Knowledge Based Systems as Tools in the Systems Engineering Process

Christian Van der Velden, Cees Bil
School of Aerospace Mechanical and Manufacturing Engineering,
RMIT University
GPO Box 2476V
Melbourne, Vic 3001
AUSTRALIA
S2108846@student.rmit.edu.au
cees.bil@rmit.edu.au

Adrian Smith
GKN Aerospace Engineering Services Pty Ltd,
Brisbane Office
Level 22, 288 Edward Street
Brisbane, Queensland 4000
AUSTRALIA
adrian.smith@au.gknaerospace.com

Xinghuo Yu
School of Electrical and Computer Engineering,
RMIT University
GPO Box 2476V
Melbourne, Vic 3001
AUSTRALIA
x.yu@rmit.edu.au

Abstract. This paper explores the use of Knowledge Based Engineering (KBE) methodologies and technologies as tools in the Systems Engineering process. Discussed is progress in the development of a system for automating the complex engineering task of electrical wiring design in aerospace vehicles. Knowledge Based Systems capture rules and engineering knowledge relating to one or more processes in the design phase and are used for design automation, verification, updating and linking product lifecycle phases to ensure compatibility. Effective information management techniques employed in Knowledge Based Systems can lead to greater product integrity, shorter development time and reduced cost. Knowledge Based Systems provide a means of ensuring compliance with requirements and performance parameters. In this example, a system for discovering paths for electrical cables or pipes through 3D structural models is developed. Numerous rules and regulations governing electrical design are automatically validated, and only valid solutions are returned to the engineer for consideration.

Keywords: Knowledge Based Engineering, intelligent algorithm, routing, Computer Aided Design

KNOWLEDGE BASED SYSTEMS

Knowledge management is of great importance within the engineering field. Knowledge Based Engineering (KBE) is a branch of engineering concerned with capturing rules and knowledge relating to one or more processes in a design cycle and implementing in software systems called Knowledge Based Systems (KBSs) that emulate human decision making processes. KBSs are used for automation, verification and integration of product design and manufacturing data, ensuring compliance with performance parameters. Use of these practices can facilitate smooth transitioning between product lifecycle phases.

KBE practices have been in use for a number of years. Numerous engineering companies such as British Aerospace, Boeing, Airbus, Jaguar and many more, have taken advantages of the competitive edge that KBSs can deliver in terms of quality, time and cost (Cooper et al., 2001). However, despite the many economic benefits KBSs can offer, a number of difficulties have prevented their widespread including, accoring to (Brown, 2006):

- Unwillingness of management to invest in activities without immediate return.
- Lack of resources to allocate to KBE activities.

- Collection and implementation of knowledge from domain experts can be tedious.
- KBSs can be viewed as a risk to job security.

The definition of what comprises KBSs varies widely between individuals and organizations. For some, KBSs are rule-based systems used for automation of low level, narrow, and repetitive tasks (eg. bracket design). To others, KBSs comprise much more and are used for high level tasks involving use of best practices, and decision rationale. The level of automation delivered by KBSs can be classified as:

1. Automation of narrow tasks.
2. Automation of model and data abstraction.
3. Automation of a documented design process
4. Discovering solutions to unique problems.

The first level involving “narrow” tasks includes automating rudimentary tasks such as model building using drawing tools (eg. lines, circles, etc.). The second level provides higher level operations which can be performed on the former which add detail, or knowledge, to the product (eg. geometry operations such as midsurface extraction and defeaturing tools, and programming tools such as Application Programming Interfaces API’s). The next level considers an complete enginnering design task consisting of a number of lower level tasks covered by either of the first two levels of automation into a single process where the user specifies critical parameters. In this level of automation, design rules are implemented through “IF {condition}, THEN {statement}” rules. The final level goes higher still and attempts to apply reasoning, or semantics, from a library of multidiciplinary knowledge and experience of varying types.

The KBS for solving routing problems described later in this paper attempts to fit somewhere between the third and fourth levels described above. The system will automate a design process while attempting to use semantics to find the solution of unique, related problems, such as new routing situations of different geometry, scale and type.

KNOWLEDGE BASED SYSTEMS AND SYSTEMS ENGINEERING

The aerospace industry is becoming increasingly globalised and competitive. Design and manufacturing contracts are awarded to companies who can deliver quality designs and components with low cost and in a short development time. Systems Engineering (SE) provides a means of managing large projects from the top down, covering all phases of the lifecycle from concept to deployment and ongoing support. Design, analysis and manufacturing technologies are evolving rapidly. Advances in computer hardware and software including Computer Aided Design (CAD) software for drawing, Computer Aided Engineering (CAE) software for analysis, and Computer Aided Manufacturing (CAM) software for automated manufacturing are changing the way companies do business. These advancements are changing the ratios between design and analysis engineers in the workplace from 4:1 about twenty years ago, to about 1:2 in today’s engineering environments (Smith et al., 2005). Competitive advantage can be gained through effective linking of the design and analysis process by forging links between the tools used to carry out the work. KBSs can provide the mechanism for linking these phases.

As mentioned previously, KBSs are used in numerous capacities. The following attempts to identify and categorise some of these.

Knowledge Based Systems as automation tools.

