An expert system for automatic design-for-assembly.

Journal:	*Assembly Automation*
Manuscript ID:	draft
Manuscript Type:	Original Article
Keywords:	Design for Assembly < Assembly, Design for Manufacture < Assembly, Assembly < Industrial Robotics

For Review Only

An expert system for automatic design-for-assembly.

Structured Abstract

Purpose - A Multi Expert System is presented that can analyse a design and provide designers with suggestions for improvement and changes to designs at an early stage in order to improve assembly later in the manufacturing process. The Multi Expert System can analyse a design and provide designers with suggestions for improvement and changes to designs at an early stage in order to improve assembly later in the manufacturing process.

Design/methodology/approach - The whole system consists of four expert systems: Computer Aided Design (CAD) Expert, Automated Assembly Expert, Manual Assembly Expert and Design Analysis Expert. The Design Analysis Expert includes a sub-system to collate the information from the Assembly Experts and to provide costs and advice

Findings - The approach and the systems can reduce manufacturing costs and lead times.

Research limitations/implications - A knowledge-based reckoning approach to design-for-assembly automation is used. The approach and systems can reduce manufacturing-costs and lead-times. The system can estimate assembly-time and cost for manual or automatic assembly and select suitable assembly techniques.

Practical implications - The system can estimate assembly time and cost for manual or automatic assembly and select a suitable assembly technique.

Originality/value - The new system models assembly, product and process design using a natural approach for capturing intelligence. The new approach categorised automated assembly and manual assembly into separate individual experts. Intelligence and knowledge from each was captured and embedded within the individual expert that represented the process. This approach enabled greater flexibility and made the sub-systems easier to modify, upgrade, extend and reuse.

For Review Only

An expert system for automatic design-for-assembly.

General Abstract

A Multi Expert System is presented that can analyse a design and provide designers with suggestions for improvement and changes to designs at an early stage in order to improve assembly later in the manufacturing process. The whole system consists of four expert systems: Computer Aided Design (CAD) Expert, Automated Assembly Expert, Manual Assembly Expert and Design Analysis Expert. The Design Analysis Expert includes a sub-system to collate the information from the Assembly Experts and to provide costs and advice. A knowledge-based reckoning approach to design-for-assembly automation is used. The approach and the systems can reduce manufacturing costs and lead times. The system can estimate assembly time and cost for manual or automatic assembly and select a suitable assembly technique.

Keywords: design, assembly, automation, knowledge-expert, knowledge-base, expert system.

For Review Only

1 INTRODUCTION

Some existing Systems have provided assistance to designers during early design stages in order to reduce product development time. A limitation of these was that they modelled product and process design as a whole, which made them inflexible for expansion when new processes were needed [1]. The new system described here was created to overcome this by modelling assembly, product and process design using a natural approach for capturing intelligence. The new approach categorised automated assembly and manual assembly into separate individual experts. Intelligence and knowledge from each was captured and embedded within the individual expert that represented the process. This approach enabled greater flexibility and made the sub-systems easier to modify, upgrade, extend and reuse.

The Multi Expert System can analyse a design and provide designers with suggestions for improvement and changes to designs at an early stage in order to improve assembly later in the manufacturing process. The whole system consists of four expert systems: Computer Aided Design (CAD) Expert, Automated Assembly Expert, Manual Assembly Expert and Design Analysis Expert.

The Design Analysis Expert included a sub-system to collate information from the Assembly Experts and provide costs and advice. A knowledge-based reckoning approach to design-for-assembly automation was used. The approach and systems can reduce manufacturing-costs and lead-times. The system can estimate assembly-time and cost for manual or automatic assembly and select suitable assembly techniques.

For Review Only

Product design lead-time has improved with the introduction of Computer Aided Design / Computer Aided Manufacturing systems (CAD/CAM) [1] but little research has been completed on designing products to make them easy to assemble with automated machinery (possible robots) during the early stages of the design process (with the possible exception of Shehab and Abdalla [2]).

Many systems use a Concurrent Engineering (CE) approach. Prasad realised that integrating a variety of life-cycles into the design process through CE systems reduced product development time [3]. Many authors have proposed different approaches for CE expert systems that could assist both experienced and inexperienced designers during early design stages and reduce product development lead-time [4-6]. These systems were designed to model the product and process-design as a whole, which made the systems inflexible for expansion when a new process such as Finite Element Analysis needed to be included in the system at a later stage.

The assembly process is one of the most important processes to affect product quality, lead-time, and cost [1, 2, 7]. More than 70% of production costs are determined during the conceptual design stage [8, 9]. Additionally, assembly cost often accounts for over 40% of manufacturing cost [10–12].

So, it was important to consider assembly early in the design process [1, 2, 7]. Design engineers do not usually have a deep understanding of assembly automation and robotics [2, 13]. They would benefit from an efficient system to support them during the design process [2, 7] and this paper describes a new system that attempts to do that.

There are overlapping considerations in design for manual assembly and design for automated or robotic assembly but there are also differences [2, 7]. Those differences are mainly due to the difference in ability between human-beings and machinery. Something that might be simple for a human-being might be impossible for a robot [2, 7, 14] (and vica versa).

Much of the research literature describes product and process design methods [14], analysis and evaluation of the ability to assemble products [15], automated assembly [16], analysis of assembly [17], sequence planning [1, 18, 19], computer simulation [20] and joining processes etc [21]. That said, according to Shehab and Abdalla [2], the best known method for design-for-assembly automation is Boothroyd [14] and for evaluation, Hitachi [15]. A disadvantage of Boothroyd is that analysis is complicated and time-consuming [2]. The procedure only considers functional analysis and not manufacturing-cost; costs associated with assembling complicated components could over-shadow any advantages in reducing other assembly costs. Hitachi is suitable for mass production but the costs of parts handling and orientation are not considered and assembly costs are vague [2].

Review papers concerning design-for-assembly include [22, 23] and descriptions of the development of knowledge-based expert systems applied to automation [24, 25] and design for automation include [1, 10, 26–28]. Generally, systems described in the literature include a design sub-system (computer-aided-design), a knowledge-acquisition-&-storage-system, and an inference-engine. The Knowledge-based design system at Lucas Engineering defines an assembly-sequence and analyses each component for ease of part-handling and assembly [2]. It also considers gripping surfaces. Daabub and Abdalla [27] presented an intelligent system for product design-for-

For Review Only

assembly that enabled designers to reduce component numbers and select assembly methods. So far, little research has been completed on product design-for-assembly automation at an early stage of the design process, with the possible exception of Tan [1], Shehab and Abdalla [2] and Sanders [7].

Few had considered the redesign of products to make them easier for a robot to assemble [2]. Most considered a completed design rather than considering assembly during early stages of design [7]. At later stages, redesign is more costly and lead-times can be increased [2]. Often designers needed to address questions about assembly that they could not answer [1].

This paper presents a design-for-assembly system that addresses the disadvantages of other systems. The new system is based on a design to cost system that was developed by Shehab and Abdalla [2, 8, 29-31]. The new system unifies cost models and design-for-assembly automation into an integrated system. The new system can estimate assembly cost; select a suitable assembly technique at an early design stage, analyse designs for the ability to automate the assembly process and provide feedback to designers (including recommendations to simplify assembly).

2 SYSTEM ARCHITECTURE

A new CAD Expert was created to interpret CAD drawing files. Overlaps and assembly points were detected that used an entities comparison technique [1]. Automated Assembly was modelled with information on parts and materials and a score was allocated. The method was successfully tested and used to change designs in order to improve assembly. This was incorporated into an Automated Assembly Expert. Once this had been completed and tested then a Manual Assembly Expert was created, initially by copying the Automated Assembly Module with one machine and tool and then modifying the machine and tool to represent human factors.

For Review Only

The new method used a novel “Expandable Knowledge Base” that allowed knowledge to be expanded over time and experience. In addition, this technique enabled a suggestion or an estimated answer to be calculated from existing knowledge even if the specific knowledge required was not available. Assembly operations were then determined. Finally, the design-for-assembly system was tested with a real product and a Fanuc S-700 robot. Robot programs were generated and compared with human assembly and against predicted time to perform tasks.

