www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

CONSTRAINT-BASED ADAPTATION FOR COMPLEX SPACE CONFIGURATION IN BUILDING SERVICES

PUBLISHED: October 2009 at http://www.itcon.org/2009/47
EDITORS: Rezgui Y, Zarli A, Hopfe C J

Benachir Medjdoub, Dr
School of the Built Environment, University of Salford, UK
b.medjdoub@salford.ac.uk

SUMMARY: *In this paper an object-based CAD programming is used to take advantage of standardization to handle the schematic design, sizing and layout planning for ceiling mounted fan coil system in a building ceiling void. In order to deal with more complex geometry and real building size, we have used a hybrid approach combining case-based reasoning and constraint programming techniques. Very often, building services engineers use previous solutions and adapt them to new problems. Case-based reasoning mirrors this practical approach and did help us deal effectively with increasingly complex geometry. Our approach combines automation and interactivity. From the specification of the building 3D BIM model, our software prototype proceeds through four steps. First, the user divides the building into zones, each zone being defined by a geometrical primitive (i.e. rectangle zone, triangle zone, curved zone, etc.). Next, for each zone a similar case is retrieved from the case library. The retrieval process will generate a first incomplete 3D solution containing some inconsistencies. Next, the incomplete solution is adapted, using constraint programming techniques, to provide a consistent solution. Finally, distribution routes (i.e. ducts and pipes) are generated using constraint programming techniques. The 3D fan coil solution can be modified or improved by the designer, while providing further contribution by concentrating on interactivity. The project has been funded by the Engineering and Physical Sciences Research Council (EPSRC) in the UK.*

KEYWORDS: *hybrid system, constraint-based adaptation, case base reasoning, design process, interactivity.*

REFERENCE: *Medjdoub B (2009) Constraint-based adaptation for complex space configuration in building services, Special Issue Building Information Modeling Applications, Challenges and Future Directions, Journal of Information Technology in Construction (ITcon), Vol. 14, pg. 724-735, http://www.itcon.org/2009/47*

1. INTRODUCTION

Medjdoub et al. (Medjdoub et al., 2003) have shown that it is possible to define and implement standard solutions to produce designs comparable with the current practice. The main beneficiaries will be engineers, manufacturers, suppliers, building users, and managers. They all stand to benefit from standardised solutions, reduced capital costs and improved performance. This project is concerned with developing a new approach to deal with complex and combinatorial problems (NP-Complete) in building services design and more precisely in space configuration of ceiling mounted fan coil systems. Currently, building services engineers solve these problems "by hand". Starting from the fresh air load, a schematic solution is defined. The engineers proceed to equipment selection, and then equipment location, followed by pipe/duct routing governed by objective requirements (e.g. minimum surface area, minimum pipe length, minimum bend number). In summary ceiling, mounted fan coil system design amounts to *layout configuration* (i.e. space layout of fan coils and diffusers) followed by *duct/pipe routing* in a 3D ceiling void. This project utilises and builds on the results from a previous project "Building Services Standard Solutions implemented in CAD" (Medjdoub et al., 2003, Medjdoub et al., 2001) to make the ceiling mounted fan coil solution useable for more complex geometry. This will bring benefits in terms of increasing the range of applications for which the solutions can be used. The extension to deal with more complex geometry has been based on hybrid approach combining case-based reasoning and constraint programming techniques (Aamodt et al., 1994; Dave et al., 1994; Sqalli et al., 1999). Very often, building services engineers use previous solutions and adapt them to new problems. Case-Based Reasoning mirrors this practical approach and did help us to deal effectively with increasingly complex geometry, where constraint programming approach has been used in the adaptation process and the duct/pipe routing.

In section 2, we discuss the related work. In section 3, we describe the ceiling mounted fan coil standard solutions. The object model of the system and the space configuration process are presented in sections 4 and 5. Before concluding, the implementation and benchmarking are presented in sections 6.

