COMPUTATIONAL DESIGN PROCESS MODELING

Marin D. Guenov*, Libish*, Dunbing Tang*, Helen Lockett*
*School of Engineering, Aerospace Engineering Group, Cranfield University,
Bedfordshire, MK43 0AL, UK
Tel: 44-1234-754612, Fax: 44-1234-758203
Email: [M.D.Guenov, Libish.2003, D.Tang, H.Lockett]@cranfield.ac.uk

Keywords: *Incidence Matrix, Design Structure Matrix, Computational Plan, Conceptual Aircraft Design, Process Modeling*

Abstract

In the conceptual design phase, relatively simple equations and functions (or compiled code) are used to describe the aircraft and to perform trade-off studies. The latter require an optimal execution sequence in order to reduce computational cost and design time, respectively. The focus of this paper is the dynamic derivation of the optimal computational plan for each study so that the designer could focus on designing the aircraft rather than managing the process flow. Two methodologies, the Design Structure Matrix (DSM) and the Incidence Matrix are used for the computational process modeling. The incidence matrix describes the relationship between variables and equations/models. The DSM has been used to express the dependency relationships between the models and also, after manipulation, to produce the solution process. The designer specifies the independent (known) variables first. Then the variable flow is modeled using the Incidence Matrix Method (IMM). It determines how data flows through the models, and also identifies any strongly connected components (SCCs). The second step is to rearrange all equations/models hierarchically in order to reduce the feedback loops in each of the identified SCCs. This is achieved by the application of a genetic-based algorithm. Subsequently all SCCs and non-coupled models are assembled into a macro model which forms a global DSM. The global DSM is further rearranged to obtain an upper triangular matrix which defines the final model execution sequence. A simple aircraft sizing example is presented to illustrate the proposed method and algorithm. Advantages of the method include improved efficiency and the ability to deal with both algebraic and numerical models as well as with multiple outputs per model.

1. Introduction

The decisions taken during the conceptual design phase commit the majority of the aircraft lifecycle costs, but also offer the greatest opportunity for innovation. The latter depends to a great extent on the ability to explore a large number of novel configurations in a relatively short space of time. Improving the conceptual design process in this respect involves several issues. In the first place, it should allow the starting point of the design study to depart from an existing configuration, otherwise the final result may end up being very similar to the original. Such freedom is often limited in practice due to the fact that many assumptions related to traditional designs may have already been hardwired into the existing compiled codes. Secondly, a greater flexibility of the computational process is needed in terms of what is considered an input or an output variable. This should depend solely on the objectives of the study. Such flexibility requires a process which would combine bottom-up composition of possibly hundreds of equations and models or black boxes (compiled chunks of modular code). These represent parametric geometry and layout configuration, aerodynamic performance, propulsion, flight dynamics and so forth. Following this, the process needs to perform a top-down hierarchical decomposition for computational process modeling and definition. Last, but not least, since any design study produces a massive amount of data, there is a need for a mechanism capable of reproducing the data derivation process for future use.

The overall aim of our work has been to provide a flexible workflow which satisfies the above stated needs. The focus of this paper is a novel approach for design computation process modeling as part of the overall effort. The text is organised as follows. The main process issues and state of the art in workflow (process) management are outlined in section 2. Our novel concept for a design computational process model is presented in section 3. The variable flow modelling and the identification of strongly connected components (SCC) are described in section 4, while section 5 contains the description of SCCs rearrangement using a genetic algorithm. A detailed example is presented in section 6 and finally conclusions and future work are outlined in section 7.

2. Workflow (Process) Management in Conceptual Design

There are several process issues related to the successful implementation of a workflow management device, including:

- Identification of the models and equations required for the analysis/trade studies,
- Selection of input and output variables,
- Identification of SCC, that is, a subset of the set of equations and models which are coupled through shared variables and cannot be solved without iteration
- Solving the SCCs
- Assembling the workflow

The rest of this section briefly describes the state-of-the-art in the development of enabling methods and technologies for workflow (Process) management systems.

