Bridging the Gap – A Design Process Case Study for an "Intelligent" Footbridge

Dirk Anderson
Architect and Post Graduate Student
Centre for Built Infrastructure Research
University of Technology, Sydney
Australia

DirkUve.Anderson@student.uts.edu.au

Keith Crews
Professor of Structural Engineering
Centre for Built Infrastructure Research
University of Technology, Sydney
Australia

keith.crews@uts.edu.au

Keywords: Interoperability, intelligent structures, sensors, computer modelling, digital architecture.

SYNOPSIS

This paper presents details of a research project that has been undertaken at the University of Technology, Sydney by a multidisciplinary team to develop a framework model that synthesises architectural, engineering and construction processes for intelligent buildings with sustainable performance characteristics. A significance aspect of this project is the bridging of the traditional gap that exists between Architecture, Engineering and Construction professions in Australia.

The focus of the first stage of this research has been to undertake design of a complex form footbridge using various software packages to undertake concept design using digital architecture, visualisation and modelling of structural performance of the finished bridge. In order to achieve this synthesis in design, a number of interoperability issues have been identified. The paper presents details of the project and discuss how various design and knowledge creation issues have been addressed, as well as how it is proposed to develop the framework model in the future, for more complex problems.

OVERVIEW

Currently the industry is struggling to achieve a standard file format that is written specifically to perform inter-operational actions between proprietary software packages with IFC and CIS/2 vying for that position. Despite considerable effort in attempting to develop software protocols, it is interesting to note that (in many parts of the world, the most widely used technique of the exchange of drawing information remains hard-copy printed format, a centuries old analogue method operating in the digital age.

The main "electronic" approach with drawing representation is the 'round trip' method of exporting DXF, Autodesk's drawing exchange format, to and from proprietary software packages. Quantity, schedule and costing information is still manually translated from drawing to word or spreadsheet applications, remaining as cumbersome and tedious as the analogue approach of the print-out.

The concept of "inter-operability" is, at least in theory, very attractive. Currently IFC is presented as one of the leading proponents of inter-operability technology - serving the needs of most architectural, general contracting, and some facilities management tasks, with varying degrees of success. However, it still lacks a complete representation of civil and structural engineering, rendering its use incomplete for one of the major design interactions that occurs for most major projects.

RESEARCH PROJECT

The research described in the paper, sets out to explore the efficiency of data flow and exchange between design and analysis disciplines by targeting a specific project and applying software integration methods. The current industry direction of interoperability is through a complete snapshot of native information from independent software packages via IFC or CIS/2 data models. In the research undertaken by the authors, the focus is on implementing database centric design using XML (Extensible Mark-up Language) and MDB (Microsoft Access Database) to store information in a number of accessible databases.

The first study model comprised of a relatively simple bridge design to test the exchange of information between a Microstation design program Generative Components (Structural) and a widely used commercially available structural engineering package.

From the case study a process has been developed that enables highly efficient interaction to occur between the Architectural and Structural Engineering software packages. The steps involved in this process are described as follows:

1. XML data is parsed to a Microstation and Space Gass MS Access database (MDB) that filters out specific structural members, elements and their respective qualities. The XML database contains all universal information such as section names, structural material, etc. that can be filtered out for software specific integration.
2. Software databases are imported into respective programs. These databases build a library of structural components and elements that the software uses either parametrically or analytically.
3. Microstation Generative Components develops a parametric model using GC features. The features are appended with additional qualities that can be used in structural analysis. This model is output directly to a shared MS Access database that both Space Gass and Microstation will utilize.
4. The Access database is imported into Space Gass, this database loads explicit information such as node locations, node fixings, member types, member material and load cases.
5. Space Gass loaded model data appended to MS Access database.
6. DataImport function imports and builds loaded model in GC
7. Loop can continue indefinitely.
8. Resulting GC model documentation can be automated using a fabrication planning feature.

CONCLUSIONS

Through investigating the key methods of translating design information, this project has undertaken research into and demonstrated the potential advantages of implementing database-centric models over document-centric and data-centric models currently used in the construction industry today.

