
Chapter 6

Integration Issues

All the work presented in the earlier chapters of this thesis related to solving problems that were relatively discipline specific. However, the real problems in composite structural optimization relate to the inter-connection of the various disciplines that are involved. This chapter addresses the issues for interfacing Computer Aided Engineering (CAE) software of different types, recent advances to the ISO standards for data transfer and advanced methods for integrated CAE-based design.

6.1 Interfacing of CAE Software

The functionality of modern engineering software is rapidly evolving and is driven by the desire to increase profitability through reducing the time taken and testing costs during the design cycle. The traditional areas of FEA, structural optimization, CAD and CAM are becoming increasingly integrated in CAE software environments through a variety of mechanisms including development of add-on modules, merging of different packages under one environment and increasing the compatibility of separate packages.

Some of the diverse computational disciplines are already linked strongly (if the designers can afford the relevant software). Two examples of strong links between disciplines are from CAD to FEA and from solid models to design drawings. Some examples of the disciplines that are weakly linked or are entirely separate are conceptual design, topology optimization, sizing and shape optimization and manufacturing process cost modeling.

6.1.1 Effectively Linking CAE Software for Composites

The three most critical issues facing designers of composite aerospace structures are manufacturing cost, weight and strength¹. Effectively linking software from these areas in a single software environment has proven difficult in the past. Schumacher & Hierold (2000) and Wang & Williams (2000) discuss some attempts to link manufacturing concerns to strength and weight optimization through numerical optimization techniques. In recent times, Object-Oriented methods have shown particular promise in combining such wide-ranging analysis concerns. With these methods, new types of software such as Knowledge-Based and Feature-Based systems are starting to help design engineers perform multidisciplinary design and concurrent engineering. Henderson *et al.* (1999) and Chep & Tricarico (1999) present some progress with Object-Oriented methods and Shah & Mantyia (1995) discuss Object-Oriented methods for CAD/CAM.

The CRC-ACS is aiming to use a system based on such approaches to capture the engineering knowledge involved in design and manufacturing processes in which its industrial partners have an interest. The objective is to achieve a time-to-product advantage through the application of the core knowledge or intellectual property that is often based on hard-won experience. The knowledge base must be captured in algorithms or data sets that are easily accessible and customizable. This is an advance upon on the methodology of the Process Link software presented in Section 5.4 where algorithms used to manipulate the data were hard-wired into the system.

6.1.2 Current Limitations

The use of advanced CAE software for aerospace design is frequently limited by the amount of human-computer interaction that is required. Although many good solutions to discipline specific problems exist, they lack the flexibility to integrate

¹ Stiffness can replace strength in specific applications.

with the other disciplines needed for a given design project. Shared data formats and application linking or embedding solve many of these problems, but in many cases, efficient and seamless interaction between the software components for different disciplines is impossible without a human operator to interpret or process data before deciding how to proceed. In the Australian automotive industry it has been estimated that non-productive transfer of data between different CAE systems wastes 15% of the design time (AUSDEC, 1999). This contrasts with the common expectation that construction and assembly data are to be passed seamlessly between the designer, manufacturer and supplier.

Figure 6.1 displays a simplified flow of information and knowledge amongst the engineers engaged in the design process using current methodologies. The solid lines indicate an exchange of electronic data about the product whereas the dashed lines indicate the exchange of verbal or textual information. The three nodes of ‘structural analysis/optimization’, ‘CAD’ and ‘costing’ are indicated by a human rooted at a computer engaged in manipulation of the data. To the right is the expert in manufacturing who converses in a non-electronic manner with the costing engineer. The bookshelves represent the knowledge that is stored by the organisation and it is inefficiently distributed throughout the organisation.