The bipartite graph method is one of the most widely used methods for variable flow modeling. Fertig and Smith [1]-[3] describe a tool named 'Design Sheet' which was developed for facilitating flexible trade-off studies during conceptual design. Design Sheet utilises the bipartite graph method and is specifically used for solving a set of algebraic equations. The Bipartite graph method provides a decomposed solution for a set of algebraic equations and inequalities, providing also flexibility in choosing the independent variables; however, it can be used only for equations and not for models which may have more than one output. Rogers [4] developed a knowledge-based tool 'DeMAID' for decomposition of complex design problems. 'DeMAID' applies knowledge-based method on an NxN matrix for reordering the design processes. The NxN matrix representation was developed earlier by Steward [5] to organize and display the interactions among the processes. However, 'DeMAID' does not deal with variable flow modeling such as identification of input and output variables. Kusiak and Wang [6] have used an incidence matrix to model the relationship between design parameters (data variables) and processes (models). The columns of this matrix represent the design parameters while the rows represent the models. A matrix element denoted by '*' indicates a relationship between a variable and a particular model. This tool minimizes the interdependency among the sub-processes which in turn enhances concurrency of the design process. Accordingly a problem is decomposed to mutually exclusive sub-processes by reordering the rows and columns. Chen and Li [7] have demonstrated an algorithm which can verify the decomposability and complexity of a design problem. The algorithm achieves an optimal number of sub-problems during decomposition which was usually determined using trial and error method. In their earlier work [8] they have proposed a formal two-phase decomposition method for complex design problems that are represented in an attribute-component incidence matrix. Although their research does not consider input and output variables or an approach for identifying and solving SCCs, the incidence matrix method has

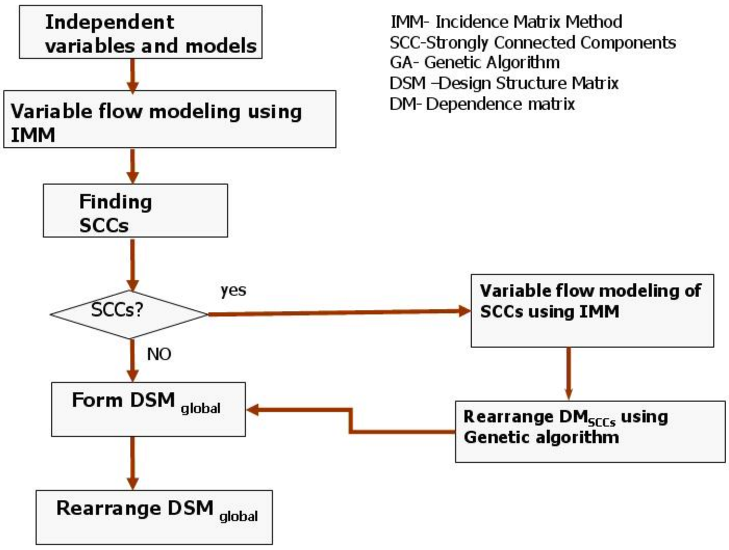


Fig. 1. Flow diagram of computational design process

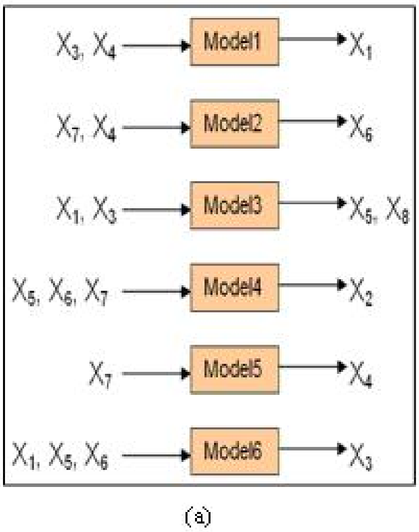
been used as a foundation for the development of the variable flow modeling presented in this work.

