ICAS-92-7.2.2

INTEGRATION OF AIRCRAFT DESIGN AND MANUFACTURE USING ARTIFICIAL INTELLIGENCE PARADIGMS*

J. S. Saggu
Artificial Intelligence Applications Unit
College of Aeronautics
Cranfield Institute of Technology
Cranfield, Bedfordshire, England

Abstract

This paper presents an approach for reducing the design cycle time and integrating design with manufacturing by focusing on the development of a design support system which employs artificial intelligence (AI) paradigms. A design support system (IKADE: (An Intelligent Knowledge Assisted Design Environment Incorporating Manufacturing and Production Information) toolkit system concept) has been developed and tested through a prototype system. The IKADE prototype provides prima facie information on the manufacturing and production implications of design decisions. A description of the possible knowledge representation schemes and inferencing strategies that are required for "intelligently" linking the design and its association downstream manufacturing and production information have been developed and tested through this prototype. A description of the software components of the prototype have been developed by using an object-oriented formalism through the Smalltalk/V software tool.

I Introduction

An aircraft design process involves a series of component descriptions at various levels of detail. This process is iterative, where each iterative cycle generates a modified superior product to its predecessors. Thus, the design is finalised only when the iterative cycle reaches the required benefit and cost targets. This cyclic process which synthesises a sequence of operations, starting from the specifications which the designer creates in an initial configuration is analysed, often with the use of computer-aided design (CAD) systems, then evaluated and possibly amended. Clearly, the more iterations that can be performed within the constraints of time and money, the better the results. Not only is the aircraft design process complex(1)(2)(3) but it also requires the expertise of many engineers from different disciplines. For example, if a good quality design is to be manufactured economically, the design process must take into account many aspects of the manufacturing/production and testing processes, how it is to be financed, marketed, installed, maintained, repaired and finally decommissioned.

The types of knowledge involved in the design of aircraft components is typically large and varied, some of it informally recorded, subjective and partly judgemental (e.g., weighting the consequences of selecting alternatives). The application of other types of knowledge is subject to standards and regulations (e.g., for the aircraft industry the airworthiness requirements).

The need to simplify the aircraft design process with its complex knowledge types has led to extending the scope of the CAD systems by new programs(4) whilst others attempt to encapsulate the the process itself within newly emerging Artificial Intelligence (AI) and CAD systems. These have focused on limited applications which have tempted researchers into extending the range to incorporate such features as manufacturing and production aspects into the aircraft design, but has met with limited commercial success(5)(6)(7).

The main difficulty in extending the integration of aircraft design with manufacture is the problem of devising suitable reasoning/inferencing mechanisms to control the information flow between them. Such mechanisms outside the design area are based on formal methods developed from well established theories(8)(9). As a result, some satisfactory Expert Systems (ES) or Knowledge-Based Systems (KBS) have been produced utilising rules and knowledge obtained from domain experts(10). However, these lack the flexibility to handle complex design problems and do not permit the construction of integrated systems enable to emulate the designers ability to reason deeply about design as well as manufacturing and production processes.

Currently, conventional CAD systems do not have facilities to associate physical attributes of the components which their model represents such as extrusions, casting and forming processes. Nor does the CAD system possess any understanding of the function of, say, aircraft components such as flaps, hinges and actuators, or have any comprehension of the process involved in producing them. Most CAD systems only support the geometrical information regarding product description and do not attempt to infer with the product entities.

Most of the hopes for achieving a breakthrough in integrating design with manufacture depends upon the relatively novel branch of programming concerned with KBS. KBS offers the prospect of equipping the computer with the means, in terms both of a database and of reasoning power, - to operate more like a designer. Presently, KBS developed for CAD purposes are specific to one particular area of applications and not general applications of integrating design with manufacture(5)(11). In recent years some investigation has been conducted in the field of KBS with CAD (12)(13) but as yet this has not been fully exploited. The traditional separation of design, manufacturing and production activities is now gradually moving towards total integration attempting to represent an "intelligent" link between CAD and KBS software. Much research is being conducted in using KBS to aid ill-structured design problems [14] [15] [16].

---

*This research was supported by Science and Engineering Research Council, UK and Cranfield Institute of Technology.

The main theme of this paper is to resolve the above problems by bridging the gap between design and manufacturing employing AI paradigms. This requires developing a knowledge representational scheme and an inferencing strategy to formulate a design support system and is described by the development of the IKADE prototype study.

II Artificial Intelligence

The term "Artificial Intelligence" is not easily defined. In fact there is no widely accepted definition of human intelligence and psychologists tend to dismiss the concept as "what an intelligence test measures" (8)(9). Allowing for this misnomer the term AI has come to encompass a wide range of methods of artificially reproducing human "intelligent" behaviour.

AI has come to signify the next step in computer programming methodology having a number of research and applications direction: natural language, robotics, KBS, psychology, etc. Computers were initially developed to solve well defined problems where data could be selected, manipulated according to a definable algorithm, and meaningful results obtained. AI programming is an attempt to go further than this. A branch of AI techniques that enables computers to assist people in analysing problems and making decisions, called knowledge-based systems (KBS) or expert systems (ES), have recently proved its value, and numerous commercial applications are now available.

The major components of an KBS are knowledge-base, inference engine, user interface mechanism (including explanation facility), and data (see Fig 1). The knowledge-base contains rules (eg. IF-THEN) and facts matches the program and the inference engine takes the statements in a knowledge-base and executes them as it contains search control and reasoning mechanisms.

