Cranfield University

Diem Huynh Lam

Application of MOKA Informal Knowledge Models for Building Knowledge Based Engineering Systems in Aircraft Wind Tunnel Test Models

School of Industrial and Manufacturing Science

MRes

Cranfield University

School of Industrial and Manufacturing Science

MRes Thesis

September 2004

Diem Huynh Lam

Application of MOKA Informal Knowledge Models for Building Knowledge Based Engineering Systems in Aircraft Wind Tunnel Test Models

Supervisor: Dr Ip-Shing Fan

Academic Year 2003 to 2004

This thesis is submitted in partial fulfilment of the requirements for the degree of MRes

ABSTRACT

Knowledge based engineering (KBE) applications are softwares that rely on some well defined engineering rules, relationships and logics for performing generative engineering tasks. A generative function is a function that is responsible for performing and obtaining generative tasks and solutions based on the given rules, relationships and logics.

Presently, it is considered that there is a lack of visibility, transparency, traceability and accountability when KBE applications are used. The present research programme aims to overcome this problem by developing and integrating the knowledge model with the design automation model of KBE applications. Thus, the inner working and design of the generative function with its engineering rules and relationships defined for the KBE application may be readily seen and understood by viewing the knowledge models. Consequently, this means a degree of visibility, traceability and accountability is achieved when KBE applications are used to perform generative engineering designs.

The knowledge model has been developed in accordance with the European MOKA's methodology and takes the form of an informal knowledge model (ICARE forms). The knowledge model (ICARE forms) serves the purpose of defining how products/assemblies/parts should be designed, processed and manufactured within a set of prescribed illustrations, constraints, activities, rules and entities.

The CATIA (Knowledgeware) KBE application is used by the present study and the design automation model residing on the KBE application has been developed using the Knowledgeware programming language, VBScript, macros and CAA IDL API, which allow a series of modelling and design tasks to be automated.

The research programme is validated by means of a case study involving an aircraft wing model supplied by industrial collaboration partners. The present study shows that KBE technology may be used to produce substantial commercial benefits in terms of time, cost

and speed. The study believes that the MOKA's methodology and ICARE forms may be used to capture knowledge for KBE applications but there is a limit on how well, easy and complete the ICARE forms can be used to depict the engineering rules and relationships that have been defined for the generative design function of KBE applications.

Parts of the work presented in this study have been demonstrated to the industrial collaborators and included in a consortium confidential DTI's research project grant report (DTI's ref. no. CHAD/002/00008) [1] on the use of KBE systems.

ACKNOWLEDGEMENTS

The author is immensely grateful for Dr Ip-Shing Fan's guidance and leadership over the course of the research and Pablo Bermell Garcia for his assistance with the study.

Many thanks to Airbus UK and Aircraft Research Association for providing the material for the case study.

Special thanks to staff from both Departments of Manufacturing Systems and Enterprise Integration for their cooperation.

Finally, the author would like to thank Cranfield University for awarding the author with the scholarship, without which this research project would not have been possible.

TABLE OF CONTENTS

FIGURES

TABLES

NOMENCLATURE

AI – Artificial Intelligence

AP – Application Protocol

API – Application Programming Interface

ARA – Aircraft Research Association

BOM - Bill of Material

CAA – Component Application Architecture

CAD – Computer Aided Design

CAE – Computer Aided Engineering

CAM – Computer Aided Manufacturing

CAPP – Computer Aided Process Planning

CIM – Computer Integrated Manufacturing

CNC – Computer Numerical Control

DFA – Design For Assembly

DTI – Department of Trade and Industry

EDM – EXPRESS Data Manager

ERP – Enterprise Resource Planning

FMS – Flexible Manufacturing System

GUI – Graphic User Interface

ICARE – Illustration, Constraint, Activity, Rule, Entity

IDL – Interface Definition Language

IGES – Initial Graphics Exchange Specification

JIT – Just-In-Time

KBE – Knowledge Based Engineering

KBS – Knowledge Based System

MML – MOKA Modelling Language

MOKA – Methodology and tools Oriented to Knowledge based engineering Application

OODBMS – Object Oriented Database Management System

PDM – Product Data Management

RADE – Rapid Application Development Environment

SDAI – Standard Data Access Interface

STEP – Standard for the Exchange of Product

UDF – User Defined Feature.

UML – Unified Modelling Language

VBScript – Visual Basic Script

XML – Extensible Markup Language

CHAPTER 1: INTRODUCTION

1.1. Aim

Currently, there is a problem experienced by many enterprises who are using and applying knowledge based engineering (KBE) [1,2] technology to their businesses. The problem lies in the fact that, besides the KBE engineers and programmers, it is difficult for others to understand the inner working of the generative modelling and design function defined for the automation model of the KBE application. This means there is a lack of understanding, visibility, transparency, traceability and accountability when KBE applications have been used. Consequently, there are tentative elements of mysterious, hidden, unknown and "lost of control" concerns when KBE technology is deployed.

The research programme aims to develop and integrate knowledge models with the design automation model from the KBE application Knowledgeware [3] which is part of the powerful CATIA [4,5] software. By integrating the automation and knowledge models, engineering designs produced from the KBE application may be performed with a degree of comprehension and transparency because engineering rules and relationships used for defining the generative design function can be viewed conveniently on the knowledge model.

Commercially, the research aims to demonstrate and support the view that KBE technology can deliver much value added benefits for industry today and tomorrow, as published by the DTI in their Best Practice Guide [2,6], see Figure 1.1.

Scientifically, the research strives to advance the development of KBE technology among the academic research communities.

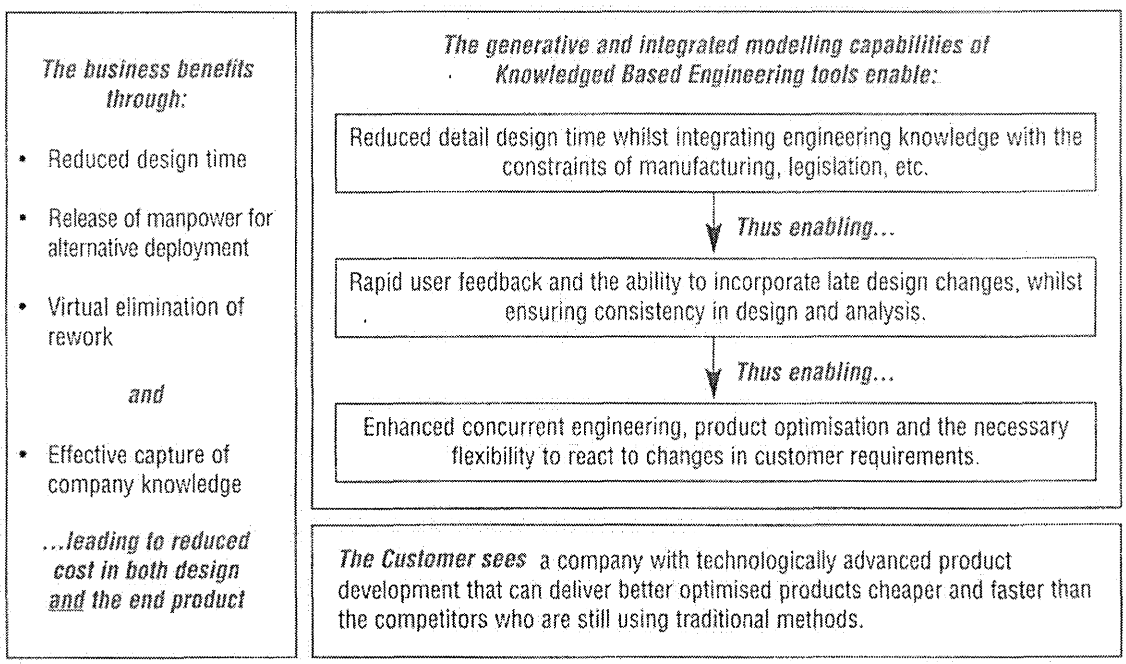


Figure 1.1 Benefits of KBE systems (Courtesy of Cooper et al) [2]

1.2. Objectives

The research programme has four prime objectives as listed below:

1. Develop Overall Framework: develop an overall framework for the proposed research programme which must address the following issues:

- What data, information, rules, constraints, relationships, parameters and specifications are required for the development of the KBE application and database?
- How to structure and in what form the data, information, rules, constraints, parameters, etc. should be stored and mapped onto the automation and knowledge models residing on the CATIA (Knowledgeware) KBE application and database, respectively?
- The knowledge and design automation models must be sufficiently adaptive to undertake a simple case study supplied by the industrial collaborators, so that the

work produced from the present study may be validated.

2. Design and Develop Knowledge Model Residing on Database: design and develop the knowledge model so that engineering knowledge may be captured and stored on the database. The knowledge model is developed in accordance with the European MOKA (Methodology and tools Oriented to Knowledge based engineering Application) [7,8] guidelines with the purpose of describing how a product should be designed and manufactured within a prescribed set of illustrations, constraints, activities, rules and entities.