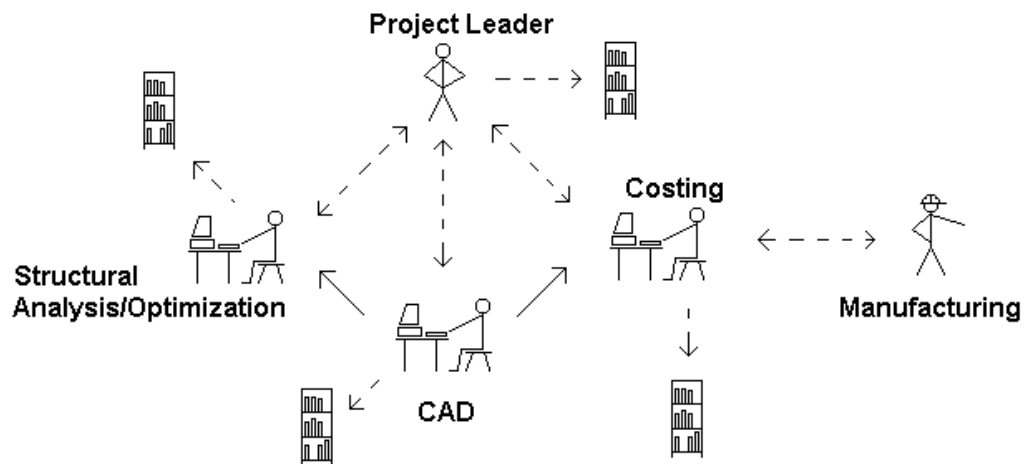


Figure 6.1, Traditional flow of information during the design process

Arguably, current CAE technology is limiting progress in engineering science. A particularly good example, is in the case of composite structural design where both FEA analysis and process modeling analyses are required to achieve the design objectives relating to cost, weight and strength. When a constraint violation is detected in a manufacturing model, it may dictate that a fundamental change be made to the product's design. This can strongly influence the other models of the product including models that organize data in ways that are entirely different from the manufacturing model. In this case extensive and repetitive re-working is needed by the designers in order to maintain consistent and accurate models of the evolving design. For any real-world design problem, such iterations are essential and time is lost whenever humans are required to link the software components together. Furthermore, the repetitive nature of the iterations can lead to errors. In many cases, the inhibiting factor is the vastly different data structures and algorithms that are used in the different analysis methods.

6.1.3 Dichotomy Between Analysis and Manufacturing Models

Part of the problem facing researchers trying to integrate the various software tools in concurrent engineering is a dichotomy between analytical modeling (FEA, etc.) and manufacturing modeling (CAD, generative CAD, process modeling, etc.). This is reflected in the attributes of the software tools used to perform these two types of modeling. Some important aspects of the software used in the respective areas are shown in the Table 6.1.

Clearly the two types of modeling are quite distinct in many fundamental ways. However, designers of composite structures routinely use both types of models. As such, the main challenge for concurrent engineering of composite structures is to overcome these differences and give the design engineer the freedom to move between different types of models at will. One obstacle to this challenge is the form of the currently available software.

	<i>ANALYTICAL MODELS</i>	<i>MANUFACTURING MODELS</i>
Use of mathematical algorithms	Extensive	Occasional
Data structures	Large, fixed	Small, mobile, object oriented
Demand for graphics	Low	High
Programming Languages	FORTAN, MATLAB, etc.	Visual BASIC, C, C++, etc.
Location of R & D	University, non-propriety	Private company, propriety

Table 6.1, The dichotomy between analytical and manufacturing models

In general, the two distinct types of modeling software do not include any scope for the user to implement algorithms that deal with issues relating to the other type. For instance, scope for new mathematical enhancements such as user-defined optimization algorithms is not included in manufacturing software; just as the option for addressing new manufacturing issues is not included in analytical software. Finding a paradigm that allows both types of models to exist in parallel is an important goal of this research.

6.2 Standards for the Exchange of Product data (STEP)

An important issue in bringing the various elements of CAE software together is the standardization of formats for exchanging electronic data relating to the structural models. Since software vendors seek profit by developing software products

independently there are often difficulties in exchanging the product data. The STEP (STAndards for the Exchange of Product model data) project is being undertaken by the International Standards Organisation (under ISO 10303) in order to improve the efficiencies of engineering data transfer. A good summary of the STEP project is seen in Gutowski *et al.* (1995).

6.2.1 The EXPRESS Language

A core part of the STEP project involves the definition of an Object-Oriented language called EXPRESS for encoding the attributes of engineering product data in a way that is portable to a wide variety of software applications and companies in a wide variety of fields. Lobo & Nazemetz (1995) discuss a number of limitations of the first version of EXPRESS. The principal limitation was the inability to specify algorithms to transform product data within arbitrary models. To address these limitations, a new phase of the STEP project was initiated in 1996. This phase involves the development of a new language called EXPRESS-2 (ISO 10303-Part 11E) that will include the capability for modeling of processes and events, definition of methods (often referred to as functions or subroutines in other languages) and the execution of scripts in a 4th generation programming language. The most recent draft version of EXPRESS-2 (Spilby & Sanderson, 1999) was used as a basis for the software developed in this research.