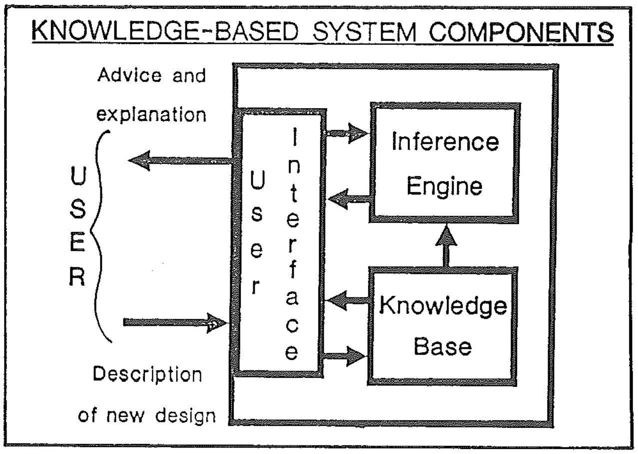


Figure 1. Components of a Knowledge-Based System

III Rationale

As previously illustrated, aircraft design is an iterative process which sometimes results in the design failing in a particular cycle, mainly due to either unfavourable design decisions made early in the cycle, or on the unsatisfactory downstream choices of manufacturing processes. Despite the fact that it is possible to backtrack in the aircraft design process, the designer needs to do this in an "intelligent" way guided by both experience and his/her knowledge accumulated during the current design process. The designer must be aware of the consequences of decisions made early in the design in order to weight the different factors involved and pass some kind of judgement towards a suitable solution.

By integrating design with manufacture, this will improve designs by bringing in considerations from areas that have traditionally been separated. In the past, the separation meant that designs took longer to develop and were of lower quality, lower product variety, poor manufacture flexibility since problems that should have been worked out in early design stages were not discovered until must later, at the start of the manufacturing set up process. Thus, their resolution required a return to a much earlier stage, making for a long overall design cycle.

Clearly reducing the design manufacturing cycle reduces the lead time by considering the downstream manufacturing implications upstream. There exists an industrial need to incorporate manufacturing in the design stage and reduce any possible disasters not only to the product but also to the organisation. For example, the Learjet aeroplane at the design stage seemed satisfactory but when it came to manufacturing, the results were devastating causing the Learjet manufacturers to go out of business. There is a need not only in the aircraft industry, but also in other industries, to watch out for disasters and reduce the gap between design and manufacture.

IV Research Objectives

An AI based approach is suggested as a means of attaining a knowledge level integration for aircraft structural component creation process with its associated manufacturing implications. AI and design of aircraft component require facilities to be able to reason about the contents of the activities engaged in integrating design with manufacture requiring a much more powerful representation scheme than that provided by geometric shapes in current CAD or computer-based design systems. Thus, to resolve these problems through the incorporation of manufacturing and production knowledge in the design process has led to the development of **IKADE:** An Intelligent Knowledge Assisted DESIGN Environment Incorporating Manufacturing and Production Information.

In order to resolve the integration issue a knowledge representation scheme and an inferencing strategy needs to be developed for a design support system, so that the downstream manufacturing and production implications can be relayed upto the design phase using AI computer techniques. The design of aircraft flap components is chosen as a template, which has a sufficiently high degree of complexity to be representative of the design process required for an effective system. As is normal in such cases, the development of a prototype is required to enable the evaluation and analysis of the knowledge representation scheme and inferencing strategy of a design support system environment to be conducted. A **prima facie** assessment approach has been adopted in order to reduce the design and manufacturing cycle time for the design of aircraft flap components.

The aim is
to create a basic Intelligent Knowledge Assisted Design Environment system which incorporates manufacturing and production information for a simple aircraft component.

V Prototype IKADE

In this section, the IKADE prototype system, developed through an Object-Oriented Programming (OOP) paradigm, allows the opportunity to assess the consequences of the design choices on the eventual manufacturing and production requirements for design of aircraft flap components is presented.

The aim of the prototype was to experiment with the concept of integrating design with manufacture that would lead towards an operational IKADE toolkit system (ie. an actual industrial working system). The operational IKADE system is envisaged on being implemented through the ICAD System^{TM} (A knowledge-based engineering system) (17). However, the ICAD System was not available during the IKADE research programme. The operational IKADE toolkit as an application system would address not only the aeronautical domain, but domains such as automotive, bridge, electrical and ship industries. The operational IKADE toolkit system would enhance the optimal integration of manufacturing functions in the design process. In this context it would encourage the design community to participate more actively in the production/manufacturing aspects of product development.

Overall IKADE Prototype Architecture

The overall IKADE prototype architecture was based around the Smalltalk/V environment and its OOP paradigm developed on the IBM PC/AT (See Fig 2). The advantage of using an OOP paradigm was that it allowed access to various modules of the IKADE prototype with relative ease.

