

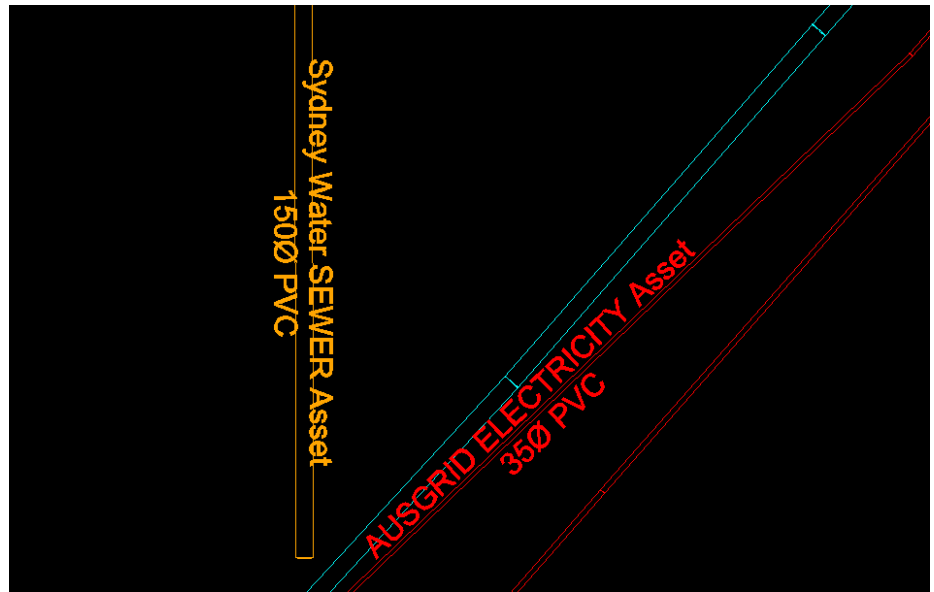
Durkin Construction Pty Ltd

SUI Locating and Survey
Requirements for Ryde
Transmission Feeder Clash
Detection

12d DIMENSIONS:

- Drafting

Ryde Transmission Feeder Clash Detection



Resulting labels after using String Labelling macro

Project Summary

A full site survey carried out to enable the design of new high-voltage Ausgrid electrical assets. An accurate 3D model of existing utilities, including drainage, as well as a basic ground model prepared in order to generate clash detection reports, long sections, and cross-sections.

SUI Engineers investigated locations and depths of underground utilities and marked information on the ground. Surveyors picked up this information during the survey process, and the post-processing team generated the required deliverables. This included an accurate 3D model of existing underground utilities showing pipe/culvert sizes.

For more information

To find out more about how you can create better designs faster with the 12d Model solution for civil engineering design, visit www.12d.com.



Australasia: Sydney
P: sales@12d.com
M: +61 2 9970 7117

The Challenge

When undertaking these construction and development projects, one of the most important preliminaries that had to be considered was utility investigation and survey, which enabled the production of a utility model. As per AS5488-2013, the information which needed to be provided for each utility were:

- Quality Level
- Utility Type
- Utility Owner
- Size
- Material
- Configuration
- Date of Installation (if known)

Although most design software packages have remarkable ways of visualising the structure of utilities, integrating the abovementioned set of information with each component proved to be a difficult task. This is the reason why the majority of the descriptive attributes are usually recorded in separate datasets. The efficiency of this kind of system is, however, very minimal. Analysing the information as a whole is very difficult since information will be coming from different sources. Because of this, inconsistencies between datasets are very likely to

be found. Also, if editing is necessary, the information needs to be updated on each of the sources. These factors render the system very costly in terms of both time and money, plus there is a lot of room for human error. Therefore, integrating the utility information into one system was deemed essential for the Durkin team.

For this specific project, although they followed the same procedures in projects involving sub-surface utility investigation and survey, extensive underground utility investigations were performed by SUI Engineers along the roads and footpaths within the scope of the project area. They opened every utility pit and traced the conduits running along them using GPR and EM methods. Then, they marked the location of each utility on the ground together with corresponding information such as depth, utility type, asset owner, configuration, pipe diameter/culvert dimensions, material, etc. To acquire the necessary data for creating a 3D model, Durkin surveyors would then pick up the location of these utilities through the ground markings, and input the corresponding information as string attributes. Data from total stations would then be processed in Magnet Tools. In preceding projects, survey data were imported in Genio format, however, attribute data is lost during 12d Model

import. With the advice of 12d Model Sales/Training experts (Extra Dimension Solutions (ExDS)), they started to export datasets in SDR format to retain string attributes. Thus, the problem of having separate datasets for string properties and descriptive attributes was solved. However, there were still a lot of flaws in this kind of system.