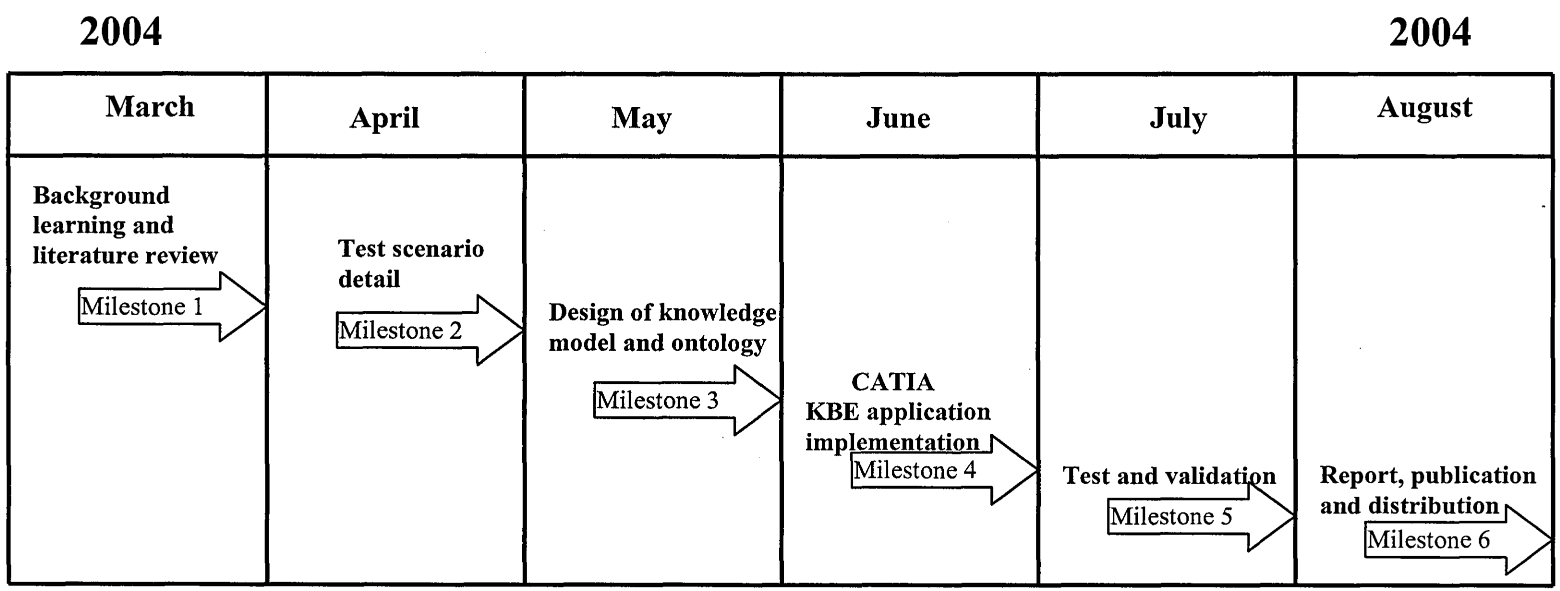
3. Design and Develop Automation Model Residing on KBE Application: design and develop automation model that resides on the CATIA application side in accordance with the programming procedures and languages offered by CATIA's own KBE application called Knowledgeware. CATIA (Knowledgeware) allows the automation model to be programmed using the Knowledgeware programming language, VBScript, macros and CAA IDL API.

4. Case Study to Validate Present Study: the research programme is validated by means of a case study involving an aircraft wing model representative of the Airbus A380, which is has been supplied by the industrial collaboration partners Airbus UK and Aircraft Research Association (ARA). The wing model is used by stress and aerodynamic engineers at a wind tunnel testing facility to calculate pressure plots of the wing structure. Validation for the present study is achieved by showing how the CATIA (Knowledgeware) KBE application may be used to perform a series of generative engineering designs and illustrating how the inner working and design of the generative design function defined for the KBE application may be represented by the informal knowledge model (ICARE forms). The case study is important because it aims to demonstrate that the engineering rules and relationships defined for the automation model may be represented by mapping and linking related illustrations, constraints, activities, rules and entities of the ICARE forms, so that engineering rules and relationships used by the KBE application may be seen and understood by viewing the human friendly knowledge model (ICARE forms).

1.3. Project Plan and Milestones

The research programme has six milestones as highlighted in the project plan given in Figure 1.2. The milestones are summarily listed as follows:

1. Background & software learning *(31/Mar/04)*
2. Detailing test scenario *(30/Apr/04)*
3. Develop ontology and knowledge model *(31/May/04)*
4. Develop design automation model on CATIA (Knowledgeware) application *(30/Jun/04)*
5. Test and validation *(31/Jul/04)*
6. Report, publication and distribution *(31/Aug/04)*



Milestone 1 - 31/Mar/04: complete background learning and literature review
Milestone 2 - 30/Apr/04: test scenario finalised
Milestone 3 - 31/May/04: complete knowledge model and ontology on KBE database
Milestone 4 - 30/Jun/04: complete CATIA application side knowledge model
Milestone 5 - 31/Jul/04: complete case study and discussion
Milestone 6 - 31/Aug/04: distribution of reports to all parties concerned

Figure 1.2 Project Plan and Milestones

1.4. Methodology

The methodology adopted for the research programme is based on a combination of literature review and direct inputs from current practitioners in industry, as listed below:

- **Literature Review:** proceed with a literature review, both paper-based and online, in the field of KBE systems, computer integrated manufacturing (CIM) and systems integration.

- **Site Visits:** obtain comments and feedback from site visits to Airbus UK (Filton), Bristol, UK and Aircraft Research Association Ltd (ARA), Bedford, UK [9] by the author and/or members of research team and evaluate the views held by practitioners in industry on the current state of KBE technology and its practical uses. Additional inputs are sought from engineers of software vendors who are active in the development of KBE related softwares, e.g. Stilo (SophX-Pack) [10] and KTI-Dassault (ICAD) [11] .

- **Case Study:** the research work is validated by means of a case study undertaken by the author with the material obtained from Airbus UK and ARA. The case study aims to show how KBE tools may be used to perform automated and integrated generative models and designs and illustrate how the inner working and design of the generative function defined for the KBE application may be represented by the knowledge model (ICARE forms) stored on the database.

1.5. Scope of Research Programme

Currently, concerns have been expressed from academia and industry that there is a perceived lack of visibility, transparency, traceability and accountability when KBE applications are used to perform generative engineering designs. The concerns are centred of KBE applications are not easily seen and understood by most people, other than the KBE engineers and programmers who developed and coded the automation models. Consequently, there are tentative elements of mysterious, hidden, unknown and "lost of control" fear when KBE technology is deployed.

Specifically, the scope of the research programme is to demonstrate through the use of a simplified case study that engineering knowledge, rules and relationships defined on the design automation model of KBE applications may be captured and presented by a MOKA informal knowledge model (ICARE forms), so that a degree of transparency is achieved between the automation and knowledge models. The case study aims to show that engineering rules, relationships and logics used by KBE applications to perform generative engineering designs may be seen and understood by viewing the human friendly MOKA's ICARE forms.

By developing and integrating knowledge models with the design automation models of KBE applications, the present study seeks to remove the mysterious, hidden, unknown and "lost of control" elements of using KBE applications and overcome the perceived lack of visibility, transparency, traceability and accountability.

The scope of the research programme is limited by the availability of time for the delivery of the work on schedule. Typically, the time constraint placed on Master degrees is approximately less than one year duration. Nevertheless, the present study seeks to illustrate that engineering knowledge may be captured and stored for future use in terms of knowledge reusability/shareability/maintainability and traceability of design changes by using and applying appropriate knowledge management tools, techniques and standards.

While intelligent systems may involve the world of artificial intelligence (AI) which includes psychological and emotional considerations, the scope of the present research will merely focus on the application of engineering rules, constraints, boundaries and logics for the implementation of KBE systems specifically in the field of engineering design and manufacturing. Figure 1.3 illustrates the scope of the present research programme within the field of intelligent systems.

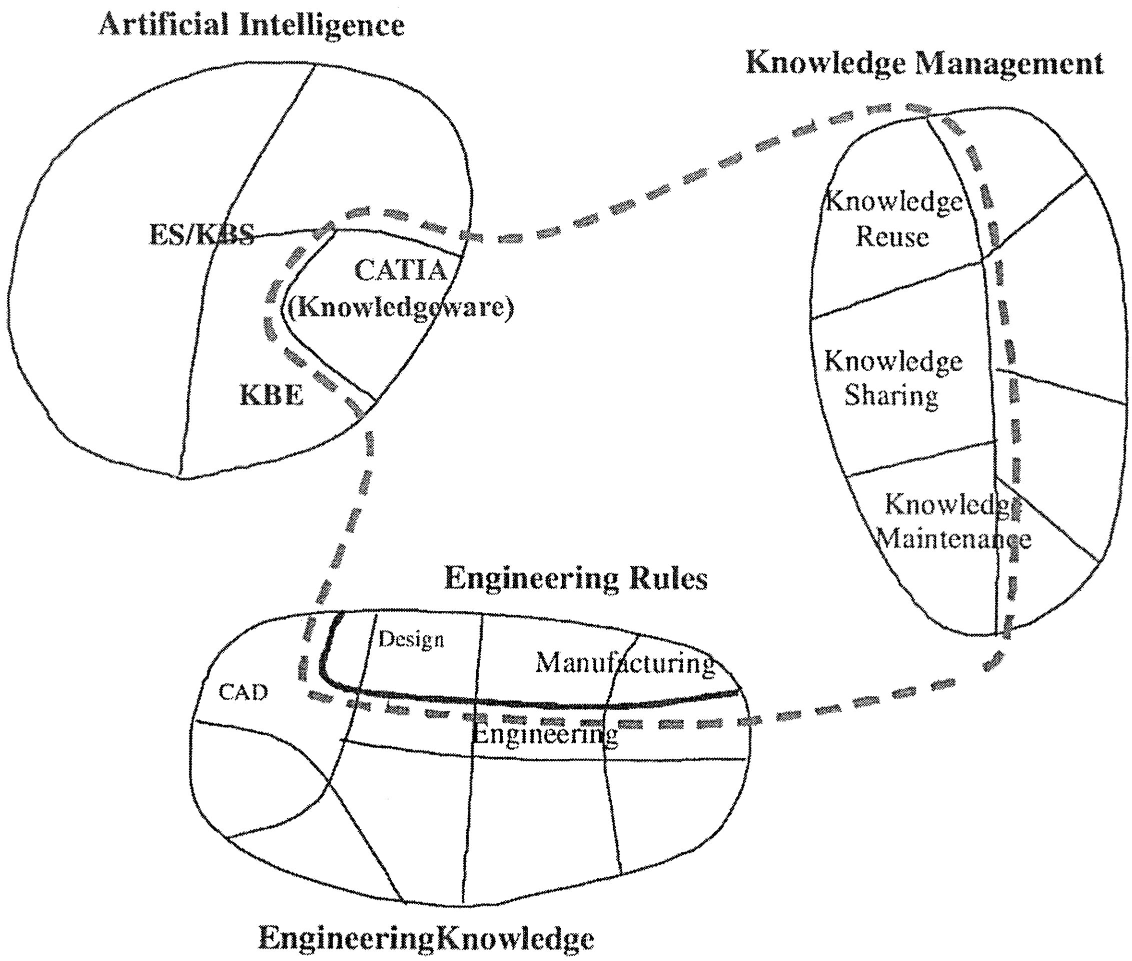


Figure 1.3 Scope of the present research programme in the field of intelligent systems