6.2.2 STEP Application Protocols

As part of the STEP project a number of application protocols were developed for modeling in specific engineering areas. These application protocols define collections of EXPRESS entities and constraints between them. Two protocols that relate to composite structural design are AP209 (Composite and Metallic Structural Analysis and Related Design) and AP222 (Exchange of Product Data for Composite

Structures). These protocols were formed to meet the pressing demands for exchanging product data in a more complete form than was possible through the existing formats such as IGES.

Hunten *et al.* (1999) reported on the use of AP209 for composite design and analysis using data from Lockheed Martin Tactical Aircraft Systems. They demonstrated how this Application Protocol could improve the current practices for the exchange of multidisciplinary CAE data for composite structures by implementing a pilot production scenario.

6.3 New Approaches

6.3.1 Feature-Based Methods

Feature-Based design was already mentioned in this dissertation during the work on Process Link in Section 5.4. Here, a more detailed description of the field and its potential is given. A thorough reference for Feature-Based design is given by Shah & Mantyia, (1995) and their work forms the basis for much of the following discussion. They define a feature as a physical constituent of a part having engineering significance, predictable properties and as something that can be mapped to a generic shape.

The concept of features was introduced to overcome some of the deficiencies of conventional CAD methods. Features can capture design intent (the utility of particular geometric entities) and allow abstraction in modeling which is useful in conceptual design. Traditionally, ‘microscopic data’ was used to model individual edges, faces, etc., of a model (e.g. IGES). Such methods become inaccurate and computationally inefficient when the models reach a certain level of complexity.

Features save time for the designer or draftsman because they automate the tedious details of constructing complex models. When a model has a large number of features, a ‘feature taxonomy’ can be used to show how properties are inherited from

higher features. Object Oriented methods naturally address this concept. An organisation that uses Feature-Based design will eventually want to create a library of features that reflect the properties of commonly occurring designs. Such a library can be very time consuming to create and hence it should be considered as an investment with the benefit being timesaving and improved products down the track.

An important property of a true Feature-Based system is that features are in the eye of the beholder. In other words, different applications will want to use different data structures to represent a given feature. Examples of cases where different representations are required for the same feature are:

- Where different levels of abstraction are necessary (e.g. solids for design, shells for FEA)
- Where incomplete description of a part is desirable (e.g. an NDI path plotter does not need to know the internal details of a composite laminate)
- Where multiple geometric models may be required for one product (e.g. part model, stock model and tooling models)
- Where solid entities are used for drafting but void entities are used for NC tool path software.

6.3.2 Feature Recognition

Conventional Feature-Based design gives an efficiency improvement over existing design practice but features must become more portable and general to fully embrace Feature-Based modeling for the entire design and manufacturing process. The concept of feature recognition is one solution to the problem of transferring information between a wide range of different design representations. Feature recognition software attempts to assign features to a design automatically from some other representation. One example of feature recognition is the automatic generation of Numerical Control (NC) machining tool paths or Constructive Solid Geometry models (CSG) from boundary representations. If feature recognition is available, the

designer can spend more time concentrating on defining features at a higher level of abstraction, and not worry about the tedious detail design features.

In order to create a featured design automatically, it is desirable to automatically map features or derive them from a purely geometric or abstract model. However, such mappings are difficult due to the abstract nature and diversity of features types. Research into feature recognition algorithms that can automatically detect the features present in a purely geometric model is at an early stage for solid machined components and non-existent for composite structures.

A number of design representations are particularly relevant to composite structural optimization - FEA models, structural optimization representations (including for both topology and sizing optimization results), thin panel models, CAD solid models and manufacturing process models. It is possible that feature recognition can assist in the rapid conversion between Feature-Based models of these different types.

One example where feature recognition might prove useful is the automatic recognition of feasible mandrel configurations and manufacturing sequences given rudimentary geometric models of a structures configuration such as seen in Chapter 4. Another example would be the automatic recognition of design features from 3D topology optimization results such as those in Chapter 2 and 3.

6.3.3 Knowledge-Based Methods

The distinction between Knowledge-Based methods and Feature-Based methods is not consistently drawn in the literature. Hence, it is proposed here that Feature-Based methods form a subset of Knowledge-Based methods that in turn form a subset of Object-Oriented methods. The principal deciding factor for each method should come down to exactly what types of information are represented by the entities stored within the system.

