

# A Driving Need for Design Automation within Aerospace Engineering

Smith, A.L.<sup>1</sup> and Bardell, N.S.<sup>1</sup>

<sup>1</sup> *GKN Aerospace Engineering Services Pty Ltd,  
850 Lorimer Street, Port Melbourne, Victoria 3207, Australia*

**Summary:** The purpose of this paper is to explain the rationale for developing engineering software tools that automate elements of the aerospace design process. The justification is described from a commercial and technological standpoint. Details of two specific examples are provided that illustrate the current state-of-the-art in design automation.

**Keywords:** Knowledge-Based Engineering (KBE), Design Automation, Engineering Software, CAD, CAE, Future-Trends.

## Introduction

GKN Aerospace Engineering Services is a global tier-one supplier of engineering services. In order to remain competitive, GKNAES is faced with the challenge of providing services that meet the time, cost and quality expectations of the large European and North American aerospace OEM companies. Meeting these expectations is becoming increasingly difficult because the aerospace OEM companies are more aggressively exploiting opportunities to reduce both engineering costs and the time-to-market by outsourcing a growing proportion of their work-share to low-cost suppliers. These low-cost suppliers are typically based in India, China and South-East Asia, where advantage can be taken of the relatively low labour costs [1, 2]. As the aerospace industry is becoming increasingly global, and technological advances erode traditional geographical and cultural barriers, this trend is certain to continue. Faced with these marketplace changes, the success of GKNAES depends largely upon gaining and exploiting a competitive advantage. This paper will address the GKNAES strategy for achieving such an advantage by exploring the rationale behind, and implementation of, design automation.

## Background

The number of new and continuing military and civil aerospace programmes worldwide has declined significantly in recent years when compared with the levels of activity that marked the Cold War years and the boom in popular tourism. Contemporary major aircraft programmes within the global aerospace industry tend to follow a “busy-then-quiet” cyclic pattern [3], often in response to events that alter the prevailing economic or political *status quo*, e.g., the September 11th 2001 terrorist attack. This pattern has a pronounced effect on the engineering workload, especially when a major aerospace programme is announced and it becomes imperative to progress the engineering design effort as quickly and efficiently as possible.

For tier-one suppliers like GKNAES to be successful in this environment, it is vital that they are able to respond positively in order to gain a share of the workload resulting from the “busy” phase of the programme. Cost, quality, and the ability to perform the work in a compressed timescale then become prime drivers in securing a successful bid. Against this background, GKNAES intends to offer a service that is stable, respected, and capable of

responding to short-term work demands. In order to understand how this aim can be, and is being, achieved it is necessary to consider the following aside.

Over the years a vast knowledge base of aircraft design principles (derived from both theory and test) has been accumulated. A large proportion of this knowledge is contained within company-specific design handbooks and manuals and represents industry best practices. With the advent of modern computer systems, these knowledge sources are ideal candidates for automation to support both interrogation and application. This is particularly germane in the case of stress analysis procedures which, for metallic structures at least, are well-defined and have not changed substantially in the last 50 years. Indeed, aerospace OEM companies such as Boeing, Airbus, Lockheed-Martin and BAE SYSTEMS have achieved an enviable degree of success in automating the tedious and time-consuming tasks that once belonged to the domain of hand calculations and slide-rules. However, there is still considerable scope for improvement, particularly in the automated application of these computerised methods and their integration with CAD and CAE systems. The ultimate aim is to automate the entire design process, and produce a platform that links and performs both the design and analysis functions in a logical sequence.

To a large extent, this process of automation has been driven by the increasing maturity of both specific and generic analysis tools, which in turn has resulted from the low cost and wide availability of reliable computing resources. The common analysis bottlenecks that were typically experienced 20 years ago are thankfully a thing of the past. Companies can now provide a high-specification PC (which is often networked) on the desk of every analyst and designer, meaning that enormous improvements in efficiency can be obtained. For example, it was only 20 years ago that the ratio of designers to analysts in a typical aerospace company numbered 4:1. This was mainly due to the time-consuming business of designing with a 2D medium, such as a drawing board or a nascent 2D CAD system, and the significant effort required to implement iterative changes as a design matured. Nowadays, the above ratio has become more like 1:2 because the productivity of the designer has greatly increased due to the benefits of, and advances in, parametric and variational 3D CAD technology.