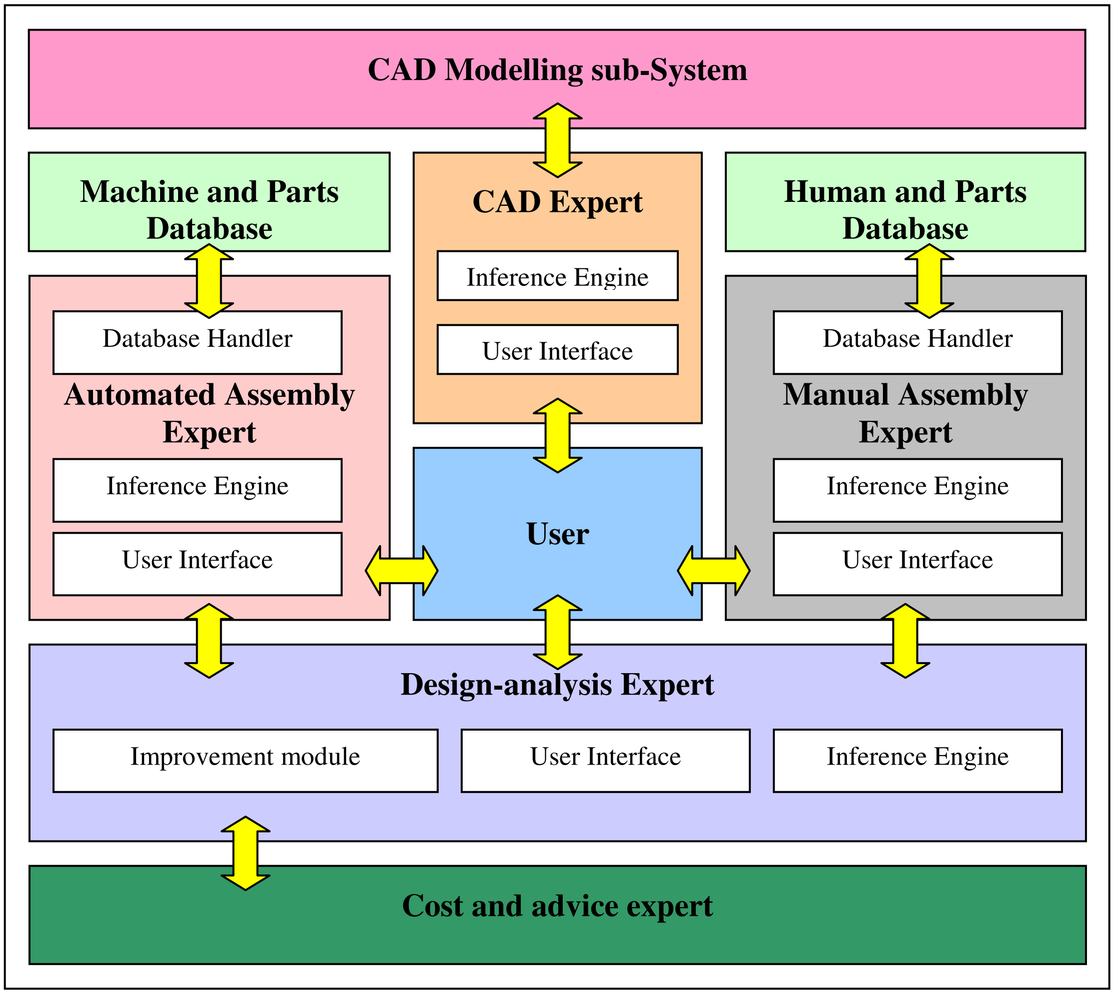


Fig. 1 Architecture of the whole system

The system architecture is shown in figure 1. It included a CAD Expert, an Automated Assembly Expert, a Manual Assembly Expert and Design-Analysis Expert. The design analysis expert considered assembly using a set of available robots within the Automated Assembly Expert and the human model in the Manual Assembly Expert.

The steps involved creating a:

- CAD Expert to interpret CAD drawing files and extract assembly data.
- User Interface (updated as new experts were added).
- Automated Assembly Expert.
- Human Assembly Expert.
- Design Analysis Expert.

Then after technical testing of the new system, the research:

- Integrated the Experts with the CAD system through the CAD Expert.
- Created a Cost and advice expert to work with the Design-analysis Expert.
- .Automatically generated time and cost predictions for Automated and Manual Assembly.
- Compared predicted times and costs with actual times and estimated costs using a Fanuc S-700 robot within an assembly cell at the University of Portsmouth.

The system allowed designers to analyse and / or modify products at any stage of the design process. The designer communicated with each sub-System via a window showing each user interface in use; that might be only one user-interface or up to four user-interfaces. For example, at the beginning of the design process only the CAD Expert was used. Later, the CAD Expert, Automated Assembly Expert and Design Analysis Expert might all be in use (and therefore displayed to the user) at the same time.

Designers specified basic product specifications such as production-volume and number of components. These data were used to select an assembly technique. The system then analysed the design for both assembly methods and presented the design analysis back to the user. The system applied design criteria for robotic assembly (using the models of robots and machines in its database) and for human asembly. The design improvement module automatically identified

For Review Only

components that may require some redesign to make the assembly more efficient, indicated features that could not be assembled by robots in the database and suggested alternatives for redesign.

2.1 Computer Programming Platform and Language

A number of software modelling methodologies were considered. This led to a review of modelling languages. That review is outside the scope of this paper but can be provided on request. In general, computer programming languages were classified into three, as shown in Figure 2.

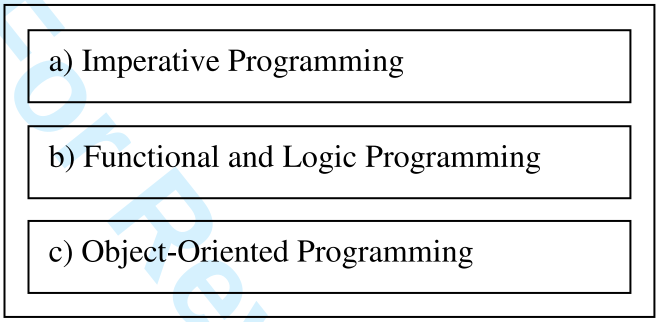


Fig 2 Computer Programming Languages classification

For Review Only

The Object-Oriented Model was selected. That was similar to an Incremental-Iterative sub-model except that it was based on objects rather than components. Objects were smaller and more generic than components (which were designed to meet specific requirements).

The history of Object-Oriented languages goes back to the mid-1960s when Ole Dahl and Kristen Nygaard in Norway created Simula [32]. However, they first came to the attention of the world in 1971 with Smalltalk. Figure 3 shows the genealogy of Object-Oriented languages, which begins with Simula. The concept was later implemented by other imperative languages such as C, Pascal and Visual Basic. The concept was a programming paradigm based on objects. The computation method used was similar to imperative languages where data was manipulated in a stepwise or sequential method. It was distinguished from Imperative languages by its object boundaries, where both data and functionality could be contained in an object. In this work, a system was modelled based on identifying real-world objects. Objects identified were created and assembled together to interact with each other to create a system [7].