Data Validation

String vertex attributes were recorded as per ground markings. However, there were quite a few inconsistencies with the attributes.

- For all the vertices running along a string, the attributes Type, Asset Owner, Material, Configuration and Pipe Diameter should be constant as they represent a single utility.
- Quality Level and Depth should make sense. QL-A is when the attributes and location of the utility are directly measured/observed. QL-B is when it is located through electromagnetic pipe and cable locators, sondes or flexi-trace, ground penetrating radar or acoustic pulse equipment. QL-C is when an interpretation of the approximate location is made using a combination of existing records or visible evidence during site survey. Finally, QL-D is when the location and attributes of a utility is obtained based from existing records, cursory site inspection or anecdotal evidence. Therefore, a QL-D point cannot have a value for the Depth attribute. Likewise, QL-A and QL-B points should have corresponding Depth values. Thus, the second problem emerges. If each vertex of every string had to be checked for blunders, it would again consume an unreasonable amount of time.

String Labelling

The other issue here was finding a way to fully customise the resulting labels from the 12d Model Label Mapfile. It was essential that the labels be of the same colour as that of the strings they represented. It was also important that the labels be placed on and of similar alignment/rotation angle with the longest segment, especially on a project wherein the utilities are very congested on footpaths. Changing the colours and alignments of these labels would take a great deal of time, and would be very problematic and confusing as the strings are running along the same direction too close to each other. Looking ahead, the team at Durkin realised they'd have a lot more projects that would require the same post-processing procedures, and doing all of this manually would definitely be unfavourable. Finding the right solutions was therefore crucial.

The Solution

Neil Perol of Durkin Construction said: *"Aside from having strong design and visualisation capabilities, 12d Model is also packed with a powerful programming language which allows users to build their own programs in the form of a Macro Language (4DML). With a vast number of intrinsic functions and an extremely helpful manual that sets out syntax and restrictions, users with basic background of C++ or any other programming language can easily create their own applications as the operations require. I started using 12d*

Model mid-January 2018, and in less than a month, I was already creating my own 12d Model applications through Macro Programming."

JUNK Model Pre-processing Macro

All the abovementioned issues with automating repetitive tasks were solved by writing and utilising 12d Model Macros. Under the supervision of our Geospatial Manager, Mr Perol created three different macros for each of those drawbacks. The first, Junk Model Pre-processing macro, is intended to iterate over all strings in the JUNK model, change the string names, line styles and colours as per RMS Customisation, and assign them to the corresponding models (per string name). When the macro is compiled and run, the macro panel opens. The user selects the JUNK model in the model input widget. The process would only run if the user chooses the model named "JUNK", no matter what the prefix is (but it should not have a postfix). This is to prevent unwanted processing of other existing models. Before clicking on the Process button, the user should also input the desired prefix for resulting models. This ensures organisation between existing models and child models.

When the process button is clicked, the iteration begins. Current string names are concatenated with the first integer/s of the string 'no attribute', following the necessary conditions. Therefore, a string named 'U', for example, is concatenated with the first character of the string no attribute. On the other hand, a string named 'PT' would be combined with the first two characters of the string no attribute. Once the proper string name is established, everything else can be matched. The line style and colour can easily be changed using a function that opens the RMS Mapfile and links the string name with the key. The last thing that is done inside the iteration is transferring the string into the correct model, which is the user-defined prefix plus the string name. With this macro, hours or (for big projects) days of manual string segregation are reduced to just a few clicks, at the same time removing the odds for personal errors.

