Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

NLR-TP-99029

The information and communication technology's contribution to the MDO project

M.E.S. Vogels, S.E. Allwright, M. Stettner, Ph. Sims and P. Bartholomew

Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

NLR-TP-99029

The information and communication technology's contribution to the MDO project

M.E.S. Vogels, S.E. Allwright*, M. Stettner**, Ph. Sims* and P. Bartholomew***

* *British Aerospace*
** *Daimler Benz Aerospace*
*** *Defense Evaluation and Research Agency*

This report is based on a presentation held at the Royal Aeronautical Society Conference on "Multidisciplinary Design Optimisation", London (UK), 26-27 October 1998.

Division: Information and Communication Technology
Issued: March 1999
Classification of title: unclassified

Abstract

When an enterprise embarks on an initiative to exploit multi-discipline analysis, design, and optimisation (MDO), the enterprise faces, next to the aeronautical challenge, challenges with respect to Information and Communication Technology to support the aeronautics work.
Experiences in the Brite-Euram MDO project, executed by 14 European aeronautics companies, institutes, and universities, have led to the identification of three key ICT areas.
Firstly, the process followed by the MDO team, being evolutionary by nature, requires the MDO process to be documented and, of course, communicated in a manner understandable and accessible for the MDO team. MDO results can only be interpreted in relation to the process in which they were generated.
Secondly, the specialists from the various disciplines need to work on *the same* design, and within *the same* timeframe, to bring together their result to come to a next design. This requires the path from the a/c specification to analysis, design, and optimisation models to be automated.
Finally, the information, tools and the operation of the enterprise’s computer network have to be arranged to form an effective and friendly environment for the MDO team to operate within. In the paper, highlights of the MDO project in the three key ICT areas are presented.

Contents

(15 pages in total)

1 Introduction

In this paper, the MDO technology is viewed from Information and Communication Technology (ICT) perspective - focusing on the unique ICT challenges an enterprise faces when it embarks on an initiative to exploit MDO.
One of the key issues is the involvement of specialists from different disciplines in the same timeframe. The requirements are different from both the predesign and the detailed design phases. In the MDO phase, specialists work on the same design, each from their own discipline perspective but combining their results to agree an improved next design. In the preliminary design phase, only predesigners are involved, building up a complete but sketchy view on the design. In the detailed design phase, the specialists work progressively to detail the different aspects of the design, thus limiting the design freedom for the later specialisms.
The specialist disciplines contributing within an MDO team have expertise and background in detailed design and detailed analysis; they have their specialist computer tools with approximations appropriate to accuracies required. They will require to use their discipline-specific analysis and optimisation tools within MDO. These tools are distributed across the enterprise's heterogeneous network of computers. The initial situation for the separate specialists can best be described as:

"lonely specialists in a hostile environment".

The above situation is considered to illustrate the typical ingredients for an enterprise when it initiates MDO activities. The Information Technology Task focuses on addressing key ICT challenges to enhance the effectiveness of MDO implementation, specifically:

- to document and communicate the MDO process followed by the design team
- to enable the MDO team to work on the same design within the same time frame
- to provide an effective and friendly environment for the MDO team to operate within

The focus and priority in all this work was set by the ongoing multi-discipline design, analysis and optimisation of aircraft work, thus ensuring that practical ICT support meeting the real needs of MDO was realised. The following sections present the highlights of the work of the project in these three key ICT areas, and then the main conclusions and recommendations are summarised.

2 Highlights of technical program

2.1 Documentation and communication of the MDO Process

The MDO Process is created by partitioning the overall process into elements that can be addressed individually and by linking these back together to as the integrated MDO process. Documentation of the MDO process is vital to ensure clear communication of the process within the team defining it and in the broader design team interacting with it.
The various drivers that dictate partitioning of the process are introduced and the N^2 methodology adopted for documenting an MDO process is described.