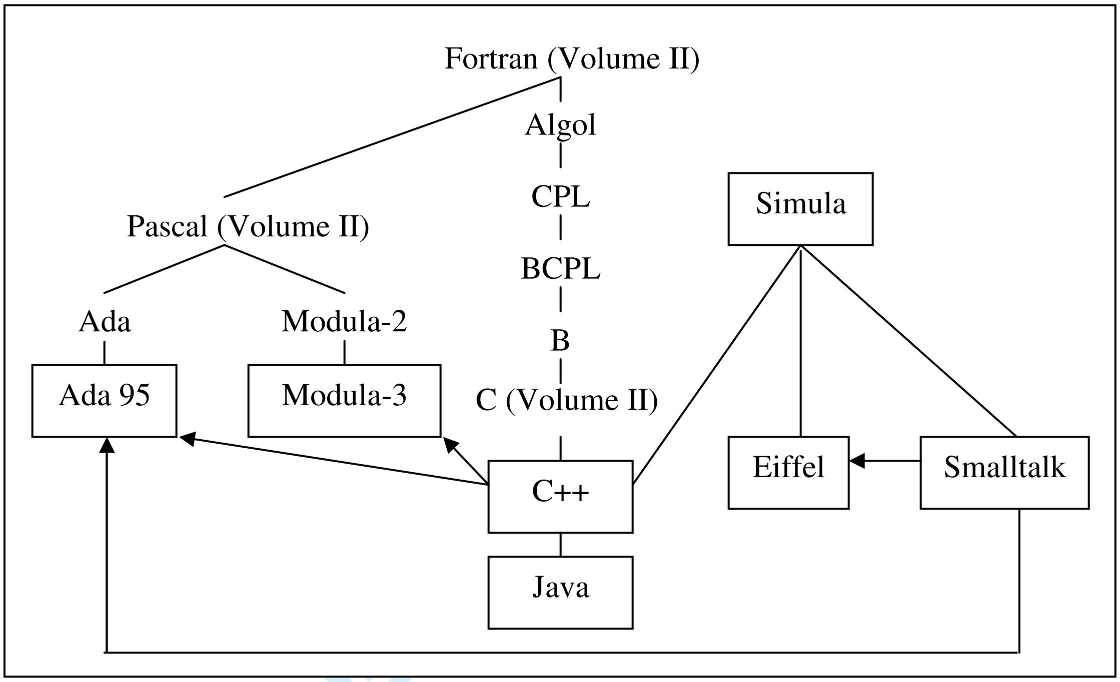


Fig 3 Family of Object-Oriented Programming Languages

[*reproduced from Tan [1]*]

Software modelling aimed to replicate tangible or intangible objects identified from real world systems. A strength was its flexibility to support iteration within phases and parallelism between phases [33]. Conversly, this flexibility sometimes led to undisciplined software development during the research as work moved more randomly between phases. Software development based on this approach was less risky and programs created were more flexible and reusable [34-36], compared to the Waterfall [37] or Spiral & Incremental-Iterative models [33]. The programming platform selected to create the software was the "Microsoft .NET" programming platform [1,7].

In general, the difference between programming in AI and Imperative languages was that in Imperative languages, a task was carried out by programs written in terms of "how to". AI programming allowed programmers to define a task in term of "what to" and the system would find out the answers using Logic or Functional reasoning. Therefore, AI languages have sometimes been considered as even higher-level languages than imperative languages [38]. The advantages of an Object-Oriented language over Imperative languages were:

- Software design was easier [1].
- Development risks were reduced for complex systems [39].
- Modelling of a software system could be based on real-world objects; hence it appeared more natural and easier to understand and explain [1].
- Software created could be maintained and upgraded easily [34].
- Object classes could be reused [35]

For Review Only

Object-oriented programming systems had several characteristics such as data abstraction, inheritance and modularity. Inheritance enabled designers to define a specific value into a higher class: each could be inherited by the lowest class of the hierarchy. Using such a technique, design and assembly techniques, such as manual assembly, automatic (robotic) assembly could be organized into classes.

Some popular AI methodologies were reviewed as part of the research and in producing a viewpoint paper [40]. That review is also available from the author. Five methodologies were adopted in this research: Rule-Based, Knowledge Based, Database, Intelligent Agents and Object-oriented. It was anticipated that greater use would often be made of hybrid tools for demanding tasks, combining the strengths of two or more of these methodologies [41].

Figure 4 shows the basic structure of the expert systems used. Each expert consisted of three levels:

- Data
- Controller
- Knowledge [42-44].

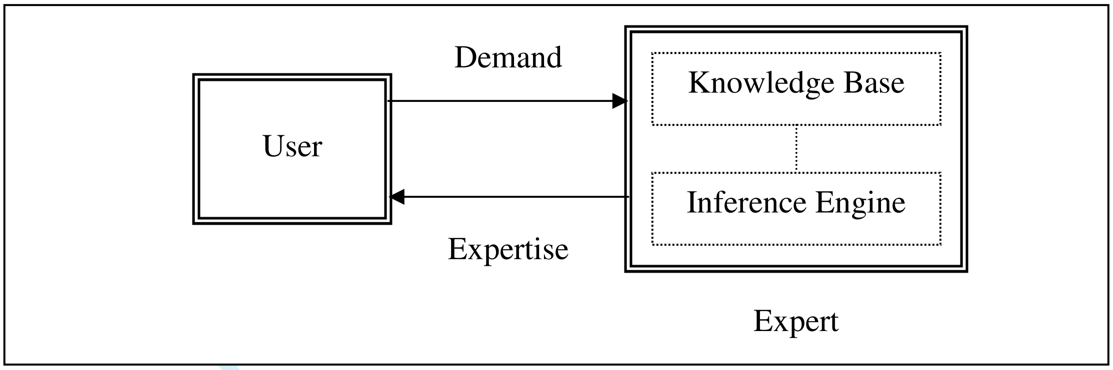


Fig 4 Basic structure of the expert systems

[*reproduced from [4], cited by [42]*]

The way in which these levels were used in this work are described and then sub-Systems are presented along with a discussion of product design improvement for robotic assembly. 1922 rules were created during the research and distributed within the experts. Within each expert, rules were connected so that the conclusion of one rule is included in the premise of another.

2.1.1 Data level.

User-interfacing was the main element of this level. It dealt with communications between the user and the system (noting that the user could be another expert system or a human designer). The data level allowed an inquiry or problem to be presented. A solution was then returned.

2.1.2 Controller level.

This level was the Inference Engine, used to guide the manipulation of knowledge in a Knowledge Base (KB). It consisted of a methodology for formulating solutions and inferential reasoning the information in the KB.

For Review Only

2.1.3 Knowledge level.

Each KB was used for storing rules and facts about the particular knowledge domain created in the level. The knowledge could be classified into: declarative and procedural. Facts, laws or terminology within a specialist domain constituted the declarative knowledge, emphasising concepts and their relations to other concepts. Declarative knowledge specified the actual answer for a problem or task whereas the procedural knowledge specified how to solve a given problem or task rather than specifying the actual answer.

2.2 CAD Expert

This was an expert created to interpret CAD drawing files. It was used to extract drawing entities including assembly data. Once all the entities were extracted, then these were assembled together to form closed shapes and to check assembly areas and volumes. This expert was an extension of work by Tan [1] and Rasol [45].

The simple user-interface initially created within a CAD Expert was not user-friendly and only provided data for technical testing. A more general and user-friendly user-interface was developed from that initial interface so that a user (even a new user) could use the system efficiently. The architecture of the CAD Expert is shown in figure 5.

2.3 User Interface

Once redeveloped then the CAD Expert opened in a window within the User Interface. Later, different experts had different user-interface windows [46] so that design analysis and recommendations for redesign suggestions could be displayed (and selected) on separate parts of a main screen [47] or on separate screens [48].

In terms of the criteria for assessing user-interfaces shown in figure 6, the selected programming environment possessed tools and functions to easily create a Graphical User Interface.