2. RELATED WORK

Computer support for layout configuration has been studied for more than twenty years. Studies include techniques such as mathematical programming (Mitchell et al., 1976), expert systems (Flemming, 1988), genetic evolution (Jo and Gero, 1998; Lee et al., 2002; Osman et al., 2003) and constraint satisfaction (Baykan and Fox, 1991; Aggoun and Beldiceanu, 1992; Charman, 1994). These approaches typically enumerate all placement solutions. Similar solutions, differing only in the precise positioning of an element on the modular grid, are considered as two different geometrical solutions. Clearly, in preliminary design, it is not necessary to distinguish between geometrically close solutions, as this generates a high number of solutions (typically several thousands or millions) which cannot be distinguished in their global aspect by the designer. Another approach based on constraint satisfaction has been presented by Medjdoub and Yannou (Medjdoub and Yannou, 2000) where the topology and the geometry are separated. This brings great flexibility in constraint utilisation since the constraint definition is separated from the resolution algorithms but deals only with mid-order combinatorial problems. Another disadvantage of most of the aforementioned approaches to space configuration is that they attempt full automation of the process. More recent approach in space planning has used case-based reasoning, very often in building services design engineers use previous solutions and adapt them to new design problems; CBR mirror very well this practice. CBR is extremely useful for knowledge acquisition and representation in less-structured domains such as design (Akin, 2002). CBR can be applied in design at different levels. The CBR cycle can be completely or partially developed within an application. For example, many applications use only the retrieval of cases, leaving the user to adapt the case to the new problem, while other applications place less emphasis on retrieval and provide complete adaptation methods. For example, EDAT (Electronic Design Assistant Tool) (Akin et al., 1997) has been developed to help students to retrieve previous building information from a case base by using criteria filter, and perform the analysis of the information to produce new design documents. However, the cases are saved as text, raster images and AutoCAD drawings so they can only be retrieved, the adaptation is done by the user. CBR has become a widely used research tool to solve space layout planning in architecture (Akin, 2002). Such typical case-based applications include ARCHIE II (Domeshek and Kolodner, 1993), FABEL (Voss, 1997), SEED (Flemming et al. 1994). However, the aforementioned CBR approaches have some limitations mainly in the adaptation process when dealing with complex and combinatorial problems. To deal with more complex problem, novel hybrid approaches have been developed combining CBR and genetic algorithm (GA) (Juan et al. 2005). GA techniques have been used at the adaptation process to generate optimal solutions. Another hybrid approach has combined CBR with constraint satisfaction problem approach (CSP) such as CADRE (Hua, 1996) and IDIOM (Smith et al. 1995). CSP approach allows more flexibility in defining the design rules while improving the interactivity of the systems. In our approach we have adopted CBR approach combined with CSP for the case adaptation to overcome the complexity of the problem and to improve the user interaction with the system.

3. FAN COIL SYSTEM STANDARD SOLUTIONS

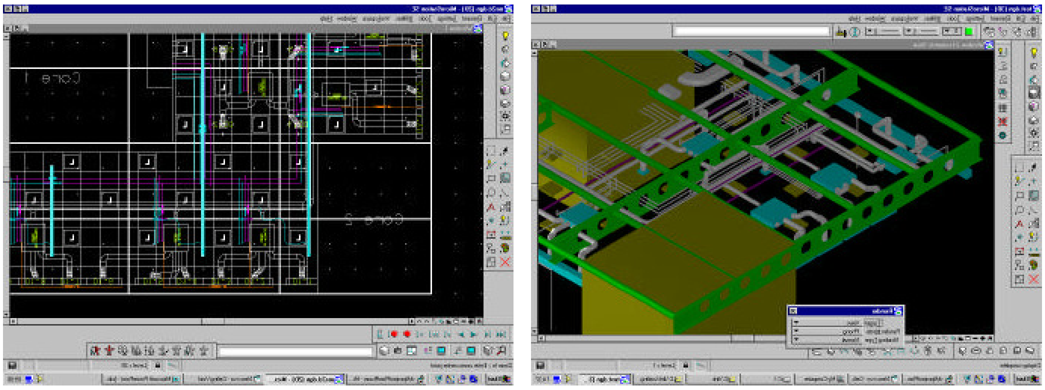


FIG. 1: 2D schematic fan coil system (left) and 3D fan coil system (right)

The ceiling mounted fan coil solution is based on a four-pipe ceiling mounted fan coil unit. Fresh air is provided by a central air-handling unit. Air supply to the space is via slot diffusers in the perimeter zones and the square diffusers internally. For ceiling voids the emphasis is more on integration with non-services elements including beams, ceiling tiles and core areas. It is necessary to define these elements first, before generating the suitable fan coil solution.
Once the non-services elements have been defined, the services are located approximately in order from the most to least geometrically constraint. For the structural solution illustrated (see *FIG. 1*), cellular beams and the distribution runs are located, next as these are constrained to pass through the beam holes. These are then followed in sequence; diffusers selected to provide the required air flow and located to integrate with the lighting; fan coils selected to meet the zone load requirements and located to ensure maintenance access from below; ductwork from fan coils to diffusers; distribution ductwork (for air flow) and pipe work (for cooling water) to the fan coil units (where not routed through beams and columns); condensate runs from the fan coils to column droppers.

The standard solutions were developed in consultation with practicing engineers from Faber and Maunsell one of our industrial partners. There are essentially a collection of rules that address the selection, sizing, location of services equipment (fans and diffusers), the duct work and pipe work in the ceiling void of the building. The rules are in a number of different forms including: Schematics; Written rules; and 2D/3D geometric layouts. Standard fan coil layouts have been defined depending on the geometry of the building (i.e. rectangular, polygon, circular, curved, etc.), the type of the building (i.e. office building, retail, accommodation, etc.), and the building cooling and heating load requirement. The design rules of the four pipes ceiling mounted fan coil system are described in the following table (see *TABLE 1*):