- Reduce low level, repetitive and tedious engineering tasks. An example of this could include design spreadsheets designed to perform calculations and return critical design parameters.
- Perform high level design tasks requiring problem solving through application of semantics and fuzzy logic. An example of this could include feature recognition systems which identify design intent and can automate section of work based on this.
- Reduce or eliminate bottlenecks in design process. KBSs can help reduce lengthy design and analysis procedures. Advancements in modeling software has led to a reduction in the ratio of designers to analysts. Linking CAD, CAE and

CAM software can help smooth the line between these distinct product development phases.

Knowledge Based Systems as verification tools.

- Automatic rule checking and validation. Automatically ensures design and component outputs are of sufficient quality to meet performance parameters and certification standards.
- A knowledge library which can be easily updated and built upon can ensure ongoing compliance as regulations are updated.
- Learning systems can also ensure knowledge retention within the industry. One problem facing many organizations is the loss of valuable knowledge and experience as personnel age and retire. Much of this knowledge is not documented and not easily grasped. This lost knowledge can take years to rebuild within the company. KBSs can help capture this information and reduce the cost as older engineers move away from the field.

Knowledge Based Systems as integration tools.

- Linking design and manufacture phases. Inconsistencies between design and manufacturability requirements require rework and can cause lengthy delays. Alerting engineers to these requirements early in the process can avoid problems later in the SE cycle.
- By linking product design and manufacturing data at early stages in the product development, feasibility of concepts can be evaluated rapidly, and can assist in tooling design.

Knowledge Based Engineering tools. As KBSs are themselves tools employed for automation, verification and integration, there exist numerous tools for the development of these systems.

- KBSs are usually developed using object-oriented programming (OOP) principles and languages. Concepts of inheritance, association and abstraction are well suited to representing knowledge within the system.
- OOP also encourages code reuse, allowing building blocks and standard libraries of generic code to be established allowing faster development of future KBSs.
- Effectively employing human resources is critical in KBS development. Teams should be multidisciplinary, with engineers and programmers working together throughout the whole process to ensure final outputs are accurate, relevant, and easy to use (Smith et al., 2005).
- Software development environments have been created specifically for building KBSs. One example is ICAD, now owned by Dassault Systemes. ICAD is based on the Lisp programming language and was a popular development tool among KBE developers some years ago. Automated component design could be achieved relatively easily. However, use of this software has diminished in recent years, with emphasis placed by on other design products published by the software provider.

Knowledge Based System Development. The development process for automating an engineering process is summarised in the following flowchart (Fig. 1).

For some organizations, KBE means rapid development of applications using standard building blocks, and existing methods and data. The end result is deliberately limited in scope of use (Smith et al., 2005). To others, system development is an evloving process with longer development time, with end results having a wider scope and a higher level of intelligence.

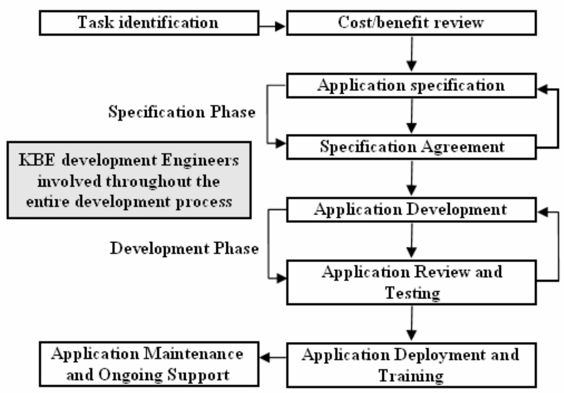

Fig. 1. KBS Development process (Smith et al., 2005)

A KNOWLEDGE BASED SYSTEM FOR ELECTRICAL WIRING DESIGN

This section describes the current progress in the development of a KBS for 3D routing in aerospace vehicles with applications including electrical wiring and hydraulic/pneumatic piping design. The aim of the system is to read model data together with source and target terminals, solve a routing problem which satisfies constraints, and output wire/pipe geometry and other information required to describe the system path. Discussed in this section is the general approach to solving the routing problem, the path finding algorithm under development, methods of reading and writing model data, and a description of the total system.

Routing Problem and Process. Electrical wiring looms in aircraft typically consist of thousands of cables and are usually routed by hand using Computer Aided Design (CAD) workstations with engineers using personal knowledge and experience of how to route cables through the structure. Numerous regulatory and functional design rules govern the design process and must be satisfied for certification. Some of these are discussed below. The routing process is highly repetitive and design outputs can vary significantly between engineers. Electrical wiring design often proceeds in parallel with principle structural design. The iterative nature of the total design process is such that structural changes are prone to occur requiring time consuming rework for any electrical cabling affected. Fig. 2 provided by (GKN Aerospace Engineering Services Pty. Ltd.) shows an example of an electrical loom design output. In a similar way, hydraulic and pneumatic pipes in aircraft are manually routed and are governed by different set of design rules. The repetitive, rule-governed nature of the routing process makes it a prime candidate for application to a knowledge based system.

The routing problem is commonly encountered in numerous fields ranging from electronics, data flow in computer networks, navigation systems, and artificial intelligence (AI). Examples include design of Printed Circuit Boards (PCBs), Very Large Scale Ingetrated (VLSI) circuits, Global Positioning System (GPS) navigation systems, computer game and robot AI.