For Review Only

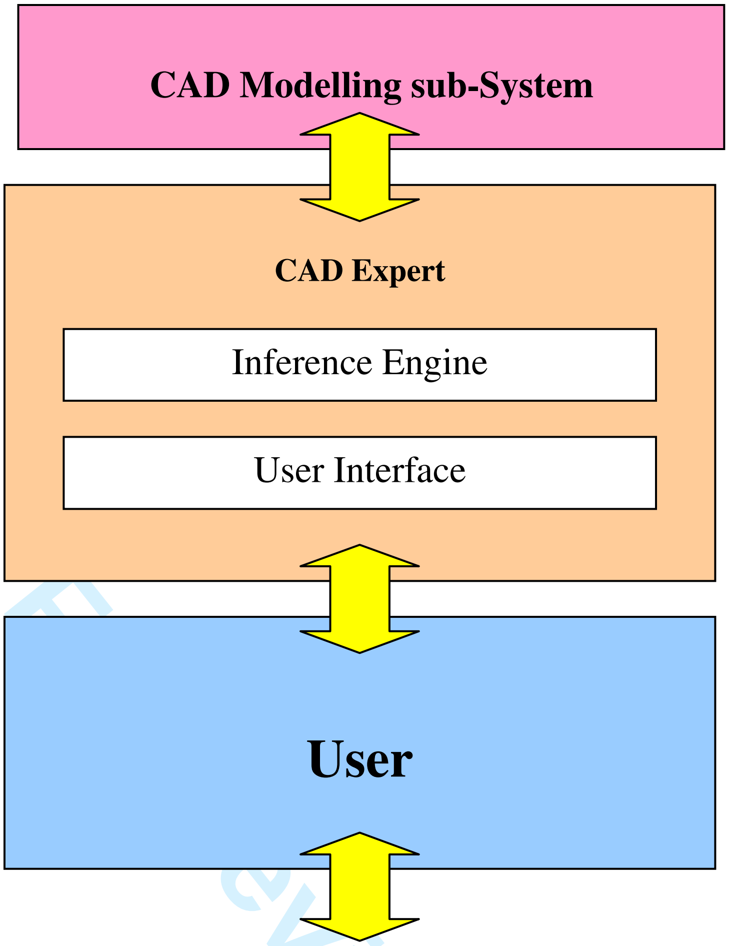


Fig. 5 CAD Expert Architecture

For Peer Review

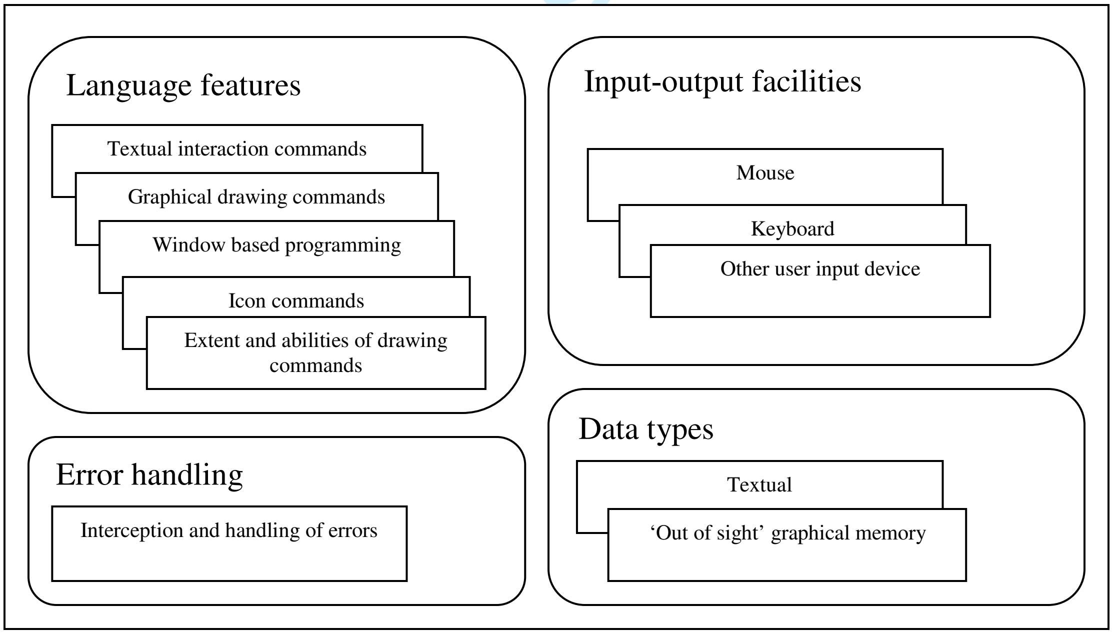


Fig 6 – Criteria for Assessing user interfaces

[*reproduced from Tewkesbury [46]*]

Testing of the user-interface was similar to [46, 47].

2.4 Automated Assembly Expert.

The Knowledge Base within the Automated Assembly Expert contained knowledge to provide assembly selection. The knowledge base was built using open literature and design handbooks [14, 23, 49, 50] in a similar way to Shehab and Abdalla [2, 8]. Other sources came from consultation with manufacturing experts in collaborating companies and manufacturing academic staff at Portsmouth University. Hybrid knowledge representation techniques were employed to represent component feeding, handling, and insertion.

Rules in the Automated Assembly Expert were connected so that the conclusion of one rule was included in the premise of another. For example, a set of Automated Assembly rules that produced a suitable outcome were:

> IF (tool-a) AND (machine-A) AND (tool-a is in place) AND (machine-A is in place) AND {(machine-A can reach tool-a) OR (tool-a can move to machine-A)} AND (parts handling YES) AND (orientation YES)
>
> THEN (Automated Assembly = YES).

The Automated Assembly Expert contributed towards design feasibility simulations by providing information to the Design-Analysis Expert to allow that expert to make improvement suggestions to allow robotic assembly to take place. For example, considering the automated assembly rules listed above… a negative return would be flagged to the User Interface to say that assembly was not possible if tool-a or machine-A were flagged as not available or if machine-A could not reach tool a and tool a could not move to machine-A. The information was sent to the Design-Analysis Expert.

For Review Only

Similarly, if parts handling devices could not deliver and remove parts or the orientation of the parts made assembly impossible then that would be flagged to the User Interface and sent to the Design-Analysis Expert.

The architecture of the Automated Assembly Expert is illustrated in Fig. 7.

Design criteria for robot assembly were applied for each component or sub-module. Assembly properties are listed in Table 1.

Assembly properties were then evaluated for each component, sub-module and the module for robotic assembly. Component, sub-module and module properties are shown in Table 2.

For Review

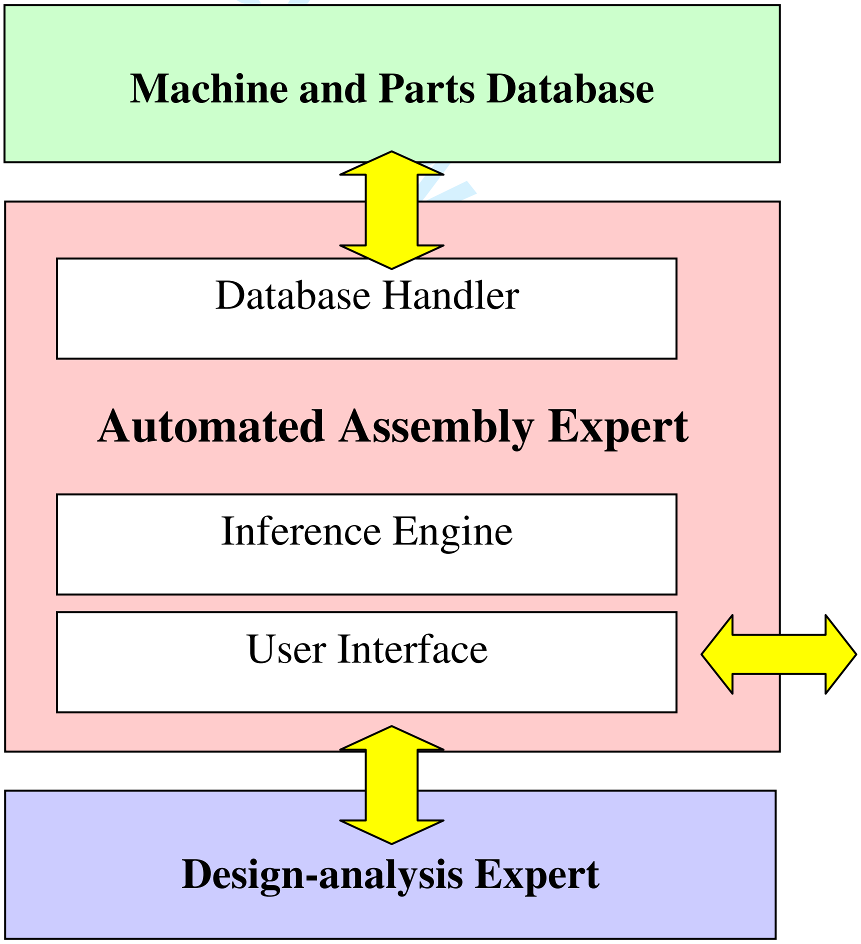


Fig. 7 Architecture of the Automated Assembly Expert

Feed
Orientation
Alignment
Insertion

Table 1 Assembly properties

For Review Only

Module	Number of sub-modules
Module	Base
Module	Dimensions
Sub-Module	Number of components
Sub-Module	Shape
Sub-Module	Size
Sub-Module	Symmetry
Sub-Module	Material
Sub-Module	Quality
Sub-Module	Weight
Sub-Module	Joining method
Sub-Module	Accuracy
Sub-Module	Force
Component	Shape
Component	Size
Component	Symmetry
Component	Material
Component	Quality
Component	Weight
Component	Joining method
Component	Accuracy
Component	Force

Table 2 Component, sub-module and module properties

YES or NO status flags were set automatically for each property based on robotic assembly criteria for a particular robot. An overall YES/NO status was set and a score representing the ability of the robot to complete an assembly task was derived for the Design-Analysis Expert.