TABLE 1: The design rules of the four pipe ceiling mounted fan coil system

	Type	Rules
General	Locations	The equipment location sequence is generally as follows: (1) Luminaries; (2) Diffusers; (3) Fan coil units; (4) Pipe work; (4) Ductwork Ensure that each equipment has a minimum maintenance space.
Fan coil units	Selection	The units are selected from manufacturers standard range using chilled water for cooling and LPHW for heating.
	Sizing	The following variables are used for selecting the size of fan coil units: Total air volume; Fresh air volume; Cooling required; Heating required; External pressure drop; Noise level A zone could require several fan coil units to meet the cooling and/or heating requirement.
	Location	• Locate the unit 50-100 mm below the slab to allow for slab inconsistencies and slope towards drain exit. • Minimize the distance between the unit and the diffusers to keep pressure drop down. • If the system includes just one unit, this should be located in

		the centre of the zone. If the system includes more then one unit, these units should be located so each unit will provide heating/cooling to a similar surface area. • Avoid grouping intakes as this can cause acoustic problems through reinforcement.
Diffusers	Selection	Perimeter zones: slot diffusers selected from manufacture standard range. Internal zones: square diffusers selected from manufacture standard range.
	Sizing	Use manufacturers sizing algorithms to meet throw pressure drop and noise requirements.
	Location	Located square diffusers to nearest possible location to suit lighting layout. Locate slot diffusers along perimeter.
Ductwork	Selection	Local ductwork runs from fan coils to diffusers circular. Distribution ductwork circular up to 200 mm, flat oval above to limit depth requirement. Use standard ISO/DW144 ranges (DW 144, 1998). Use fittings as defined by BSRIA (BSRIA, 1995) Standard Details project.
	Sizing	Use CIBSE Guide (CIBSE, 1986). Local ductwork from fan coils to diffusers sized as plenum spigot connection subject to a maximum velocity of 3 m/s. 5 m/s maximum velocity limit for ductwork distribution to fan coils.
	Location	Run duct and pipe work headers out from riser and tap off to fan coil units. Run fresh air to 200 mm behind fan coil – better diffusion of air supply into fan coil and avoids balancing problem with fresh air supply. Duct bends should, where possible, be located away from fan coils and diffusers. Route ductwork through cellular beams if possible, otherwise run below beams. Route ductwork down centre of area to be served. Branch supply ductwork local to risers to facilitate crossovers.
Pipe work	Selection	Use fittings as defined by BSRIA (BSRIA, 1995) Standard Details project.
	Sizing	For LPHW and CHW pipe work use CIBSE Guide C (CIBSE, 1986). Base the sizing on 200 Pa/m and never exceed 250 Pa/m. Size condensate at 20 mm from units, 40 mm for 2 or more units and 50 mm in risers.
	Location	Try to run in pairs (side by side) for F&R but not essential. Share commissioning sets if units are adjacent and similar (cost and commissioning benefits). Route Pipes through cellular beams if possible, otherwise run below beams. Run condensate lines @ 1:100 gradients to column droppers.

4. OBJECT MODEL OF CEILING VOIDS SOLUTIONS

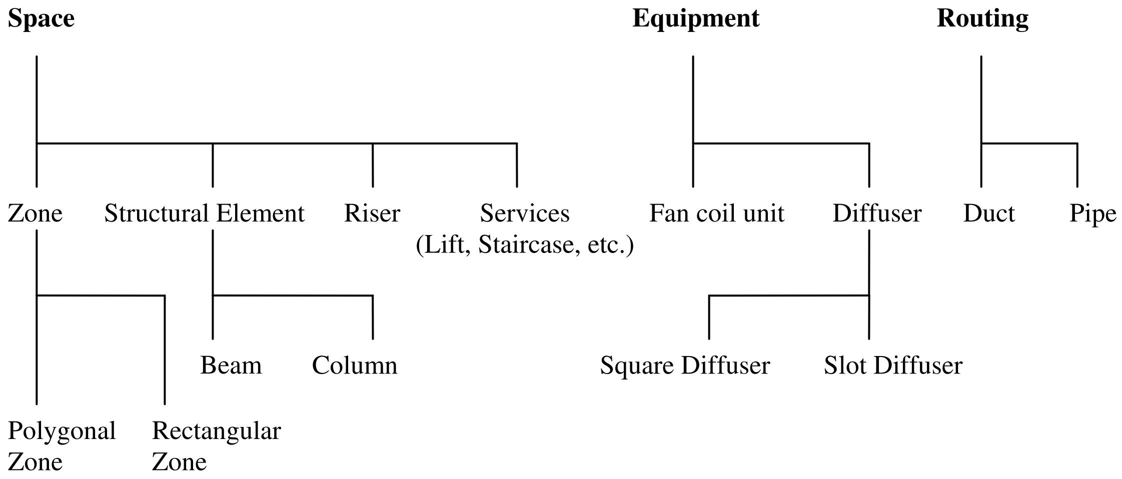


FIG. 2: Object model hierarchy

Ceiling mounted fan coil system design amounts to equipment location (i.e. space configuration of fan coil units and diffusers) followed by duct/pipe routing in the building ceiling void. Our object model holds three main classes representing the ceiling voids space, equipment and duct/pipe (i.e. routing) work. Each defined class is characterised by attributes, methods and class constraints. As indicated in the object structure (see *FIG. 2*), subclasses have been defined from the three generic classes. Thus, from space class we have defined zone class (subdivision of a space), service class, riser class and structural element class. From equipment class we have defined fan coil unit class and diffuser class, and finally from routing class we have defined two subclasses: duct and pipe.