Partitioning Drivers The requirements for competitive performance and functional integrity naturally partition the aircraft design process into classical aerospace engineering disciplines, yielding Contributing Analyses (CA's) like unsteady aerodynamics, static structural analysis, or aircraft performance. However, partitioning may also be dictated by available computational resources, manpower, or assigned responsibilities within a single company or a joint venture of multiple partners. In aircraft design, the availability of a specific aeroelastic optimisation package within a company suggests its utilisation for the aeroelastic sub-problem despite disciplinary redundancies with specialised in-house software for aerodynamic and structural analysis. A group of engineers at distributed sites might have gained substantial experience in co-operating on the solution of a particular sub-problem and therefore be considered an operational unit; or the development of aircraft components may be distributed amongst international partners according to each manufacturer's previous experience. As a result, there is no unique way to pursue an MDO activity and design data can only be interpreted in combination with the followed MDO process.
The end results of the partitioning process is typically a diagram indicating the various separate components and the flow of information between them (see Figure 1).

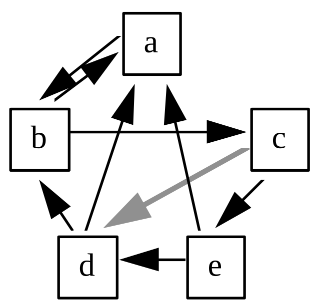

Fig.1 Connectivities between five contributing analyses

<u>N^2 *documentation methodology*</u> The approach adopted within the MDO Project to document MDO Processes is the so-called "N^2 approach". Each *contributing analysis* (CA) is described by the information it generates and the input data it requires to perform this task. Matching all input requirements with all generated information is therefore mechanical in nature - it can be supported by a number of graphical tools. A common approach is to represent CA's by boxes, which are then connected with lines depicting interfaces. These charts are useful to document CA connections, but usually result in a confusing, unstructured network of lines and boxes which does not highlight the information flow structure of the problem (Figure 1). A more descriptive representation is provided by N^2-Diagrams, as explained in the following example.

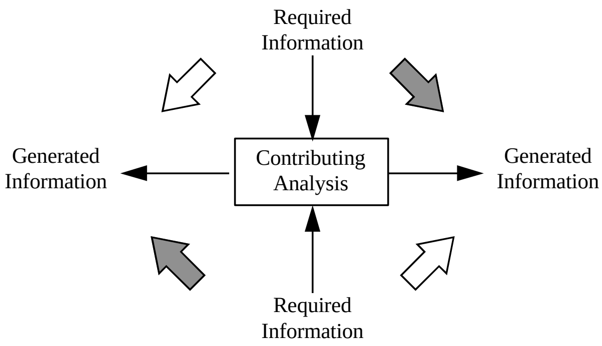

Fig. 2 Contributing Analysis

Figure 2 shows the process of information flow through a generic contributing analysis. The CA receives data through *"input ports*" at its upper and lower sides, converts this information, and supplies data at *"output ports"* at its left and right side. These boxes are then arranged on the main diagonal of an imaginary matrix. Input and output requirements are matched between contributing analysis. Figure 1 showed that in this sample case, the CA *"d"* requires input data, which is generated by *"c"*.

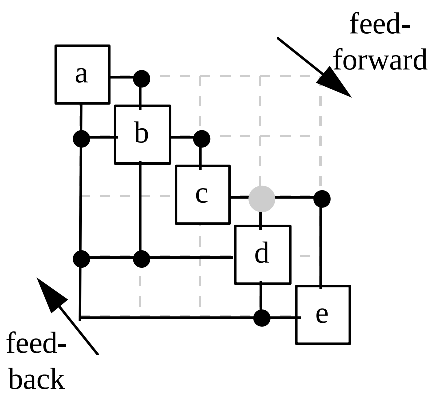

Fig. 3 N^2-Diagram

In the N^2-Diagram, Figure 3, this situation is represented by the enlarged grey dot in the row associated with "*c*" (the generating CA) and the column of *"d"*. Data (such as this) above the main diagonal depicts the forward-feed of information from *"a"* towards *"e"*, while those below the diagonal indicate a feedback of information.