VLSI routing automation practices provide a good starting point for addressing electrical loom and pipe design problems in aerospace vehicles. Computer processors consist of millions of logic components interconnected using very fine wires within a very small space. The VLSI routing problem is considered NP-complete, commonly employing powerful heuristics which can find near optimal solutions. In the case of electrical looms in aircraft, the number of connecting wires to be routed, or nets, is several orders of magnitude less than computer chips therefore routing algorithm run time would be expected to be significantly less.

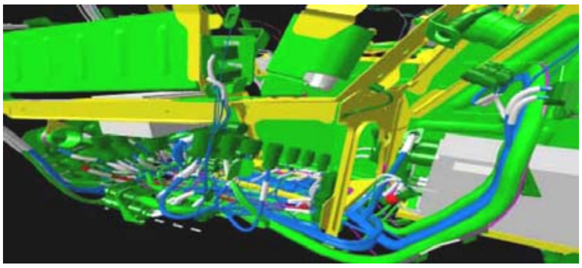

Fig. 2. Example of routed electrical loom. (GKN Aerospace Engineering Services Pty. Ltd.)

In general, once physical component layout is defined and routing requirements given (usually in the form of a netlist), the routing process consists of four main steps according to (Groeneveld, 2005):

- Definition of regions: problem divided into smaller routing problems.
- Global routing: planning phase which assesses and prioritises nets to maximize completion rate (proportion of solvable nets), and minimize total path length, especially for critical nets.
- Ordering regions: determines order in which regions are routed such that congestion will be avoided.
- Detailed routing: determines the exact path taken by wires including layers, connecting vias.

Routing Rules. As mentioned previously, design of electrical wiring systems for aircraft is a complex tasks with hundreds of rules and best practices which must be adhered. Currently, there is no dedicated section of the Federal Airworthiness Requirements for transport category aircraft (FAR-25) dedicated to wiring design practices. Instead, a number of sections touch on the subject including

25.1301/1309, 25.1529, 25.1353, 25.869, AC 43.13-1b, AC 25-16, AC 25-10, and policy memos (Sadeghi, 2003). Engineers must sift though a large amount of data to single out the rules applicable to wiring and monitor any updates to these rules made by governing bodies. For military aircraft, a different set of rules apply and are contained in MIL-W-5088L: Military Specification– Wiring, Aerospace Vehicle, last updated in 1991.

The KBS under development aims to collect these rules and implement them in a path finding algorithm which will return valid paths. For the system to have maximum flexibility, it will need to incorporate methods of updating existing rules and adding new rules as necessary, with minimal effort. The following, obtained from civil and military airworthiness requirements (FAR-25) and (MIL-W-5088L), lists just some of the areas of consideration when designing electrical wiring systems:

- Electrical Loads
- Breaker/Wire Sizing
- Wire Routing
- Clamping
- Tie-wraps
- Bend radii
- Splicing
- Wire terminations
- Grounding & Bonding
- Wire Marking
- Connectors
- Conduits
- Wire Insulation
- Wire Separation
- Chafing
- Unused wiring
- Riding on structure
- Riding on other wires
- Passing through lightening holes
- Slack
- Corrosion / contamination
- Documentation

Routing algorithm. Numerous algorithms are used for a variety of routing applications and for different stages of the routing process, as well as specialised applications. The main categories of path finding algorithm include maze routers, channel/switchbox routers, and line search/probe routers.

In its basic form, the algorithm used by the KBS to find paths is based on the classic breadth-first, grid-based maze algorithm or Lee's algorithm described by (Lee, 1961). Lee's maze router is proven technology and returns the shortest path for a single net in a given search space with obstacles. The algorithm functions by propagating a wave from source and/or target terminals over a grided search area and assigns a value to each node depending on its distance from the source or target. A backtracking phase then determines the shortest path between the two terminals (see Fig. 3).

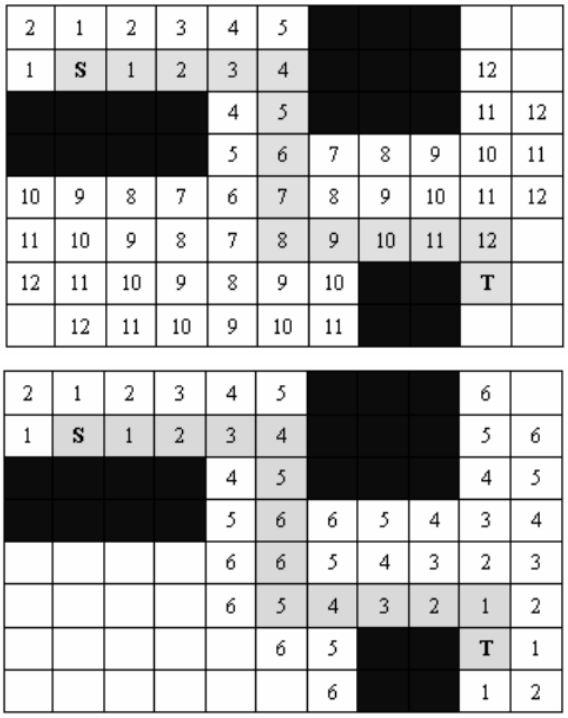

Fig. 3. Example of maze routing. (Top) wave propagation from source. (Bottom) wave propagation from source and target

This basic maze algorithm has numerous limitations of which the main problem is sensitivity to the order in which nets are routed. Paths from routed nets form obstacles for subsequent nets and in some cases can prevent nets from being completed. In cases where this is encountered, a rip-up-and-reroute procedure can be used which removes routed nets and retries in a different order. In addition to this, the algorithm is inefficient when routing more than two terminals in a single net. In the case of multi-terminal nets, the connection between two terminals is found, then the partially routed net is treated as the source for remaining terminals. Also, the algorithm is inefficient when routing terminals in large empty spaces.