3. The Computational Design Process

The flow diagram shown in figure 1 depicts the concept behind the computational design process model. It combines two methodologies, the design structure matrix [5] and the incidence matrix [6]-[8]. The incidence matrix describes the relationship between variables and equations/models. The DSM has been used to express the dependency relationships between the models and also, after manipulation, to produce the solution sequence. The process starts with the selection of a set of equations and/or models representing single discipline analysis (or parts thereof) with the aim of assembling these in a workflow for multidisciplinary aircraft analysis or trade-off studies. The designer then specifies the set of independent, i.e., known variables which will be used as inputs to the solution process. (The case where the number of selected independent variable resulting in an over or under determined system of models is not considered here). Following this, the variable flow is modeled using the Incidence Matrix Method (IMM). It determines how the data flows through the models, and also identifies any SCC. SCCs result from the coupling of models through shared variables. We propose an algorithm for identifying SCCs. The second step is to arrange all equations/models hierarchically in order to reduce the feedback loops in each of the identified SCCs. We apply a GA-based algorithm for resolving the couplings. Subsequently the SCCs are grouped together to form a global design structure matrix. Using Tang's DSM rearrangement algorithm [9], the global DSM is further rearranged to obtain an upper triangular matrix which defines the model execution sequence. Finally appropriate mathematical treatments are applied on the models in which inputs and outputs are swapped. Iteration treatments are applied on the SCCs for solving. The following sections explain the IMM, SCC identification, as well the processes for SCC rearrangement and solution in detail.

4. Variable Flow Modeling and SCC Identification

Variable flow modeling is the process of identifying how the information should flow throughout the models, in order to calculate those variables which are not selected as independent. As mentioned earlier, in order to



	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Model1	*		*	*				
Model2				*		*	*	
Model3	*		*		*			*
Model4		*			*	*	*	
Model5				*			*	
Model6	*		*		*	*		

(b)

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Model1	*		i	*				
Model2				*		*	*	
Model3	*		i		*			o
Model4		o			i	i	i	
Model5				*			*	
Model6	*		i		*	*		

(c)

Fig. 2. (a) Test case; (b) Initial incidence matrix for the test case; (c) Propagated incidence matrix

calculate certain values in a particular analysis or a trade-off study, sometimes models have to be solved in reverse order, which implies that some of the outputs of the models have become inputs and vice versa. This process can be very laborious in practice where hundreds of models may need to be assembled for a particular study. This section describes the incidence matrix method for efficient variable flow modeling and SCC identification.

An Incidence matrix for a study has models in the rows and variables in the columns. The association of a variable in the column with a model in the row is denoted by a '*' marked in the corresponding cell. Solving the incidence matrix demands substituting these *`s in each cell with either 'i' (input) or 'o' (output) depending on whether the variable in the column should be an input or output of the model in the row. This solving is based on the heuristic rules stated below.

Rule 1

Independent variables should be always input to models.

Rule 2

If a variable is associated only with one model and if it is not an independent variable then it should definitely be output of that model.

Rule3

If a model is associated with only one variable then that variable should definitely be an output of that model.

Rule 4

Each variable should be output of only one model.

Rule 5

Each model in the variable flow modeling process should produce the same number of outputs as the number of equations or functions embedded in the model.

Figure 2 shows an example which illustrates variable flow modeling and SCC identification. The input and output variables of the models are shown in figure 2(a), and the initial incidence matrix of the case is illustrated in figure 2 (b). Figure 2(c) shows the final populated incidence matrix, obtained with X_3 as the independent variable after applying the five rules stated above.

The models, i.e. the rows in figures 2(c), containing '*' in the final updated incidence matrix will form a SCC. Thus in the above example, models 1,2,3,5 and 6 form a SCC. These are coupled through data variables X_1, X_4, X_5, X_6, X_7. A design problem can have more than one mutually exclusive SCC. In such a case each SCC will be considered as an

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Model1	*		i	*				
Model2				*		*	*	
Model3	*		i		*			o
Model5				*			*	
Model6	*		i		*	*		

(a)

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Model1	i		i	o				
Model2				i		o	i	
Model3	i		i		o			o
Model5				i			o	
Model6	o		i		i	i		

(b)

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Model1	i		i	o				
Model2				i		o	i	
Model3	o		i		i			o
Model5				i			o	
Model6	i		i		o	i		

(c)

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Model1	o		i	i				
Model2				*		i	*	
Model3	i		i		o			o
Model5				*			*	
Model6	i		i		i	o		

(d)

Fig. 3. (a) Incidence matrix of the SCC; (b) Incidence matrix of SCC with X_1 chosen as output for model6; (c) Incidence matrix of SCC with X_5 chosen as output for model6; (d) Incidence matrix of SCC with X_6 chosen as output for model6

aggregated model with its own input and output variables through which it will be linked with other models or SCCs in the design study.