The first database created was for the Automated Assembly Expert and that was later used as a model for the creation of the other databases (*the database for the Manual Assembly Expert was an exact copy*). It included details about robots (*initially only one but later two*) and the various components, sub-modules and modules.

For Review Only

2.5 The Design-Analysis Expert.

The architecture of the Design-Analysis Expert is shown in Fig. 8.

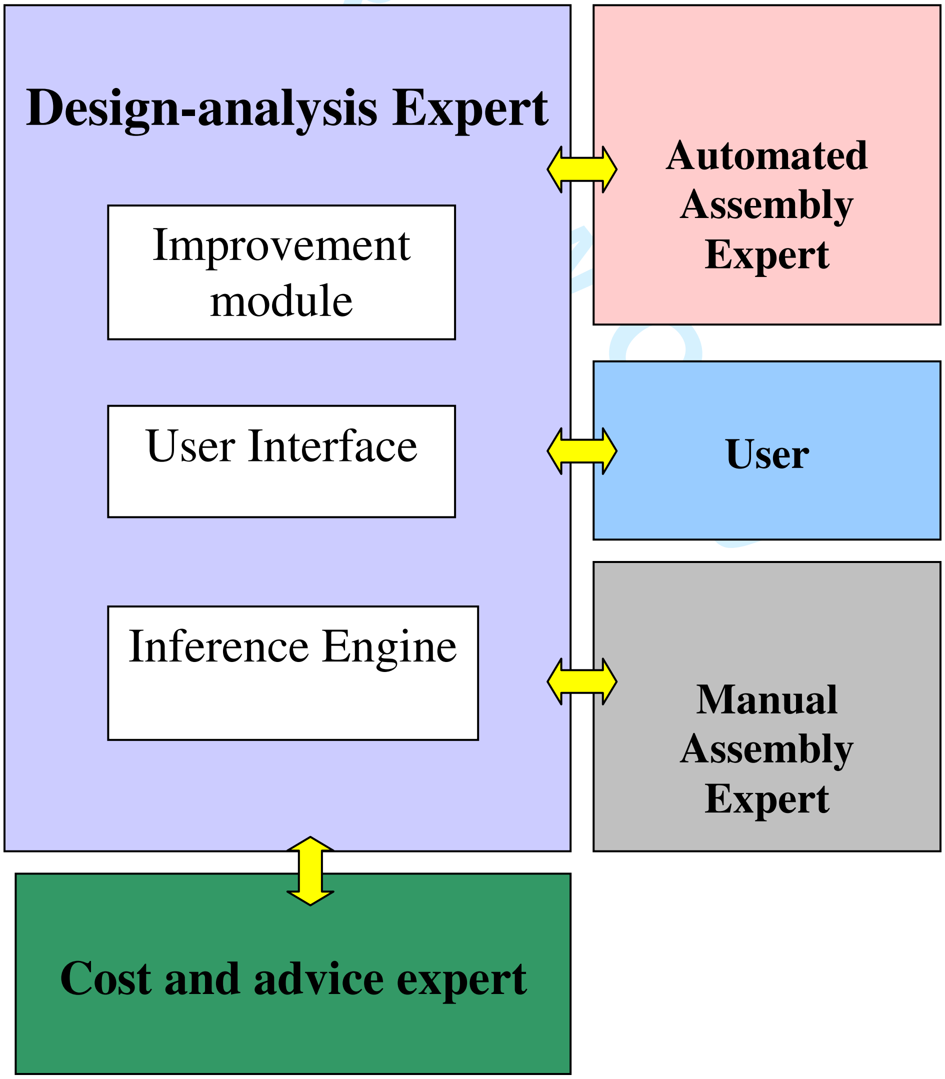


Fig 8 Architecture of the Design-Analysis Expert

Knowledge bases within the expert contained knowledge to provide assembly time and cost estimation, design analysis, and heuristics of redesign suggestions. Design-Analysis Expert rules were also inter-connected. Considering the example used in Section 2.4, a simple example of rules used to estimate assembly time was:

IF (tool-a) AND (machine-A) AND (parts handling YES)

THEN (Automated Assembly Time = AATaA).

The Expert provided design improvement suggestions that could reduce time and / or cost for robotic assembly or that could rearrange things to make assembly possible. For example, if Automated Assembly = NO for a design, then rules were interrogated. If tool and machine were available but machine-A could not reach tool-a then another tool could be suggested that was available but that might require a design change. If the tool could reach or move to the machine then different orientations might be considered. Alternatively a different machine could be suggested that could complete the operation.

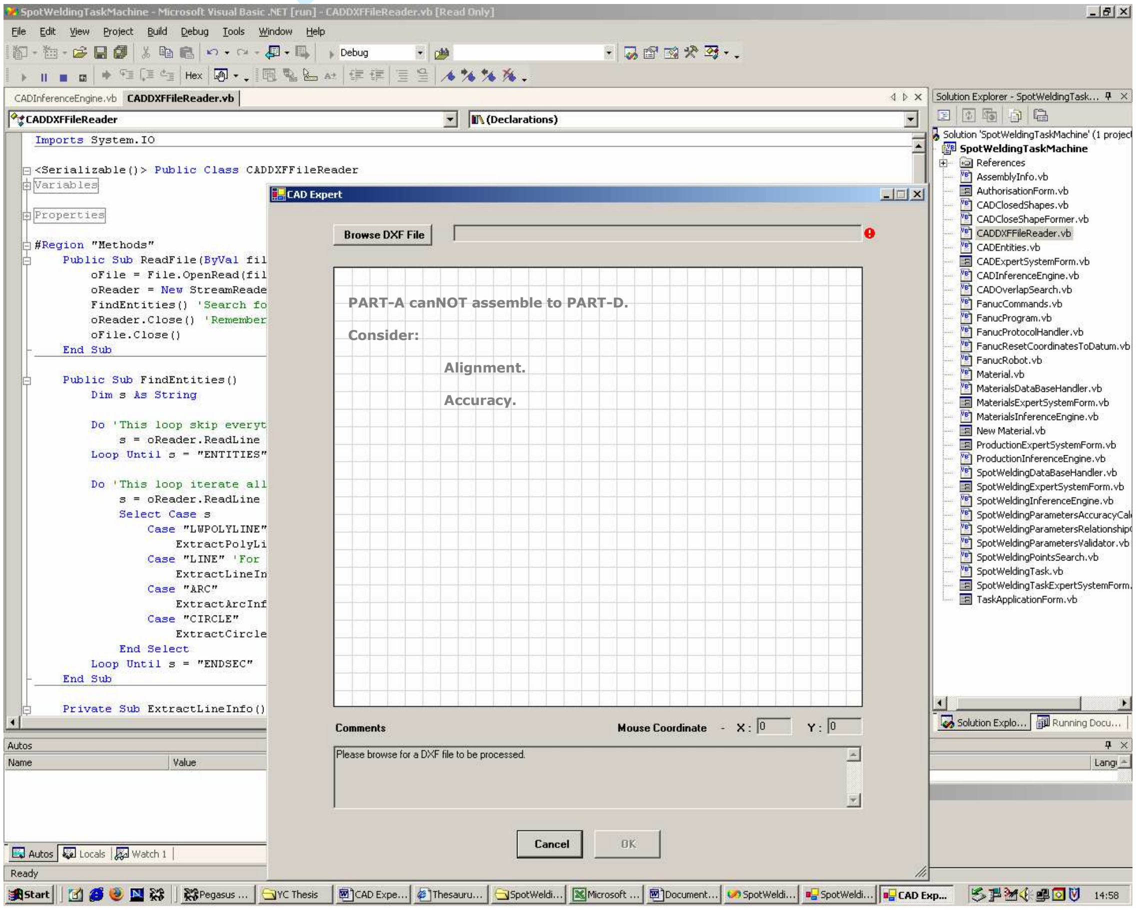
Feedback was displayed on the user-interface. The designer could redesign to improve the ability of the robots in the database to assemble the module. Suggestions for changes to design depended on knowledge gained from experienced engineers and needed several iterations and versions (and this could be improved indefinitely). In suggesting redesign, the expert firstly considered problems with component assembly for each sub-module and provided suggestions for redesign to simplify robotic assembly. The database of possible suggestions for improvement became so large and complicated that a separate expert was created to consider advice (that expert was also used later to provide cost information for the Design-analysis expert). Options for redesign are those listed in Table 2.