To address shortcomings of the basic maze algorithm, alternative path finders have been developed based on Lee's basic principles, employing intelligent searching techniques, and are used in numerous applications. One example is A* (A star), used extensively in computer game navigation which employs a "best-first" search technique to determine path steps (Fig. 4, top) as given in the tutorial by (Lester, 2005). A score is given for each node (F) based on minimum distance to target (H) and distance from source (G). Nodes with lower scores are favoured. Another example is Hadlock's algorithm which uses a "greedy" search technique and adds penalties for every deviation away from the target (Fig. 4, bottom) (Tehranipoor, 2005).

G = 4 H = 8 F = 12	G = 3 H = 7 F = 10	G = 2 H = 6 F = 8	G = 3 H = 5 F = 8		G = 9 H = 3 F = 12	G = 10 H = 2 F = 12	G = 11 H = 3 F = 14
G = 3 H = 7 F = 10	G = 2 H = 6 F = 8	G = 1 H = 5 F = 6	G = 2 H = 4 F = 6		G = 8 H = 2 F = 10	G = 9 H = 1 F = 10	G = 10 H = 2 F = 12
G = 2 H = 6 F = 8	G = 1 H = 5 F = 6	S	G = 1 H = 3 F = 4		G = 7 H = 1 F = 8	T	G = 9 H = 1 F = 10
G = 3 H = 7 F = 10	G = 2 H = 6 F = 8	G = 1 H = 5 F = 6	G = 2 H = 4 F = 6		G = 6 H = 2 F = 8	G = 7 H = 1 F = 8	G = 8 H = 2 F = 10
G = 4 H = 8 F = 12	G = 3 H = 7 F = 10	G = 2 H = 6 F = 8	G = 3 H = 5 F = 8	G = 4 H = 4 F = 8	G = 5 H = 3 F = 8	G = 6 H = 2 F = 8	G = 7 H = 3 F = 10
G = 5 H = 9 F = 14	G = 4 H = 8 F = 12	G = 3 H = 7 F = 10	G = 4 H = 6 F = 10	G = 5 H = 5 F = 10	G = 6 H = 4 F = 10	G = 7 H = 3 F = 10	G = 10 H = 4 F = 14

5	4	3	3	3	3	3	4	5
4	3	2	2	2	2	2	3	4
3	2	1	1		2	2	3	4
2	1	S	0		2	2	3	4
2	1	0	0				4	5
2	1	0	0		2	T	3	4
3	2	1	1		2	2	3	4
4	3	2	2	2	2	2	3	4
5	4	3	3	3	3	3	3	4

Fig. 4. Extensions to maze algorithm. (Top) A* algorithm (Bottom) Hadlock's algorithm (Tehranipoor, 2005)

The algorithm under development uses object-based programming principles and implements intelligent search techniques such as best-first and greedy searches. As far as possible, the algorithm uses intuitive, plain English terminology in its implementation. The algorithm will access a knowledge base of design rules and best practices. Whereas most VLSI routing algorithms are based on a multilayered 2D approach, algorithm used in this system can solve 3D mazes of any dimension. Multiple nets can be routed in the same search space. Multi-terminal nets can also be routed by routing the path between two terminals first and then connecting additional terminals to the original path.

Reading and writing model data. To apply a grid-based algorithm to the routing problem, the 3D model of surrounding structure and obstacles with source and target positions is to be discretised. Currently two methods of geometry preparation are under consideration which use Finite Element (FE) and Voxel modelling techniques.

Finite Element. The first method uses a Finite Element (FE) modelling approach which is common procedure in the engineering process of analysing the response of structures to loading. For this method, empty space in the model is meshed using solid elements (for example CHEXA, Fig. 5), and this mesh searched for a valid path between given source and target nodes. Property cards are used to distinguish between structural mesh and search space mesh. The advantage of this method is that mesh generation is standard practice in the engineering design process, allowing existing knowledge and software to be used. Mesh coarseness can be varied depending on accuracy required. The main drawback of this method is irregular mesh shapes.

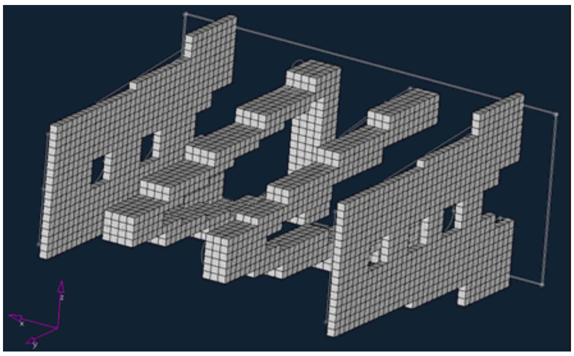

Fig. 5. FE solid mesh

Fig 6. shows the searching process. It begins by searching and selecting the starting node, then querying the element list for attached elements. The algorithm would then select the "best" attached element based on direction of the target and and other applicable crtieria contained in the knowledge library. From this element, attached nodes can be determined and the best node selected based on similar rules. This process of selecting nodes then attached elements continues until the target is found and all conditions satisfied.