All these benefits can be traced directly to improvements in computing software, hardware, peripheral devices, and the networked office. Moore's Law<sup>1</sup>, which approximately translates to a doubling in computer power every 18 months, has been proved valid, and will likely remain so for the foreseeable future. This means that continuing improvements may be anticipated in the capability, application, efficiency, and quality of future automated systems.

## **Strategic Approach**

### **Overview**

GKNAES has developed a strategic approach to the application of automation technology. This strategy is based around the structured capture and execution of company-specific engineering design principles and processes using software technology – this is referred to as Knowledge-Based Engineering (KBE). KBE applications are developed for a diverse range of engineering processes that generally involve one or more of the following key characteristics:

- repetitive design or analysis process;

---

<sup>1</sup> The observation made in 1965 by Gordon Moore, co-founder of Intel, that the number of transistors per square inch on integrated circuits had doubled every year since the integrated circuit was invented. Moore predicted that this trend would continue for the foreseeable future. In subsequent years, the pace slowed down, but data density has doubled approximately every 18 months, and this is the current definition of Moore's Law, which Moore himself has blessed. Most experts, including Moore himself, expect Moore's Law to hold for at least another two decades.

- complex engineering process requiring simplification/standardisation;
- integration of engineering tools and data sets (CAD/CAE/CAM);
- reduction in lead-time/time-span on critical path tasks.

Applications are developed for a diverse range of engineering tasks including conceptual and detailed design, structural analysis, jig and tooling design, and manufacturing support (NC and CMM programming). In each case the KBE application attempts to fulfil the following objectives:

1. *Responsive*: Applications are developed quickly to meet project timescales and provide immediate relief for scheduled and un-scheduled problems.
2. *Cost Effective*: Application development costs are justified by comparison with the cost of conventional solutions and timescales – the aim is to provide a substantial return-on-investment.
3. *Existing Methods and Data*: Applications utilise agreed methods and accepted data sources. Application specifications are agreed prior to development to ensure company best-practice is being captured and implemented.
4. *Point Solution*: Application scope is deliberately limited to ensure that only the specific problem is solved and additional “nice-to-have” functionality is not incorporated. This is in complete contrast to commercial software products that attempt to address many different problems and often suffer from being too general.

To support this capability, significant effort has been devoted to the creation of a generic code library, which has been developed using a combination of software tools including Common Lisp [4] and Microsoft .NET [5]. This code library contains software modules that act as generic *building-blocks* to enable new KBE applications to be rapidly configured. Hence every application is custom-built to fulfil a particular need; no particular effort is made to enable the tool to be developed beyond its intended purpose. The creation of a KBE application relies on the effective implementation of an *agile software*<sup>2</sup> development process [6]. The key characteristics of agile software development as applied to KBE are:

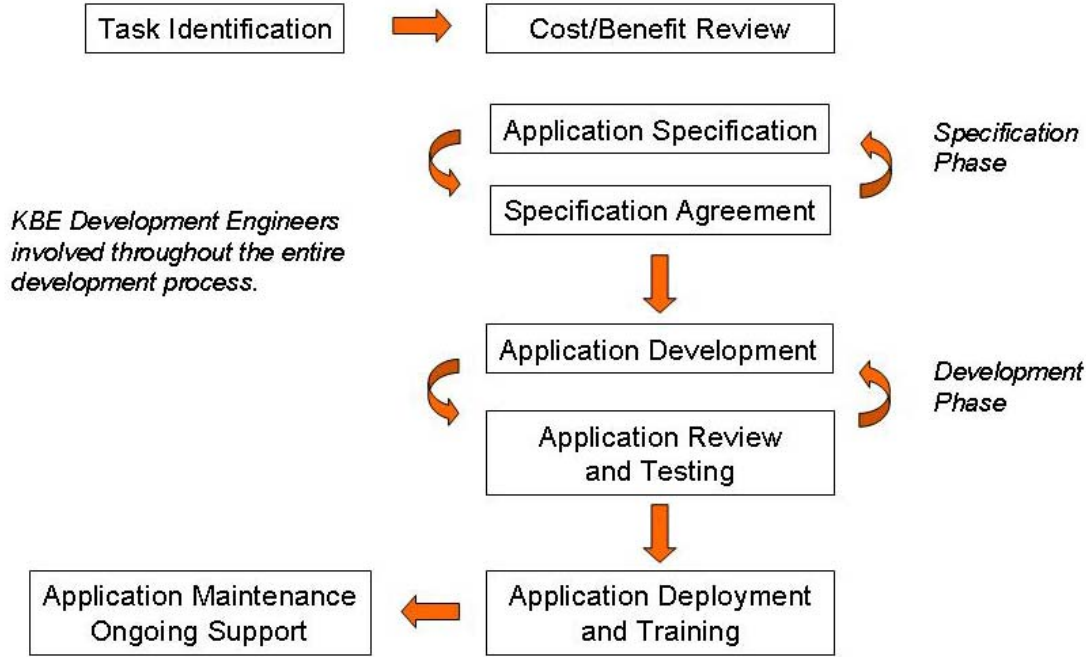
- code modularity based on object-oriented software technology;
- iterative development processes that enable rapid verification and correction;
- time-bound short interaction cycles;
- incremental development processes that ensure the developing application is always working;
- continuous scope review;
- people-oriented collaborative and communicative working style;
- elimination of all unnecessary activities.

Such agile software development methods and principals ensure that KBE applications are firstly responsive, but also cost-effective, thereby addressing objectives (1) and (2). Objectives (3) and (4) are achieved through effective management and a development team that comprises skills from both specialised engineering and software disciplines. GKNAES has built up a mixed team of aerospace/mechanical engineers and computer scientists to support this capability. Although the end product is a software application, it is critical that engineers are involved at all levels during its development to ensure its relevance, ease of use,

---

<sup>2</sup> A general term referring to light-weight rapid software development methods.

and applicability within its intended engineering environment. An overview of the development process is illustrated in Figure 1.



**Figure 1: The typical process of automating an engineering process.**

As with all attempts at automation, it is important to recognize that not everything can, or indeed should, be automated. Some human input will always be required - the real benefit of automation is its ability to provide vast economies of scale on repetitious tasks or tasks with a high degree of similarity, rather than simply de-skilling an engineering task. The aim is to complete the majority of the task in a fraction of the allotted time, thereby achieving a significant cost reduction benefit to the job. If this can be achieved with confidence, then it confers on GKNAES a competitive cost advantage when bidding for certain contracts.

Another strategic benefit of developing custom applications is that licensing issues are rarely encountered with in-house software developments<sup>3</sup>. Consequently, if a job requires additional resource at short notice, additional copies of a given KBE application can simply be made available to staff – there are none of the costs or waiting time impacts that can occur when using large commercial software packages.

The successful application of engineering automation demands a high level of skill and innovation, tempered with the practical and commercial needs of industry. Key staff members within GKNAES have developed working associations with academic/research institutions in order to take advantage of the latest developments in computer science and engineering. This is a positive symbiotic relationship that provides academic/research institutions with an industrial focus.

<sup>3</sup> In some instances there are strategic and commercial advantages to incorporating a third party software product into a KBE application. However, third party products often carry licence issues that must be properly understood before commercial usage or external deployment.

## Examples

The following two examples demonstrate the efficacy of recently developed KBE applications. Each example focuses on a different area of engineering to illustrate the breadth of subject matter that can readily be automated:

### Automated Composite Panel Tooling Design

The fixed-trailing-edge wing structure of the Airbus A380 is constructed in part from 44 unique composite panels. The manufacture of these panels requires the creation of 44 sets of tooling. A tooling set comprises a mould tool, trim fixture, NC program and a CMM program. While there is an obvious case for automation based simply on the number of panels involved, there is additional justification arising from the schedule-critical nature of the tooling and any late changes that inevitably will impact the tooling design.