The next step is to resolve the local incidence matrices corresponding to the SCCs for obtaining the complete solution. Note that only SCCs are considered for further solving. Figure 3(a) shows the incidence matrix for the SCC identified above. Solving of the incidence matrix for a SCC is performed by choosing a '*' cell as output from any one of the corresponding models (rows) in the SCC. This is done according to a new axiomatic rule defined as follows:

Rule 6

In the incidence matrix of a SCC always those models should be chosen for further solving in which inputs are defined different from the inputs of the embedded equations or functions in the model. If such a case doesn't exist then the incidence matrix could be populated with inputs and outputs the same as those of the embedded equations or functions of each model of the SCC.

In figure 3(a), X_3 is input to model 6, but the embedded equation/function of model6 has X_3 as output (referring to figure 2(a)). According to rule 6, model 6 is chosen for further solving. In model 6, either X1, X_5 or X_6 can be chosen as the preferred output for further solving. Solutions obtained for each case are shown in Figure 3 (b), (c) and (d) respectively.

It can be noted in figure 3 (d) that no further solution is possible. Thus either model 2 or model 5 has to be chosen for further solving. According to axiomatic Rule 6, model 2 is chosen. Thus either X_4 or X_7 of model 2 can be further chosen as output and consequently two solutions are obtained as shown in figure 4(a) and 4(b).

There are now in total four solutions for the current example. The question then is which one of the four solutions is the best one? The following section discusses how to choose the best solution with the best rearrangement of the SCCs.

5. Rearrangement of SCCs

In aircraft conceptual design, solving of SCCs is one of the major challenges during trade-off and optimisation studies. Due to the inherent feedback loops, solving a SCC requires iteration. The more feedback loops the more time is required for solving. Thus reducing the feedback loops can reduce the time and cost for solving a SCC. Rearranging the models in a SCC can reduce the feedback loop considerably. In our approach, a permutation

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Model1	o		i	i				
Model2				o		i	i	
Model3	i		i		o			o
Model5				i			o	
Model6	i		i		i	o		

(a)

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Model1	o		i	i				
Model2				i		i	o	
Model3	i		i		o			o
Model5				o			i	
Model6	i		i		i	o		

(b)

Fig. 4. Incidence matrix of SCC with X_6 and furtherX_4 and X_7 chosen as output

	Model1	Model2	Model3	Model5	Model6
Model1	0.5	1	0	1	0
Model2	0	0	0	0	1
Model3	0	0	0	0	1
Model5	0	1	0	0.5	0
Model6	1	0	1	0	0.5

(a)

Genetic algorithm for rearrangement

	Model1	Model5	Model2	Model6	Model3
Model1	0.5	1	1	0	0
Model5	0	0.5	1	0	0
Model2	0	0	0	1	0
Model6	1	0	0	0.5	1
Model3	0	0	0	1	0

(b)

Fig. 5. (a) Initial dependence matrix of SCC; (b) Rearranged dependence matrix of SCC

based genetic algorithm [10] is used for rearrangement of SCC. Genetic algorithms have a major advantage compared to other optimisation methods, in that the former are independent of problem formulation. In this way different objective functions can be formulated for different scheduling architectures.

Currently feedback length is considered as the fitness function to be optimised by the genetic algorithm for the SCC rearrangement. The equation formulated for calculating the feedback length is given in equation 1 below:

$$J = \sum_{i=2}^{n}\sum_{j=1}^{i-1} DM(i,j)*(i-j) + \sum_{k=1}^{n} DM(k,k) \qquad (1)$$

Here *DM* is the dependence matrix created from the incidence matrix of the SCC, *J* is feedback length, and *n* is the number of models.