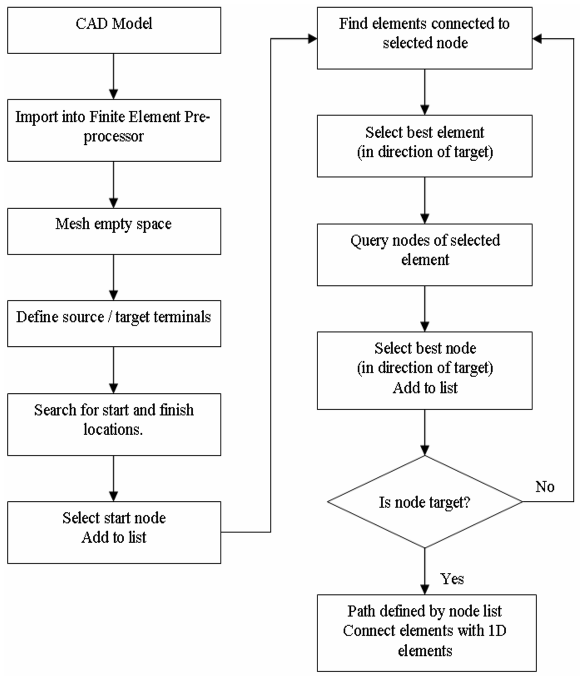

Fig. 6. Searching process using FE methods for geometry discretisation

Voxels. 3D models in virtual environments such as CAD/CAM applications and computer games are generally represented using surface graphics which are composed of polygon meshes (usually triangles) over hollow shells. For high quality visualisation of models, a large number of polygons is required. For each polygon, complex computations are performed to determine shading, requiring high end computer graphics hardware for display.

Volume graphics is an emerging technology in 3D model representation which uses stacks of 3D cube elements called voxels (or volume pixels) which are analogous to pixels in 2D images (Stevenson, 1996). Voxels are defined by a number of characteristics including size, address (in x, y, z coordinates), state (on or off), colour, density, etc. (Fig. 7). Voxel rendering engines assume all voxels face the “camera” as a model is rotated, reducing the memory requirement to a single position and colour for each element. Whereas surface graphics require complex computations to determine shading, volume graphics can be displayed without use of 3D acceleration hardware.

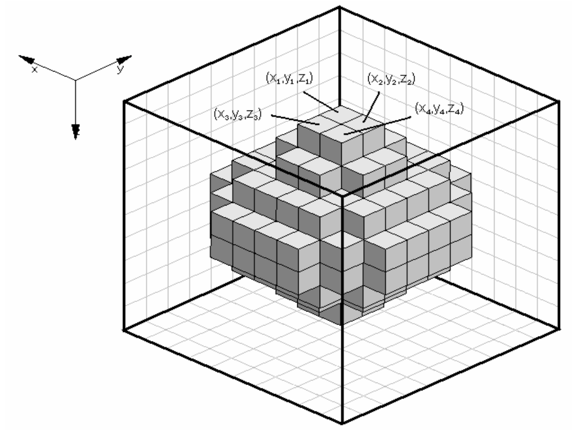

Fig. 7. Voxels represented in 3D space.

Whereas surfaces modelled in traditional CAD software use complex relationships, volume graphic representation uses a fixed x, y, z integer-based address for each element. If a model to be routed could be expressed in such a format and the data structure accessed, a grid-based algorithm could be directlty applied to the problem. This method would be advantageous in its simplicity. The searching process is shown in the flowchart in Fig. 8.

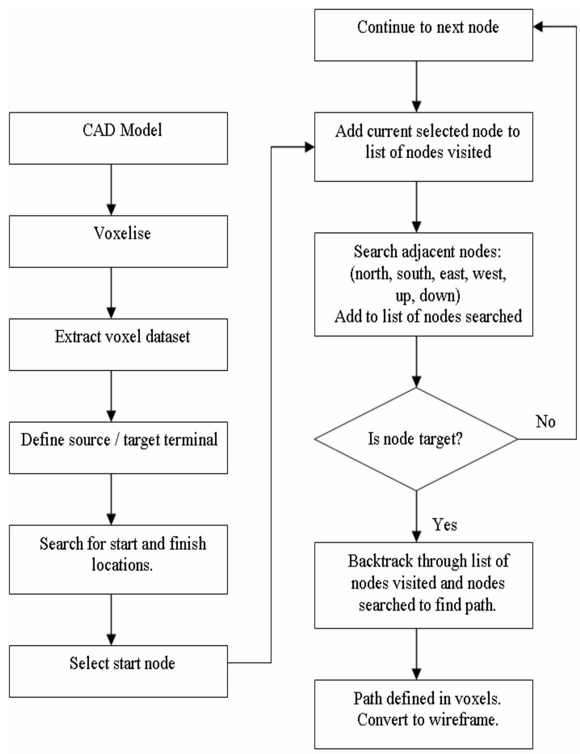

Fig. 8. Searching using voxel methods for geometry discretisation.