Advantages of the data base-centric approach include:

- Live Data - database-centric models hold live information, enabling a build and assessment of models in multiple disciplines simultaneously from its content.
- Flexible Modelling - Generative Components implements flexible modelling techniques, easily generating many iterations of the design model with data that is ready assessed within the structural analysis software.
- Open Source - Data-centric models such as DXF and IFC formats require the proprietary software developer to implement these formats in their packages. However, where a software package can export to a database, the structure of its data organisation is clear and openly accessible. This paves the way for interoperability between packages as users can ultimately filter and send explicit information between software through database methods.
- Reduction of Errors - Database-centric models can minimise the risk of error by reducing the amount of manual handling during the translation process. In some cases data-centric and document-centric model techniques may be more time efficient with simpler models.

This paper presents a viable alternative to current “inter-operability approaches”, suggesting that database-centric models might well be the future for software users.

Bridging the Gap – A Design Process Case Study for an "Intelligent" Footbridge

Dirk Anderson
Architect and Post Graduate Student
Centre for Built Infrastructure Research
University of Technology, Sydney
Australia

DirkUve.Anderson@student.uts.edu.au

Keith Crews
Professor of Structural Engineering
Centre for Built Infrastructure Research
University of Technology, Sydney
Australia

keith.crews@uts.edu.au

Summary

This paper presents details of a research project that has been undertaken at the University of Technology, Sydney by a multidisciplinary team to develop a framework model that synthesises architectural, engineering and construction processes for intelligent buildings with sustainable performance characteristics.

The focus of the first stage of this research has been the design of a complex form footbridge using various software packages to undertake concept design using digital architecture, visualisation and modelling of structural performance of the finished bridge. The paper includes discussion on various design and knowledge creation issues, as well as how it is proposed to develop the framework model in the future, for more complex problems.

Keywords: Interoperability, intelligent structures, sensors, computer modelling, digital architecture.

1. Introduction

Achieving interoperability relies on the ability to identify, gather, structure, generalize, and formalize information that is exchanged between a variety of building design and engineering applications [1].

As software and hardware developed for the construction industry evolves, shifts in the current trend of building design and construction move towards an ever increasing complexity not only in the design, but also in the building environment itself. The management of information has again moved to the forefront of critical building strategies, requiring a higher level of co-ordination of ideas and data in the design / build process.

By looking at the key methods of translating design information, this paper aims to research the potential advantages of implementing database-centric models over document-centric and data-centric models currently used in the construction industry today.

2. Interoperability in the Construction Industry

2.1 Current Trends

The difference between a traditional geometric data model and a building data model that inter-operability aims to achieve is illustrated in Figure 1. In order to define an effective database-centric model, an understanding of the current trends of interoperability in the industry needs to be established.

Currently the industry is struggling to achieve a standard file format that is written specifically to perform inter-operational actions between proprietary software packages with IFC and CIS/2 vying for that position. Despite considerable effort in attempting to develop software protocols, it is interesting to note that (in many parts of the world) the most widely used technique of the exchange of drawing information remains hard-copy printed format, a centuries old analogue method operating in the digital age.

The next popular approach with drawing representation is the 'round trip' method of exporting DXF, Autodesk's drawing exchange format, to and from proprietary software packages. Quantity, schedule and costing information is still manually translated from drawing to word or spreadsheet applications, remaining as cumbersome and tedious as the analogue approach of the print-out.