It is proposed that a Feature-Based method stores and manipulates only ‘features’ as per the definition given at the beginning of Section 6.3.1. Knowledge-Based methods are considered to apply a generalization of the ‘feature’ information to include design attributes that are not physical constituents of the product yet still contain engineering relevance (e.g. rules, manufacturing sequences, analysis results etc.). A further generalization is Object-Oriented methods that remove the restriction of having engineering significance. However, this level of generalization is not required in this research since only engineering issues are under consideration. Figure 6.2 indicates the nesting of the various concepts discussed above.

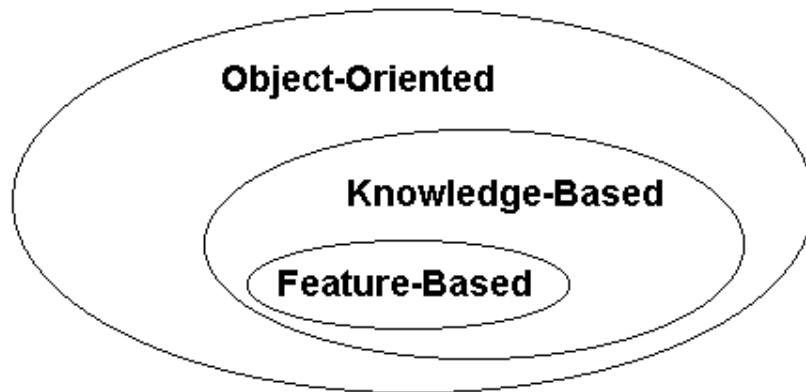


Figure 6.2, The proposed relationship between Feature-Based, Knowledge-Based and Object-Oriented methods.

To better express the nature of a Knowledge-Based system, the question “What types of knowledge exist in design?” was considered. Some answers to this question are presented in Table 6.2.

<i>KNOWLEDGE TYPE</i>	<i>DESCRIPTION</i>	<i>EXAMPLES</i>
Highly organized data	Data that can be represented in a conventional database	Nodal deformations in a finite element model
Loosely organized data	Data that is too ill-defined, irregular or complex to be represented in databases	Technical reports, verbal interactions between designers
Hierarchical data	Data that can be organized into a tree graph	Families of products, nested information, file systems
Procedures and algorithms	Methods that enable knowledge generation	Manufacturing procedures, FEA, solid modeling operations
Relations, constraints	Methods that express relationships between the entities of a design	Frameworks for organizing data, design requirements, interacting design decisions

Table 6.2, Types of knowledge in design

Of all the types of knowledge suggested in Table 6.2, the most difficult to integrate into a software system is undoubtedly loosely organized data. This is due to the manner in which current-day computer systems store their information. To best accommodate design knowledge in modern computer systems, the following key aspects of a Knowledge-Based CAE system are suggested:

- Knowledge entities – having engineering significance
- Algorithms that operate on the entities
- A user interface – allowing creation, modification and deletion of entities
- File transfer – any information with engineering significance eg. CAD data, design tables, material property databases, technical reports.

The above aspects do not consider the important issues related to adding organized knowledge to the system. However, Knowledge-Based systems are of little

use unless the knowledge of the designer is progressively captured. Section 6.4.3 will discuss the customization of the knowledge contained in such systems.

Appendix C contains a report on investigations into the functionality and flexibility of the CATIA V5 software package – a market leader in the above-mentioned technologies. The appendix also serves as a good indicator of the state-of-the-art for commercial, Knowledge-Based and Feature-Based CAD systems.

6.4 An Integrated Concurrent Optimization Environment

The desired properties of an integrated environment for the concurrent engineering of composites are discussed in this section. This discussion formed a basis for the specifications of the system that is reported in Chapter 7. Additionally, Chapter 7 presents an application of this system to a real design problem, typical of those faced by the CRC-ACS and its industrial partners.

6.4.1 Geometric Modeling

Engineering often involves 3D data and many design decisions require the best possible understanding of 3D issues. Humans (engineers arguably more than others) are adept at manipulating 3D objects in space. However, 2D computer screens can make these manipulations difficult and tedious. Hence, an intuitive interface between the product data and the 3D attributes of the data is desirable.