A combination of feed-forward and feedback between two CA's indicates an iteration loop. Iterations are time-consuming and expensive, and the project planner or designer will therefore attempt to avoid them as far as possible. One approach is to minimise the number of feedback loops by re-arranging contributing analyses.

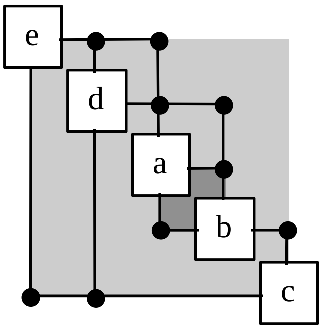

Fig. 4 N^2*-Diagram, re-ordered*

It can be seen from Figure 4 that the large, intertwined iteration structures detectable in the first N^2 diagram have been simplified by re-ordering the CA's. It is apparent that an **inner iteration** (dark grey) is nested in an **outer** iteration (light grey)- the order of execution is much clearer than in the previous figure. Tools for aiding in this re-arrangement exist; they may be based on linear algebra (by virtue of the matrix representation in the N^2-Diagram), knowledge bases, or genetic algorithms.

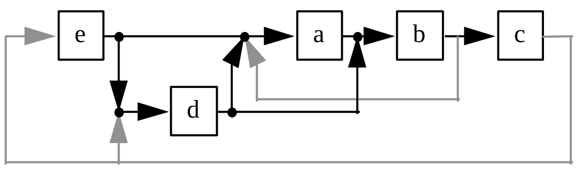

Fig. 5 Data Flow Chart

The final version of the N^2-Diagram is easily converted into a data flow diagram, Figure 5. This representation is commonly used for describing macro-level design process information flow in graphical user interfaces of computational design frameworks. The term "macro-level" refers to the low level of detail used in data representation - no information about single data values, or design model "attributes", is contained in these charts. The level of definition is increased to the "micro-level" when interface modules or tools between contributing analysis tools and the framework are specified.

2.2 Product Model and Multi Model Generation

In order to enable the MDO Team to work on the same design, a formal approach is adapted whereby the design is represented as a single **integrated product** model. This product model is shared across the disciplines - each accessing and contributing data (see Figure 6).

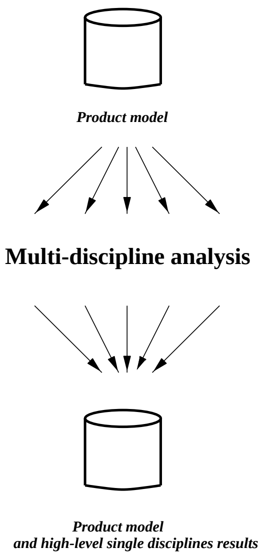

Fig. 6 Information flow around Multi-discipline Analysis

The Technical Data Modeller and Browser `tdmb` ([1]) was selected by the ICT group for the implementation of the integrated product model. The analysis models required for each individual discipline, such as the aerodynamics model or the aeroelastics model, are derived from the product model. The derivation has been automated completely, by a jointly developed Multi-Model-Generator `mmg, which` directly integrates the `tdmb` database for all technical information required for the model generation process. The `mmg` contains modules for:

- the definition of the general arrangement of the whole aircraft (`def`)
- the generation of the (wing) surface shape, and the aerodynamics model (`ssg`)
- the generation of the structural layout, loads, and preliminary sizing (`slls`)
- the generation of the finite-element model (`feg)`
- the generation of the aeroelastics model (`aeg`)

and enables the separate disciplines to work together on the same design in the same time frame. The initial architecture of the mmg was devised by the ICT group to support the joint Analysis Task and featured integration of the modules within a single computer program. The architecture was then updated and generalised as a linked suite of programs to provide the flexibility to support the development of the alternative modules required for the research work of parallel Optimisation Tasks (see Figure 7).