For Review Only

The report included a series of design modifications to assist in simplifying the assembly process. The design suggestions were initially displayed for a component and then once the components had been cleared, then for a sub-module, and then for a module. That completed advice to redesign a product.

The Design-analysis Expert provided two sorts of suggestions. First Level Suggestions were made for changes that would allow a product to be assembled if assembly was not possible at all. Second Level Suggestions were made to make design changes to improve assembly once assembly was possible. An example of a First Level re-design suggestion is shown in Fig. 9.

For



```
Imports System.IO

<Serializable()> Public Class CADDXFFileReader
Variables

Properties

#Region "Methods"
    Public Sub ReadFile(ByVal fil
        oFile = File.OpenRead(fil
        oReader = New StreamReade
        FindEntities() 'Search fo
        oReader.Close() 'Remember
        oFile.Close()
    End Sub

    Public Sub FindEntities()
        Dim s As String

        Do 'This loop skip everyt
            s = oReader.ReadLine
        Loop Until s = "ENTITIES"

        Do 'This loop iterate all
            s = oReader.ReadLine
            Select Case s
                Case "LWPOLYLINE"
                    ExtractPolyLi
                Case "LINE" 'For
                    ExtractLineIn
                Case "ARC"
                    ExtractArcInf
                Case "CIRCLE"
                    ExtractCircle
            End Select
        Loop Until s = "ENDSEC"
    End Sub

    Private Sub ExtractLineInfo()
```

PART-A canNOT assemble to PART-D.

Consider:

Alignment.

Accuracy.

Please browse for a DXF file to be processed.

Fig. 9 . An example of a First Level re-design suggestion.

3 ANALYSING A PRODUCT FOR DESIGN-FOR-ASSEMBLY

Once the product specifications had been defined then the CAD Expert interfaced with the CAD modelling system to extract and create initial assembly information. The two Assembly Experts selected assembly techniques for that product. The human user could select between manual assembly and automated assembly and the Design-analysis Expert provided cost and time data for both. In practice an early selection was often made by designers testing and using the system because design-for-assembly requirements for each type of assembly differed, and other factors may also influence the decision, for example forces required to assemble [51] or previous work scheduling within a factory [1,2]. A diagram showing the general flow of information for the system is shown in figure 10.

Production rules had been developed for each Assembly Expert and they provided data to assess components in the design for ease of insertion and / or handling. Considering the Automated Assembly Expert, once automatic handling and insertion had been approved by the Automated Assembly Expert then assessment within the Design-Analysis Expert depended on assembly time and machine running cost for each component, sub-module or module. The Design-Analysis Expert considered design feasibility and provided suggestions for changes to the design for easier assembly. Design criteria for robotic assembly were applied to each component, sub-module and module and then evaluated. The system automatically identified components that could not be assembled and suggested First Level Design Changes. Once component assembly was possible then sub-modules that could not be assembled were identified. Once component and sub-assembly was possible then components and sub-modules that were the most difficult to assemble were identified and further Second Level design changes were suggested for redesign.

The system provided designers with an ability to select economic assembly techniques at an early stage for a product based on initial product specifications, production data and available equipment. Designers do not usually know enough about assembly automation and robotics to create designs that can be easily assembled [2]. The system analysed product designs and provided suggestions

For Review Only

for changes that would simplify later assembly of the product [7]. The system also estimated assembly time and provided an estimate of cost for various assembly techniques.

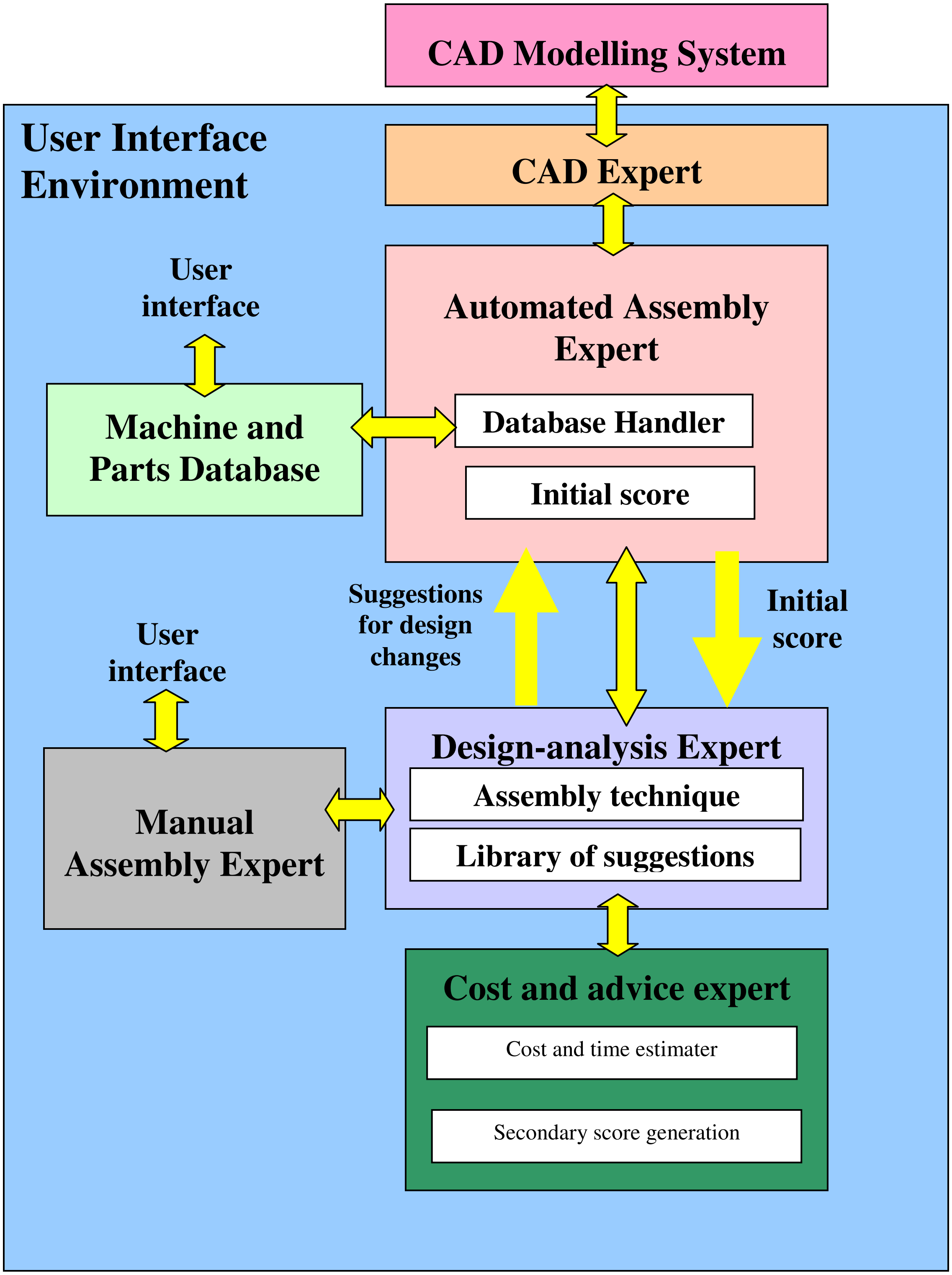


Fig 10 General flow diagram for the system

4 TESTING WITH A NEW SIGNATURE CAPTURE DEVICE

A new Signature Capture Device was being created at a collaborating company. That product is considered as an example and the constructed CAD Drawing is shown in figure 11.