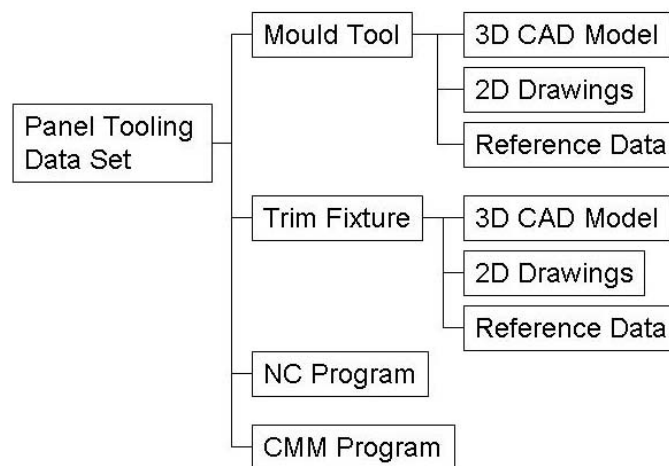


Figure 2: Composite tooling data set automatically generated.

GKNAES developed a KBE application that automates the creation of 3D CAD models and drawings for mould tools and trim fixtures and also generates the associated NC and CMM programs (see Figures 2 and 3). By using the application, the cost for the entire activity was reduced by 43% and the time by 77% compared with the cost and time estimates based on a conventional approach. Additionally, the application provided the flexibility to respond to late (un-scheduled) changes in the upstream panel design.

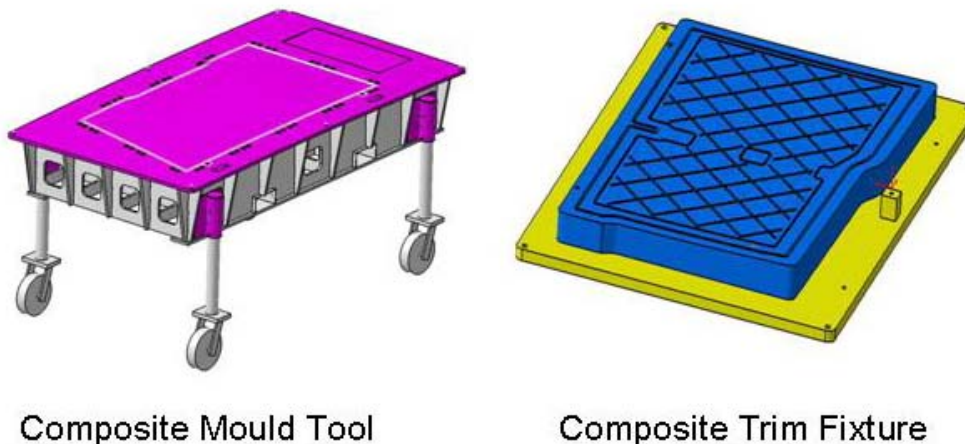


Figure 3: Composite panel tooling generated automatically.

## Structural Loads Generation from Incompatible Finite-Element Models

Aerospace structural analysis relies heavily on the interrogation of loads from finite-element models as the basis for component stress analysis. Typically the manoeuvre loads are applied to a coarse-grid whole-vehicle FE model, while component-specific loads are applied to a detailed fine-grid component FE model. Generally these two sets of models are topologically incompatible, as illustrated in Figure 4. Frequently the results from these models need to be combined and manipulated in order to calculate the equivalent loads that are applied to the physical structure. This is a difficult task to perform manually - apart from being very repetitive, it is also prone to human error.