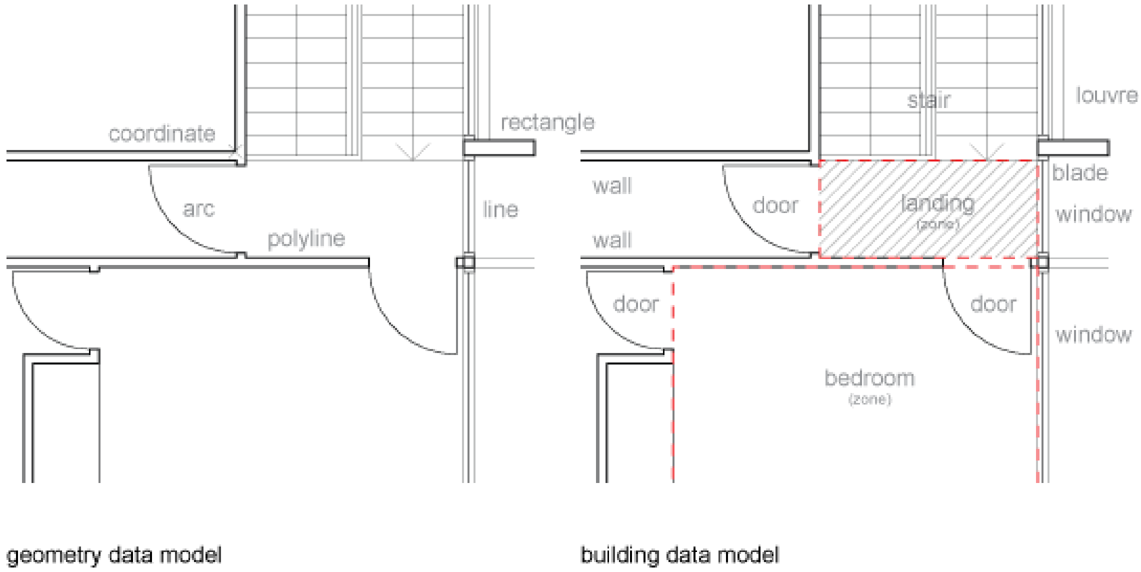

Fig. 1 *Geometry Data Model which purely describes geometry in space and a Building Data Model which can include such information as building zones, quantity calculations, material finishes, etc*

2.2 DXF (Drawing Exchange Format)

Although introduced in 1982 and still relying on similar principles of conversion, DXF is still the most common file format used to translate information between CAD applications. Developed by Autodesk, The DXF format was developed to represent of all the information contained within the AutoCAD DWG drawing file. Although it specifies that 'Virtually all user-specified information in a drawing file can be represented in DXF format' this is not the case. DXF is primarily concerned with representing geometry data and lacks the depth to support interoperability of other virtual building elements such as materials, parametric components and building zones to name a few.

How much information does DXF translate? The DXF translation process differs with every program export to every program import. On a simple export from a design program (Rhino) to a documentation program (Microstation), out of 23 elements exported, 6 had lost most of their information and 7 had not even been included in the file, over 50% of the file had been incomplete. Using more sophisticated techniques such as BIM in Microstation, almost all of the parametric and database information may be lost whist translating through DXF.

2.2.1 Disadvantages of DXF

Only DWG compatible information is maintained with DXF export and program specific data does not carry through with the file. As a general user, it is quite difficult to include specific information carried within the native file, a user may be able to customise basic settings such as curve, surface and mesh export but elements such as parametric objects and data (variables) used in contemporary practice are not included.

2.3 IFC Model

The International Alliance for Interoperability (IAI) have worked with a number of partners to define a single building information framework combining building elements, their properties and interrelationships – the IFC (IFC) model. Currently IFC serves the needs of most architectural, general contracting, and some facilities management tasks, but still lacks a complete representation of civil and structural engineering, rendering its use incomplete.

The IFC file is data model that is structured around virtual building elements and their relationship with each other. Rather than just representing geometry, it comprises of virtual building components such as walls, building zones, window elements, door types, etc. to construct a complete virtual building.

3. Design Model Types

3.1 Document-Centric Models

The Document Centric Model is a traditional design interchange method which focuses on the document as the unifying object in the workflow process. Documents between sketch design, analysis and documentation are independently developed with only representational information transferred between consultants. Digital techniques (CAD) currently widely used in the industry employ a tighter connection between the design and documentation models, however, interchange of documents between consultants is predominantly of analogue methods – hardcopy paper prints or non transferable formats such as Adobe PDF.

In a survey conducted in 2005 by Harris Interactive Polling Organization it was shown that traditional methods of hardcopy document exchange (fax / mail) dominated the AEC industry. One of the major difficulties in document exchange that the survey noted was the number of different file types and applications that they needed to deal with, resulting in an analogue output choice for most surveyed members.