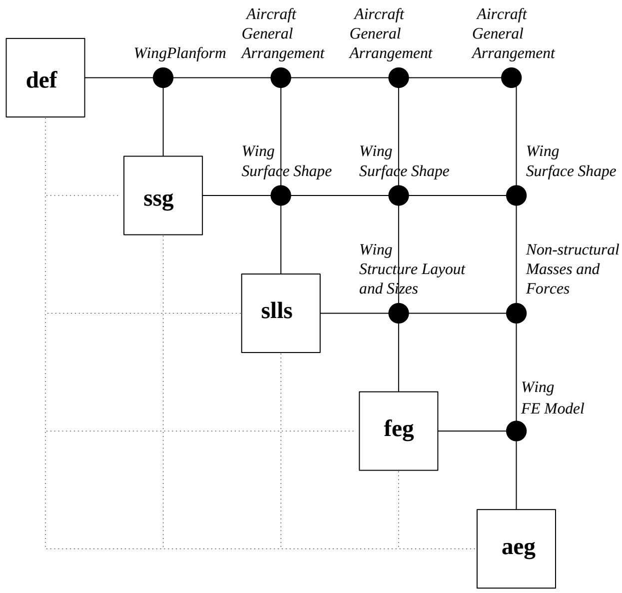

Fig. 7 Architecture of MultiModelGenerator

The management and integration of the software modules developed by the MDO Consortium partners was carried out by the ICT group.

The comparison of designs is enabled by the transfer of selected, high-level results from the contributing disciplines back into the Integrated Product Model for use at the multi-discipline level. Examples of single discipline results on the multi-discipline level include drag breakdown and pressure distributions at pre-set wing sections, optimised wing mass breakdown, mode shapes and flutter damping.

2.3 MDO Environments

In the experience of the MDO partners it has become clear that the prime purpose of an MDO environment is that of **integration** of data and processes. Where in the detailed design phase, specialists groups are shielded from the software, data, and reports from other specialists groups, now all information used for the MDO activities are shared among the MDO design team.

In working towards a common MDO environment, the project has identified three levels:

- architecture layer
- software layer
- hardware layer

The primary focus of the project has been in the architecture and software layers - as costs, timescales and company ICT strategy issues become an increasing constraint for the lower layers.

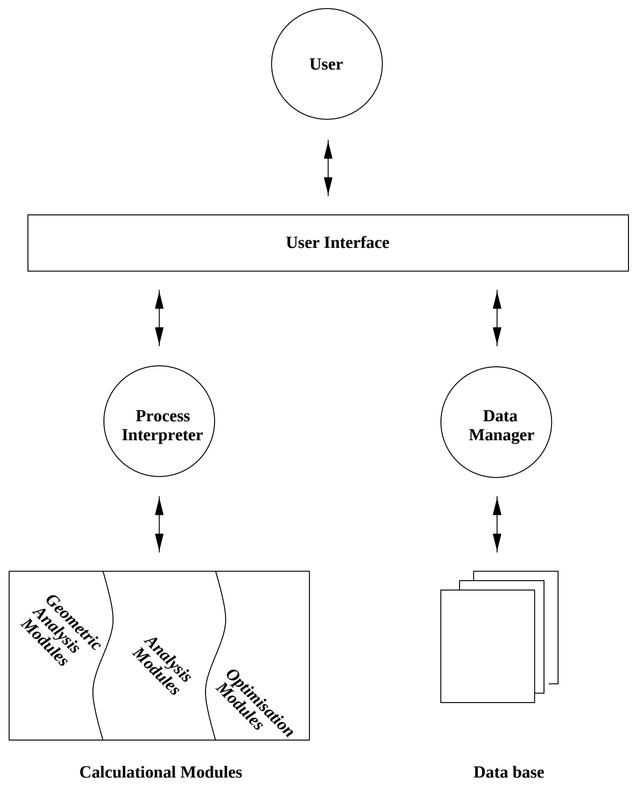

Fig. 8 (Software) Architecture of MDO Environment

Architecture Layer The architecture of a populated MDO framework is given in Figure 8. The Users interact with the MDO Environment through a User Interface:

- to use the MDO Process Interpreter to control the execution of Calculation Modules (including Design, Analysis, and Optimisation**)**
- to use the Data Manager to control design data relating to the exploitation of MDO.

The specific calculation modules required for a particular class of MDO problem will in general be developed independent of the MDO Environment. These will include both design, analysis, and optimisation tools relevant to the disciplines involved.

The primary architecture components required then of any generic MDO environment are:

- Process Interpreter
- Data Manager
- and User Interface

A standard set of evaluation criteria was identified by which to review all MDO environment implementations investigated by the ICT group.

Software Layer The software layer is the ICT implementation of the architecture layer, populated with data and software: the MultiModelGenerator `mmg`, the data manager `tdmb`, company own design and analysis modules, and optimisation modules.
A combination of commercially available, partner proprietary and new approaches for the user-interface and process interpreter have been investigated and compared during the course of the MDO Project, viz.:

- SiFrame - team working and work flow control
- iSight - optimisation algorithms and optimisation user interface
- SPINE - single user interface across a networked environment
- TOSCA - robust execution and data/process integration

All approaches were evaluated against the same generic evaluation criteria and in addition each approach was applied *"for real"* in at least one of the aeronautical optimisation research work.
This practical application highlighted the second fundamental requirement on an MDO environment, viz. that of *flexibility* of tool operation to support the evolution of new processes. Starting from a single-design analysis process consisting of separate analysis steps, support is required for automation of compound processes:

- chaining the separate analysis tools to perform Multi-Discipline Analysis
- performing calculations for all design variables
- chaining calculations together to perform Optimisation cycles

while maintaining the elementary possibilities for

- performing calculations for a single design
- extending the range of individual calculations

The latter, for example, enables the aerodynamics specialists to check the validity of the parameter tuning for the CFD tool in a design point on the boundary of the design space, and to introduce new and improve calculation procedures. Specifically, *flexibility* of tool operation was an unacceptable bottleneck limiting practical exploitation of MDO early in the Project.

However, it has been addressed satisfactorily by various environment builders and further valuable enhancements are under development.

Hardware Layer On the hardware-layer, the information from the MDO specialists groups is distributed across the Enterprise's heterogeneous computer network. The integration-level of the MDO information depends heavily on the Enterprise's computer and network infrastructure.
In the current MDO-work, the MDO partners' networks have been used as they are. Attention has been focused on the software layer. Still, the two layers can not be treated separately as, for example, a heavily loaded file server may seriously damage the performance of the MDO software environment. Enterprises should especially take care that security does not become synonymous with "impossible".

The scale of MDO computations carried out during the course of the project is characterised in Table 1.

Table 1 Characterisation of MDO activities

Number of disciplines	2
Number of design variables	7
Order of optimisation cycles	10
order of CPU-time per discipline/ DV/optimisation cycle	1 hour

Given the finite difference technique to estimate the derivatives of the objective function, the above amounts to 14-21 CPU-hours per optimisation cycle; that is, one optimisation cycle per day is feasible in case of a limited number of design variables.
Because of the low number of optimisation cycles, the role of Optimisation is not dominant over the role of Analysis. As a result, the requirements on the Process Interpreter in the system architecture are restricted to the chain-structure; iterative and conditional control structures are less relevant.

3 Conclusions, and Recommendations for Future Work

The development of complex MDO processes and the interpretation of the design solutions requires clear documentation of the MDO process.