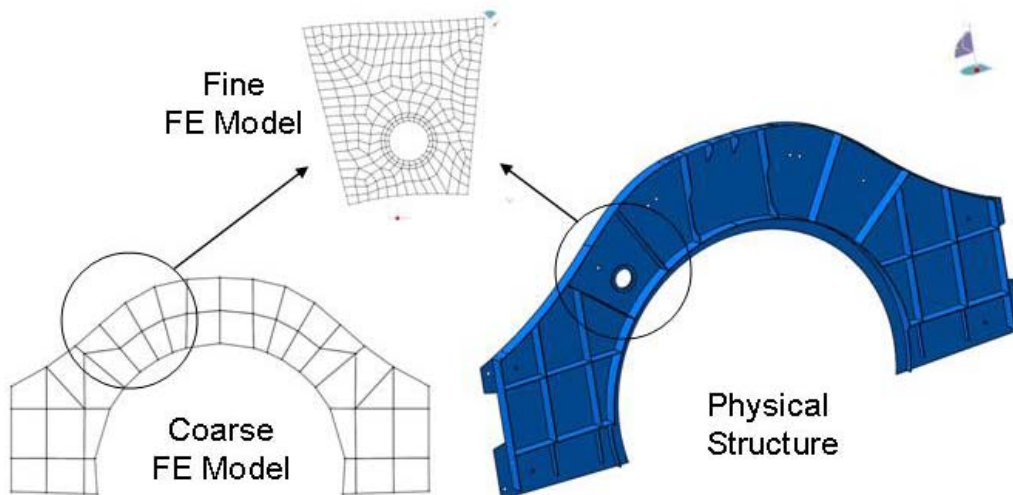


Figure 4: Typical aerospace structures and the equivalent fine and coarse FE models.

GKNAES has developed a KBE application that captures the engineering process for combining the results from incompatible FE models and then calculates the equivalent structural loads. The application was intended for processing a large number of load cases (typically 5000) for which equivalent loads for panel and edge type structures were required. This process is shown schematically in Figure 5. Since its deployment, the use of this toolset has significantly reduced the time taken to complete panel and edge type analysis when compared with conventional methodologies. Additionally, the application eliminates the need for the analyst to *envelope* the loads, thereby reducing conservatism within the design.

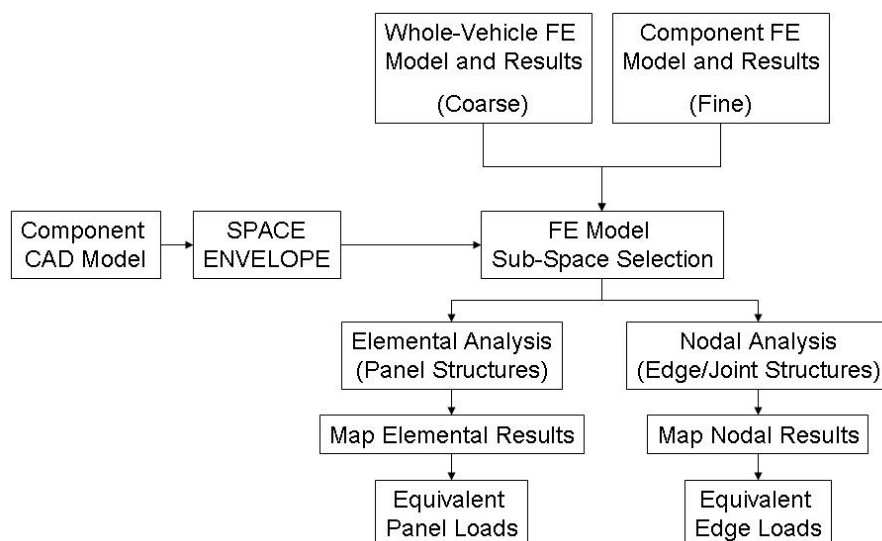


Figure 5: Process overview for extracting equivalent structural loads from incompatible FE models.

## Verification and Testing

One of the major hurdles to overcome with any software application is validation, and engineering automation applications are no exception. The degree to which applications are tested needs to be tailored to the type of data that is being generated and what testing is most appropriate. For example, an application that manipulates geometric data and generates CAD models could be validated by verifying the output for a predefined range of inputs as opposed to validating the actual process. This may be both practical and acceptable as the range of possible geometric inputs could be impractical to test, and the output CAD model would be checked in accordance with normal design procedures. Conversely, more rigorous tests are required for applications that are neither transparent nor provide easily validated output. In such cases, unit-tests are written to test specific code modules against known results obtained from independent sources. Additionally, application level tests are performed to ensure that the interaction between the code and the user interface functions properly. In all cases, the type and level of testing are matched to the requirements of a specific application.