1.6. Synopsis of Thesis

Chapter 1 sets the scene for the proposed research programme by establishing the aim, objectives, milestones and methodology. **Chapter 2** presents the background and literature review related to the field of intelligence systems such as artificial intelligence (AI), KBE and heuristic-inference systems. The chapter also highlights the numerous distinctive features and characteristics typical of KBE systems. **Chapter 3** defines the proceedings for designing and building the MOKA's informal knowledge models (ICARE forms), as well as narrating the procedures for designing and coding automation models residing on the CATIA (Knowledgeware) KBE application. The chapter describes the design considerations required for building ICARE forms and how to structure the rules and relationships of automation models in the CATIA (Knowledgeware) software.

Chapter 4 narrates how the work presented in this study is validated by means of a case study involving the design of an aircraft wing model obtained from Airbus UK and ARA. The case study demonstrates that transparency is attained between the KBE application and knowledge models, so that engineering rules, relationships and logics defined for the generative design function on the KBE application may be represented by mapping and linking the appropriate illustrations, constraints, activities, rules and entities on the knowledge models (ICARE forms). The chapter describes the work involved for implementing the MOKA-based knowledge model, follows by the work required for developing the design automation model for the CATIA (Knowledgeware) KBE application.

Chapter 5 discusses the subsequent results and findings obtained from the validation process performed in the case study by the author involving the design of the aircraft wing model. The results and discussion highlight on the level of transparency achieved between the automation and knowledge models, as well as verifying that the proposed KBE system can be used to perform generative designs rapidly and efficiently.

Chapter 6 gives the conclusion and ascertain whether the aim of the study has been met and what is the impact, if any, of the work presented in this study in relation to the wider context of KBE systems. The chapter also identifies the limitations of the present study before ending with a list of recommendations for further studies.

CHAPTER 2: LITERATURE REVIEW

2.1. Roles of Automated/Integrated/Concurrent/Holistic Systems

For many industries, e.g. aircraft, automotive, oil, etc., it takes a large team of multi-disciplinary designers and engineers from all departments to design, develop and manufacture any product. Many enterprises realise that it is important to adopt a holistic/concurrent/integrated approach in the design-to-end of life cycle, which may be summarised by the following sequence:

- Identify the demand in the market place
- Planning and feasibility studies
- Conceptual design
- Detailed design
- Build prototype
- Test
- Develop
- Production and roll out
- Troubleshooting
- Decommissioning

The full cycle of the design-to-end of life process invariably consumes immense time, money and human and technological resources. For most products, design changes are inevitable as the product go through its design and development stages, e.g. due to quality function development purposes, design improvements to augment the efficiency of the design for assembly (DFA) as suggested by Boothroyd and Dewhurst [12], etc. Any manufacturing enterprise that can rapidly cope with design changes and reduce the overall design-to-end of life process time without introducing defect will have a competitive edge over their competitors. Such an enterprise will attain maximum customer responsiveness and ability to cope with product changes, varieties and demands.

In many industries the design team often consists of a large number of multi-disciplinary personnel from different departments, each using their specialised CAD/CAM/CAE/CAPP tools to accomplish their specific tasks. In this case, there is an obvious need for technology integration to achieve efficiency and fluidity of information flow in the product development process.

Fuelled by globalisation [13,14] and international competition [15], today's manufacturing industry is highly competitive in all sectors, which means manufacturers must compete for their share of the market place. This is usually achieved by pursuing means to reduce direct labour costs, direct material costs and overheads with raised productivity, hastened customer response time, shorter time to market and increased quality and reliability.

One considered strategy of gaining a highly competitive edge is to operate a superbly lean and agile business, which may be attained by integrating the appropriate technologies, methodologies and practices together to ensure that it delivers a highly proficient and rapid design-to-realisation process.

In the present competitive economic climate, many enterprises, such as BMW [16], Jaguar [17], Chrysler [18], Ford [19], Airbus and BAE Systems, believe technology and integration of technology is the key to gaining the competitive edge [20] in current market place. However, it is insufficient merely using technology, because it is critical that any technology adopted must be correctly integrated in parallel with the lean [21,22], agile [23,24,25] and just-in-time (JIT) [26] doctrines of manufacturing. Enterprises have sought to gain the competitive edge by relying and investing on state of the art technology and integration of technologies, e.g.

- Integration of robotic [27] and automation [28] technologies for delivery of flexible manufacturing system (FMS) [29,30].

- Rapid prototyping [31], knowledge based systems (KBS) [32], knowledge based engineering (KBE) [33] and expert [34] systems to streamline the design and development process.

- Product data management (PDM) and enterprise resource planning (ERP) systems, e.g. Metaphase [35], Windchill [36], Enovia [37], SAP [38] and Baan [39], for generating the bill of material (BOM) and keeping track of all the parts and components used, as well as providing the means to control the logistics and supply chain processes.

Integration of systems goes hand in hand with interoperability and exchangeability of data and information between systems and softwares. This is where standards such as IGES [40] and ISO 10303 (STEP) [41] come into consideration for the delivery of a truly concurrent product development process. Presently, STEP is the most well-defined neutral format standard that is now supported by many of the large engineering software vendors. One of the foremost STEP OODBMS is the EXPRESS Data Manager (EDM) from EPM Technology [42] in Scandinavia, which is favoured by many of the largest engineering enterprises and defence contractors.

The STEP standard of course contains many Parts and Application Protocols (APs) allowing software vendors to implement specific parts and APs relating to their softwares, e.g. clear text encoding for the exchange of STEP files (Part 21), C++ binding of the standard data access interface – SDAI (Part 23), 3D geometries (AP 203), structural analysis (AP 209), data representation for systems engineering (AP 233), etc.

2.2. Knowledge

For many people, when the word knowledge is used in the course of any subject, it is reasonable to say that this implies know-how, skills and/or experience is somehow involved.

According to the Collins English dictionary [43], the word knowledge is defined as follows:

"1. Facts, feelings, or experiences known by a person or group of people,
2. State of knowing,
3. Awareness, consciousness, or familiarity gained by experience or learning,
4. Erudition or informed learning,
5. Specific information about a subject,
6. Become known to one"

Soltan[i] defines knowledge as a three-stage sequence **Data→ Information → Knowledge**. In this sequence, data is something that is rather abstract, has no meaning by itself and can take the form of any alpha-numerical values, measurements, readings, words, symbols, etc. Information is obtained after the data has been analysed, correlated, rationalised and/or synthesised to produce something that has meaning. Knowledge is attained when information is combined with the ability to make decision or involvement of logic to undertake tasks and generate new knowledge.

2.3. Knowledge Based System (KBS)

This section endeavours to fathom what is meant by the term "*knowledge based system*" (KBS). The previous section gives several definitions for the word "*knowledge*" from

---

[i] *Lecture notes by Soltan H, "Knowledge Management: Invest Your Corporate Knowledge Asset, the Return is High", Cranfield University, 1995.*

different sources, which suggests that knowledge is associated with the following list of items:

- Know-how
- Skills
- Experience
- State of knowing
- Facts
- Learning
- Feelings

Evidently, it appears that any system that is knowledge based must somehow acquire the use of knowledge. This would imply that any KBS must somehow also be associated with the above list of items. Consequently, it transpires that any KBS must have the ability to capture, harness and store knowledge.

Incidentally, KBS may also be called expert systems. As the name suggests, an expert system is a system that is an expert usually in some specific domain, e.g. an expert system for injection moulding [44], process control [45], feedback control [46], etc.

Some people may wonder what is the difference between a database and KBS? The difference lies in the ability to store, handle and utilise complexity of data, information and logic and what to do with them. A typical database merely stores data, information and logic for simple uses. Whereas, KBS store data, information and logic with the ability to make complex decision to perform highly complicated tasks to deliver either an individual or set of highly intricate results and outcomes.

In practice, many researchers readily identify a KBS as a smarter cousin of the database and KBS are used to perform tasks, make decisions and come up with new solutions by some form of automation and integration with a set of given rules, relationships and logics coded for the system. Without using the KBS, tasks, decisions and solutions will have to be performed manually which can consume considerable amount of time and effort.

2.4. Knowledge Based Engineering (KBE) Systems

Knowledge based engineering (KBE) systems are a subset of knowledge based systems (KBS) because they are considered as specialised version of KBS and may contain engineering rules, relationships, logics, constraints, parameters, formulas, specifications, procedures and processes for the design and manufacturing of any product.