Salomans (1995) used a Feature-Based design methodology to link design features with manufacturing features, focussing on machined solids (as opposed to thin panel, composite structures). It was noted that when a designer draws a CAD model, they are only putting part of their knowledge into an electronic form. The rest of their knowledge is inefficiently tapped later when the process model is built from the CAD model. A more efficient scenario would be to capture more of the engineering knowledge and design intent in an electronic form from the outset. This is a strong

reason for incorporating a geometric modeler as an integral part of a Knowledge-Based system. In such a system, manufacturing features could be recognized more easily since extra information is available to make manufacturing decisions.

6.4.2 Mathematics Libraries

Mathematics of some sort lies at the heart of almost any engineering activity. This is particularly important when complex analysis is required such during FEA of composite structures. An integrated concurrent engineering system would require some sort of support for mathematical functions such as the basic operators of addition, subtraction, multiplication and division; in addition to other operators for exponentiation, trigonometric functions and matrices, etc.

The “MATLAB”TM software (Stateflow) provides a very powerful environment for mathematical analysis, modeling, optimization and data display. Ideally, a concurrent engineering system for composite structures would have the same level of mathematical interactions as is provided in that software. However, the inclusion of a more rudimentary mathematical library would still provide vast improvements over many conventional CAE systems.

6.4.3 Customization in CAE software

Modern commercial software of all types can be customized by a variety of methods. These include buttons, macros and user-defined data structures. Customization is used to speed up often-performed actions by creating an environment that suits the particular task at hand. A classic example is the word processing software used to write this dissertation, whereby custom arrangements of buttons on the toolbars were used to allow faster access to text formatting and file handling functionality. In most modern commercial CAE software, similar levels of customization are available. Examples for which the author is familiar with are “AutoCAD” (AutoDesk), “ANSYS” (ANSYS Inc.), “SolidWorks” (SolidWorks Corporation) and “Strand 7” (G+D Computing).

Frequently however, the customization does not go very far beyond the GUI configuration. Hence, there are certain limitations to what activities can and cannot be automated. The more flexible the software, the further the user can customize. However, when the CAE user talks about customizing features and algorithms (such as in the “Response to Process Link”, Section 5.4.2), more than just a customizable GUI is required. In the ideal case, the user could create customized features to model any arbitrary design. Full support of user-defined features would require all the following modes of operation (Shah & Mantyia, 1995):

- Feature creation and deletion
- Feature property editing
- Assigning relationship between features
- Validating constraints
- Geometric possibilities equal to underlying geometric modeler

Most commercial Feature-Based CAD systems compromise on at least some of these areas. Shah and Mantyia (1995) indicate that the following systems are approaching flexible creation of user defined features; “The KBO EnvironmentTM with ICAD© Built In” (KTI), “KEE knowledge-based shell” (Intellicorp), “CATIA V5” (Dassault Systemes) and “AutoCAD” (AutoDesk).

In general, if the user of the software wants to go to a greater level of customization, they must at some stage resort to some sort of programming. Importantly, this programming should not require a degree in computer science or an equivalent level of specialist knowledge. Programming could be as simple as setting up the cells in a spreadsheet program to perform the desired calculations. However, in this research, the use of spreadsheets was considered to be too inflexible for modeling the diverse types of data involved in composite structural optimization. Hence, customization by programming in a more complete sense was investigated.

6.4.4 Customization through Programming

Feature customization programming techniques are of two types - procedural feature definition and declarative feature definition. Procedural feature definition methods use a procedural programming language such as FORTRAN and feature properties and their relationships are combined through procedural algorithms. The methods investigated in this research are of this type. On the other hand, declarative feature definition uses non-procedural languages such as LISP and Smalltalk and constraint satisfaction algorithms are used to assign properties. The declarative definition methods are more modular than procedural definition methods but require more complex software architecture.

The choice of programming language was a fundamental decision when designing a system for Knowledge-Based concurrent engineering for composite structures. Hence, a survey of some existing languages that show some potential for flexible customization was made. The results of this survey are presented in the following paragraphs.

Programming with large and complex software systems is traditionally seen to be the realm of computer scientists however recent advances are having some success at reversing this trend. The JAVA language (Sun Microsystems) is a step down in complexity for the programmer over existing Object-Oriented languages such as C++. A number of technical advancements were behind these changes, however a discussion of these issues is beyond the scope of this thesis. JAVA is an Object-Oriented language that can be run from script files without special compilation software. Also, JAVA can be linked to graphical software for the display of geometric models. However, one major drawback of JAVA is that it is slow to execute and cannot efficiently perform large calculations such as geometric modeling or FEA. Also, linking JAVA to external software is overly complicated through currently available JAVA compilers.