4.1 Space

The space class contains four sub-classes: the zone class, the structural element class (e.g. columns and beams), the risers and the services. The zone geometry can vary from a simple rectangle to an irregular polygon with known obstructions, structural elements and external walls. The attributes of a zone depends on its shape, if the zone is a polygon it will be defined by a set of points *(X[n], Y[n], Z[n])* which represent the polygon vertexes, where *n* equals to the number of all vertexes. The internal structure is characterised by a reference point, its length and its width.

4.2 Equipment

The equipment class (see *FIG. 3*) includes the fan coil and the diffuser classes. These classes are characterised by a reference point (X,Y,Z), a length, a width, a Height and an orientation attribute defined as a discrete constrained variable defined over the domain *{0°, 90°}*. We have defined two types of fan coils; the internal fan coil and the perimeter fan coil. The perimeter fan coil and its slot diffusers (see *Fig. 3* middle) are laid out along the perimeter (window) side, and the internal fan coil and its square diffusers (see *Fig. 3* left) are laid out in the internal area of the floor.

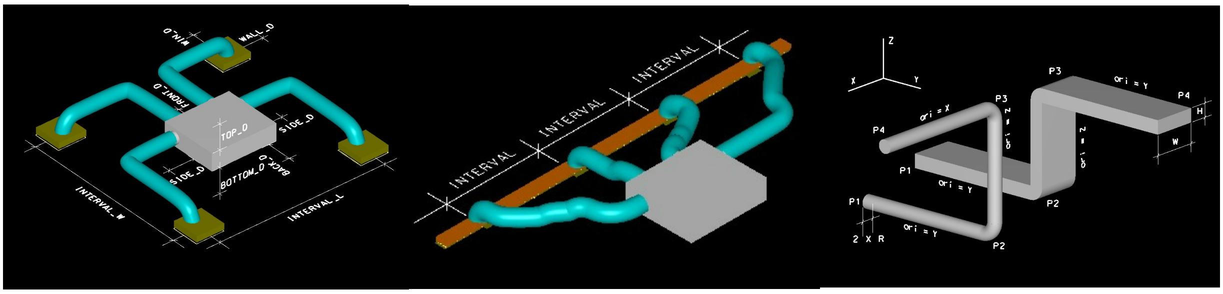


FIG. 3: Internal fan coil and its diffusers (left). Perimeter fan coil and its diffusers (middle). Geometrical representation of duct/pipe with set of 4 points (P1–P4) corresponding to 3 segments, 2 bending angles (right).

4.3 Duct and pipes

The pipe and duct classes represent the ventilation ducts and cooling water pipes. The pipes and ducts classes are defined by a set of points *(X[n], Y[n], Z[n])*, a radius *R* in the case of cooling water pipes or a width *W* and a height *H* in the case of ventilation ducts. Each pair of successive points defines a segment of the pipe or the duct *(i.e.* (P_1P_2), (P_2P_3) *and* (P_3P_4) are the three segments of the pipe shown in FIG. 3 right). An orientation *ori* is associated with each segment. All the attributes are integer constrained variables except the orientation which is a discrete constrained variable defined over the domain *(x°, y°, z°)*. A class constraint is defined to ensure that two successive segments have different orientation.

5. SPACE CONFIGURATION PROCESS

From the 3D specification of the building geometry, our approach proceeds through four steps. First the user divides the building in zones, each zone being a simple geometrical primitive (i.e. rectangle, curved zone, etc.). Next, for each zone a similar case is retrieved from the case library. The retrieval process will generate a first incomplete solution. Next, the incomplete solution is adapted, using constraint programming techniques, to transform it in a consistent solution. Finally, distribution routes (i.e. ducts and pipes) are generated using constraint programming technique. Through the interactive user interface, the 3D ceiling void solution can be modified or improved by the designer.

5.1 Case library

The case library is composed by a set of standard solutions defined with our partners from Faber and Maunsell (Section 3). The cases are stored in a database as objects. Each case is characterized by a set of parameters which cover:

- The geometry of the layout: Our approach could deal with rectangular layout, irregular polygon layout and circular layout.
- The floor function which determines the building cooling and heating load requirement
- Type and size of the fan coils and diffusers: from the floor function and known cooling and heating load requirement a class constraint of the case will instantiate the appropriate type and number of fan coils and diffusers.
- The description of the 3D standard solution is stored as a set of constraints.

5.2 Floor zoning

In the case of a building within a complex geometry, through the user interface the user will divide the building floor area in zones. As indicated in *Fig. 4* the zones are geometrical primitives (e.g. circle, rectangle, triangle, trapezoid, circle, and polygon). The advantage of this method is to divide the complexity of the problem in a set of sub-problems. Each zone will be processed separately and will be considered as a separate case to retrieve and adapt. In the current practice, the engineers in building services use a similar zoning method to approach their design.