A number of authors [47,48] have provided some detailed descriptions of typical KBS/KBE systems. Publications by Boullart et al [49] and Stephanopoulos and Han [50] provide a thorough discussion on the various roles, designs, operations and paradigms of intelligence systems such as artificial intelligence (AI), KBS/KBE and heuristic-inference systems, and a glossary from Jovic [46] lists a plethora of terms that one may come across when these intelligent technologies are used.

Perhaps one of the best descriptions of what is a KBE system and its relevance is given by Cooper et al [2] in their DTI's Best Practice Guide. The guide provides a simple, succinct and precise description of typical KBE systems and suggests that such systems typically have two distinctive characteristics:

- **Object-Oriented Paradigms:** the use of object-oriented programming paradigms to model products and their components in a modular manner as objects, see Figure 2.1 and refer to article published by Boullart [51].

- **Generative and Integrated Modelling:** the ability to perform generative and integrated modelling (Note that integrated modelling is also known as virtual product or total product modelling), see Figure 2.2.

The DTI's Best Practice Guide produced by Cooper et al [2] advocates that well developed KBE systems should attain the complete knowledge for generating and replicating any engineering design by capturing, storing and harnessing all engineering, financial and regulatory relevance with the ability to perform generative and integrated modelling. In other words, any robust KBE system must acquire the complete knowledge for the design, development, manufacturing and cost calculation for the delivery of any product.

Delving further into the detail, Cooper et al [2] reason that KBE systems should contain all the knowledge jigsaws such as engineering rules, manufacturing constraints, design specifications, product structure, engineering analysis, product cost, etc., see Figure 2.2.

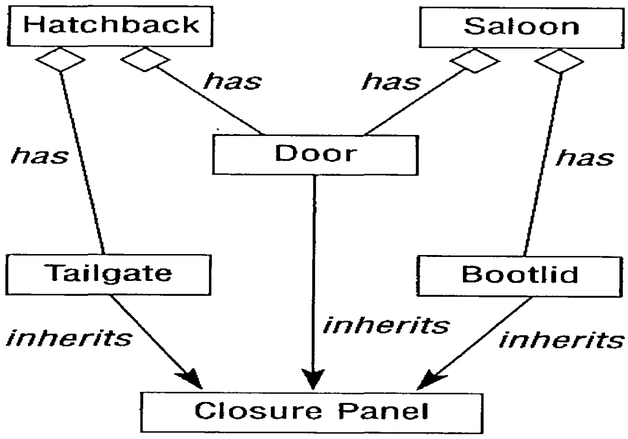


Figure 2.1 Example of object-oriented modelling of KBE systems (Courtesy of Cooper et al) [2]

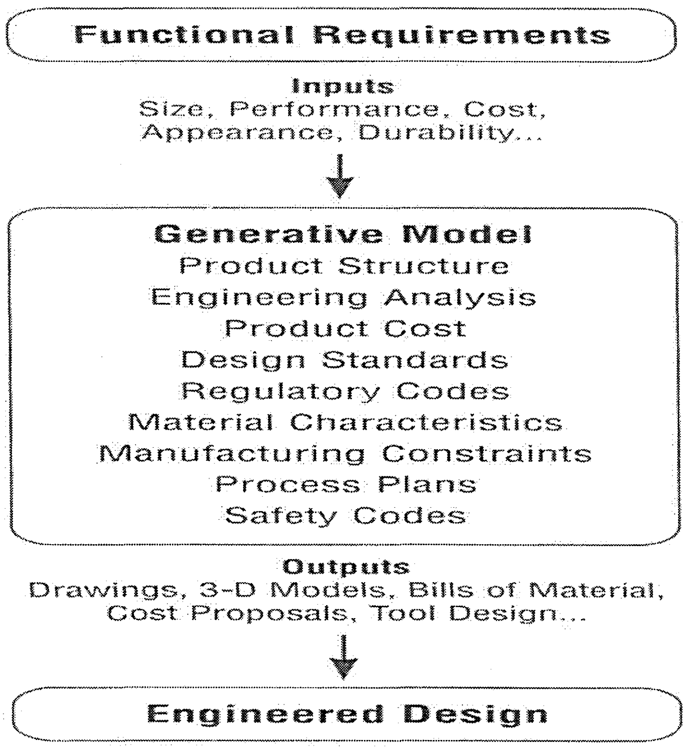


Figure 2.2 Typical inputs, outputs and knowledge involved in KBE systems (Courtesy of Cooper et al) [2]

2.5. Knowledge Reusability, Exchangeability, Shareability, Maintainability and Management

The KBE system allows human knowledge, skills, know-how and arguably experience to be captured and stored in a database by building the ontology and knowledge model of an engineering design, which may contain rules, relationships, logics, constraints and parameters defining specific or generic designs. Therefore, it is lucidly clear that KBE systems provide the means for engineers to reuse any knowledge, information and data that have been stored previously in the system for future uses.

Data exchangeability and shareability is a major issue as the reliance on software computing technology such as CAD, CAM and CAE increases, because this can hinder the seamless flow of data and information between softwares and systems in integrated/automated/concurrent systems. In other words, rapid delivery of design-to-realisation process can only be accomplished when softwares and systems can share and exchange data and information rapidly and effortlessly. Otherwise, costly delays, inaccuracies and errors can plague any project team when a large number of specialist softwares and systems are used during the product development and manufacturing processes.

The data exchangeability and shareability issue may be overcome by using well defined and reputable neutral format standards such as IGES and ISO 10303 (STEP). Nevertheless, there remains the possibility of inaccurate and/or loss of information and data when data/information is exported from one software/system to another, because most softwares have their own peculiarities for representing certain data and information which are not standardised across the software industry. The effects and implications caused by inaccurate and/or loss of information and data may or may not be critical in different circumstances and it is the responsibility of the individual users to determine whether STEP and IGES files are the best and appropriate means to achieve data exchangeability and shareability between softwares and systems.

The KBE system prevents or lessens any inaccuracy/error/mistake from manifesting itself through the design, development and manufacturing phases of the design-to-realisation cycle, because it should alert the design team of any major design digression from the predefined set of rules, relationships, constraints, parameters and logics. This of course assumes that the original knowledge with its rules, relationships, constraints, parameters and logics are accurate, correct and properly maintained and managed.

Proper maintenance and management of the knowledge with its rules, relationships and logics is important because complex engineering designs often required a level of consistency and updating, particularly when a large number of partners and cross-functional teams are involved in the design-to-realisation process. Long term benefits of using KBE systems can only be delivered when major, erratic and unexpected changes to the rules, constraints and parameters are avoided or eliminated, otherwise havoc will reign with regards to the engineering designs and no engineering team can possibly function smoothly and work under these uncertain circumstances. Paradoxically, it is essential that existing rules, constraints, parameters and logics must be modified and updated rapidly whenever it is necessary to do so. The question is of course how often one has to modify and update the rules and logics stored on the KBE application without causing inconsistency and havoc in the engineering design?

Finally, it is critical that any information and data fed into the KBE systems must be properly vetted, maintained and managed with the assumption that only authorised personnel are allowed to modify any of the engineering rules and logics used by the KBE application.

2.6. Lifecycle Development of KBE Systems

The European MOKA consortium has proposed a 6-stage lifecycle for the design, development and implementation of KBE systems, see Figure 2.3.

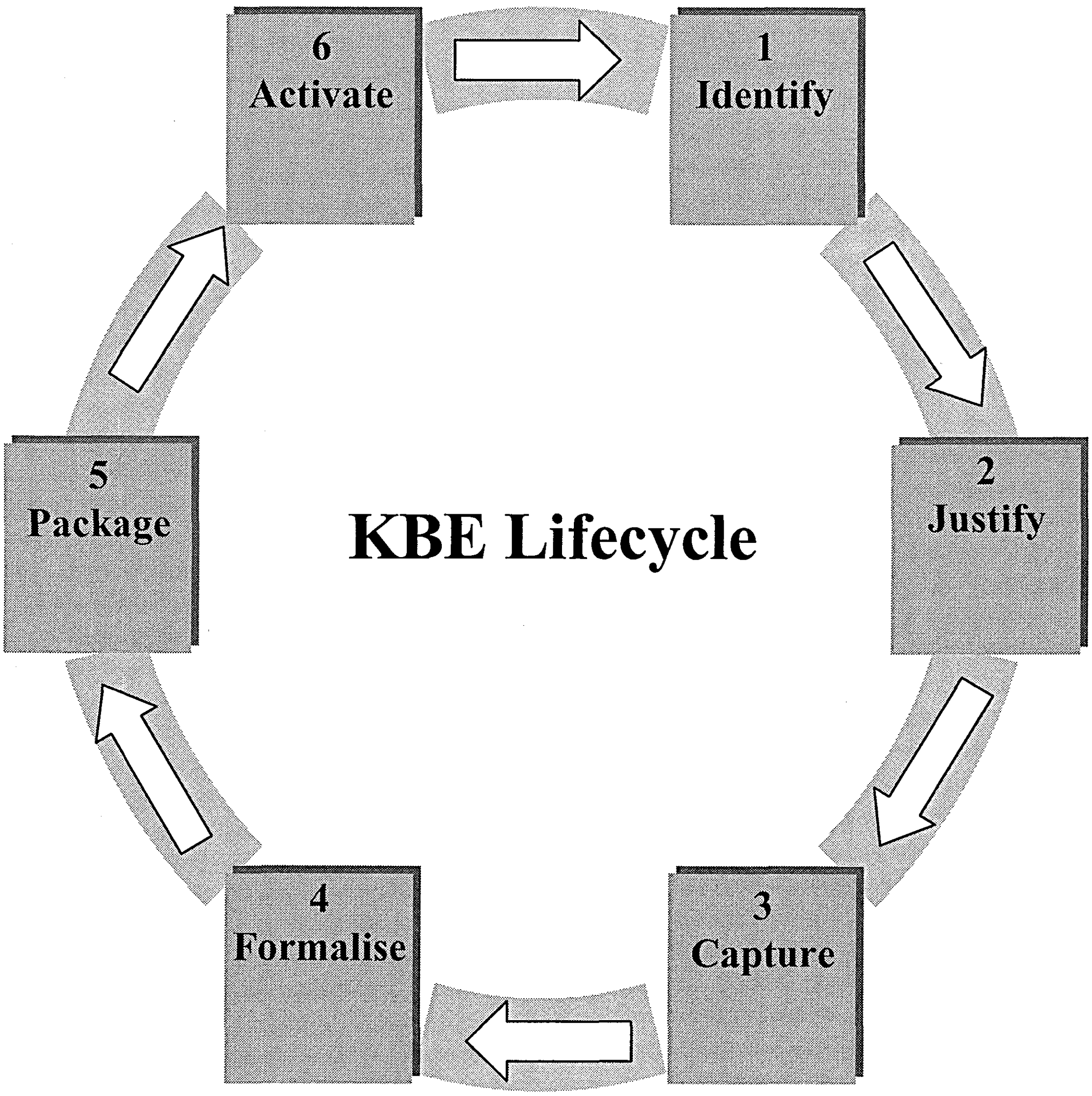


Figure 2.3 MOKA consortium's 6-stage lifecycle for the design, development and implementation of KBE systems [8]