3.2 Data-Centric Models

With data centric models, explicit information such as geometric, material, building zoning, etc. properties can be embedded within virtual building elements. The focus with this method is to maximise data transfer and continuity between design, construction and even into facilities management stage. This type of model is illustrated in Figure 2.

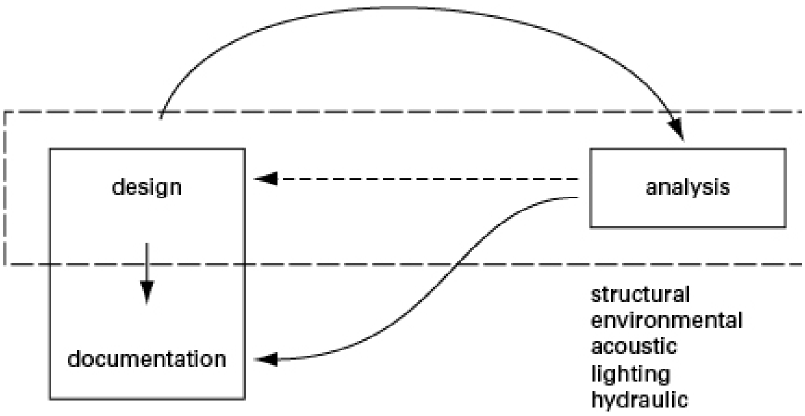

Fig. 2 *Current Data-Centric Process*

A limited example of a current data centric model is the use of DXF as a file format. DXF can combine geometry, object tags and layer structure within the model and this data can be effortlessly shared between various applications of design and analysis. There are limitations to set by the DXF model and any additional information will be lost when transferring in this format.

An advanced example would be the use of an IFC model to embed extensive amount of information such as Quantity Surveying, Project Management, Structural, Environmental and Building Operations data. Lossless information can be implemented within a range of applications as long as the proprietary software is IFC enabled.

3.3 Database-Centric Models

The theory of the Database-Centric Model has been pursued in since the late 1970's, known as Building Product Model. It was later coined as the Building Information Model (BIM) after AEC software developers such as Autodesk and Bentley (Triforma) moved to incorporate the technology within their suites. BIM heavily relies on Datasets that group and aggregate information, making this available to a number of packages. The data-centric process is illustrated in Figure 3.

Both Autodesk and Bentley structured their development around single database to serve a distributed team and application set. With the complexity of projects, diverse applications and international teams collaborating on projects, the single database becomes too large to work efficiently and is quite difficult to aggregate with data from a number of partners. Bentley are now looking towards developing federated databases that can serve clusters of teams internationally rather than the single database structure.

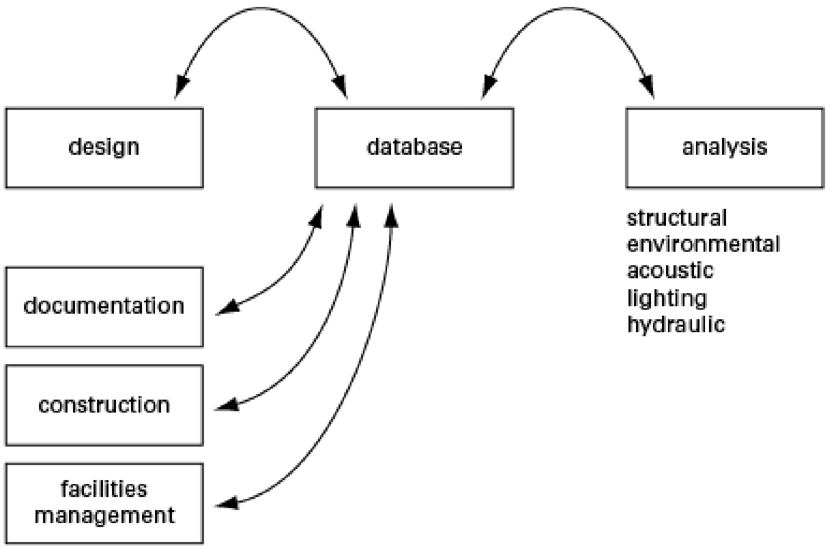

Fig 3. *Database-Centric Process*

4. Generative Components Case Study

4.1 Research Outline

The research described in the following sections, sets out to explore the efficiency of data flow and exchange between design and analysis disciplines by targeting a specific project and applying software integration methods. The current industry direction of interoperability is through a complete snapshot of native information from independent software packages via IFC or CIS/2 data models. In the research undertaken by the authors, the focus is on implementing database centric design using XML (Extensible Mark-up Language) and MDB (Microsoft Access Database) to store information in a number of accessible databases.