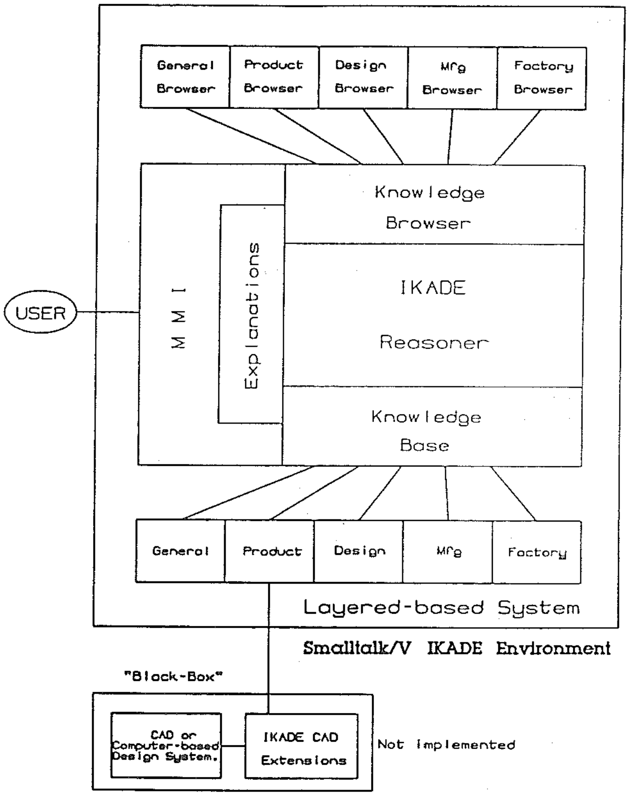


Figure 2. IKADE Prototype Architecture

With reference to Fig 2 the prototype architecture is divided in two halves, the upper half containing the developed IKADE prototype system modules, and the lower half containing the CAD or computer-based design **black-box** system which has a CAD or computer-based design system together with the IKADE's CAD Extensions module residing in it. This **black-box** has not been implemented within the prototype mainly due to the lack of availability of the appropriate CAD software tools.

The developed and implemented section of the architecture depicted in Fig 2 is the IKADE prototype part. This constitutes four main modules: the IKADE prototype Reasoner, which acts as the inference engine; the Knowledge Browers; the Knowledge Base; and the Man-Machine Interface (MMI). A small module within the MMI is the Explanations module which contains explanations and help facilities for the IKADE prototype system. The periphery modules on the upper portion of the architecture: General browser, Product browser, Design browser, Manufacturing browser and Factory browser, contain the knowledge of the aircraft flap components which is browserable by the designer/user through the Knowledge Browser and the MMI modules. The Knowledge Browser module acts as a controller for the five browser modules.

The lower five modules: General, Product, Design, Manufacturing and Factory contain the knowledge, rules, etc., accessible by the IKADE prototype Reasoner through the Knowledge Base. These five Knowledge Base modules are arranged in layers developed on the layered-based knowledge representation scheme (explained later).

The IKADE prototype Reasoner performs functions similar to those of an ES inference engine, but also performs the execution process and maintains a cohesive flow of information between the various modules of the IKADE prototype system. The designer/user can access all of the modules through the MMI windowing facilities incorporated within the Smalltalk/V environment. Finally, the whole prototype system is built around a user-friendly environment accommodating additions, removals, and modifications of data/knowledge.

Smalltalk/V Object-Oriented Programming Language

Smalltalk/V is an OOP Language (18). . In an object-oriented system, abstract constructs, called objects, encapsulate a set of related data, called attributes, and a set of related procedures, called methods. Attributes are simple data values or pointers to other objects, and methods are procedures that define the behaviour of objects. Processing in an object-oriented system is initiated by inter-object communications called messages. An object can request, through a message, an attribute from another object, or the executation of a method in another object. Individual objects are defined as instances of a class, in which default attributes and default methods are specified. The instance object can use the default attributes and methods, in which case it is said to inherit the attributes and methods.

The advantage of using Smalltalk/V for the IKADE prototype system was that it is extremely good for prototyping programming ideas and has facilities for reuseability of program code. Because of its OOP paradigm, Smalltalk/V provides modularity for ease of development. The user-friendly environment helped develop windows and menus fulfilling the MMI feature of the IKADE prototype system.

Although there are many advantages in using Smalltalk/V, one of the major disadvantages faced in using it to develop the IKADE prototype system, was that a considerable amount of time was taken to learn the system before it could be efficiently used as a programming tool.

Development of the Prototype

The development of the IKADE prototype system began by initially performing the knowledge elicitation and acquisition processes for the aircraft flap component. The next stage involved mapping the aircraft flap design and manufacturing knowledge into a knowledge representational scheme by evaluating and accessing existing methodologies. At the same time an inferencing strategy for the IKADE prototype system was developed. The final stage involved the programming of the developed layered-based knowledge representation scheme and the inferencing strategy and drawing together the MMI module to produce an effective prototype system.

Knowledge Acquisition Process and Knowledge Sources

Knowledge from end-users as well as designers/experts was required to build up the knowledge base and the sequence of design and manufacturing processes for the IKADE prototype system. The target end-users were MSc students on the Aerospace Vehicle Design course at Cranfield Institute of Technology and designers in the aerospace industry. The knowledge sources for the IKADE prototype system were: British Aerospace Plc designers and manufacturing engineers, Aeronautical staff at Cranfield Institute of Technology, including personnel from the drawing office and the engineering workshops; lecture notes on aircraft design from Cranfield Institute of Technology; documents such as books, reports, past MSc and Ph.D theses; and from the author's previous engineering experience. Furthermore, on-site work practices were studied at several British Aerospace Plc branches and other engineering sites. The knowledge encapsulated was then documented into reports(18). This was followed by studying the various knowledge representation possibilities in order to represent the aircraft flap design and manufacturing processes. Gathering this knowledge from numerous sources required the use of knowledge acquisition techniques which helped to select, segment, aggregate, condense and reproduce this knowledge.

Knowledge Types

The type of knowledge that was required to be extracted from the domain experts and from the documentation, etc., stated above are the implications of manufacturing on design of aircraft flap components. The implications of manufacturing on design was directed towards the flow of aircraft flap components as products, along a manufacturing and production line which was represented by an aircraft manufacturing/production factory. This enabled the many complex relations and interactions between design and manufacture to be considered. Factors such as flap loading, structural weight, cost reliability, strength and stiffness, operational characteristics and aircraft type, etc., were considered in the design process, because some of these factors were dependent upon the downstream manufacturing implications. The knowledge which was required to be encapsulated was found to be dispersed among several design and manufacturing engineers, each having encountered a unique set of problems.