The six stages of the lifecycle development of KBE systems proposed by the MOKA consortium are succinctly described below and a full description is given in Chapter 3 of reference [8]:

Stage 1 – Identify

This stage involves ascertaining the business needs, opportunities and goals of the company with clear aim, objectives, tasks and scope of the proposed KBE system. Stakeholders and resources identified, criteria defined and feasibility studies undertaken with conceptual specifications in place. At the end of this stage, there is a "Go/No Go" decision to take whether to proceed further with the considered KBE system or not.

Stage 2 – Justify

This stage involves generating a global project plan with a clear business case for management approval in the development of the KBE system.

Stage 3 – Capture

This is a major and critical task in the lifecycle development of KBE systems. The stage involves the capture and preparation of raw knowledge into a suitable form for storage on a database. The knowledge is stored in a well defined structure known as the ICARE (Illustration, Constraint, Activity, Rule, Entity) forms which represent the informal knowledge model of the KBE system. The intention of the informal knowledge model (ICARE forms) is to map out and link up all the relationships that exist between relevant illustrations, constraints, activities, rules and entities. The expert and knowledge engineers are expected to work closely to develop the ICARE forms. A knowledge book is produced at the end of the stage when the informal knowledge model has been validated against the set criteria.

Stage 4 – Formalise

This stage involves using the MML (MOKA Modelling Language), which is an extension of UML (Unified Modelling Language), to formalise the informal knowledge model (ICARE forms), thus yielding a formal knowledge model. The formal knowledge model exhibits the analysis and design of the KBE system required for the coding implementation by software programmers. The formal knowledge model itself represents two types of model, namely the product and design process models. The product and design process models describe, respectively, the WHAT and HOW to-do in the design process.

Stage 5 – Package

This stage predominantly involves the software developers whose task is to assume the code implementation of the KBE system from the design defined in the formal knowledge model. The work involved at this stage includes dividing the coding implementation work into manageable packages, developing appropriate graphic user interfaces (GUIs) for the KBE application and selecting particular technical platforms for the KBE system, e.g. STEP, XML, Java, etc. Note that the MOKA consortium does not consider this stage as one of its core research work.

Stage 6 – Activate

This stage deals with the distribution, installation and utilisation of the developed KBE application. Training is expected to be provided for users when the KBE application is released for utilisation and the business success of the KBE system can only be determined after a prolonged period of time such as months or years. Again, this stage falls outside the core research work undertaken by the MOKA consortium.

CHAPTER 3: DESIGN OF KNOWLEDGE & AUTOMATION MODELS

3.1. Design of MOKA Informal Knowledge Model (ICARE Forms)

One of the major tasks involved in building any KBE system is the capturing and storing of the required knowledge on the KBE database in an appropriate form. For the MOKA KBE researchers, this form is often referred to as an informal knowledge model which is one of the core components of developing any KBE system. The intention of the informal knowledge model is to structure the required knowledge residing on the KBE database in an orderly manner.

The design of the knowledge model used in the present research project is based on the methodology proposed by the European MOKA consortium. The work for the design of the informal knowledge model is situated at the 3rd stage of the MOKA's proposed KBE development lifecycle, see Figure 2.3 of Chapter 2.

According to the MOKA paradigm, the required knowledge for any product design may be captured and stored on the database by an informal knowledge model. The informal knowledge model is obtained by filling out the appropriate ICARE forms. The acronym ICARE is briefly summarised as follows:

- **I – Illustration:** description of any relevant information, examples, case studies, etc.
- **C – Constraint:** limitations on the Entities
- **A – Activity:** description of the WHAT in the design process
- **R – Rule:** means of regulating Activities and it is the HOW in the design process
- **E – Entity:** objects representing, e.g. assemblies, components, parts, etc., of the product

The aim of the ICARE forms is to link the Rules and Constraints with the appropriate

Entities and Activities. Illustrations can be used anywhere to provide additional useful information. The layout of the Illustration, Constraint, Activity, Rule and Entity forms of the ICARE forms are depicted in more detail in the following pages and a complete description is given in the Chapter 7 of reference [8] which has been produced on behalf of the MOKA consortium.

3.1.1. I-Form (Illustration)

The I-form serves the purpose of providing any additional or relevant information which may make it easier for other people to understand what information and knowledge are being captured and stored on the database. Note that this form is not used during the actual coding implementation of the KBE system itself because it is merely a means of providing additional information. The table below shows the layout of the I-form and its data fields.

I-Form (Illustration)	
Name	Name of illustration
Reference	Reference no. of illustration
Context, info, validity	Explanation where this illustration can be applied
Description	Description and/or illustration example
Related Constraints	List of linked constraints
Related Activities	List of linked activities
Related Rules	List of linked rules
Related Entities	List of linked entities
Information Origin	Original source of this illustration
Management	Author: Date Version No: Status: (i.e. in progress/complete/verified)

Table 3.1 Layout of I-form and its data fields

3.1.2. C-Form (Constraint)

The C-form describes any constraint that exists on the informal knowledge model. Constraints are linked to the appropriate entities (Constraint→Entity) and serve the purpose of placing limitation on the entities. Constraints may be applied locally or globally, i.e. the constraints may be applied to an individual part, assembly or the entire product, which means they may be used to define a relationship between a single entity or group of entities. The table below shows the layout of the C-form and its data fields.

C-Form (Constraint)	
Name	Name of constraint
Reference	Reference no. of constraint
Objective	Purpose of the constraint
Context, info, validity	Explanation where this constraint can be applied
Description	Description: Function/Algorithm: Context of application: Validity field: Other additional info:
Related Illustrations	List of linked illustrations
Related Rules	List of linked rules
Related Entities	List of linked entities
Information Origin	Original source of this constraint
Management	Author: Date Version No: Status: (i.e. in progress/complete/verified)

Table 3.2 Layout of C-form and its data fields

3.1.3. A-Form (Activity)

The A-form is used for describing the activities that are required in the design process, thus recording the WHAT to-do in the design process (the HOW to-do in the design process is recorded on the R-form). Activities are linked to the appropriate rules (Activity→Rule). The table below shows the layout of the A-form and its data fields.

A-Form (Activity)		
Name	Name of activity	
Reference	Reference no. of activity	
Objective	Purpose of this activity	
Trigger	Event that triggers this activity	
Input	Information available at the start of the activity	
Output	Information produced at the end of the activity	
Input requirements	Requirements for achieving the desired outcome	
Potential failure modes	Criteria for assessing the success completion of this activity	
Context, info, validity	Explanation where this activity can be applied	
Description	Description of the activity for main tasks and sub-activities. Give list of sub-activities and their global description.	
Related Activities	Parent activity: Sub-activities: Preceding activities: Following activities:	List of related activities which are part of the current activity or previous and following activities.
Rules involved	List of rules involved during the activity execution	
Entities required	List of entities required during the activity execution	
Related Illustrations	List of linked illustrations	
Information Origin	Original source of this activity	
Management	Author: Date Version No: Status: (i.e. in progress/complete/verified)	

Table 3.3 Layout of A-form and its data fields

3.1.4. R-Form (Rule)

The R-form is used to define the rules that exist in the design process, thus recording the HOW to-do in the design process (the WHAT to-do in the design process is recorded on the A-form). Rules are linked to by the appropriate activities (Activity→Rule). The table below shows the layout of the R-form and its data fields.

R-Form (Rule)	
Name	Name of rule
Reference	Reference no. of rule
Objective	Purpose of the rule
Context, info, validity	Explanation where this rule can be applied
Description	Description of the rule: Function/Algorithm: Context of application: Validity field: Other additional info:
Related Activities	List of activities where the rule is used
Related Entities	List of entities that are affected by the rule
Related Illustrations	List of linked illustrations
Linked Constraints	List of linked constraints
Linked Rules	List of linked rules
Information Origin	Original source of this rule
Management	Author: Date Version No: Status: (i.e. in progress/complete/verified)

Table 3.4 Layout of R-form and its data fields

3.1.5. E-Form (Entity)

The E-form serves to describe all the objects, i.e. parts, components, assemblies, sub-assemblies and/or products, which are required in the design process, along with their structure, behaviour (transitional state) and functional aspects. Entities have constraints applied to them (Constraint→Entity) which define their boundaries and limitations. The table below shows the layout of the E-form and its data fields.