Considering the previous example with X_3 as the independent variable, the SCC incidence matrix in figure 3(b) (which is one of the four solutions obtained) is then converted to dependence matrix (DM) as shown in figure 5(a). The marker '1' appearing in a cell denotes *DM(i,j)*=1, that is, the model in the corresponding column needs an input from the model in the corresponding row, otherwise *DM(i,j)*=0. The mark '1' when appearing above the main diagonal of the matrix symbolizes a feed forward loop while the '1's below the main diagonal represent feedback loops. In addition, a '0.5' marked on a main diagonal element denotes that the corresponding model has its inputs and outputs reordered as a result of variable flow modeling, i.e., *DM(k,k)*=0.5, otherwise *DM(k,k)*=0. This signifies the additional time required for solving these models. In summary, feedback length '1' indicates coupling between two adjacent models, while '0.5' is currently chosen for models whose inputs and outputs are swapped, and would therefore require inner iterations to solve.

The feedback length calculated using equation 1 for the dependence matrix shown in figure 5(a) is 9.5.

	Model1	Model3	Model5	Model2	Model6
Model1	0.5	0	1	1	0
Model3	1	0.5	0	0	1
Model5	0	0	0.5	1	0
Model2	0	0	0	0	1
Model6	0	1	0	0	0.5

(a) Feedback length=6 (For incidence matrix from figure 3(b))

	Model1	Model3	Model6	Model2	Model5
Model1	0	1	1	0	0
Model3	0	0	1	0	0
Model6	0	0	0.5	1	0
Model2	1	0	0	0.5	1
Model5	0	0	0	1	0.5

(b) Feedback length=5.5 (For incidence matrix from figure 4(a))

	Model1	Model3	Model5	Model2	Model6
Model1	0	1	0	0	1
Model3	0	0	0	0	1
Model5	1	0	0	1	0
Model2	0	0	1	0.5	0
Model6	0	0	0	1	0.5

(c) Feedback length=5 (For incidence matrix from figure 4(b))

Fig. 6. Rearranged dependence matrix for each solution obtained from incidence matrix

The Genetic algorithm is then applied with feedback length as the fitness function. Final feedback length obtained after optimisation is 5.5 and the rearranged dependence matrix is shown in figure 5(b).

In the example considered in Section 4 to demonstrate the incidence matrix method for solving a SCC, we have obtained three more solutions along with the one shown in figure 3(b). Figure 6 shows the final results obtained for the other three cases (figure 3(c), 4(a), 4(b)) after converting the corresponding SCCs incidence matrixes into dependence matrixes and rearranging these matrixes using genetic algorithm with the fitness function from equation 1.

From the four results obtained, the one with the least feedback length is the solution obtained from the dependence matrix in figure 6(c), which has 5 as the feedback length. Therefore, the final execution sequence of SCC after rearrangement is: model 1, model 3, model 5, model 2 and then model 6, as shown in figure 6 (c).

6. Example

A simplified set of aircraft sizing equations from [11] is considered for testing the computational process management concepts developed in this research. The equations are as follows:

$W_e=W_o*2.61*W_o^{(-0.1)}*(W_o/S_{ref})^{(-0.05)}$ (a)
$W_o=W_f+W_e$ (b)
$W_{alt}=0.985*W_{LO}$ (c)
$W_x=0.995*W_{ec}$ (d)
$W_f=1.06*(1-W_x/W_o)*W_e$ (e)
$W_{LO}=0.97*W_o$ (f)
$W_{ec}=Exp[0.00043R]*W_{alt}$ (g)

Where:

W_e - the empty weight,
W_O - the gross take off weight,
S_{ref} - the wing area,
R - the range,

W_* - weight of aircraft at each position of the mission shown in figure 7.