IKADE Knowledge Representation Scheme

The knowledge representation scheme was fundamental to the development of the IKADE prototype system. Using Smalltalk/V's OOP environment, a layered-based approach was developed and adopted for the representation of aircraft flap design and manufacturing knowledge. The layered-based knowledge representation approach enables programmers to represent the domain knowledge in a modular fashion (See Fig 3). In addition, this representation enables programmers to add, remove, and modify particular parts of coding offering substantial prototyping capabilities for the IKADE prototype system development.

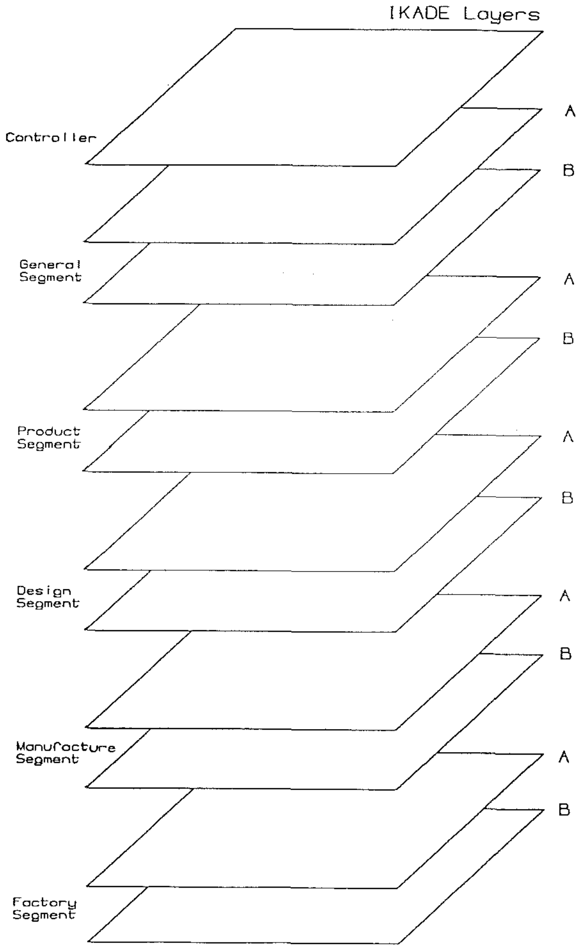
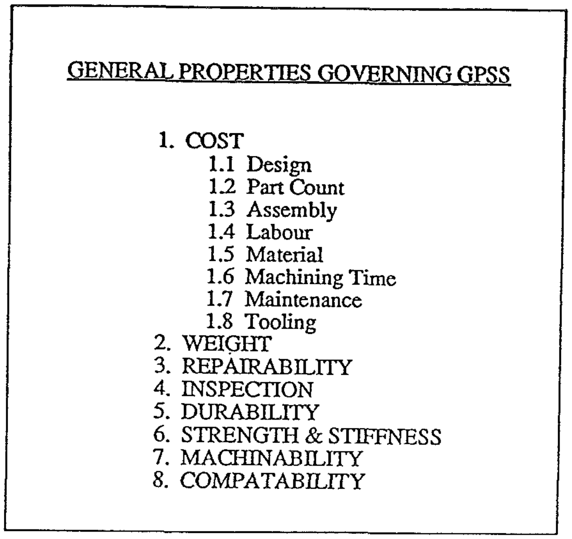


Figure 3. IKADE Knowledge Representation Layers

IKADE Inferencing Strategy

The developed inferencing strategy for the IKADE prototype system relays the downstream manufacturing and production implications of aircraft flap products into the design stage. A number of different inferencing strategies were studied with different existing and developed knowledge representation schemes.

The inferencing control was performed by the use of "weighting factors", referred to within the IKADE prototype system as the General Priorities Scoring Scheme (GPSS)(See Fig 4). The GPSS values ranged between 0-9 (zero to nine) and were set up by introducing criteria that governs the design and manufacturing decisions in the construction of aircraft flap components. GPSS values are set up prior to the execution process of the program so that different values can be assigned in order to represent a factory scenario. The GPSS approach forms constraints on the design and manufacturing process/information that was most applicable to the tasks at hand. As some of the knowledge was encapsulated in the form of rules, inferencing using chaining methods was performed.



GENERAL PROPERTIES GOVERNING GPSS

1. COST
    - 1.1 Design
    - 1.2 Part Count
    - 1.3 Assembly
    - 1.4 Labour
    - 1.5 Material
    - 1.6 Machining Time
    - 1.7 Maintenance
    - 1.8 Tooling
2. WEIGHT
3. REPAIRABILITY
4. INSPECTION
5. DURABILITY
6. STRENGTH & STIFFNESS
7. MACHINABILITY
8. COMPATABILITY