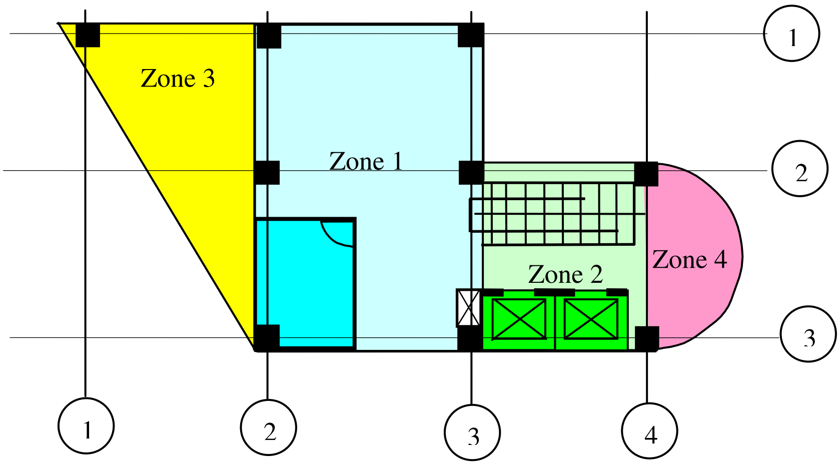


FIG. 4: the floor has been divided into 4 zones, which are assigned different colours

5.3 Case retrieving

After having defined the zones and their attributes, the next step consists of retrieving the most similar case from the case library. Retrieving a case starts with a problem description and ends when a best matching case has been found. As mentioned in Section 3, standard solutions of ceiling mounted fan coil systems corresponding to the basic geometrical floor shapes (rectangle, triangle, trapezoid, circle, etc) are stored in the case library. The match between the floor zones and the retrieved cases is based on the following similarities: the geometry of the zone and the floor function (i.e. office building). Next, from the retrieved case a first inconsistent solution (*FIG. 5*) is generated and will need further adaptation to make it usable.

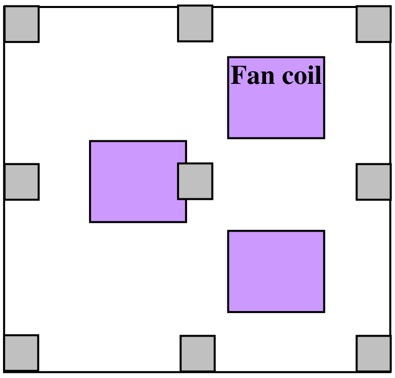


FIG. 5: Incomplete solution generated by the retrieved case, the problem consists of a fan coil overlapping with a column

5.4 Constraint-based case adaptation

The adaptation process is based on constraint programming techniques. The method used is by substitution, where we replace some parts of the old solution that does not fit the current requirements. Thus from the incomplete solution from the case retrieving step, the system will identify the inconsistent parts (i.e. a fan coil overlap with a column). Next this inconsistency is solved applying simple constraints (e.g. inclusion constraint, non-overlapping constraint, dimensional constraint) to make it consistent (Medjdoub et al., 2000; Medjdoub et al., 2001). The advantage of this approach is that each inconsistency is solved separately, which decrease drastically the complexity of the problem. To do so, we have developed an algorithm to check the consistency of the solution using the same set of constraint to adapt the solutions, these constraints are described bellow:

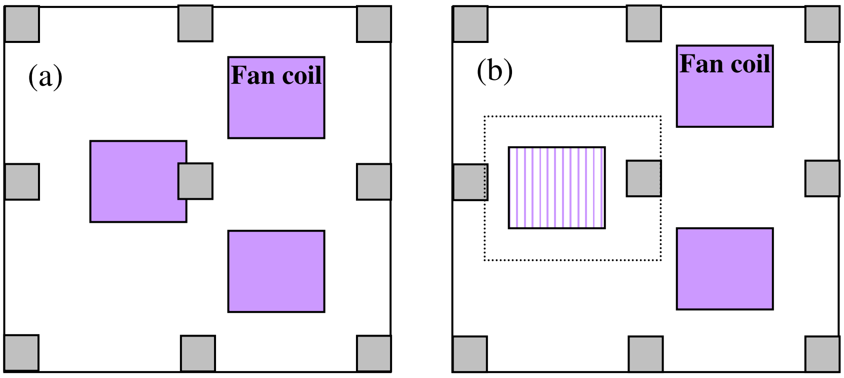


FIG. 6: (a) incomplete solution, (b) solution after local adaptation within the use of a non-overlapping constraint between the fan coil and the column.

Dimensional Constraints: Dimensional constraints assign a minimal or a maximal value to the constrained variables. This constraint is expressed by equality or inequality, i.e. fan.X.position>=100.

Inclusion constraints: Constraints the fans, diffusers and ducts/pipes to be inside the appropriate zone or local area. These types of constraints are important for finding the locations of fan coils and diffusers, where they must be located within their local grid areas.