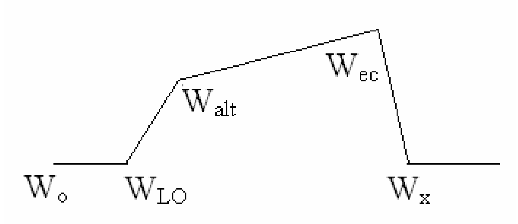
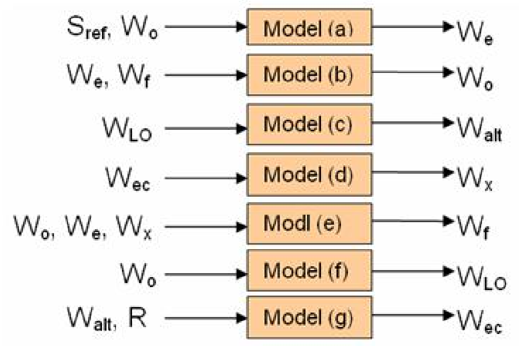


Fig.7. Mission profile for small plane sizing problem

The equations are compiled and formed as models. The inputs and outputs for each model are shown in figure 8(a). Figure 8(b) shows the result of the variable flow modeling on the incidence matrix with *R* and W_{alt} as independent variables. It can be seen that models b and e form a SCC. For this SCC, there are two further solutions shown in figure 9.

In figure 9, the incidence matrix of SCC is converted to its corresponding dependence matrix. For solution 1, model b's inputs and outputs are swapped and therefore it has a value of 0.5 in the diagonal element. In the same way for solution2, both models have 0.5 in their diagonal elements. For both solutions, rearrangement has not reduced the feedback length. Feedback length of solution1 is less than that of solution2, and thus solution1 is chosen for the SCC solving. The SCC is merged together with the remaining models to form the global DSM shown in figure 10(a). Using Tang's algorithm [9], the rearrangement of the global DSM in the upper-triangular form is obtained in figure 10 (b). Therefore the final model computation sequence is either g→ c→ f→ d→ b→ e→ a or g→ c→ f→ d→ e→b → a.



(a)

	S_{ref}	W_o	W_e	W_f	W_{lo}	W_{alt}	W_{ec}	W_x	R
a	o	i	i						
b		i	*	*					
c					o	i			
d							i	o	
e		i	*	*				i	
f		o			i				
g						i	o		i

(b)

Fig. 8. (a) Models for simplified aircraft sizing problem; (b) Propagated incidence matrix

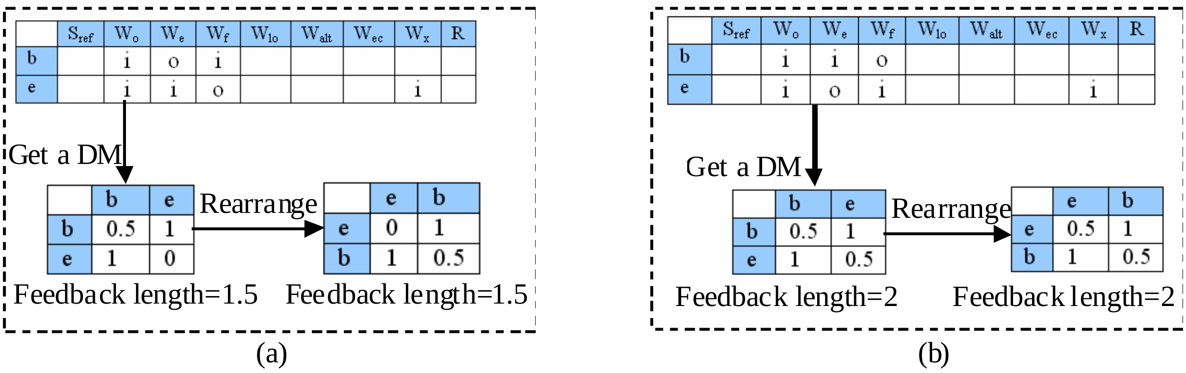


Fig. 9. (a) Incidence matrix and dependency matrix (DM) of SCC for solution1; (b) Incidence matrix and dependency matrix (DM) of SCC for solution2

	a	c	d	f	g	SCC
a	x					
c		x		x		
d			x			x
f	x			x		x
g			x		x	
SCC	x					x

(a)

	g	c	f	d	SCC	a
g	x			x		
c		x	x			
f			x		x	x
d				x	x	
SCC					x	x
a						x

(b)