Figure 4. General Priorities Scoring Schemes (GPSS) Attributes

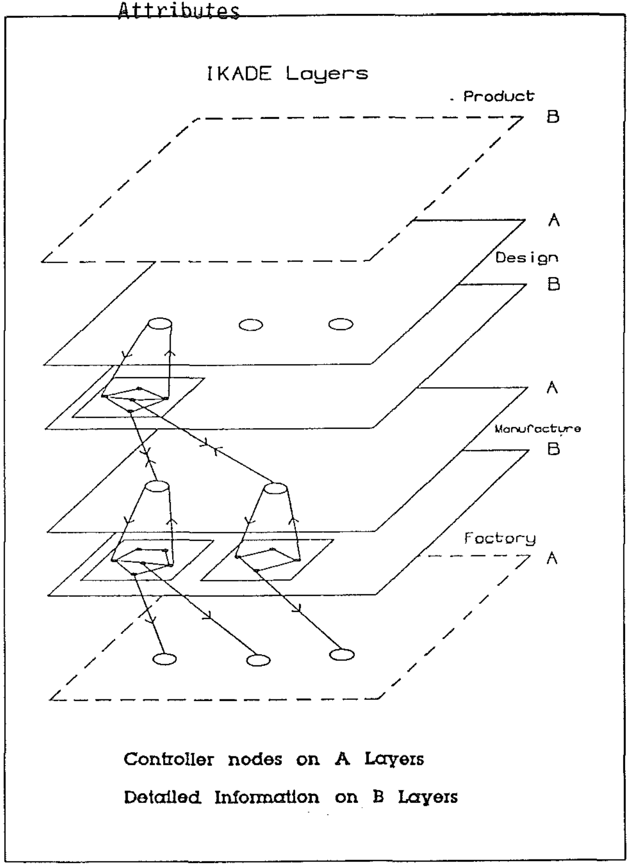
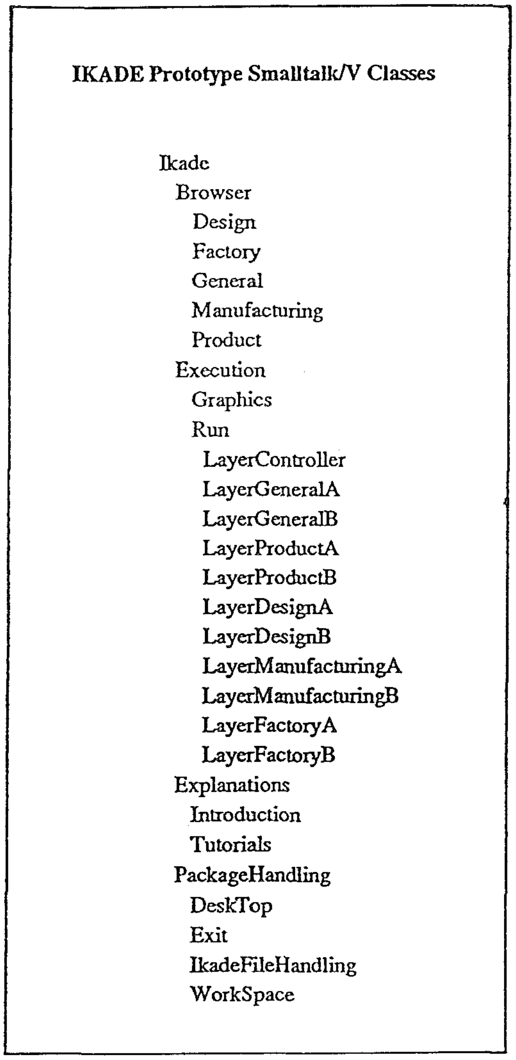


Figure 5. IKADE Knowledge Representation Layers with Network Trees of the Cells

The control flow was performed by adopting a method similar to that of the **critical path analysis technique.** This was used in association with the GPSS (weighting factors) values to find the optimum "best" path through the network-tree structure of the cells in the layered-based knowledge representation layers (see Fig 5). This effectively acts as an ordering strategy focusing on the relevant areas of knowledge and processes.

Prototype IKADE Modules

The IKADE prototype is made up of four main modules represented by the developed Smalltalk/V classes. These four main classes are: Browser, Execution, Explanations and Package-Handling. Each of these four classes are further divided into subclasses which constitute the complete IKADE prototype system. These classes, which make up the modules of the IKADE prototype architecture (see Fig 2), are depicted in Figure 6.



IKADE Prototype Smalltalk/V Classes

- Ikade
    - Browser
        - Design
        - Factory
        - General
        - Manufacturing
        - Product
    - Execution
        - Graphics
        - Run
            - LayerController
            - LayerGeneralA
            - LayerGeneralB
            - LayerProductA
            - LayerProductB
            - LayerDesignA
            - LayerDesignB
            - LayerManufacturingA
            - LayerManufacturingB
            - LayerFactoryA
            - LayerFactoryB
    - Explanations
        - Introduction
        - Tutorials
    - PackageHandling
        - DeskTop
        - Exit
        - IkadeFileHandling
        - WorkSpace

Figure 6. IKADE Prototype Smalltalk/V Classes

The MMI of the IKADE prototype system is represented by the Smalltalk/V environment using multiple windows, pop-up menus and access to external files. The Knowledge Base and the Knowledge Browser are made up of external files written through the

Smalltalk/V environment and held at the MS-DOS level. Smalltalk/V has no restrictions in developing a new knowledge representation scheme and inferencing strategy for the IKADE prototype system.

The implementation of the IKADE prototype system has been discussed through describing the functionality of the prototype's modules. Hence, with reference to Figure 6 the classes that make up the IKADE prototype system together with their functionalities are described below.

Ikade

Ikade is the top level class which initiates all the relevant files, declares the size of the array variables and other relevant variables, and basically introduces the IKADE prototype system to the designer/user (operator). The Ikade class contains the Master menu and description of the main options that are available within the IKADE prototype system. From the Master menu, the designer/user is able to access the different modules of the IKADE prototype system, i.e., via the classes.