Non-overlapping constraints: Non-overlapping constraints represent the fact that two objects cannot overlap each other; it is automatically applied to all pairs of objects in our application (i.e. fan coils to structures, fan coils to fan coils, etc.) *Fig. 7* shows the position permitted for e2.x2 and e2.y2[1] by the non-overlapping constraint between element e1 and e2. This constraint is dependent on the minimal space dimension notion. The minimal space dimension is, at any moment, equal to the smallest dimension value (width or length) of all objects. This value is used in order to constrain two spaces to be adjacent, or to be separated by a sufficient distance which allows another object to be inserted in between. The non-overlapping constraints between equipment (i.e fans and diffusers) introduce new constrained variables with a domain including four values { east, west, north, south } as they are all on the same level. Between the pipework and the equipment the domain of these constrained variables include six values {east, west, north, south, above, under}, as the pipework can be located above or under the equipment. These variables divide the space surroundings into four or six parts (see FIG. 7) but not symmetrically. Indeed, the north and south values give more solutions than the east and west values. This asymmetry is made to avoid any redundant solution. It is the instantiation of these non-overlapping variables which, if proven consistent, gives a topological solution. We can consider the following equivalence: non-overlapping (e1,e2)=Adjacent (e1 e2 d1 d2) (1) with $d1 \in [0+\infty]$ and $d2 \in [0 +\infty]$.

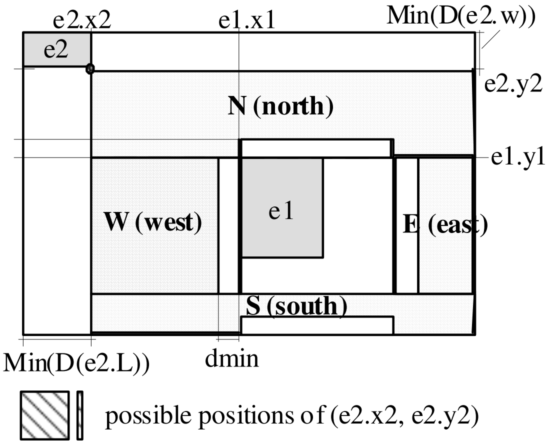


FIG. 7: Positions permitted for point (x2, y2) of e2 after the non-overlapping constraint applied between e1 and e2. The partitioning of the surroundings of e2 in {E,W,N,S} is given.

---

[1] *e2.y2* represents the constrained variable *y2* of element *e2*.

In order to generate the solutions, first we instantiate the non-overlapping constrained variables. This process will generate one topological solution for each sub-problem to adapt. We converge to the following definition of a topological solution: *"Each space layout Constraint Satisfaction Problem (CSP) where the n.(n-1)/2 (n being the number of spaces) non-overlapping variables and adjacency variables are instantiated and which remains geometrically consistent (i.e. for which at least one geometrical solution exists) is a topological solution."* Next, we instantiate the geometrical constrained variables to generate the geometrical solutions,. Finally this process leads to the adaptation of an inconsistent solution to a solution meeting the design requirements.

5.5 Pipe and duct routing

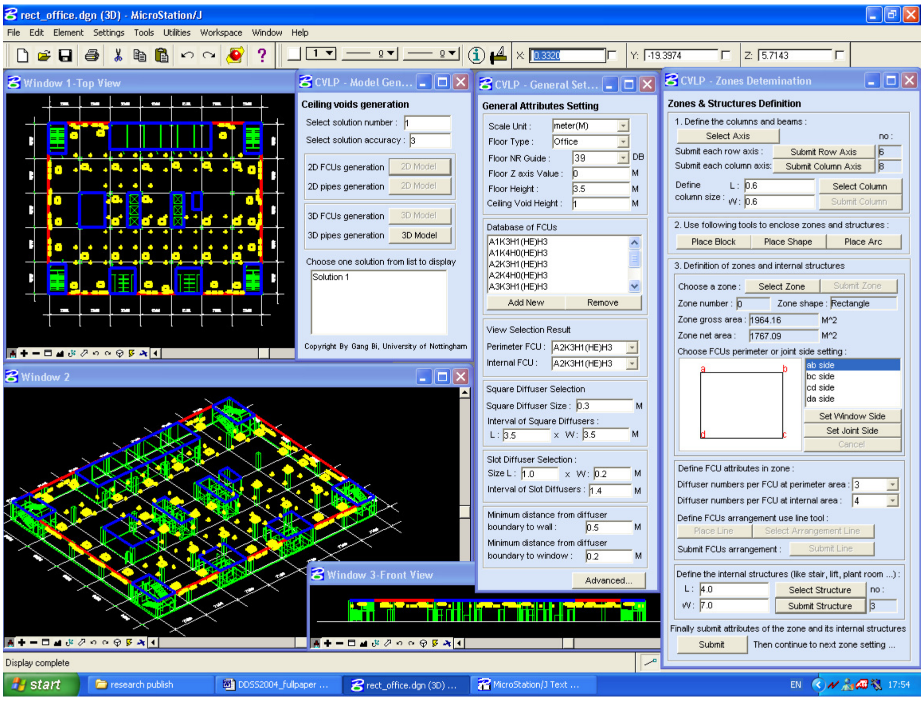
The pipe and duct routing is the last step of the enumeration. The algorithm proceeds in three steps. First the bends number variables are instantiated, the smallest values being chosen first. Then the non-overlapping variables between pipes, ducts, equipment and structural elements are instantiated. Finally, we instantiate the pipe reference points minimising the length of each pipe using the "branch and bound" algorithm (Carpaneto et al., 1995). This is the most common approach used in constraint programming to find the optimal solution. First, we create a constrained variable representing the objective function and find an initial solution, then we introduce a new constraint that the value of the objective variable must be better than in the initial solution. We repeatedly solve the new problem and tighten the constraint on the objective variable until the problem becomes insoluble: the last solution found is then the optimal solution.