Fig. 10. (a) Initial global DSM; (b) Rearranged global DSM

7. Conclusions

A novel workflow management concept has been presented with emphasis on the process management aspects of the system. It incorporates a novel variable flow method based on the incidence matrix concept. Unlike other existing methods the variable flow method can handle multiple outputs of a particular constituent model as well as the identification of strongly connected components in the entire multidisciplinary set of models and equations describing the aircraft. Traditionally, variable flow modeling and rearrangement of SCC are performed separately and the only link is the transfer of variable flow model results to the rearrangement process. Our novel approach based on the incidence matrix method is capable of exploring a number of feasible variable flow models according to the objectives of the particular design study. Furthermore all these variable flow models are subjected to rearrangement thus obtaining the solution strategy with the shortest feedback length.

Currently the focus of our research is on the design process modeling and management for trade-off studies. Future work will concentrate on process management for multidisciplinary optimisation and especially on the decomposition of large design problems to sub problems. The advantages are that smaller sub-problems can be easily managed and also could be run in parallel. In the second place, our genetic algorithm for rearrangement of SCCs currently uses feedback length as the fitness function to be optimised. Feedback length does give an approximate idea about the complexity of feedback loops in SCC. However, it does not consider the execution time for each model, number of iterations required for solving an SCC, crossed and embedded iterative loop, etc., which could potentially make significant impact on the computational time and cost of a study. Therefore an improved fitness function for rearrangement will be developed which will acknowledge the above mentioned factors

Acknowledgements

This research was conducted as part of VIVACE (Value Improvement through a Virtual Aeronautical Collaborative Enterprise)-integrated project FP6-502917, partly sponsored by the Sixth Framework Programme of the European Community under priority 4, “Aeronautics and Space”.
We wish to thank the anonymous reviewers for their helpful comments.

References

[1] Buckley M, Fertig K and Smith D. Design Sheet: An environment for facilitating flexible trade studies during conceptual design. *AIAA 92-1191 (1992 Aerospace design conference)*, American Institute for Aeronautics and Astronautics, Washington D.C., 14p, 1992.

[2] Reddy S, Fertig K and Smith D. Constraint management methodology for conceptual design tradeoff studies. *Proceedings of the 1996 ASME design engineering technical conferences*, California, 1996

[3] Reddy S and Fertig K. Managing function constraints in design sheet. *Proceedings of 1998 design engineering technical conferences*, Georgia, 1998.

[4] Rogers J, Salas O and Weston P. A Web-based Monitoring System for Multidisciplinary Design Projects. *Seventh AIAA/NASA/ISSMO symposium on multidisciplinary analysis and optimisation*, St. Louis, Missouri, NASA/TM-97-206287, pp.18, September, 1998.

[5] Steward D. *Systems analysis and management: structure, strategy and design*. 1st edition, Petrocelli Books Inc., 1981.

[6] Kusiak A and Wang J. Decomposition of the Design Process. *ASME Transactions: Journal of mechanical design*, Vol. 115, No. 4, pp. 687-695, 1993

[7] Chen L and Li S. Analysis of decomposability and complexity for design problems in the context of decomposition. *Journal of Mechanical design, Transactions of the ASME*, Vol. 127, No. 4, pp 545-557, 2005.

[8] Chen L, Ding Z and Li S. A formal two-phase method for decomposition of complex design problems. *Journal of Mechanical Design, Transactions of the ASME*, Vol. 127, No.2, pp 184-195, 2005.

[9] Tang D, Zheng L and Li Z. Re-engineering of the design process for concurrent engineering, *Computers and Industrial Engineering*, Vol. 38, No. 4, pp 479-491, 2000.

[10] Altus S, Kroo M. and Gage J. A genetic algorithm for scheduling and decomposition of multidisciplinary design problems, *Journal of Mechanical Design, Transactions of ASME*, Vol. 118, pp 486-489, 1996.

[11] Raymer D, *Aircraft Design: A Conceptual Approach.* AIAA Educational series, New York, NY, 1999.