Attribute Data Validation Macro

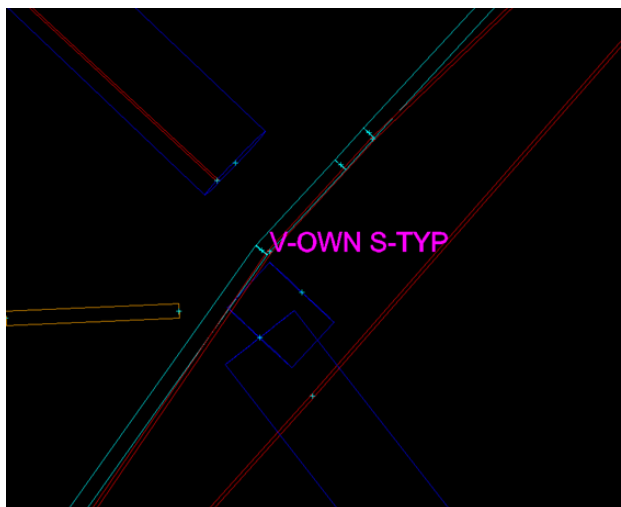
Checking vertex Depth against Quality Level

The second macro that was prepared was the Attribute Data Validation macro. The script for this macro was very lengthy compared to the others as there were a lot of conditional statements, and three different features. The first part is for validating vertex Depth and Quality Level attributes. After running the macro, the user first selects the View where the models to be validated are shown. When the user clicks on the Check(V) button, a function with multiple nested loops is executed. The outer-most loop iterates over all the models included in the selected view. Inside this loop is another loop that iterates over all the strings of each model. Finally, there is the last loop, inside the latter, which iterates over all the vertices of each string of every model. The Depth and Quality Level attributes are checked for each vertex. If an anomaly is detected (i.e. QL-D with depth or QL-A without depth), the function will add, to the vertex, an attribute called Error which indicates the inconsistency. The function will also create a text string which shows

the quality level and depth for the erroneous vertex. These text strings are assigned to the Vertex Errors model. After every single vertex is accounted for, the iteration's end and the Vertex Errors model are added to the selected view. This allows for easily pinpointing erroneous vertices, instead of checking them all manually one by one. There is also a Clear(V) button that enables the user to delete all Error attributes along with the Vertex Errors model.

Transferring Vertex Attributes to String Attributes

The next feature of this macro is for transferring vertex attributes to string attributes, which is very important for automated labelling. The Transfer button runs a function that checks all the vertices and whichever has a value for that specific attribute will be taken as a string attribute. For example, if the Type attribute is found on the second vertex while the Asset Owner attribute is found on the third vertex, the function will take both of these values as string attributes. Note that the function checks vertices chronologically. The moment it finds an acceptable value, it will ignore the rest of the remaining vertices. This is done for all the strings of each model on the selected view. There is a Clear Attributes button which undoes this operation.



Validating String Attributes and Checking Vertex Attributes' Consistency

The last feature of this macro is for checking the string attributes. For this part, a lot of conditions are expected to be met; otherwise, a list of errors shall be created. Again, a triple-nested loop is run when the user clicks on the Check(S) button to check all vertices of each string. Attributes such as Type, Asset Owner and Material are expected to be constant throughout the string. Therefore, if two or more vertices have different values for these attributes, an error shall be added to the list of errors. After checking the consistency of vertex attributes, string attributes are then validated. A pool of values for each attribute was first created. If the value for a specific attribute is not found in the list of allowable values, an error shall be added to the list of errors. For example, a string named 'EU' should have 'Electricity' as Type and cannot have 'Jemena' as the Asset Owner since 'Jemena' is not in the list of allowable Asset Owner values for a string with 'EU' as key. At the end of each string iteration, the list of errors is added to the attributes as Errors. A text string showing the list of errors is also created in the first vertex of the string under the String Errors model which is added to the selected view after all strings are accounted for. This allows us to easily pinpoint the errors. There is also a Clear(S) button that

enables the user to delete all Errors attributes along with the String Errors model.

String Labelling Macro

The last macro is for labelling each utility string based on attributes. After execution, the macro panel opens. The user then chooses the view containing the models to be labelled, as well as the prefix for resulting label models. When the Label button is clicked, the macro iterates over each string of every model for the selected view. By looking into string attributes, it creates a text variable which would eventually be the value for the label. When the text value is set, the macro then creates a text string containing the label. The colour is set based on the utility being labelled (*i.e.* EU is coloured red). The next task is to find the longest segment of the string and find its midpoint where the label would eventually be positioned. The angle of the text will be the same as that of the segment, unless it is between 90 and 270 degrees where it would be of reverse direction to avoid upside-down texts (see headline image on p.1 for a sample result). The only problem left is when the labels get too congested; we have to manually organise to make the plan more visually appealing. The good thing is that it would be a lot easier to

re-organise congested labels as they follow the same colour and the same angle as the strings they represent.

All of the macros that were created have message box widgets at the bottom of their respective panels. These widgets show processing status, as well as error-handling prompts. The string labelling and attribute data validation macros both have Info buttons which opens text boxes that show information regarding the macro.