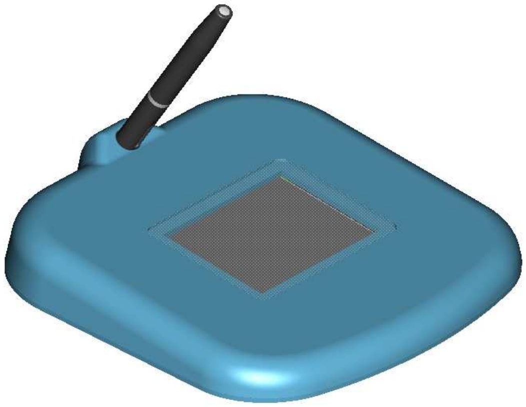


Fig 11 Signature Capture Device

For Review Only

Modular design techniques were used at the collaborating company to identify:

- Module Selection.
- New modules to be created.
- New modular circuit design and prototyping.
- PCB generation.
- Interfacing between modules.
- Constraints for new modules.

The new system then analysed the prototype product design for ease of assembly, recommended an assembly technique and provided associated time and cost for assembly. Components and sub-modules that needed to be redesigned in order for a sub-module or module to be assembled were identified and suggestions for changes were generated. Then Second Level suggestions considered improvements to the design for later ease of assembly.

The Signature Capture Device consisted of two sub-modules that existed already from previous products at the collaborating company: a digitiser and an LCD Screen. A new interface circuit with nine components (including sockets) was added to interface between the sub-modules. Finally, four screws secured the casing. Once the outer casing was closed then internal shaping held parts in place.

A first requirement was to consider assembly of electrical components onto the circuit board. After that, the sequence of assembly operations was determined by the order of placement of the sub assemblies within the casing.

Electrical components and the sockets were mounted on one side of the circuit board. The Design-analysis Expert approved use of either Automated Assembly or Manual Assembly but at the Second Level, suggested aligning the resistor and capacitor components. This was done and the manufacture of the PCB was then out-sourced to sub-contractors.

The processes to assemble the product were:

1. Mount front casing onto a fixture tool.
2. Insert
    - 2a. New interface PCB.
    - 2b Digitiser.
    - 2c LCD Screen.
    - 2d Back casing.
    - 2e (i – iv) Screws.
3. Tighten screws.

For Review Only

Fig 12 Front casing with the sub modules mounted under the LCD Screen Module.

For Review Only

The assembly is shown with the back of the case removed in figure 12.

4.1 Selecting assembly method for the Signature Capture Device

The collaborating company already used a modular design system to generate early prototypes by reusing existing sub-modules. That avoided using separate discrete components that would tend to lead to higher design and assembly costs. If new components were needed after that process then these were combined to create a new sub-module (as was the case for the new Interface PCB) or existing sub-modules would be modified to incorporate them.

Some operations that were easy for a human assembly worker may be difficult for automated machines (such as a robot) [2, 8, 14]. The system recommended assembly techniques early that were possible and then provided information on the most cost effective assembly method. The selection of the assembly technique depended on production volume, production quality, required quality, number of production shifts, number of sub-modules, number of components, market, market life, costs and overhead expenditure.

For the Touch Screen GPS these were:

Production volume	A	per year.
Production quality	Medium	(Low/Medium/ High).
Required quality	High	(Low/Medium/ High).
Number of production shifts	3	per working day (five days per week).
Number of sub-modules	4	
Number of components	0	
Market	High	(Low/Medium/ High).
Market life	4	years.
Costs	Medium	(Low/Medium/ High).
Overhead expenditure	B	pounds.

Actual values for A and B were commercial in confidence.

For Review Only

Procedures for assembly technique selection [7] were similar to those described by Shehab in his PhD Thesis [22] and suggested by Tan in his PhD Thesis [1].

4.2 Design analysis expert

The Design-analysis Expert then used the Cost & Advice Expert to estimate assembly costs and to provide suggestions for changes. Signature Capture Device sub-modules were considered for ease of robotic and human assembly by considering placement, assembly direction and orientation, and component insertion.

4.2.1 Cost estimation

Available parts-handling devices included a programmable feeder, programmable conveyor carrying sub-modules on pallets or trays, and a manual magazine or separate manually delivered pallets and trays. The user needed to select a parts-handling device to deliver sub-modules and components. Estimating costs for human and robotic assembly required costs for purchasing, using and running equipment. The Cost & Advice Expert considered costs of: initially one (and later three) assembly robots with grippers, controllers and sensors; operating times; parts-placement and transfer. The system only considered one workstation but that could easily be expanded. A user could estimate the cost of new designs and investigate alternate materials or processes. The various cost estimation techniques were similar to those presented in [22].

4.2.2 Suggestions for design changes

Although all the assembly operations were possible by a human operator, two assembly operations were highlighted for automated assembly: Add Digitiser sub-module; and Add LCD Screen sub-module. Problems were not actually with the sub-modules named but with their insertion into the new interface PCB sockets. That insertion required a higher degree of accuracy than was possible with the robotic systems.

It was an assembly problem rather than a design problem!

Design of the product needed to be reconsidered if automated assembly was to be used. Suggestions for changes to the product were made without affecting functionality. Properties highlighted for improvement were alignment-difficulty and insertion-difficulty. In the original design, the two sub-modules could be inserted into the new interface PCB easily by a human operator but could not be easily inserted by a robot (if at all).

Design modifications were made to increase the size of each subsequent layer placed within the product case so that guides could easily be made in the sides of the case for the sub-modules during manufacture of the casing. A Fanuc S-700 robot was then programmed and used to demonstrate that the robotic assembly was possible.

For Review Only

5 CONCLUSIONS

The system successfully estimated assembly cost, selected assembly techniques early in the design and analysed and provided advice for useful design changes to the product. Although that was not a powerful artificial intelligence [40], it did combine a variety of technologies to give machines an ability to make some decisions.

Four main areas were successfully integrated during the research: Programming Language, Artificial Intelligence (AI) and Design Methodology. The methodologies successfully used for each were:

- Programming Language.
    - Object-Oriented Programming.
- Artificial Intelligence.
    - Rule-Based.
    - Knowledge-Based.
    - Database.
    - Intelligent agents.
    - Object-Oriented.
- Design Methodology.
- Concurrent Engineering.

The insertion of the digitiser sub-module and LCD Screen sub-module into the new PCB were difficult to automate because the tolerances required for insertion into the sockets on the interface board were too tight. Changes were required and made so that one simple motion was all that was required for each insertion. The sub module was guided into place by ledges shaped into the casing.

The system was tested using a real product and a robot and satisfactory results were obtained. Suggestions for modifications to the Signature Capture Device design were successfully made to make it easier to assemble automatically. As an example, an assembly was successfully completed.

For Review Only

REFERENCES

1 Tan Y.C. Multi expert concurrent engineering system to assist a designer. PhD Thesis, Portsmouth University, Portsmouth, UK, 2006.

2 Shehab EM and Abdalla HS. A cost-effective knowledge-based reasoning system for design for automation. Proc. Instn Mech Engrs, Part B: J. Engineering Manufacture, 2006 220, 729–743.

3 Prasad, B. Concurrent Engineering as an Integrator for Managing Product Development/Delivery Process. Int Jnl of Materials & Product Technology, 1997, 12(4-6), 279-287.

4 Gayretli, A. & Abdalla, H. S. An Object-Oriented Constraints-Based System for Concurrent Product Development. Robotics and Computer-Integrated Manufacturing 1999, 15, 133-144.

5 Pham, D. T. & Gologlu, C. (A Computer Aided Process Planning System for Concurrent Engineering. Proc. Instn Mech Engrs, Part B: J. Engineering Manufacture 2001, 215, 1117-1131.

6 Tewkesbury, G. Design using Distribution Intelligence within Advanced Production Machinery. PhD Thesis, Portsmouth: University of Portsmouth UK. 1994.

