP BRIDGE - UNDERSIDE

SHORT
- A-445.dwg
- A-445-View-1.dwg
- A-445-View-2.dwg
- A-445-View-3.dwg
- A-445-View-4.dwg

LONG
- CPSL-10-01_NDC_CENTRAL_2019 - Reflected Ceiling Plan - OVP BRIDGE - UNDERSIDE.dwg
- CPSL-10-01_NDC_CENTRAL_2019 - Reflected Ceiling Plan - OVP BRIDGE - RCP.dwg
- CPSL-10-01_NDC_CENTRAL_2019 - Floor Plan - OVP BRIDGE - ROOF PLAN.dwg
- CPSL-10-01_NDC_CENTRAL_2019 - Floor Plan - OVP BRIDGE - PLAN.dwg

AGENCY STANDARD
- rpd-a-445-wwy_ptd.dwg
- s-way_xvp bridge - underside_pln.dwg
- s-way_xvp bridge - rcp_plan_pln.dwg
- s-way_xvp bridge - rcp_pln.dwg
- s-way_xvp bridge - plant_pln.dwg
ISSUE: SHEETS

1. Referenced used generic incremental naming convention i.e. X1, X2, X3.
2. The default name “Layout 1” was used rather than the Revit drawing number.
3. The sheet size is determined by AutoCAD’s default plotter settings
ISSUE: REFERENCES

1. Reference views are all attached to their respective sheets about the internal origin.
2. Drawings with multiple references have a seemingly arbitrary offset away from each other.
3. When references move to their geospatially correct location, viewports would need to be updated.
4. Sheets with multiple views of the same plan at different elevations, overlap with each other.
2ND DAY ON THE JOB

THIS IS FINE.
DYNAMO

1. Archilab (2019.2.21)
2. Genius Loci (2019.3.11)
3. SpringNodes (132.2.8)
PYTHON 3.7
(defun RegExpSet (pattern ignoreCase global / regex)
  (setq regex
    (cond
      (null (vl-load-vm))
        (vl-load-vm)
        (vl-load-vm (vl-eval-vm 'vscript.regex))
      ))
  (vl-put regex 'Pattern pattern)
  (vl-put regex 'IgnoreCase asTrue)
  (vl-put regex 'IgnoreCase asFalse)
  (vl-put regex 'Global asTrue)
  (vl-put regex 'Global asFalse)
  regex
)
(defun RegExpTest (string pattern ignoreCase)
  (= (vl-catch (RegExpSet pattern ignoreCase nil) 'Test string) -1))
(defun RegExpExecute (string pattern ignoreCase global / sublist list)
  (vl-catch-all-match (vl-catch-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatches)
    (vl-catch-all-match 'SubMatc
.NET CORE & API
TECHNICAL DETAIL
EXPORT SCRIPTS

Create Collections of views from sheet set sorting by no viewports, a single viewport, or multiple viewports

- No viewports
  - Sheet View Number
  - Sheet View Type Name
  - Boolean value (isPlan?)

- One viewport
  - Sheet View Number
  - Sheet View Type Name
  - Number of Viewports
  - Boolean value (isPlan?)

- Multiple viewports
  - Sheet View Number
  - Sheet View Type Name
  - Number of Viewports
  - Boolean value (isPlan?)

Extract Project and Survey Base point information and assign a LISP variable for the delta of each constituent

- PROJECT BASE POINT
  - X Coordinate
  - Y Coordinate
  - Angle To True North

- SURVEY BASE POINT
  - X Coordinate
  - Y Coordinate

Combine view set data with "baked" LISP files to create final output. Export additional files for use in Scriptpro.
RENAMING WITH PYTHON

1. Relatively simple Python script
2. Used Regex to match patterns for renaming files
3. Deleted duplicates during renaming
4. Hardcoded directory structure based on ProjectWise
5. Found and moved images and DWG files
6. Simple log file to keep track of duplicates, and verify which files were moved and where
7. Later version ran from executable placed in same directory using PyInstaller
TRANSFORM PLANS