The first study model comprised of a simple bridge design to test the exchange of information between a design program Generative Components (Structural) and a widely used commercially available structural engineering package, known as Space Gass [2].

4.2 Data-Centric Case Study

A Bentley structural model was developed for design of the bridge, using structural elements within the software interface. The two structural shapes used were a RHS150x50x6 and a CHS140x5. The file was exported as a DXF file format (Release 14). However, in order for the structural model to be imported successfully into Space Gass it was necessary to rebuild the model in Bentley Structural as a wire-frame model, since Bentley structural shapes or blocks cannot be read into the engineering package. Once inside Space Gass, the wire-frame must then be set up as structural members, defining the location of supports and applying the relevant loads before an analysis can be undertaken.

Exporting the data as DXF allows for the members to be sent as 2d or 3d lines (member outlines) or 3d faces. The model is then imported back into Bentley Structural where the structural skeleton has been replaced by a number of lines that represent the structural member outlines.

4.2.1 Graphical Process and Reverse Engineering

To understand the syntax and structure of the data that Space Gass operates with, a simple structure was exported as an MDB Access File. The most important features noted were the table names, node positions and the format in which the data of members were organised. This gave a clear understanding of how the structure of the initial model should appear. Design

The process started off with a 2 sets of nodes generated by an algorithm to produce the form. A feature within Generative Components (GC) called StructFormVB can be used to produce a structural component between a start and an end node that defines the spatial location of a single member. A script was used to gather the node sets and create a number of edge structural members in this arrangement, as illustrated in Figure 4. An alternate script sorted out paired nodes and placed cross structural members between the pairs.

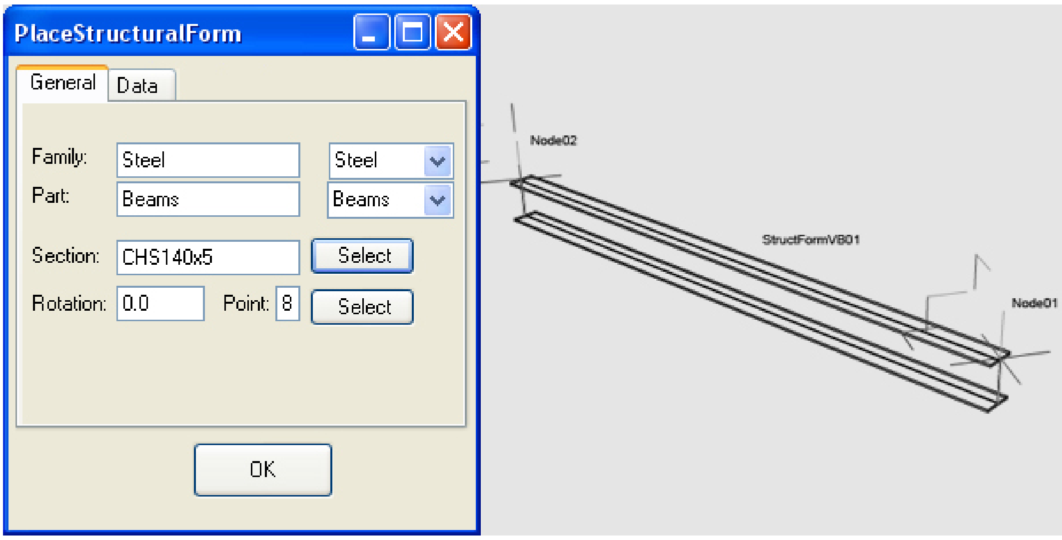

Fig 4. *Structural Form Dialog Box and Graphic Representation*

4.2.2 Generative Components Output

A script was then developed to filter and list all nodes that supported members. All members are sorted and their supporting node information is appended to their data set. Both lists are sent out using the DataExporter feature, where tables for both nodes and members are and lists filling their respective tables (Members, Nodes). All other tables such as Material, Section, etc are aggregated from the universal database (Figure 5).