The N^2 diagramming technique provides a simple but very effective method supporting evolution, definition and communication of MDO processes - that also provides practical support for implementation. The method is applied at a level sufficient to identify the dependencies within the process without being laborious or impracticable to update as the process evolves.

The adoption of an integrated product model `tdmb` and the automatic generation of analysis models for multiple disciplines `mmg` has provided the core technology required to allow partners to connect together their analysis and optimisation tools to form the MDO capability demonstrated in e.g. Reference [5]. These aspects are an important and unique innovation of the MDO Project - and have allowed the consortium partners to collaborate and push ahead in the implementation of MDO.

A combination of commercial, partner proprietary and new approaches to MDO environments was applied and demonstrated successfully during the course of the project - the common MDO architecture prevented unnecessary duplication of effort. All addressed the issue of *Integration*, but focusing in from different viewpoints:

- SiFrame - team working and work flow control
- Sight - optimisation algorithms and optimisation user interface
- SPINE - single user interface across a networked environment
- TOSCA - robust execution and data/process integration

Each of these demonstrations and viewpoints identify important end-user requirements that must be accounted for in steering the future development of the MDO environments. The practical application of all frameworks identified that the specific issue of *flexibility* of tool operation to support evolution of MDO processes was a bottleneck, and this is being addressed by environment builders.

On the hardware layer, most partners had access to appropriate computing resources to address simple MDO problems, however, network problems and heavily loaded systems sometimes frustrated routine execution of these large calculations. The problem must be alleviated on the software-layer by providing flexibility of tool operation:

- SPINE includes features to manage the execution of calculations on remote computers (e.g. supercomputers) in a manner that is invisible to the user,
- TOSCA includes features allowing the restart of complex optimisation processes after network (or software) problems.

With these software solutions in place it will become feasible to address increasingly complex, increasingly realistic and increasingly valuable MDO design optimisations. Hardware performance will become an increasingly critical limitation and it is recommended that partners ensure alignment of MDO exploitation plans with hardware development plans.
An MDO environment that integrates the complementary functionalities of SPINE and TOSCA is recommended for demonstration to the consortium at the Final Project Review Meeting.
The project has established an MDO capability that is competitive, that has integrity from an ICT perspective, and that has the inherent flexibility required to support the evolution of new MDO processes. From ICT perspective no fundamentally new problems are expected when other disciplines are added, or when specialists tools are changed.

New fundamental challenges for ICT will arise as the technology for MDO matures and as organisations evolve to secure better exploitation of the technology:

- support of the relation between the MDO phase, the preliminary design phase and the detailed design phase**;**
- the management of people, tools, and procedures (as required in larger teams**);**
- MDO as an activity in the Extended Enterprise**.**

Relevant developments have already started on the areas of workflow management, Internet/Intranet, STEP/GEM, CORBA. The application of the new technologies to MDO will be part of the standardisation of the European aeronautical sector to a common way of working in the European Aeronautical Extended Enterprise, throughout the entire development and lifecycle of aircraft.

References

1. Allwright, S. (1996): Technical Data Management for Collaborative Multidiscipline Optimisation:published in *Proceedings of the AIAA Conference on Multidiscipline Optimisation, Seattle, 1996.*

2. Allwright, S. (1998): Introduction to the MDO project; *Royal Aeronautical Society MDO Conference,* 1998.

3. Baalbergen, E.H. (1998): `SPINEware`: a practical and holistic approach to metacomputing; published in the poster section of the *Proceedings of the HPCN98 Conference,* Springer 1998.

4. Loeve, W., Vogels, M.E.S., Ven van der H., Baalbergen, E.H., (1997): Network middleware for enterprise enhanced operation; published in *Proceedings of the 4th International Conference on Computer Integrated Manufacturing ICCIM'97 Singapore; Springer 1997 ISBN 981-3083-68-9.*

5. Gould (1998): Aerodynamics and shape optimisation; *Royal Aeronautical Society MDO Conference*, 1998.