E-Form (Entity)		
Name	Name of entity	
Reference	Reference no. of entity	
Entity type	Entity-Structure, Entity-Function or unspecified	
Function	Names and references of key functions associated	
Behaviour	Names and references of key behaviours associated	
Context, info, validity	Explanation where this entity can be applied	
Description	Description of entity with text, figures and/or geometries. Give list of attributes, properties, information, tolerances and forms.	
Related Entities	Parent:	any parent entity of this entity
	Child:	any child entity of this entity
	Undefined:	any other related entity
Related Illustrations	List of linked illustrations	
Related Constraints	List of linked constraints	
Related Activities	List of linked activities	
Related Rules	List of linked rules	
Information Origin	Original source of this entity	
Management	Author: Date Version No: Status: (i.e. in progress/complete/verified)	

Table 3.5 Layout of E-form and its data fields

3.2. Design of Automation Model on CATIA (Knowledgeware) KBE Application

The powerful 3D CAD modelling/design/analysis CATIA software is used in the project to perform generative modelling and design tasks. The CATIA software is widely used in industry, particularly within the aerospace and automotive sectors. Knowledgeware is the KBE engine of the CATIA software that formed one part of the CATIA suite of functionalities which includes drafting, design, analysis, CNC manufacturing, etc.

The CATIA (Knowledgeware) KBE application allows users to perform generative modelling and design based on the inputted rules, relationships, constraints, parameters, etc. which means some of the engineering modelling and design tasks may be automated as defined on the automation model. The degree of automation depends on the methods chosen to access and execute the software and how the automation models have been developed and built, e.g. writing script files, macros or using the CATIA's API to access and interface with the software.

This section of the chapter describes the main aspects of CATIA (Knowledgeware) KBE application where parameters, formulas, rules, constraints, laws, design tables, etc. may be defined to design the automation models that reside on the KBE application. The Knowledgeware part of CATIA contains all the relevant workbenches, e.g. Knowledge Advisor, Knowledge Expert, Product Knowledge Template, etc., required to develop the design automation models.

3.2.1. Parameters & Formulas

Parameters are important in CATIA (Knowledgeware) because the design automation models developed within Knowledgeware employ the rules, relationships, checks, constraints, laws and design tables to manipulate and alter the parameters. There are two types of parameters in CATIA (Knowledgeware):

- **Default Parameters**
- **User Defined Parameters**

The default parameters are automatically set by the software for the geometries contained in the 3D CAD model generated from CATIA, e.g. the default parameter PartBody\Sketch.1\Panel_Width\Length is automatically created for a part design which contains a Panel_Width constraint whose measurement is in Length unit. Figure 3.1 shows a table displaying some of the default parameters created automatically for a 3D panel generated from CATIA.

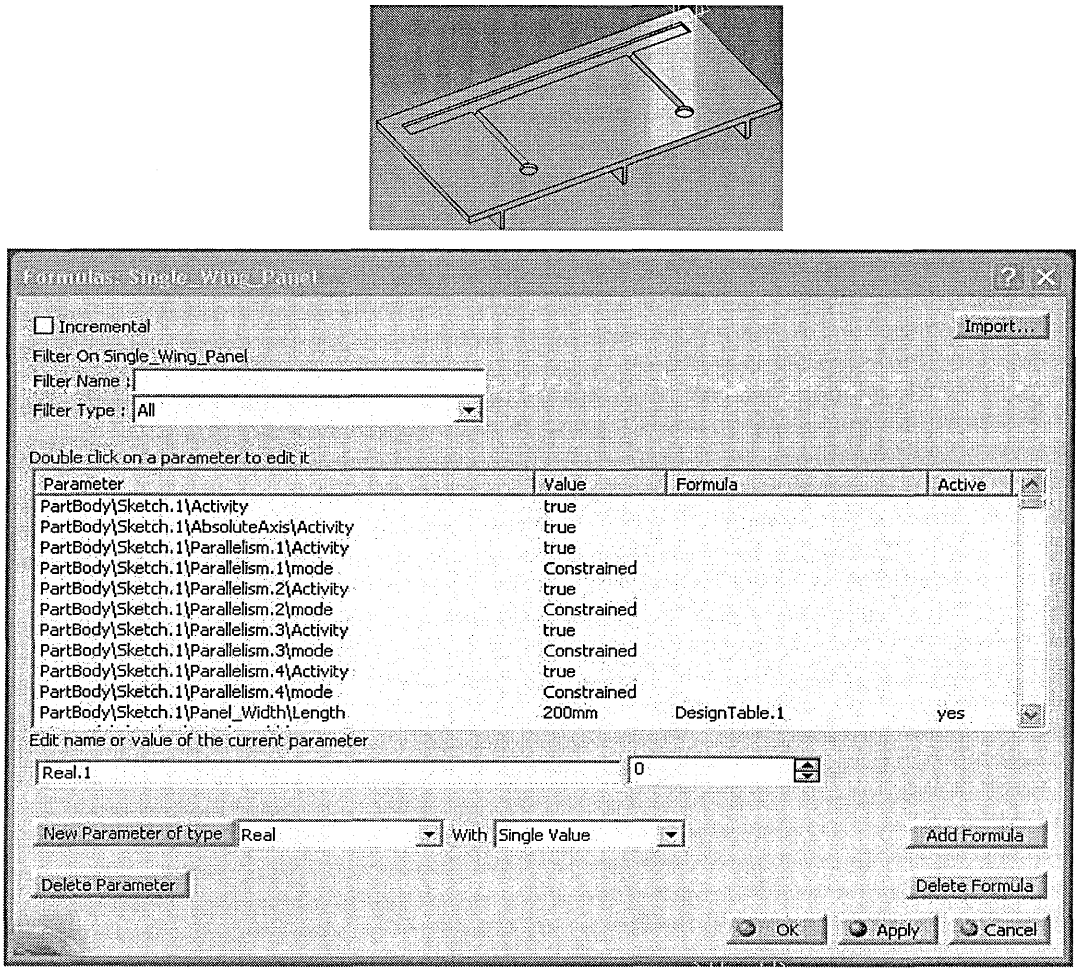


Figure 3.1 Table from CATIA displaying some default parameters for a wing panel

User defined parameters are additional parameters defined by the users themselves for further manipulation of the model. Parameters may have explicit value(s) assigned to them or linked to formula(s) which calculates their value(s). This is illustrated in Figure 3.2 where parameters such as Force_Longitudinal and Force_Lateral have explicit values assigned to them and parameters such as Pressure_Skin and Mass are linked to formulas that determine the skin pressure and mass of the panel.

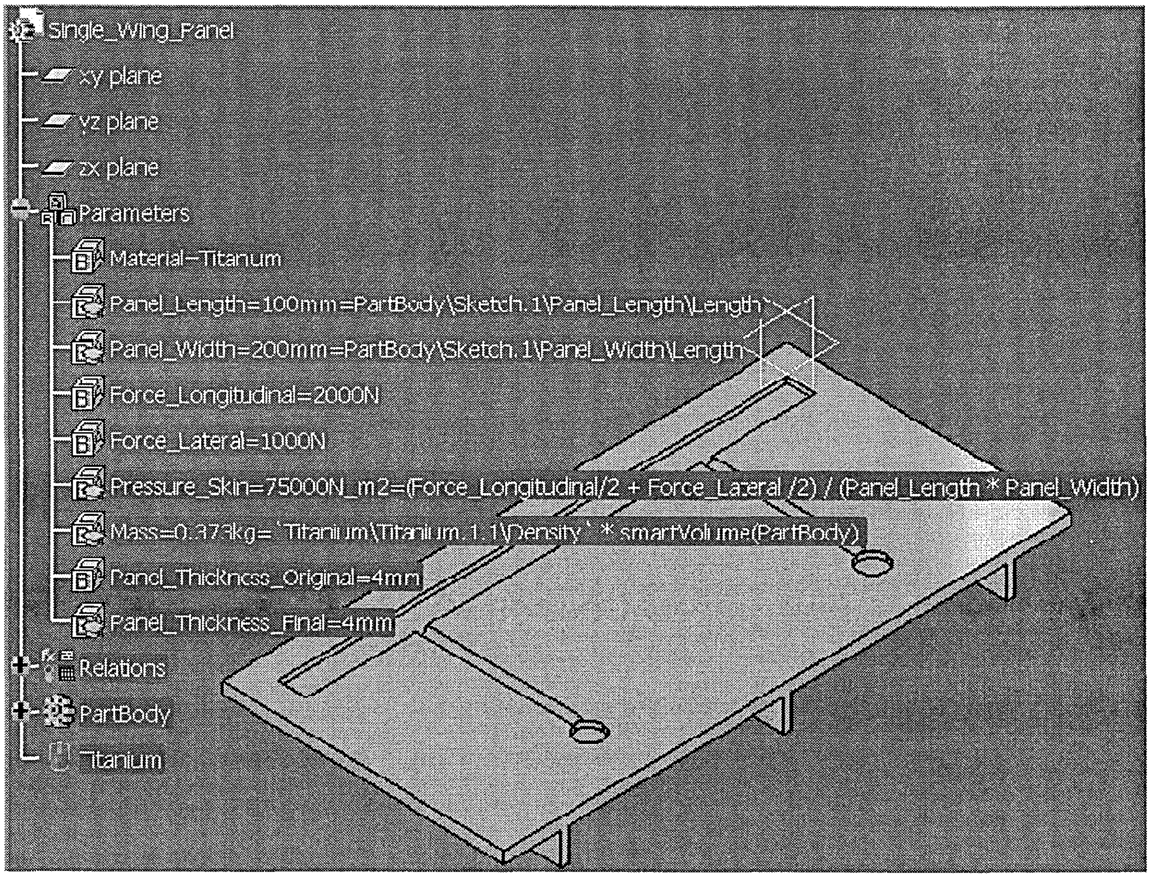


Figure 3.2 Specification tree in CATIA showing default and user defined parameters for a wing panel model

3.2.2. Relationships, Rules, Checks, Laws, Constraints & Design Tables

The design automation models developed within CATIA (Knowledgeware) rely on the relationships, rules, checks, laws, constraints and design tables defined for the models to perform the generative modelling and design. Constraints are set for geometries so that they may be manipulated by the rules, checks, laws, formulas and design tables in Knowledgeware. Figure 3.3 illustrates several dimensional constraints (shown in green) set for the spacing and thickness of the stiffeners in a stiffened wing panel.