## Return on Investment

In order for such an automation strategy to succeed, it has to demonstrate significant cost benefits when compared with traditional techniques. Whenever GKNAES deploys a new KBE application, one of the key parameters that is monitored very closely is the Return-on-Investment (ROI) index, as defined by Equation 1.

$$ROI = \frac{t_{conventional}n}{t_{code} + t_{new}n} \quad (1)$$

Here  $t_{conventional}$  is the time taken to complete the task using conventional methods,  $t_{code}$  is the time take to develop the application and  $t_{new}$  is the time taken to complete the task using the new application. The number of times the task is completed is represented by  $n$ . Generally, the ROI for a given GKNAES application is usually of the order of 2, meaning that a 100% time saving can be achieved.

Before embarking on a new KBE application, a clear case has to be made to justify the investment. For example, a simple 2 man-month job involving well-established procedures may not warrant developing a bespoke KBE tool. However, a significantly larger task of long duration involving repetitious analysis of similar parts would be a strong candidate for some level of automation. For example, GKNAES is currently contracted to carry out stress analysis justification of simple brackets carrying simple loads. Once a number of hand analyses had been performed, and the problem was well-understood, it became apparent that there was the potential to greatly improve the efficiency of the job with the assistance of a KBE tool. The resulting application combined CAD geometry with an in-house stress analysis code to provide reactions, attachment bolt loads, bending stress on any user-defined section, and deformation information. Toe-and-heeling effects were fully accounted for, and a standard stress report was automatically generated containing margins-of-safety computed for limit loading and ultimate loading.

However, with all such innovation, the law of diminishing returns will invariably set in at some future juncture and the scope to realise the sort of cost savings required to remain competitive will probably reduce. However, innovation in other areas will help offset this to a large degree, and other avenues of automating design engineering are currently being explored with this in mind.

## Conclusions

This paper has sought to explain the compelling case for the need to automate design engineering wherever possible in order to retain a competitive edge over low-cost competition. Whilst human intervention and decision-making will always be required, the stated aim of KBE is to develop an application and then perform the majority of the work in a fraction of the time it would otherwise have taken. Two specific examples have been given to show that this is already a realistic and achievable goal, and is set to become even more central to the future engineering strategy of GKNAES.

However, it would be wrong to conclude that automation is a panacea for design engineering - it isn't, and doesn't pretend to be. What it does do, however, is to provide levels of leverage and efficiency gains that permit a first-tier supply company like GKNAES to remain profitable and competitive in the world aerospace marketplace. It is slightly ironic that in order to achieve this there are two principal challenges of a *human* nature that have to be faced and overcome: (i) there is a need to capture the knowledge and processes that lead to an analytical or design solution in a form that readily lends itself to automation, and (ii) gaining acceptance of such techniques by the customer and by the relevant certification authorities.

## References

1. AFP News, India Set to Emerge as a Major Outsourcing Hub [Online February 2005]  
[http://news.yahoo.com/news?tmpl=story&u=/afp/20050213/bs\\_afp/indiaairoutsourcing\\_050213223823](http://news.yahoo.com/news?tmpl=story&u=/afp/20050213/bs_afp/indiaairoutsourcing_050213223823)
2. Web Pro News, India invites aerospace outsourcing, [Online February 2005]  
<http://www.webpronews.com/business/topbusiness/wpn-54-20050209IndiaInvitesAerospaceOutsourcing.html>
3. Aerospace America, Grand Theft Aero, [Online January 2003]  
<http://www.aiaa.org/aerospace/Article.cfm?issuetocid=287&ArchiveIssueID=34>
4. Graham, P., *ANSI Common Lisp*, Prentice Hall, 1996.
5. Microsoft .NET Framework [Online March 2005]  
<http://msdn.microsoft.com/netframework/technologyinfo/overview/>
6. Cockburn, A., *Agile Software Development*, Addison-Wesley 1997