The disadvantage of this method is the availability of voxel software. Some open source voxel engine software is available freely on the internet, however the power of these applications is limited, requiring expensive commercial software (such as that used in laser scanning of three dimensional models). Many commercial software packages include conversion utilities from common CAD model formats (such as IGES, DXF, STEP) to voxelised representation.

System description. As previously discussed, the KBS for electrical wiring in development is a mixture of the third and fourth levels of knowledge implementation described by (Brown, 2006), discussed in the first section. Included will be system mechanics required to read and write 3D model data, and an intelligent path finding algorithm to navigate the structure. The system will include a library containing knowledge and experience necessary to solve routing problems of varying complexity, and will be easily interchanged allowing different rule and knowledge sets relating to different routing applications to be implemented.

The knowledge based routing system comprises a number of steps which are as follows and are summarised in the flowchart in Fig. 9. Firstly, physical structure is designed and modelled using CAD software by structural engineers. Electrical or piping requirements are defined in terms of start and finish locations and other relevant characteristics such as category of load etc., and are given in the form of a netlist. The 3D model is then exported from the CAD package using a neutral file format such as IGES or STEP. The CAD model is converted to a discrete format suitable for applying a grid-based search algorithm using either FE or voxel techniques discussed above. The discrete data set is extracted and fed into the maze algorithm which determines paths for multiple nets given constraints stored in the knowledge library. This module can be interchanged for different routing applications, not necessarily limited to aircraft (eg. a rule set for air conditioning ducts in buildings could be developed). After execution of the routing algorithm, the output path (defined in FE or voxel elements) is converted to wireframe geometry which is imported into a CAD package and detail added according to the knowledge base consulted in the process.

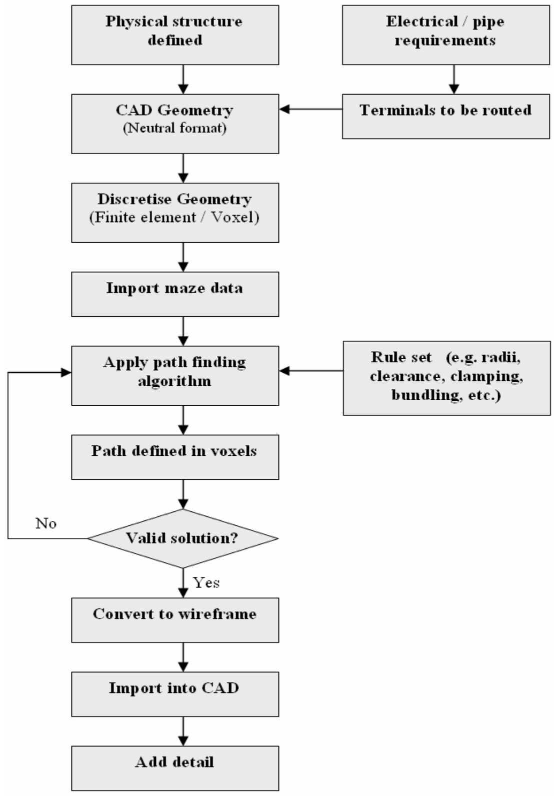

Fig. 9. Knowledge Based router process flow.

Current Progress. The KBS for wiring design described in this paper is a work in progress. Currently the system is capable of reading 3D models in a particular format, discovering the path for one or more source/target sets which may be single or multi-terminal nets, and writing the model to a format which can be viewed in an FE pre/post processor. A screen capture of a 3D demonstration applet is given in Fig. 10. To visualise results in 3D, the applet writes a Finite Element input file for NASTRAN which defines the path with nodes connected by 1D PROTEL elements. Obstacles are represented using 3D CHEXA elements with the solution space bounded using a CHEXA element. An example of the finite element visualization of the problem shown in Fig. 10, is given in Fig. 11. At this stage no optimisation of the algorithm has been attempted thus the efficiency is limited to O(d^3). However, for the applet shown in figure 5, the main cost in terms of processing time is the visualization using the standard windows graphics. Visualization of results will be improved with the use of OpenGL in the near future.

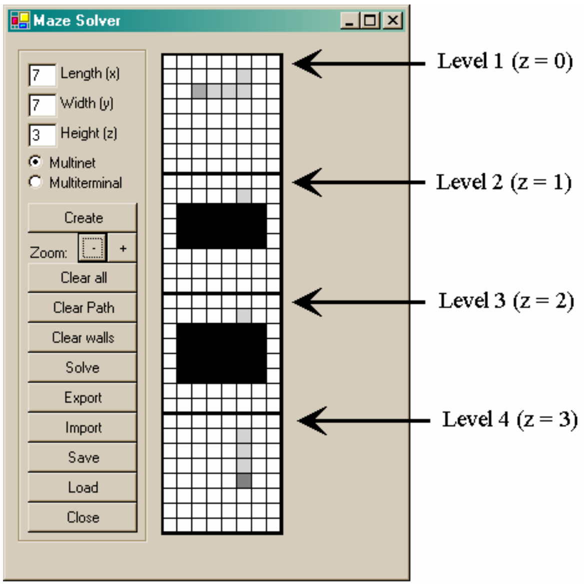

Fig. 10. Path finding demonstration applet.