Browser

The Browser (Knowledge Browser) is the subclass of Ikade. The Knowledge Browser which has been set up using the Smalltalk/V environment allows the designer/user to wander through the large amount of information held within its data files while giving access to information at several different levels. In order to develop the Knowledge Browser three alternatives were examined: (a) searching links and opening windows successively to examine their contents, (b) searching the Knowledge Browser data base for some string, and (c) searching the Knowledge Browser graphically. The scheme chosen was (a), which proved suitable for implementation within the Smalltalk/V environment. Rather than having many windows open to hold the bits of data/information, a multi-pane window was employed. The Smalltalk/V provided in addition to its editing facilities the manipulation of objects and classes, which proved useful for wandering through the Knowledge Browser.

The Knowledge Browser for the IKADE prototype system was structured using a frame-based approach. Here frames were represented as data/information structures in which all the knowledge about a particular entity was stored. In addition, this representation was particularly useful for providing modularity, accessibility and modifications to the Knowledge Browser's entities. With reference to Figure 7, the several panes in the Browser window were used to list options and display textual information according to the product or process. The entities were selected one at a time from each pane and its corresponding list then presented in a new pane, to the right of the entities pane and on selecting the entity from the right pane its corresponding information is depicted in the lower half of the window pane.

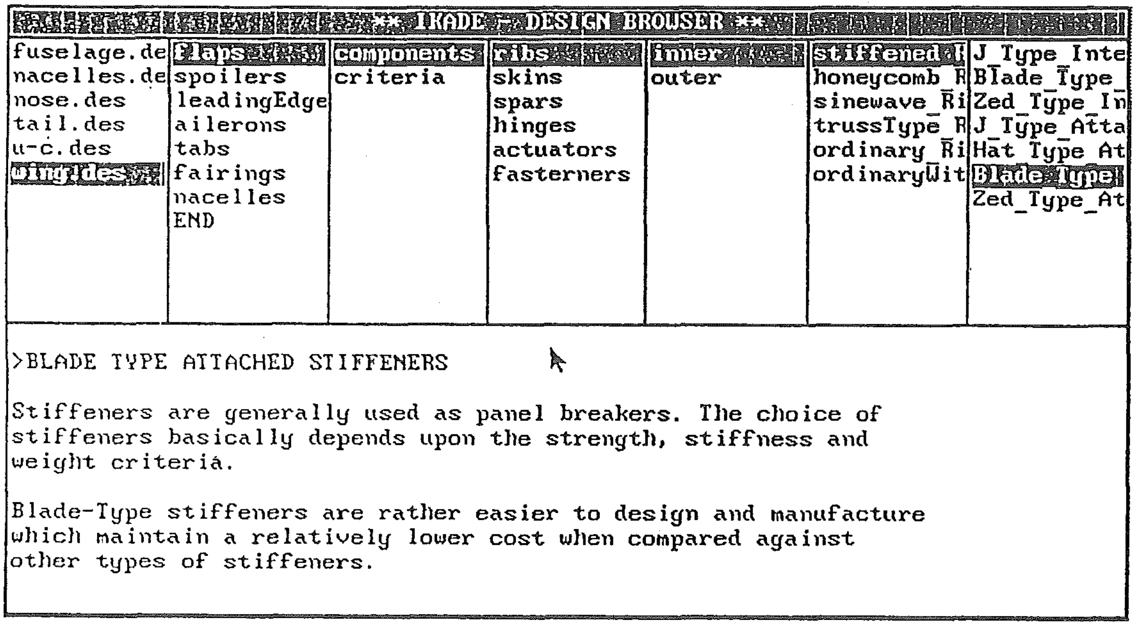
There are five Knowledge Browser options: General, Product, Design, Manufacturing and Factory, which are represented by the subclass of the Browser class. These subclasses of the Browser formulate the Knowledge Browser of the IKADE prototype system and are discussed in the subsections below.

General

The General subclass of the Browser class contains methods to represent the general information of the aircraft flap component design and manufacture. Furthermore, the General Knowledge Browser contains the information related to the aerodynamic data, performance characteristics, and loading data. It also contains general information regarding different types of flap configurations. The five flap configuration types are: plain type, single slotted, double slotted, triple slotted and Fowler.

Product

The Product subclass contains the aircraft flap product information. This information is in the form of text files and refers to the various product characteristics; material, special features, constraints, standard part features, etc.. Several panes of the Product Knowledge Browser window allow the



>BLADE TYPE ATTACHED STIFFENERS

Stiffeners are generally used as panel breakers. The choice of stiffeners basically depends upon the strength, stiffness and weight criteria.

Blade-Type stiffeners are rather easier to design and manufacture which maintain a relatively lower cost when compared against other types of stiffeners.

Figure 7. A Typical Prototype Browser Window

designer/user to focus on the product feature details.

Design

The Design subclass of the Browser class has Smalltalk/V class methods in order to represent a browser system containing the design knowledge of aircraft components. The knowledge encapsulated in the Design Knowledge Browser involves the applications of fundamentals of aerodynamics, material and structures and analytical steps (used for examining various flap configuarions). Moreover, it contains design process information required to design the aircraft flap components, including its detailed component attributes such as lightening holes, rib bend radii and stiffeners.

A typical design process information contained in the Design browser of an aircraft for a pressed sheet metal type may require the following design parameters:-

i) Bend radii.

ii) Outside corners rounded off.

iii) Flanges (five times the web thickness).

iv) Joggles (for parts to pass over one another).

v) Lightening holes (saving weight with small loss of strength.