A very popular method for customizing existing commercial software applications through programming is to use a relatively simple language such as Visual Basic to interact with the Application Programming Interface (API) of the existing package.

This approach is less popular in engineering than in general business, perhaps due to the extra complexity inherent in engineering systems. Two weaknesses of this approach are that BASIC is not fully Object-Oriented² and that the customization and bug-fixing cycles can be slow when the compiler is not part of the finished application.

A good example of a flexible scripting environment for analytical computing is “MATLAB”TM (Stateflow). The MATLAB scripting language has proven very useful in a wide variety of analyses (both scientific and engineering) and has the flexibility to efficiently perform almost any type of calculation that can be done in other languages. Its principal limitations with respects to forming the basis for a Knowledge-Based system are a lack of Object-Oriented behaviour³ and no in-built support for 3D geometric modeling.

The ANSYS Parametric Design Language (APDL) is another example of a powerful scripting language for analytical computing. This language also has the drawback of not being Object-Oriented although it does include low-level geometric modeling commands that would be necessary for the proposed system. Most significantly however, in the work presented in Chapter 4, it was found that the algorithmic capabilities of APDL are quite restrictive when working with models that have a range of topologies such as the spoiler designs of Figure 4.4.

“SolidWorks” (SolidWorks Corporation) is a modern generative CAD package that allows some limited form scripting. However, the user is unable to customize all of the geometric modeling capabilities of the software. Also, subroutines or functions to perform complex mathematical algorithms often used in engineering calculations such as matrix algebra and differential equations are impossible to define in its in-built scripting language.

² Although Visual Basic software frequently manipulates objects, the behaviour of these objects is beyond the control of the user.

³ The most recent versions of MATLAB claim to support Object-Oriented behaviour however they do not support inheritance, casting and other functionality of true Object-Orient systems.

“CATIA V5” (Dassault Systemes) is another generative CAD package that allows Visual Basic scripting control. It also facilitates the automatic execution of functions, rules and checks after changes are made to the design. These options allow the designer to capture more knowledge about the design. However, the CATIA V5 model for Visual Basic scripting is not ideal since the features that make up the designs are hard-wired into the system. Appendix C contains a review of the CATIA V5 software product that was conducted by the author.

The International Standards Organization is currently working on the EXPRESS-2 language that aims to include analytical, manufacturing and other data types in one scripting language. It contains standards for the specification of algorithms and the definition of time-dependant models (i.e. events, reactions, sequencing constraints, etc.) A recent draft of EXPRESS-2 is presented in Spilby & Sanderson (1999). For reasons of compatibility, all files written using the original EXPRESS language can also be used in EXPRESS-2. Hence, in some later parts of this thesis the two languages are referred to interchangeably, depending on the specific methodology being discussed.

6.4.5 Knowledge Flow

The integrated concurrent engineering environment targeted by this research is illustrated by Figure. 6.3. This contrasts with Figure 6.1 of Section 6.1.2 in a number of ways. At the heart of the environment is the ‘Knowledge Base’. This is an extensive collection of electronic data files that stores the engineering knowledge relating to the product’s design. A significant proportion of the company’s effort is devoted to upkeep of the system through the ‘System Integrator’ who is a specialist in Knowledge-Based system design. The ‘Project Leader’, ‘Costing’ and ‘Structural Analysis/Optimization’ nodes of the organization are all connected to the same knowledge base, manipulating the evolving design through features that belong to the same feature library. The feature library is an integral part of the knowledge base and must be continually upgraded by the ‘System Integrator’ who adds new features to meet the needs of the experts in various disciplines. The importance of verbal and

textual communication is not forgotten since the human nodes are still linked in this way.