7 Sanders DA. Notes for design for automation. University of Portsmouth Rep - Mech-312-06, 2006.

8 Shehab, E. M. and Abdalla, H. S. A design to cost system for innovative product development. Proc. Instn Mech Engrs, Part B: J. Engineering Manufacture, 2002a 216(B7), 999–1020.

For Review Only

9 Asiedu, Y. and Gu, P. Product life cycle cost analysis: state of the art review. Int. J. Prod. Res., 1998, 36(4), 883–908.

10 Zha X. F., Du, H. J., and Qiu, J. H. Knowledge-based approach and system for assembly oriented design. Part I: the approach. Engng Applic. Artif. Intell., 2001a, 14(1), 61–75.

11 Li, R. K. and Hwang, C. L. A framework for automatic DFA system development. Computers Ind. Engng, 1992, 22(4), 403–413.

12 Venkatachalam, A. R., Mellichamp, J. M., and Miller, D. M. A knowledge-based approach to design for manufacturability. J. Intell. Mfg, 1993, 41, 355–366.

13 Barnes, C. J., Jared G. E. M., and Swift K. G. Decision support for sequence generation in an assembly oriented design environment. Robotics Computer- Integ. Mfg, 2004, 20(4), 289–300.

14 Boothroyd, G. Assembly automation and product design, 1992 (Marcel Dekker, Inc, NY).

15 Miyakawa, S., Ohashi, T., and Iwata, M. The Hitachi new assembly evaluation method (AEM). Trans. NAMRI/SME, 1990, 352–359.

16 Abdel-Malek L. L. Design for automated assembly of circular parts: an algorithm. Int. J. Prod. Econ., 1995, 41(1–3), 45–50.

17 Strobl, G. and Dodenhoft, D. DFA-XPERT: A knowledge- based approach for design analysis in mechanical engineering. Computers Artif. Intell., 1992 11(5), 425–456.

For Review Only

18 Barnes, C. J., Jared, G. E. M., and Swift, K. G. A pragmatic approach to interactive assembly sequence evaluation. Proc. Instn Mech. Engrs, Part B: J. Engineering Manufacture, 2003, 217(B4), 541–550.

19 Molloy, E., Yang, H., and Browne, J. Feature-based modelling on design for assembly. Int. J. Computer Integr. Mfg, 1993, 6(1), 119–125.

20 Hu, M., Lin, Z., Lai, X., and Ni, J. Simulation and analysis of assembly processes considering compliant, non-ideal parts and tooling variations. Int. J. Machine Tools Mf, 2001, 41(15), 2233–2243.

21 Brown, N. J., Swift, K. G., and Booker, J. D. Joining process selection in support of a proactive design for assembly. Proc. Instn Mech. Engrs, Part B: J. Engineering Manufacture, 2002, 216, 1311–1324.

22 Shehab, E. M. An intelligent knowledge-based cost modelling system for innovative product development. PhDThesis,DeMontfort University, Leicester, UK, 2001.

23 Boothroyd, G. and Alting, L. Design for assembly and disassembly. Ann CIRP, 1992, 41(2), 625–636.

24 Sanders DA and Hudson AD. A specific blackboard expert system to simulate and automate the design of high recirculation airlift reactors. Mathematics and computers in simulation 2000, 53 (1-2): 41-65.

For Review Only

25 Sanders DA, Hudson AD, Tewkesbury GE, et al. Automating the design of high-recirculation airlift reactors using a blackboard framework. Expert Systems with applications 2000, 18 (3): 231-245.

26 Zha, X. F., Du, H. J., and Qiu, J. H. Knowledge-based approach and system for assembly oriented design. Part II: the system implementation. Engng Applic. Artif. Intell., 2001b, 14(2), 239–254.

27 Daabub, A. and Abdalla, H. A computer-based intelligent system for design for assembly. Computers Ind. Engng, 1999, 37(1–2), 111–115.

28 Zha, X. F., Lim, S. Y. E., and Fok, S. C. Integrated knowledge-based approach and system for product design for assembly. Int. J. Computer Integr. Mfg, 1999, 12(3), 211–237.

29 Shehab, E. M. and Abdalla, H. S. An integrated prototype system for cost effective design. Concurrent Engng: Res. Applic. (CERA): A Special Issue on Cost Engineering and CE, 2001, 9(4), 243–256.

30 Shehab, E. M. and Abdalla, H. S. Manufacturing cost modelling for concurrent product development. Robotics Computer Integr. Mfg, 2001, 17(4), 341–353.

31 Shehab, E. M. and Abdalla, H. S. An intelligent knowledge based system for product cost modelling. Int. J. Advd Mfg Technol., 2002. 19, 49–65.

32 Beaudouin-Lafon, M. Object-Oriented Languages: Basic Principles and Programming Techniques. 1994. London: Chapman & Hall (ISBN: 0-41-255800-9).

For Review Only

33 Schach, S. R. Classical and Object-Oriented Software Engineering with UML and C [Plus Plus]. (4th ed.). 1999. London: WCB / McGraw-Hill (ISBN: 0-07-290168-3).

34 Chatzigeorgiou, A. Performance and Power Evaluation of C++ Object Oriented Programming in Embedded Processors. Information and Software Technology 2003, 45, 195-201.

35 Ng, K. W. & Luk, C. K. A Survey of Languages Integrating Functional, Object-Oriented and Logic Programming. Microprocessing and Microprogramming 1995. 41, 5-36.

36 Tan, Y. C., Sanders, D. & Tewkesbury, G. Control, Programming and Automation of Intelligent Production Machines using a Task Oriented Approach. Proceeding of the 2nd International Conference on Artificial Intelligence in Engineering & Technology 2004. pp. 162-166. ISBN: 9-83-264338-4.

37 Priestly, M. Practical Object-Oriented Design with UML. (2nd ed.). 2003. Berkshire: McGraw-Hill Education (ISBN: 0-07-710393-9).

38 Le, T. V. Techniques of Prolog Programming with Implementation of Logical Negation and Quantified Goals. 1993. Canada: John Wiley & Sons (ISBN: 0-47-159970-0).

39 Tan, Y. C., Sanders, D. & Tewkesbury, G. Creating a New Smart Workplace using Intelligent Task Machines. 7th International Conference on Work with Computing Systems. 2004. 120-125. ISBN: 9-83-417420-9

40 Sanders D. Progress in machine intelligence. Industrial Robot – An international journal 2008, 35 (6), 268-268.

For Review Only

41 Pham, D. T. & Pham, P. T. N. Artificial Intelligence in Engineering. International Journal of Machine Tools & Manufacture 1999, 39, 937-949.

42 Hudson, A. D. (1997). A New Blackboard Framework Applied to a HRAR Design Program. PhD Thesis. Portsmouth: University of Portsmouth, UK.

43 Stott I and Sanders D New powered wheelchair systems for the rehabilitation of some severely disabled users. Int Jnl of Rehabilitation Research 2000, 23 (3) 149-153.

44 Bergasa-Suso J, Sanders DA, Tewkesbury GE Intelligent browser-based systems to assist Internet users. IEEE Transactions 2005, 48 (4): 580-585.

45 Rasol, Z. A New Automatic System for Simple Spot Welding Tasks. MPhil Thesis, 2000, Portsmouth: University of Portsmouth, UK.

46 Tewkesbury, G. & Sanders, D. A New Robot Command Library which Includes Simulation. Industrial Robot 1999, 26(1), 39-48.

47 Sanders DA, Urwin-Wright SD, Tewkesbury GE, et al. Pointer device for thin-film transistor and cathode ray tube computer screens Electronics Letters 2005, 41 (16): 894-896.

48 Stott I and Sanders D The use of virtual reality to train powered wheelchair users and test new wheelchair systems. Int Jnl of Rehabilitation Research 2000, 23 (4): 321-326.

49 Rampersad, H. K. Integrated and simultaneous design for robotic assembly, 1994 (John Wiley & Sons, New York Ltd).

For Review Only

50 Boothroyd, G. and Radovanovic, P. Estimating the cost of machined components during the conceptual design of a product. Ann CIRP, 1989, 38(1), 157–160.

51 Sanders D. Industrial Robot – An international journal 2007, 34 (4): 268-268.

For Review Only