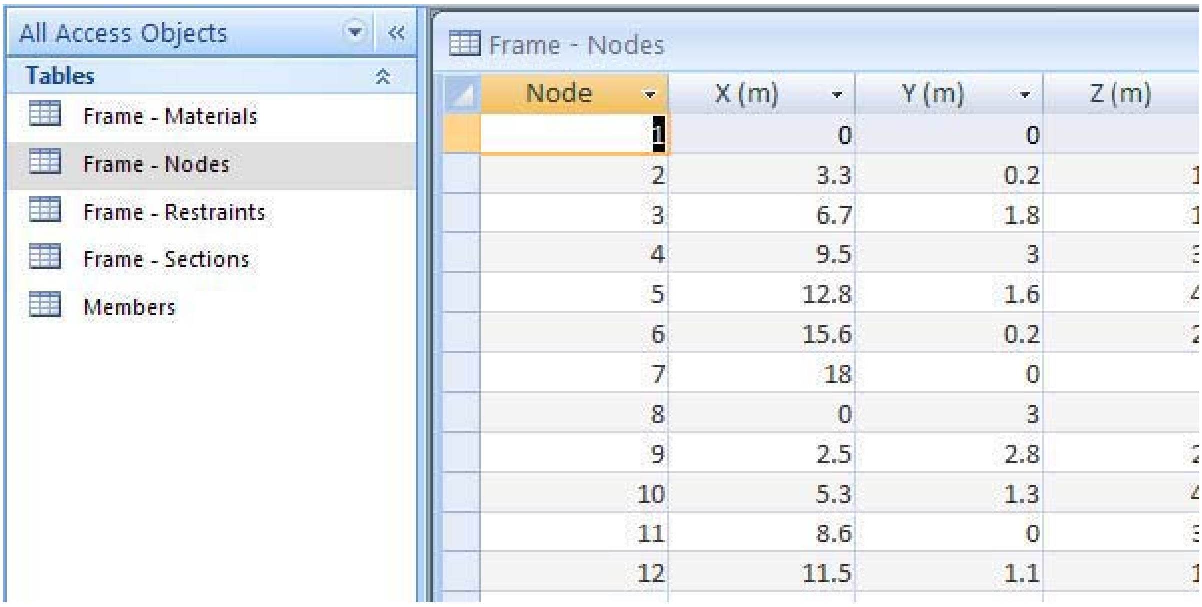

Fig 5. *Access Database structure and Node table*

By importing the database bridge.mdb, Space Gass looks at the relevant tables such as Materials, Nodes, Restraints, Sections and Members to develop a full working structural model, ready to apply loading and additional calculations. All loadings and forces can however, be a feature built into Generative Components and can either be applied or read as vectorial information within GC.

After applying loads, an analysis solution can be generated in Space Gass.

4.2.3 Structural Analysis Feedback

The new loads, displacements, reactions, etc can be added to the bridge.mdb database by exporting to MS-Access. This will append a number of new tables to the database such as Loads and Results All the new load and distribution data can be loaded into GC with the DataImporter feature. Using a script, a new list can be built to organise the displaced nodes and members. This feed back is significant, because it enables analysis of the variation between the unloaded and the loaded structure. This effect is shown in Figure 6, with a comparison between the Space Gass deflected shape and the updated data regenerated in GC. The structure may even be completely rebuild or altered in Space Gass and the new structure can be loaded directly via the database.