Manufacturing

The Manufacturing subclass of the Browser class contains manufacturing process information/data. This information in the Manufacturing Knowledge Browser consists of the manufacturing processes which are built by linking together machines, material handling systems, manufacturing practices, machining processes, etc.. It does not contain process planning information for aircraft flap components. Furthermore, the Manufacturing Knowledge Browser also contains the manufacturing processes of welding, riveting, adhesive bonding, attaching stiffeners, and other details associated with completely manufacturing the aircraft flap components.

Factory

The Factory subclass contains details of the manufacturing resources. This is the extensive information/data of the manufacturing equipment available to the system and consequently the designer/user. The overall manufacturing resources of the related manufacturing processes and implications are held in the Factory Knowledge Browser together with the available manufacturing machines and associated information which make up the factory scenario and is set up by the IKADE programmer.

Execution

The Execution class is the subclass of the Ikade class which has two subclasses, Graphics and Run. It contains the variables that are specifically used for the execution process. The functionality of the Graphics and Run subclasses are explained in the following subsections.

Graphics

This Graphics subclass of the Execution class has been added in order to present nodal graphical representation.

Run

The Run class, which is the subclass of the Execution class, contains subclasses which accommodate the knowledge representation layers of the IKADE prototype system. Global class variables used for the knowledge representation layers and the execution process are initialised in this class. The five main knowledge representation pairs of layers containing layers A and B, expressed by subclasses are described in the following subsections. The A layers are control layers and B layers encapsulate detailed information of particular processes. Only one layer exists for the main controller layer and this with the five pairs of layers make up eleven knowledge representation layers of the IKADE prototype system.

LayerController

The LayerController class is a subclass of the Run class which contains Smalltalk/V methods for the General Priorities Scoring Scheme (GPSS) "weighting factors" for controlling the inferencing process for the IKADE prototype system.

LayerGeneralA and LayerGeneralB

The LayerGeneralA and LayerGeneralB are subclasses of the Run class. They contain the general information regarding aircraft flaps, such as air loading, different flap configuration performance and characteristics. From these subclasses the designer/user selects the type of flap configuration. In addition to the general information of flaps in these classes, further information for different aircraft components: ailerons, fins, fuselage, etc., may be subsequently added.

LayerProductA and LayProductB

The LayerProductA and LayerProductB subclasses contain the information that would be relayed tothe IKADE's CAD Extensions module and then mapped onto the CAD or compuer-based design system. As the CAD Extensions module of the IKADE prototype system has not been designed and implemented, the information that was envisaged to be displayed on the CAD or computer-based design system was presented in textual format in the Production Representation window, by using these subclasses.

LayerDesignA and LayerDesignB

In the LayerDesignA and LayerDesignB classes, the Design Knowledge Representation layers, a preliminary design of aircraft flap components was performed. Detailed design or finite element analysis (FEA), etc., was not conducted in this or any phase of the execution process. Instead, a general design process was performed allowing sufficient detail to be determined before moving to the manufacturing processes in the Manufacturing Knowledge Representation layers. Once the component had been designed and its dimensions established, these dimensions were then mapped onto the Product Representation window in a textual format.

LayerManufacturingA and LayerManufacturingB

In the LayerManufacturingA and LayerManufacturingB classes the flap components which had been designed in the Design layers were assessed against possible manufacturing processes. For example, if a flap rib was designed on the basis of employing sheet metal construction, then the related manufacturing process for sheet metal construction was reviewed.

LayerFactoryA and LayerFactoryB

In the LayerFactoryA and LayerFactoryB classes, the Factory knowledge representation layers contained the manufacturing resources information. This information was related to the manufacturing equipment available within the factory. The factory equipment information was set up by the IKADE programmer in order to represent a particular factory scenario.

Tutorial

The Tutorial subclass of the Explanations class contains the recapping facility, and its purpose is to recap the information acquired from a previous design session allowing the designer/user to reflect on the information provided or deduced by the prototype system.

The Remaining Ikade Classes

The remaining Ikade classes that constitute the components of the IKADE prototype system: Explanation, Introduction, PackageHandling, DeskTop, Exit, IkadFileHandling and WorkSpace are components very similar to the facilities provided by current PC desktop software packages. These facilities were incorporated in order to make the IKADE prototype appear as a complete PC software package.

Trial Implementation

Several trial implementation of aircraft flap rib components has been accomplished, and to give details is beyond the scope of this paper. As an example, one trial focused on the aircraft flap ribs that are designed and manufactured from sheet metal alloy material(20). The rib attributes, such as lightening holes, stiffeners and attachment processes were investigated through the prototype's knowledge base.

VI Conclusions

The paper has been concerned with the development of the IKADE prototype system. The overall IKADE prototype architecture based around the Smalltalk/V environment and its OOP paradigm has been specified and implemented on the IBM PC/AT machine. The main advantages of using Smalltalk/V's pure OOP capabilities promoting modularity, prototyping and its methods enabling it to reduce dramatically the size of the programming code.

The initial part of the IKADE prototype's Knowledge Base and Knowledge Browser development was achieved through a knowledge acquisition process. This involved identifying the knowledge sources and adopting elicitation techniques. The development of the MMI for the prototype was also discussed. Here the Smalltalk/V environment proved beneficial in developing the MMI which was also accomplished by employing existing classes and methods for constructing windows and pop-up menus.

The IKADE prototype system is built around classes, subclasses, methods and objects which make up the specified architecture. Their functionalities have been discussed in terms of Smalltalk/V classes and with respect to the prototype system's modules. Finally, a trial implementation using the developed layered-based knowledge representation scheme and the inferencing strategy for the IKADE prototype system used through an OOP paradigm was discussed by describing the functionalities of the prototype's modules. The scope of the prototype covered the aircraft flap rib components from sheet metal material.