This LISP routine translates and rotates all drawing objects using data from Project/Survey Basepoints.
ADJUST VIEWPORTS
Using ScriptPro, you can apply a set of commands to multiple drawings by simply specifying a script file and the list of drawings that you would like to apply the script to.
RESULTS

- TOTAL: 1152.33
  - MANUAL: 63.98
  - AUTOMATED: 1188.31
- UPDATE SHEETS (108 DRAWINGS): 720.00
  - MANUAL: 35.60
  - AUTOMATED: 755.60
- UPDATE PLANS (44 DRAWINGS): 220.00
  - MANUAL: 9.47
  - AUTOMATED: 229.47
- RENAME FILES (387 DRAWINGS): 193.50
  - MANUAL: 0.08
  - AUTOMATED: 193.58
- EXPORT FROM REVT (387 DRAWINGS): 18.83
  - MANUAL: 18.83

MINUTES: 0 200 400 600 800 1000 1200 1400

MANUAL AUTOMATED
.NET CORE & API
• Asset Deliverables
• ArcGIS Visualization Deliverables
Conversion had two goals

- Satisfy 3D model Visualization Desire of Client
- Create a Compatible link to Asset Information
## Layer to Asset Mapping

<table>
<thead>
<tr>
<th>Asset Abbreviation by Stantec</th>
<th>Category</th>
<th>Sub-Category</th>
<th>Asset</th>
<th>Definition</th>
<th>GIS Geometry</th>
<th>Discipline</th>
<th>Path to xref File</th>
<th>Layers that make up Asset</th>
</tr>
</thead>
<tbody>
<tr>
<td>INF_STN_FAC_PAS_ESCALATORS</td>
<td>Facilities and Stations</td>
<td>Passenger Stations</td>
<td>Escalators servicing the passenger stations</td>
<td>AM Information only</td>
<td>HVAC and MEP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INF_STN_FAC_PAS_PUBLICADDRESSSYSTEMS</td>
<td>Facilities and Stations</td>
<td>Passenger Stations</td>
<td>Public Address (PA) Systems</td>
<td>PA systems meant to be audible within the passenger stations</td>
<td>AM Information only</td>
<td>Communications</td>
<td>com-stn.cpl.pln.dwg</td>
<td>SPEAKER TEXT</td>
</tr>
<tr>
<td>INF_STN_FAC_PAS_CCTVSYSTEMS</td>
<td>Facilities and Stations</td>
<td>Passenger Stations</td>
<td>CCTV Systems</td>
<td>Video surveillance equipment used to monitor the passenger stations</td>
<td>Centerpoint</td>
<td>Communications</td>
<td>sec-stn.cpl.pln.dwg</td>
<td>SECURITY-TEXT</td>
</tr>
</tbody>
</table>
Metadata Extraction
Spatial Data Transformation
AGOL Web Application
Azure hosted Infor Assets
Using application components developed in partnership with Autodesk, our team was able to pull in a station model from InfraWorks into ArcGIS Pro.
1. Export 3D Model from InfraWorks as COLLADA

- InfraWorks exports models by type as a single model, i.e. 8 Revit models and 4 Sketchup models in InfraWorks as “Building” feature class.
- Examining the XML structure reveals that the geometry is grouped by individual feature and can be parsed.
2. Parse COLLADA file with tool and import in ArcGIS Pro
   - Coordinate System is preserved from InfraWorks
To Summarize

- Automated data transfer and versioning between 3 collaboration platforms
- Enabled models from several different design platforms to be federated in BIM 360
- Centralized data could then be transformed into client defined assets
If you live through defeat, you are not defeated. If you are beaten but acquire wisdom, you have won. Lose yourself to improve yourself. Only when we shed all self-definition do we find who we really are.

RZA