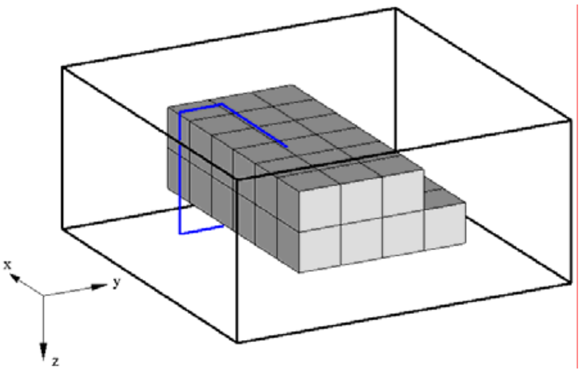

Fig. 11. Representation of search space, path and obstacles using Finite Element software

Currently, the procedure for solving a path finding problem is as follows. Firstly, the part or assembly is modeled using CAD software as is standard engineering practice (Fig 12, step 1). The model is exported using a standard file format, IGES. This model is imported into a FE pre/post processor and external space around the model meshed (Fig 12, step 2). Start and finish locations for a wire/pipe are entered as entity sets in the FE model. The model is exported as a FE mesh. Using the Windows based application discussed earlier, the model is read into the solver. The solver interprets start and finish locations and attempts to find a path between them. Once found, the path is exported as an FE model consisting of 1D elements (Fig 12, step 3). Wireframe elements are converted to geometry and exported as an IGES model. The completed path is imported back to the CAD design software and detail added according to knowledge gained in the process (Fig 12, step 4).

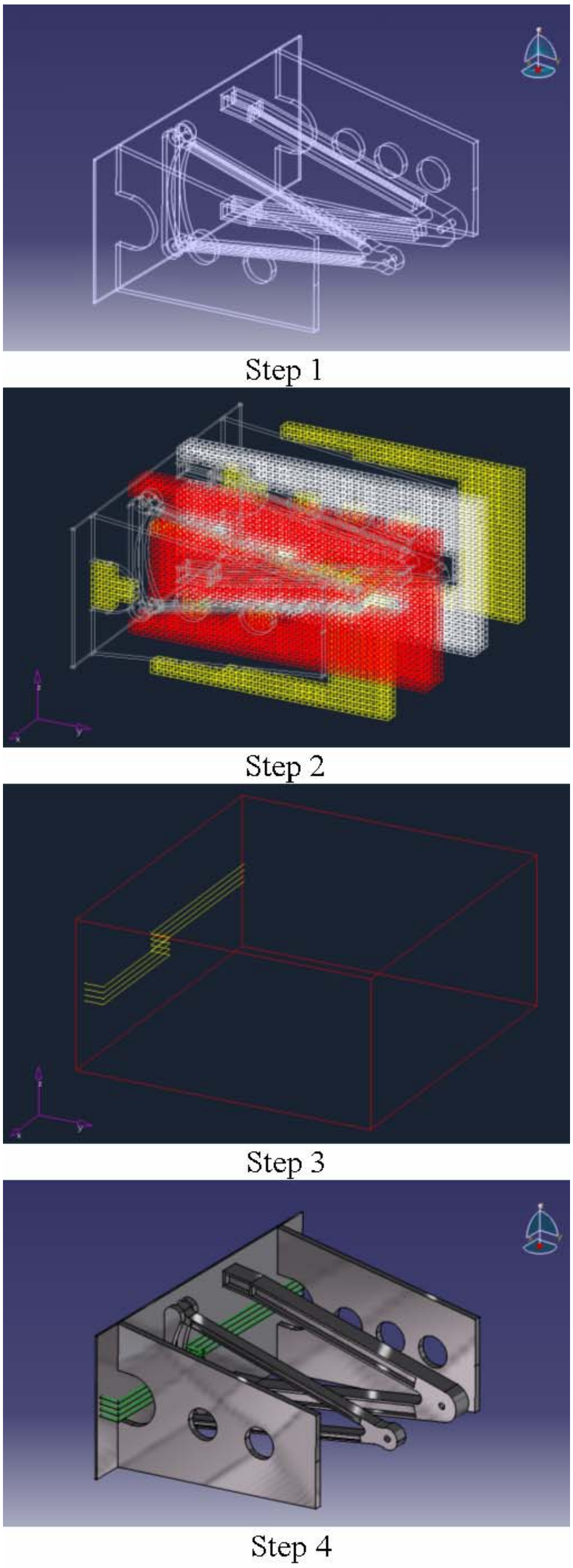

Fig. 12. Process for solving routing problem.

CONCLUSION

This paper has discussed the use of Knowledge Based Systems (KBSs) as tools in the Systems Engineering process. One of the bottlenecks in today's engineering environments is transitioning between phases in the systems engineering cycle including design, analysis and manufacture. Use of KBSs as linking mechanisms between CAD/CAE/CAM software, can assist in smoothing these transitions, providing a saving of both time and cost. Also discussed in this paper was the development of a KBS for electrical design automation for aerospace vehicles. The system itself consists of a number of elements including a knowledge library for storing routing rules, best practices and constraints, a method of discretising geometry using either finite element or voxel modelling techniques, and an intelligent path finding algorithm to navigate the structure and return a valid path for the wire/pipe.