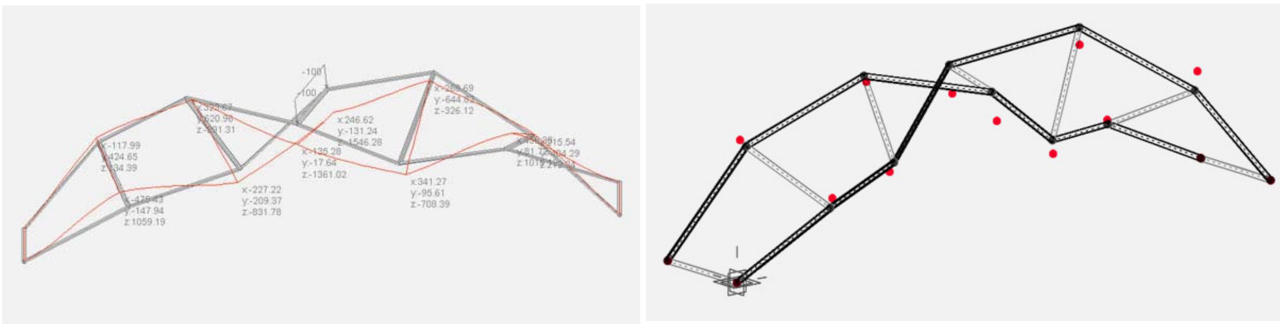

Fig. 6 – *Comparison of Loaded shape – Space Gass (left) and GC Drawing (right)*

5. Process in developing a Generative Components Model

5.1 Required Modelling Stages:

From the case study described above, a process has been developed that enables highly efficient interaction to occur between the Architectural and Structural Engineering software packages. The steps involved in this process are described as follows:

9. XML data is parsed to a Microstation and Space Gass MS Access database (MDB) that filters out specific structural members, elements and their respective qualities. The XML database contains all universal information such as section names, structural material, etc. that can be filtered out for software specific integration.
10. Software databases are imported into respective programs. These databases build a library of structural components and elements that the software uses either parametrically or analytically.
11. Microstation Generative Components develops a parametric model using GC features. The features are appended with additional qualities that can be used in structural analysis. This model is output directly to a shared MS Access database that both Space Gass and Microstation will utilize.
12. The Access database is imported into Space Gass. This database loads explicit information such as node locations, node fixings, member types, member material and load cases.
13. Space Gass loaded model data appended to MS Access database.
14. DataImport function imports and builds loaded model in GC
15. Loop can continue indefinitely.
16. Resulting GC model documentation can be automated using a fabrication planning feature.

5.2 Further Iterative Inclusions to the Database-Centric Model

The process outlined above can be further refined to enhance an "integrated" approach to design for architects and engineers. Generating and working from a common data base required the following processes to be implemented:

1: Database Build

- Compile an XML universal database with tagged information that can serve Microstation and Space Gass databases.

2: Scripting Complexity - Microstation

- Build upon study model to include: Jigs, Scaffolds & Features to utilize the power of Generative Components as a design tool.
- Build lists to translate Parametric Space (position on surface, etc) to Cartesian Space (X,Y,Z coordinates)
- Include additional information such as loads to members and node fixings within GC.
- Automate DataExporter feature to address all nodes, members, loads and node fixings to write all data to file

3: Interaction and Iteration for Optimising Design

- Automate DataImporter feature to include all data and build resulting model
- Analyse differentiation between original model and loaded model to adjust original model in a more economical form.
- Continuation of Feedback loop - adjusted model exported / imported back into Space Gass for *nth* round of analysis

An example of the type of complex bridge form that has been generated and analysed using the process developed in this research project is presented in Figure 7.