The Result

Mr Perol feels that: *“Macro Programming has proven itself to be a very powerful tool in automating repetitive and conditional tasks. It gives 12d Model immense versatility and efficiency in doing sophisticated operations. I am sure that I would be making more macros that would make our processes faster, and our datasets more reliable. With this, we now have a highly efficient integrated system for utility modelling. Descriptive attributes are embedded on each utility string and post-processing is almost fully-automated. This makes our procedures a lot simpler, our datasets easier to manipulate, and at the same time greatly reducing time and monetary costs.”*



Roads and Highways

12d Model's design option is the smarter solution for the design, modification and maintenance of Road and Highway projects.

Enjoy advanced 3D tools to design local and major roads, intersections, roundabouts, highways, interchanges and much more.



Ports and Dredging

12d Model is the solution for port infrastructure and dredging, easily managing the very large datasets and complex volume calculations often required by these projects.

A complete range of flexible and customisable volume calculation tools allow teams to extract and present the information quickly and easily.



Land Development

12d Model is the most versatile solution for the creation of sustainable land development projects, including residential, commercial and industrial developments, recreational areas, landfills, and agriculture projects.

Easily manage all aspects of your land development project from earthwork quantities, road design utilities and drainage design.



Airport Infrastructure

12d Model provides a solution for the design, construction and analysis of new airports, as well as the upgrade and maintenance of existing runways and airport infrastructure.

Easily manage large airport infrastructure projects and share data across multi-disciplinary teams.



Rail

12d Track has been specifically designed for the survey, design and construction of light, heavy and high speed rail projects.

Extensive railway tools in 12d Track allow the rail designer to quickly and easily design their projects. These options are built on the existing 3D modelling and design tools available in 12d Model.



Mining Infrastructure

12d Model's powerful set of exploration, site investigation, survey and analysis tools are crucial for the initial design, construction and ongoing operation of mining projects.

Comprehensive tools for the survey, design and construction of access roads, railways, earthworks and services allow for the coordinated design and management of mining infrastructure from within 12d Model.



Drainage, Sewer and Utilities

12d Model provides comprehensive tools for the design, analysis and optimisation of stormwater and sewer projects using rational, dynamic (hydrograph) and 2d drainage methods.

Powerful clash detection management allows for efficient 3D modelling of service networks such as gas, electricity, telecommunications and water prior to construction.



Surveying

12d Model is a complete surveying package providing the tools to manage all facets of surveyed data including LIDAR, topographical, as-built, conformance, traversing, geodetics, data mapping, labelling and much more.

The 12d Field option runs on a ruggedized tablet and gives the user access to full 12d Model functionality, allowing you to take the entire project into the field with the most comprehensive pick-up and set-out tools.



Oil and Gas

12d Model assists with the design, construction and mapping of oil and gas pipelines, original site exploration and the wide range of infrastructure required for oil and gas projects.

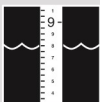
Accurate 3D modelling and the ability to share data between users allow teams to quickly and easily coordinate designs.



Construction

12d Model is the ultimate software for construction with powerful set-out options, direct interfaces to machine control and detailed conformance reporting and auditing.

Manage 3D data and control volumes, quantities and progress claims with 12d Model. Set-out your project and undertake conformance and as-built surveys live on-site using 12d Field.



Rivers, Dams and Hydrology

12d Model handles very large datasets and interfaces with a wide range of analysis packages, making it perfect for flood studies and the management of rivers and dams.

12d has partnered with industry leading analysis software, allowing users to apply 2D drainage analysis from within 12d Model.



Environmental

12d Model's ability to handle very large datasets combined with flexible and comprehensive 3D analysis and modeling tools make it perfect for a wide variety of environmental projects.

Existing workflows can adopt 12d Model easily as it allows users to directly interface with GIS systems and most software packages and file formats.

Why Choose 12d?

- **Powerful data processing & intelligent functionality.**
- **Modular, easy to update & completely customisable.**
- **Seamless integration with major industry software and hardware.**
- **Used in over 55 countries worldwide.**
- **Friendly support & training from industry experts.**

Australasia: Sydney

P: sales@12d.com

M: +61 2 9970 7117

12d Solutions Pty Ltd
PO Box 351 Narrabeen
NSW 2101 Australia
© 2020 12d Solutions Pty Ltd