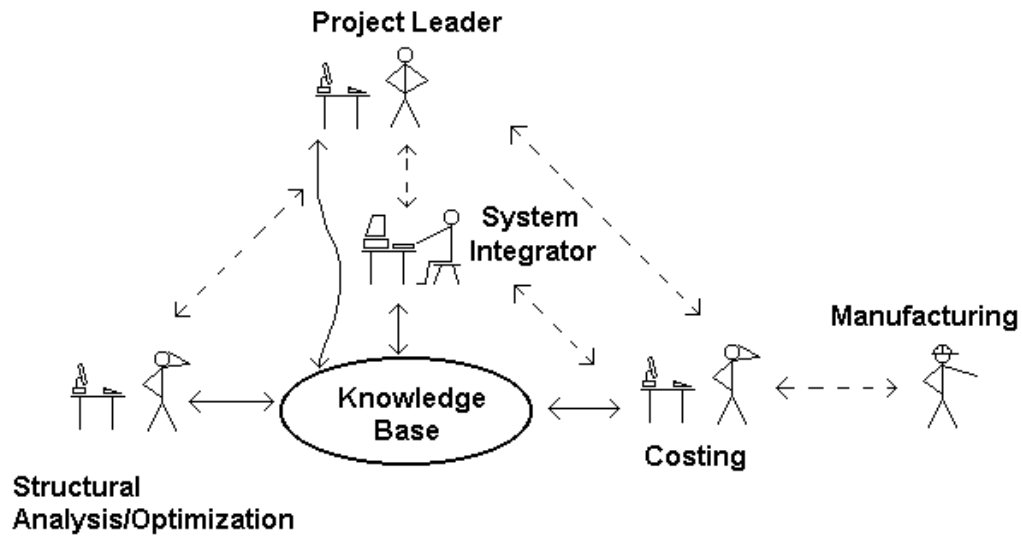


Figure 6.3, Information flow in a Knowledge-Based concurrent engineering environment
(solid lines represent the flow of electronic information and dashed lines represent verbal or written textual information)

6.5 Conclusion

There are a number of significant interfacing issues between the various existing CAE environments that are utilized during the design and optimization of composite structures. The dichotomy in Section 6.1.3 goes some way to explaining the difficulties and limitations currently facing the integration of these methods. The ISO project known as STEP was researched and a number of its key developments in this area were summarised. It was determined that there is a strong potential for STEP to meet the increasing demand for exchanging product data in electronic form between diverse CAE software applications.

Various other new approaches to integrating the various disciplines involved in composite structural design were discussed and it was proposed that a highly centralized, Knowledge-Based system could be used to incorporate manufacturing issues into structural optimization of composites. One final point to note is that CAE software is rapidly becoming more integrated and that engineering companies must stay abreast of the changes to remain competitive. In fact, one beneficial effect of the investigations reported above was to prompt new research activities within the CRC-ACS in the area of Feature-Based and Knowledge-Based methods.

References

- AUSDEC, "AUSAP – Information Circular 2", *The Australian STEP Automotive Demonstration and Awareness Project*, 1999.
- Chep A., Tricarico L., "Object-Oriented Analysis and Design of a Manufacturing Feature Representation", *Int. J. Prod. Res.*, Vol. 37, No. 10, pp 2349-2376, 1999.
- Gutowski T. G., Neoh E., Polgar K. C., "Adaptive Framework for Estimating Fabrication Time of Advanced Composite Manufacturing Processes", *Technical Report, Laboratory of Manufacturing. & Productivity*, MIT, 1995.
- Henderson J.L., Guerdal Z., Loos A.C., "Combined Structural and Manufacturing Optimization of Stiffened Composite Panels", *Jnl. of Aircraft*, Vol. 36, No. 1, 1999.
- Hunten, K., Klintworth, J., Pitre, N., Mack, T., "New Standards Based Data Exchange "Bridge" for Design (CAD), Analysis (CAE) and Manufacturing (CAM) of Composite Structures", *Proceedings of MSC Aerospace Users' Conference*, 1999.
- Lobo M., Nazemetz J. W., "Information Modeling in EXPRESS", *School of Indust. Eng. & Management*, Document CATT-95-01, Oklahoma State University, 1995.
- Salomons, O. W., 'Computer Support in the Design of Mechanical Products', *Ph.D. thesis, University of Twente*, Holland, 1995.

Schumacher A., Hierold R., “Parameterized CAD-models for Multidisciplinary Optimization Processes”, *Proc. 8th AIAA/USAF/NASA/ISSMO Symposium*, 2000.

Shah J., Mantyia M., *Parametric & Feature-Based CAD/CAM*, John Wiley & Sons, N.Y., 1995.

Spilby P., Sanderson D., “EXPRESS Language Reference Manual”, ISO TC184/SC4/WG11 N61, 1999.

Wang L., Williams A.F., “Aircraft Wing Structural Optimization with Manufacturing Considerations”, *Proc. 8th AIAA/USAF/NASA/ISSMO Symposium*, Long Beach, USA, 2000.