Recommendations

Object-oriented formalism has been used in the development of the prototype IKADE. This has advantages of re-usability of the software components. If the IKADE prototype is to be further developed, for example, by adding more aircraft components (ailerons, elevators, fuselage, etc.), then existing software components from the prototype may be used. However, in order to employ this type of re-usability, difficulties mut be noted: first, component correctness becomes more critical since errors are replicated whenever components are reused; second, components must be understandable, i.e. well written and well-documented; third, components must be easily adaptable for different uses, either in original or modified form.

The IKADE prototype system stores the design and manufacturing results from the execution process in a window of the prototype's environment. These results can be displayed fully on the computer screen or a hardcopy produced in order to review, by the designer, the design and manufacturing processes. It would be advantageous to have the IKADE system learning from the previous designs.

From the present research work has identified the difficulties in developing a generic engineering design support system. Further work is required which will address a small number of engineering applications rather than all.

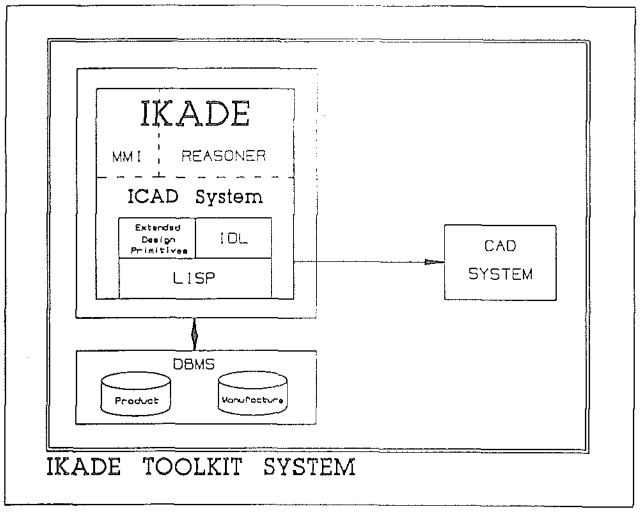


Figure 8. Operational IKADE Toolkit System Starting Architecture

REFERENCES

1. Nicolai, L.M., "Fundamentals of aircraft design", METS. Inc., Xenia, Ohio, USA, 1975.

2. Stinton, D., "The anatomy of the aeroplane", Collins, 1983.

3. Torenbeek, E., "Synthesis of Subsonic Airplane Design", Delft University Press, Martinus Nijhoff Publishers, 1986.

4. Saggu, J.S., Fielding, J.P., "An Integrated Package for the design and construction of aircraft structural components using a parametric CADCAM method", CAD in the aerospace industry, Computing and Control Division Colloquium, IEE, April 1989.

5. Medland, A.J., "The Computer Based Design Process", Kogan Page Ltd., pp. 129-141, 1986.

6. Swift, K.G., "Knowledge-Based Design for Manufacture", Kogan Page Ltd., 1987.

7. Saggu, J.S., "Knowledge-Based Systems in Engineering", BCS SGES-South East, April Workshop held at the College of Aeronautics (AI Group), Cranfield Institute of Technology, Cranfield, Beds, UK., Workshop leader: Saggu, J.S., 1990.

8. Barr, A., Feigenbaum, E.A., "The Handbook of Artificial Intelligence", Vol.I, II, and III, Los Altos, California, USA., William Kaufmann Publisher, 1981.

9. Shapiro, S.C., "Encyclopedia of Artificial Intelligence", New York Chichester, Wiley, J., Publishing Company, 1987.

10. Tong, S.S., "Design of Aerodynamic Bodies Using Artificial Intelligence and Expert Systems Techniques", AIAA Paper 85-0112, Reno, Nev, 1985.

11. Nicklaus, D.J., Tong, S.S., Russo, C.J., "Engineous: A Knowledge directed computed aided design shell", IEEE, 1987.

12. Kempf, K.G., "Manufacturing and Artificial Intelligence, McDonnell Douglas Automation Company", Bernold, T., Albers, G., (Editors)., Artificial Intelligence Towards Practical Applications, pp.1-17, 1985.

13. Woodward, J.A.J., Corbett, J., "An expert system to assist the design for manufacture of die cast components", ICED '89, C377/088, pp.959-73, 1989.

14. Swift, K.G., Matthews, A., Firth, P.A., Syan, C.A., "Artificial Intelligence in Engineering Design", Proceedings of the 13th International CAM-I Conference, Florida, USA, pp.521-30, Nov 1984.

15. Alsina, J., Aninos, A., Erol, N., Fielding, J.P., Morris, A.J., "Progress Towards a Commercial Aircraft Design Expert System", DGLR Annual Conference, Berlin, Oct 1987.

16. Smithers, T., "Artificial Intelligence and Product Creation: An AI-Based View", DAI Research Paper No.342, Workshop on Expert Systems and their Impact on Human Conditions, Zurich, 21-23 Nov 1987.

17. Rosenfeld, L., "ICAD System^{TM}", Many articles from ICAD Inc., Suite 3250, 400 West Cummings Park, Woburn, Massachusetts 01801, USA, 1985.

18. Goldberg, A., "Smalltalk-80: The Interactive Programming Environment", Addison-Wesley, New York, 1983.

19. Saggu, J.S., "IKADE for Aircraft Wing Artefacts (Flaps)", College of Aeronautics (AI Group), Cranfield Institute of Technology, Cranfield, Beds, UK, July 1988.

20. Saggu, J.S., "IKADE Prototype User Manual", College of Aeronautics (AI Group), Cranfield Institute of Technology, Cranfield, Beds, UK, August 1990.