6. IMPLEMENTATION & BENCHMARKING

The result of our research has been implemented in a software prototype. It has been developed in JMDL (*Java Modeling Language*) as embedded in MicroStation (Microstation is a trademark of Bentley Systems one of our project partner), and Ilog Jsolver (trademark of Ilog) the constraint programming system. Microsoft ACCESS, the relational database application has been used to store and manage the case library. To communicate between ACCESS and the JAVA programs we have used ODBC data source, which supports the data transfer between SQL database and the system files. We have used MicroStation to hold the object model and for the 3D rendering.

Our approach focuses largely on interactivity. As soon as the system generates the first satisfactory solution from the defined ceiling void geometry, it is possible to make further modification through the interactive user interface (*Fig. 8*). Then the user can modify the location of a fan coil interactively by simply dragging the fan, the system generates a new solution while updating the distribution routes automatically.

The solutions generated by our prototype have been tested against conventional solutions in a benchmarking exercise within our industrial partners (Faber and Maunsell). Thus several real scenario projects have been tested. The results have shown that it is possible to define and implement standard solutions to produce designs comparable with current practice. This benchmarking exercise has underlined many advantages and made some suggestions for further development.



X: 0.3320
Y: -19.3974
Z: 5.7143

Ceiling voids generation

Select solution number : 1
Select solution accuracy : 3
2D FCUs generation
2D pipes generation
3D FCUs generation
3D pipes generation
Choose one solution from list to display
Solution 1
Copyright By Gang Bi, University of Nottingham

General Attributes Setting

Scale Unit : meter(M)
Floor Type : Office
Floor NR Guide : 39 DB
Floor Z axis Value : 0 M
Floor Height : 3.5 M
Ceiling Void Height : 1 M

Database of FCUs
A1K3H1(HE)H3
A1K4H0(HE)H3
A2K3H1(HE)H3
A2K4H0(HE)H3
A3K3H1(HE)H3

View Selection Result
Perimeter FCU : A2K3H1(HE)H3
Internal FCU : A2K3H1(HE)H3

Square Diffuser Selection
Square Diffuser Size : 0.3 M
Interval of Square Diffusers :
L : 3.5 x W : 3.5 M

Slot Diffuser Selection :
Size L : 1.0 x W : 0.2 M
Interval of Slot Diffusers : 1.4 M

Minimum distance from diffuser boundary to wall : 0.5 M
Minimum distance from diffuser boundary to window : 0.2 M

Zones & Structures Definition

1. Define the columns and beams :
no :
Submit each row axis : 6
Submit each column axis: 8
Define column size : L : 0.6 W : 0.6

2. Use following tools to enclose zones and structures :

3. Definition of zones and internal structures
Choose a zone :
Zone number : 0 Zone shape : Rectangle
Zone gross area : 1964.16 M^2
Zone net area : 1767.09 M^2
Choose FCUs perimeter or joint side setting :
ab side
bc side
cd side
da side

Define FCU attributes in zone :
Diffuser numbers per FCU at perimeter area : 3
Diffuser numbers per FCU at internal area : 4
Define FCUs arrangement use line tool :
Submit FCUs arrangement :

Define the internal structures (like stair, lift, plant room ...) :
L : 4.0
W : 7.0
no : 3

Finally submit attributes of the zone and its internal structures
Then continue to next zone setting ...

FIG. 8: User interface panels to fan coil layout

The main advantages are:

- The system deals with complex floor geometry.
- The retrieval process of the similar case is done in reasonable time (less than 1 second).
- The constraint-based adaptation approach is done sequentially which decrease the complexity of the problem.
- The output from the solutions such as the 3D data model is beneficial to other parties in the supply chain.
- The overall time saving is around 50% of the design time.

The main improvements needed:

- To enrich the case library including bspline surfaces.
- To be able to retain a case (new solution) by incorporating it in the case library.
- To improve the duct/pipe routing and more precisely the interactivity of the duct/pipe modification. The designer should be able to modify interactively the position of bends through the 3D interface.

7. CONCLUSION

The approach presented in this paper shows an interactive system for *ceiling mounted fan coil system in a building ceiling void*. The system is simple to use with interactive modification of the 3D parametric model. This approach has shown the potential to significantly reduce design costs by reducing design time by 50%, improve the quality of the solution, and produce additional benefits elsewhere in the supply chain. On the computational part, the integration of CBR and CSP approaches did achieve a synergy, which produces results that could not be obtained if each mode were operating separately. Further developments are being done and concern mainly the case library enrichment and the complex problem of pipe routing.