REFERENCES

Arnold, M. H., and Scott, W. S. *"An Interactive Maze Router with Hints"*. Lawrence Livermore National Laboratory, University of California, 1988.

Brown, D. H. and Associates. *"Knowledge-Based Engineering Systems: Applying Discipline And Technology For Competitive Advantage"*. Gardner Publications, Inc. United States. [Online May 2006] `http://www.mmsonline.com/articles/0600sup.html`

Cooper, S., Fan, I., and Li, G. *"Achieving Competitive Advantage through Knowledge-Based Engineering"* Department of Enterprise Integration, Cranfield University, 2001.

Groeneveld, P. *"Electronic Design Automation, Part 1"*. Technische Universiteit Eindhoven, The Netherlands, 2005.

Guruswamyl, M., and Wong, D. F. *"A General Multi-layer Area Router"*. The University of Texas at Austin, Department of Electrical and Computer Engineering, 1991.

Joobbani, R., Siewiorek D. P. *"WEAVER: A Knowledge-Based Routing Expert"*. Department of Electrical and Computer Engineering, Carnegie-Mellon University, 1985.

Kaas, E. *"The NGRAIN Technology Difference Explained A Whitepaper For Technical Evaluators Of Visualization And Simulation Technologies"*. NGRAIN Corporation. Vancouver, Canada.

Kang, S.-S., Myung, S. and Han, S.-H. "A Design Expert System for Auto-Routing of Ship Pipes". *Journal of Ship Production*, 1999.

Lee, C. Y. "An Algorithm for Path Connections and its Applications". *IRE Transactions on Electronic Computers* vol. EC-10, no. 2, 1961.

Lester, P. *"A* Pathfinding for Beginners"*. Almanac of Policy Issues website. [Online May 2006]. `http://www.policyalmanac.org/games/aStarTutorial.htm`

Lunow, R. E. *"A Channelless, Multilayer Router"*. Lawrence Livamore National Laboratory, California, 1988.

"MIL-W-5088L: Military Specification – Wiring, Aerospace Vehicle". Department of Defense, United Stares. 10 May 1991.

Sadeghi, M. *"Electrical Wiring Practices"*. Federal Aviation Administration. Aircraft Electronics Association Convention, 2003.

Stevenson, A. *"Voxels And Volumetric Representation"*. The University of British Columbia, Vancouver, Canada, 1996.

Tehranipoor, M. *"CAD Algorithms – Routing"*. Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, 2005.

Vakil, D., Zargham, M. R. *"An Expert System for Channel Routing"*. Computer Science Department, Southern Illinois University, 1988.

Van der Velden, C., Bil, C., Yu, X. Smith, A. "Towards A Knowledge Based Cable Router for Aerospace Vehicles". *Information and Knowledge Engineering Conference*, Las Vegas, United States, 2006.

Van der Velden, C., Bil, C., Yu, X. Smith, A. "Mathematical Techniques Applied to Knowledge Based Engineering Design Systems". *Engineering Mathematics and Applications Conference*, Melbourne, Australia, 2005.

Yoshimura, T., and Kuh, E., S. "Efficient Algorithms for Channel Routing". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 1982.

BIOGRAPHY

Christian Van der Velden. Christian received his BE from RMIT University in Aerospace Engineering in 2004. He is currently working on his PhD which focuses on the application of Knowledge Based Engineering (KBE) technology to intelligent design engineering systems. The project aims to develop a knowledge based system for automation of electrical wiring design. The PhD is a joint project with a scholarship involving RMIT University, GKN Aerospace and the Australian Research Council (ARC).

Cees Bil. Cees has a Master of Science and a PhD in aerospace engineering from Delft University of Technology in the Netherlands. He is Associate Professor in Aircraft Design at RMIT University. He is involved teaching and research in aerospace design, computer-aided engineering, stability and control, and design optimization. He is Deputy-Director of the Sir Lawrence Wackett Centre for Aerospace Design Technology at RMIT University. He is chairman of the Melbourne Branch of the Royal Aeronautical Society and Faculty Advisor for the RMIT Student Chapter of the American Institute of Aeronautics and Astronautics..

Adrian Smith. Adrian Smith is a Technical Specialist with GKN Aerospace and is responsible for the development of the Engineering Design Automation capability within the business. Adrian received his BE and PhD from The University of Queensland in Mechanical Engineering. His PhD focused on the analytical prediction and experimental measurement of aerodynamic forces and moments in hypervelocity wind tunnels..

Xinghuo Yu. Prof Xinghuo Yu is the Associate Dean (R & I) of Science, Engineering and Technology at RMIT Australia. His research interests include control systems and intelligent systems. He has published over 300 papers and coedited 9 books. Prof Yu has served as an Assoc Editor of IEEE Trans Circuits and Systems I and IEEE Trans Industrial Informatics. He has been on the program committees of more than 50 conferences, and chaired VSS'00, AI'04, Complex'04, Complex'98, INDIN'05. Prof Yu was the sole recipient of the 1995 Central Queensland University Vice Chancellor's Award for Research. He is also a Fellow of Institution of Engineers Australia.