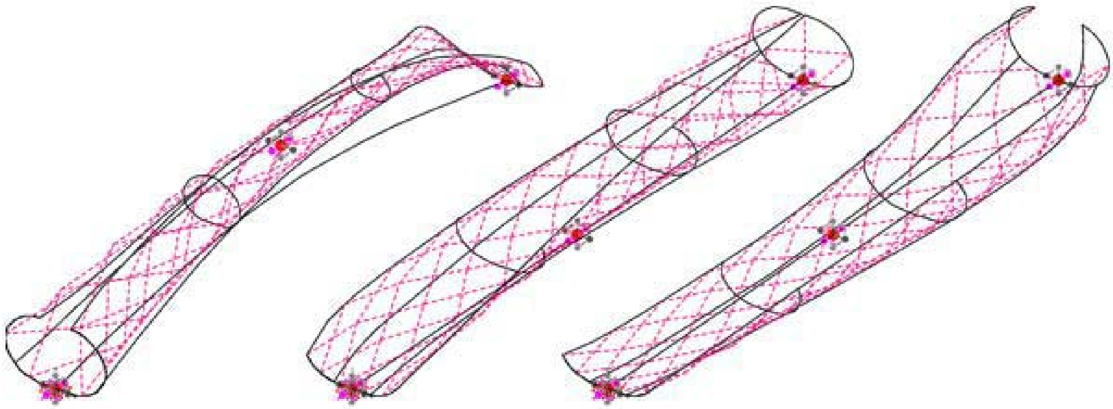

Fig. 7 – GC *Spline used to develop a foot bridge, with a basic skeletal structure of CHS steel*

By adjusting the parameters of the edge rail distribution, a number of outcomes can be rapidly developed. The power of Generative Components is that the structural arrangement can be driven by the scaffold, edge rails, base spline or any other variable higher up in the geometries history. By adjusting the point series count, the structural members will follow change further down the history chain. This variable can be automatically adjusted by a check with the node displacements generated and sent back from the structural analysis program.

6. Conclusions from Results of Data-Centric & Database-Centric Case Studies

6.1 Live Data

It is often the case that each party involved in the design process holds snapshots of the model at different stages of revision. This is easily seen when the design is rapidly evolving and the models between say an Architect and an Engineer are not being developed concurrently. A database-centric model can be updated with every design change or at stepped intervals, giving design partners access to the up to date information. The database can be located on a web server, giving worldwide read / write access to the model data.

This project has demonstrated that while data-centric models provide a snapshot of the model at a stage in the design, database-centric models hold live information, enabling a build and assessment of models in multiple disciplines simultaneously from its content.

6.2 Flexible Modelling

Generative Components implements flexible modelling techniques, easily generating many iterations of the design model. Using custom features within GC, the same scripts gather all information in each of the iterations, organising the data into exactly the same structure format within the database. This new data is ready to be assessed within analysis software.

Document-centric and data-centric solutions require an almost total rebuild of the translated file, creating a bottleneck in the design / analysis process - iterative design is proven useless when it requires quick feedback from analysis software.

6.3 Open Source

Data-centric models such as DXF and IFC formats require the proprietary software developer to implement these formats in their packages. After more than 10 years if IFC incubation, many software developers still have not embraced the format. An incomplete IFC structure may be the reason that it has not become a standard, or it could be the complexity of file structure that is holding the format from success.

Software developers such as Autodesk and Bentley are now taking the approach of Integration over Interoperability with the purchase and internal development of structural analysis programs to compliment their design, documentation and co-ordination packages. This still does not resolve the issue of universal interoperability.

This project has presented a viable alternative, suggesting that database-centric models might well be the future for software users. As long as a software package can export to a database, the structure of its data organisation is clear and openly accessible. This paves the way for interoperability between packages as users can ultimately filter and send explicit information between software through database methods.

6.4 Reduction of Errors

Ability to automatically send explicit data between packages also means an exact translation of information such as co-ordinates, area calculation, displacement values, loads, material qualities, etc. can be made with no chance of error. Document-centric and data-centric models require almost all the information to be rebuilt with each translation – more manual input also means increased chance of incorrect or missing data in the new model.

Database-centric models can minimise the risk of error by reducing the amount of manual handling during the translation process. In some cases data-centric and document-centric model techniques may be more time efficient with simpler models. The start up time to sketch a design in alternate design software may be quicker than with Generative Components, although, if we are assessing not only the economy of time but look at the risk in accuracy of information translated or the consistency of information gathering, organisation and structure, the database-centric model has shown that it far exceeds the other methods in effectiveness.

6.5 References and Acknowledgements

[1] Ali Malkawi & Godfried Augenbroe, *Advanced Building Simulation*. Taylor and Francis – 2004; ISBN 0415321220

[2] Space Gass, *3D Structural Analysis and Design Software Version 10.5*, Integrated Technical Software P/L, Geelong Victoria 3220, Australia. www.spacegass.com

The authors wish to acknowledge the enabling support of a Challenge Grant (funded by the Pro Vice Chancellor – Research) from the University of Technology, Sydney.