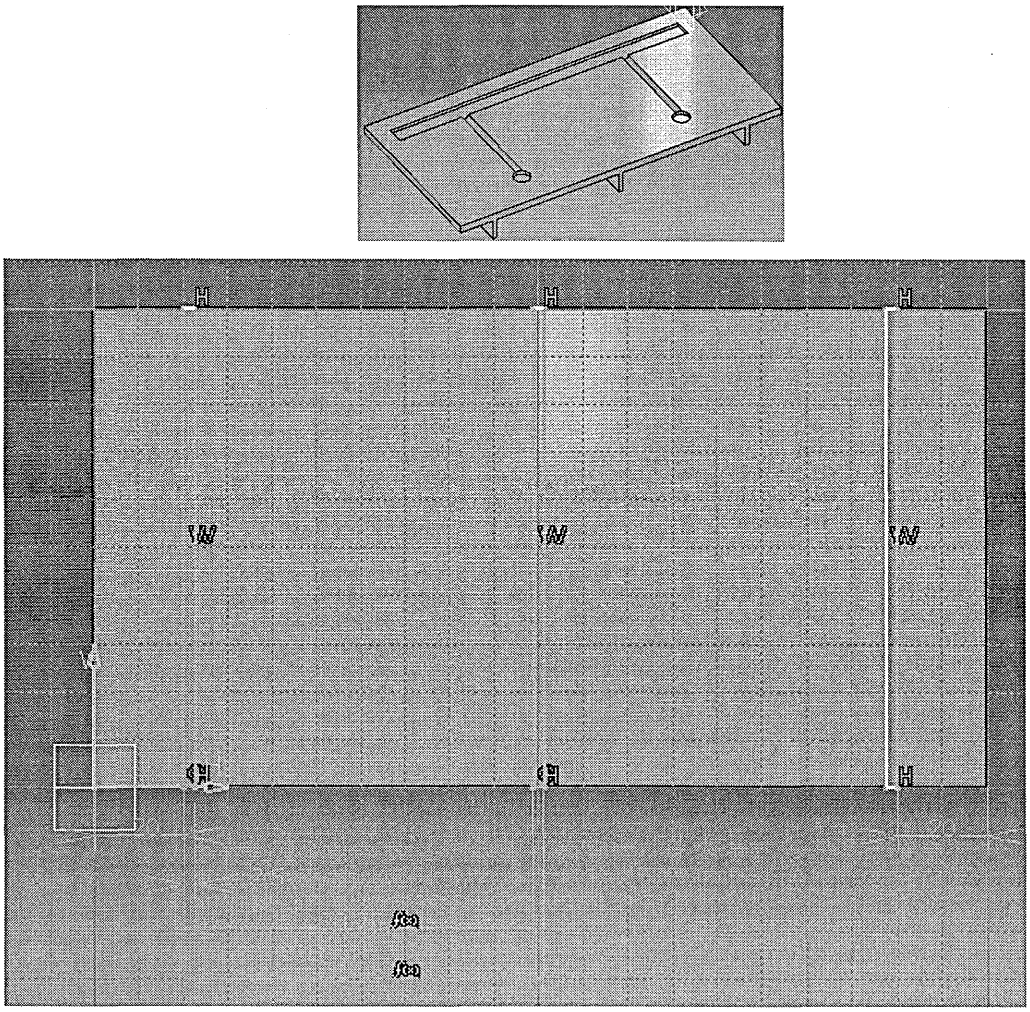


Figure 3.3 Constraints set for a wing panel model in CATIA

Rules are written with logics which determine how/which/what the outcome of the design should be depending on the values or conditions of the parameters. Figure 3.4 shows a rule written with the Rule Editor in the Knowledgeware Advisor workbench of CATIA to manipulate the offset distance of the stiffeners based on the loading conditions.

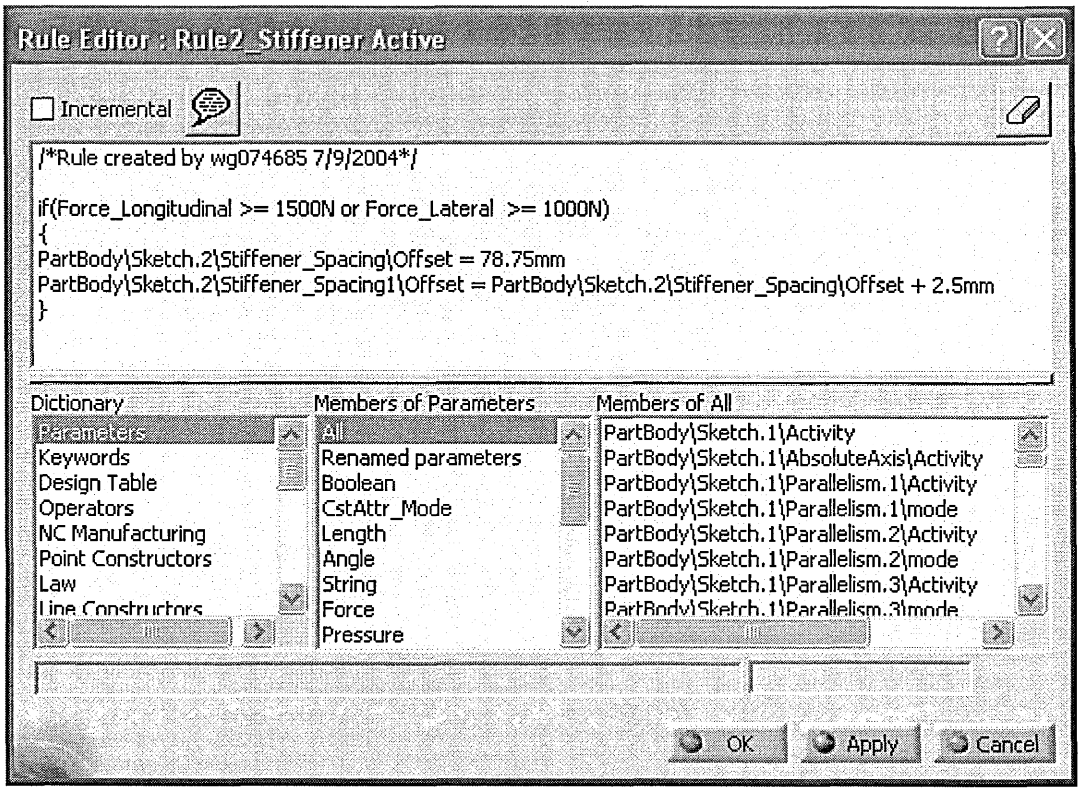


Figure 3.4 Rules written in CATIA (Knowledgeware)

Rules are used to change the design of the model based on the current dimensions and conditions of the model. Checks are similar to rules in CATIA but differ in the sense that they are only used to warn and inform users whether the model has passed or failed a design check in the automation model. Note that checks written from the Knowledge Expert may be used to modify the design of the model if desired. Figure 3.5 illustrates a design check written with the Check Editor in Knowledgeware. The figure shows the model contains two design checks as listed under the Relations specification tree, Check1_Skin is highlighted with a green light which means the model has passed the design check and

Check2_Stiffener_Spacing has a red light indicating that the model has failed the design check.

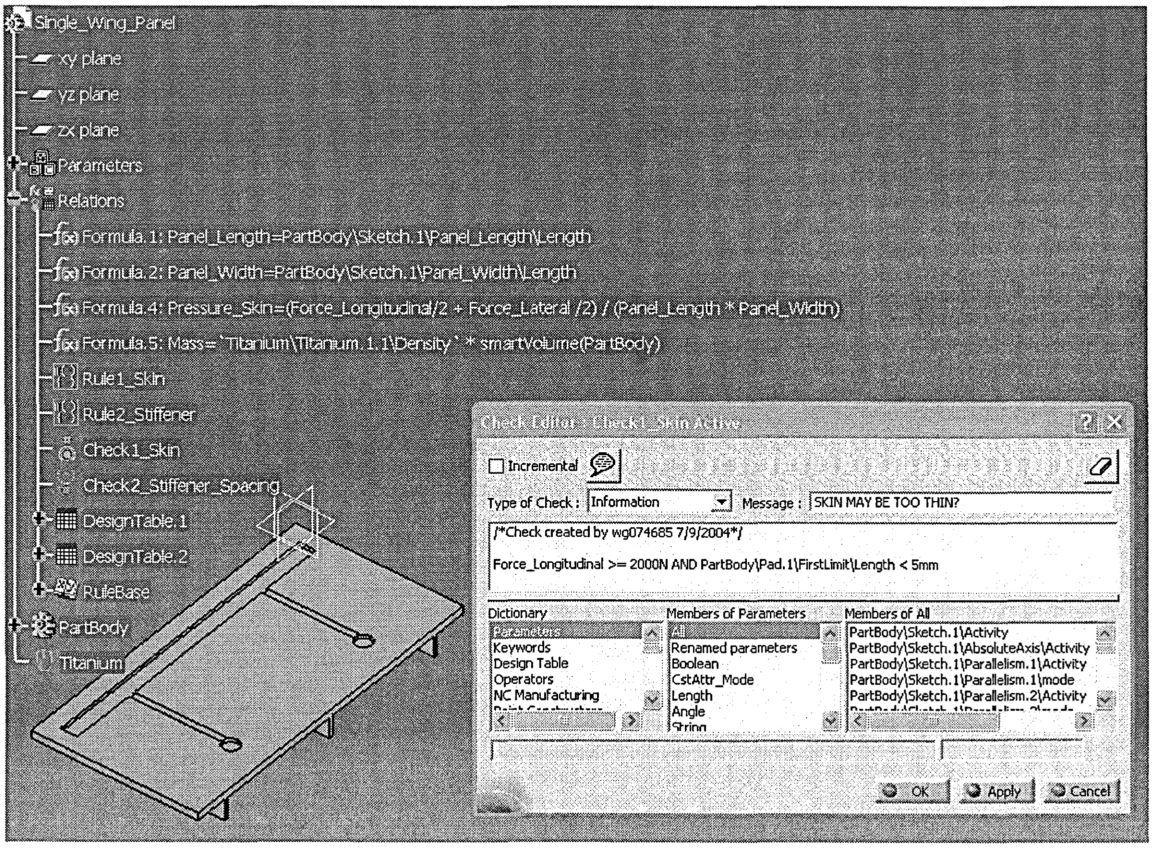


Figure 3.5 Checks written in CATIA (Knowledgeware)