8. ACKNOWLEDGEMENTS

This project is funded by the EPSRC, and involves three industrial partners including Faber and Maunsell, Bentley Systems and Biddle Air Systems. The author would like to thanks Mr Nick Barnard from Faber and Maunsell and Mr Mike Price from Biddle Air Systems for their valuable help and comments about various aspects of this research project.

9. REFERENCES

Aamodt A. and Plaza E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, AI Com – Artificial Intelligence Communications, Vol. 7, No. 1. p. 39-59.

Aggoun A. and Beldiceanu N. (1992). Extending CHIP in Order to Solve Complex Scheduling and Placement Problems, Journées françaises de la programmation logique, Marseille.

Akin Ö. (2002). Case-based instruction strategies in architecture, Design Studies 23 (2002), Elsevier Science, p.407-431, 2002.

Akin Ö., Cumming M., Shealey M. and Tuncer B. (1997). An electronic design assistance tool for case-based representation of designs, Elsevier, Automation in Construction 6, p..265-274, 1997.

Baykan C. and Fox M. (1991). Constraint Satisfaction Techniques for Spatial Planning, Intelligent CAD Systems III, Practical Experience and Evaluation.

BSRIA. (1995). Rules of thumb, Technical Note TN17/95, UK.

Carpaneto G. Dell'Amico M. and Toth P. (1995). A Branch-and-bound Algorithm for Large Scale Asymmetric Travelling Salesman Problems, ACM Transactions on Mathematical Software 21, p. 410-415.

Charman Ph. (1994). Une approche basée sur les contraintes pour la conception préliminaire des plans de sol, PhD report, CERMICS-INRIA, France.

CIBSE Guide C. (1986). Reference Data, Section C4: Flow of Fluids in Pipes and Ducts, CIBSE, UK.

Dave B., Schmitt G., Faltings B. and Smith I. (1994). Case based design in architecture, Artificial Intelligence in Design – AID' 94, Kluwer Academic, p. 145-162.

Domeshek E. A. and Kolodner J. L. (1993). Using the points of large cases", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 7(2), p.87-96.

DW 144. (1998). Specification for sheet metal ductwork – Low, medium and high pressure/velocity air systems, Published by the Heating and Ventilating Contractors Association (HVCA), UK.

Flemming U. (1988). A generative expert system for the design of building layouts, Artificial Intelligence in Engineering: Design, Ed. Elsiever, New-York.

Flemming U., Coyne R. and Snyder J. (1994). Case-Based Design in the SEED System, in Computing in Civil Engineering. Vol. 1, Proceedings of the First Congress held in conjunction with the A/E/C Systems `94, Washington, DC, p.446-453.

Hua K., Faltings B. and Smith I. (1996).CADRE: case-based geometric design, Artificial Intelligence in Engineering 10, Elsevier, p.171-183.

Jo J.H. and Gero J.S. (1998). Space Layout Planning Using an Evolutionary Approach, Artificial Intelligence in Engineering, 12(3), p. 149-162.

Juan Y. K., Shih S. G. and Perng Y. H. (2005). Decision support for housing customization: A hybrid approach using case-based reasoning and genetic algorithm, Elsevier, Expert Systems with Applications, 1-11.

Lee Y. H. and Lee M. H. (2002). A shape based block layout approach to facility layout problems using hybrid genetic algorithm, Computers and Industrial Engineering, 42, p.237-248.

Medjdoub B. and Yannou B. (2000). Separating topology and geometry in space planning, Computer Aided Design, 32(1), Elsevier, p. 39-61.

Medjdoub B. and Richens P. (2001). Building Services Standard Solutions: Variational Generation of Plant Room Layouts, CAADfutures, Kluwer Academic Publishers, p. 479-493.

Medjdoub B. and Yannou B. (2001). Dynamic space ordering at a topological level in space planning, Artificial Intelligence in Engineering, 15, Elsevier, p. 47-60.

Medjdoub B., Richens P. and Barnard B. (2003). Generation of Variational Standard Plant Room Solutions, Automation in Construction Journal, 12(2), Elsevier, p. 155-166.

Mitchell W.J., Steadman J.P. and Liggett R.S. (1976). Synthesis and Optimization of a Small Rectangular Floor Plans, Environment and Planning B, (3): p. 37-70.

Osman H. M., Georgy M. E. and Ibrahim M. E. (2003). A hybrid CAD-based construction site layout planning system using genetic algorithms", Elsevier, Automation in Construction 12, p. 749-764.

Smith I., Lottaz C. and Faltings B. (1995). Spatial composition using cases: IDIOM, Lecture Notes In Computer Science; Vol. 1010, p. 88-97.

Sqalli M.H., Purvis L. and Freuder E.C. (1999). Survey of Applications Integrating Constraint Satisfaction and Case-Based Reasoning, Paper presented at PACLP99: The First International Conference and Exhibition on the Practical Application of Constraint Technologies and Logic Programming, 19-21 April, London, UK.

Voss A. (1997). Case design specialists in FABEL, In: Maher, M.L., Pu, P., editors. Issues and applications of CBR in design, London: Lawrence Erlbaum, p. 301-335.