Design tables provide the means to input, alter, update and drive any parameters defined in CATIA from external sources using Excel files or tabulated text files (e.g. from Notepad). This functionality offers versatility and flexibility because the spreadsheet can contain a range of values which may be used conveniently by selecting the appropriate values. Figure 3.6 shows a model of a wing panel with the length and width dimensions imported from an Excel design table.

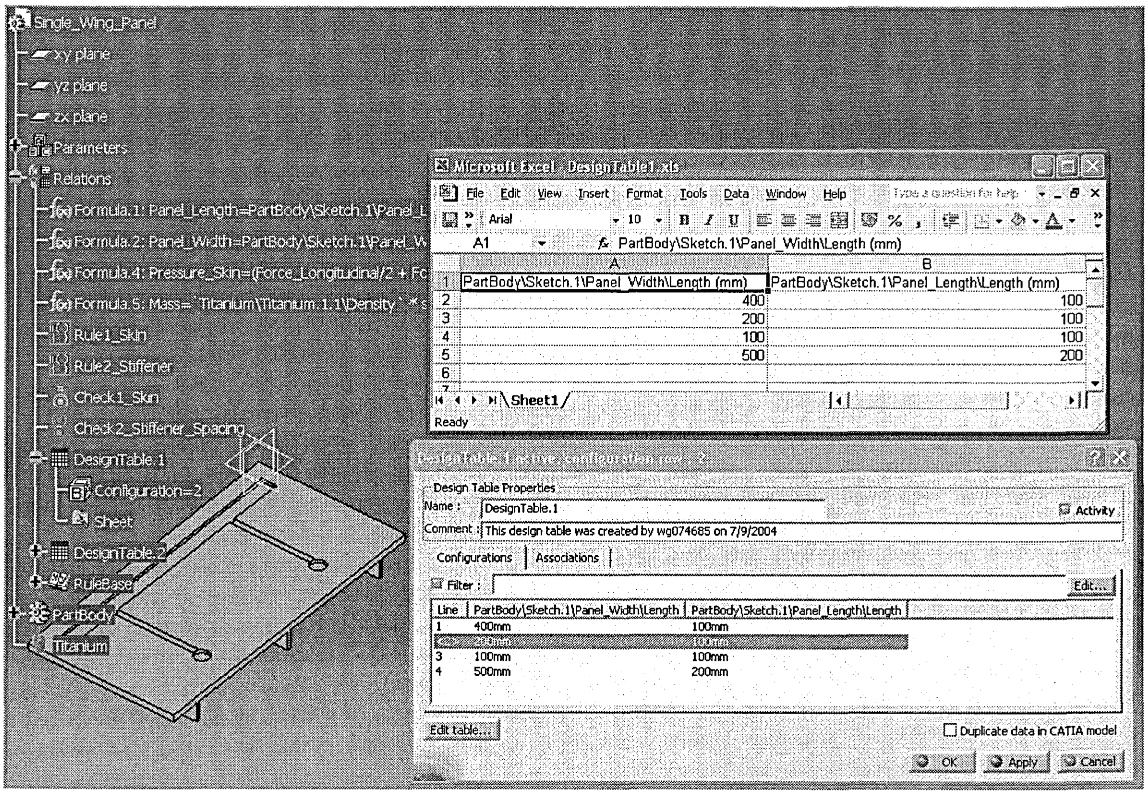


Figure 3.6 Design table imported from an Excel file into CATIA for the dimensions of a wing panel

Knowledge Advisor laws allow relationships to be defined between parameters. This is particularly useful in design problems where there are intertwined relationships between different geometries and their parameters, e.g. a law can be created to represent the relationship between diameters of a hole and screw so that they can be fitted properly.

3.2.3. Knowledge Reusability via Catalogs, PowerCopies & User Defined Features (UDF)

One of the purposes of developing and using KBE systems is that knowledge, information and data stored on the KBE application may be reused intelligently for successive engineering tasks. Knowledge reusability may be achieved by several manners when using the CATIA (Knowledgeware) KBE application, e.g. Catalogs, PowerCopies and user defined features (UDFs).

Catalogs are files where any information, data, parameter, rule, check, geometry, etc. may be stored and catalogued so that they can be used and reused by other people by importing and linking the catalog file into their model. Figure 3.7 shows a catalog is being imported and linked by a model using the *"import with link"* option, which means a permanent link is formed between the model and catalog and changes made to the catalog will be updated automatically in the model.

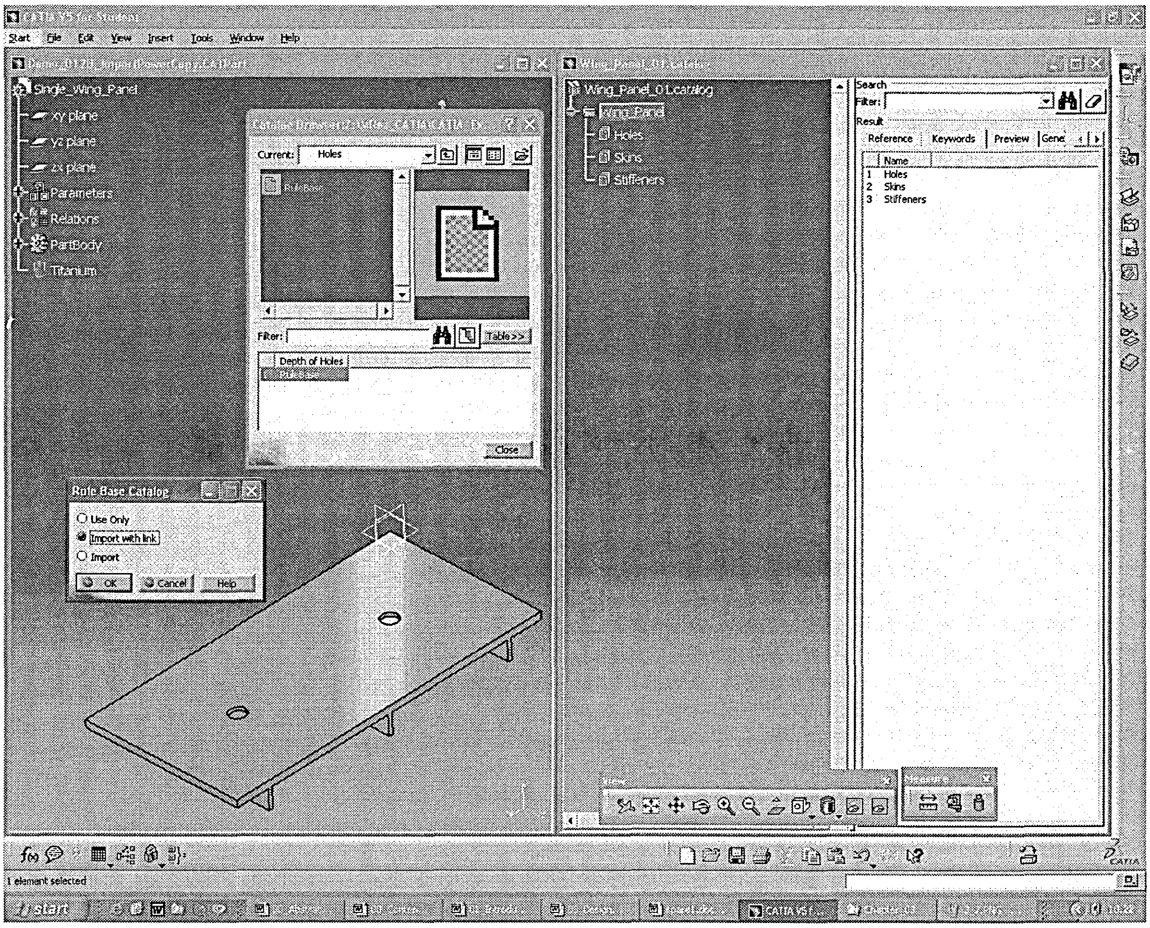


Figure 3.7 Catalog being imported and linked by a model in CATIA (Knowledgeware)

PowerCopies may contain simple geometrical information and data or complex rules, checks and formulas that may be used to develop automation models for performing generative models and designs. Knowledge, information and data stored on PowerCopies may be used and reused by other models by simply inserting the PowerCopies into other models. Figure 3.8 shows a PowerCopy is being created by a model to store the Rule1_Skin rule which uses four parameters. Figure 3.9 shows another model is about to reuse the knowledge, information and data contained in this PowerCopy by inserting it into its own design.

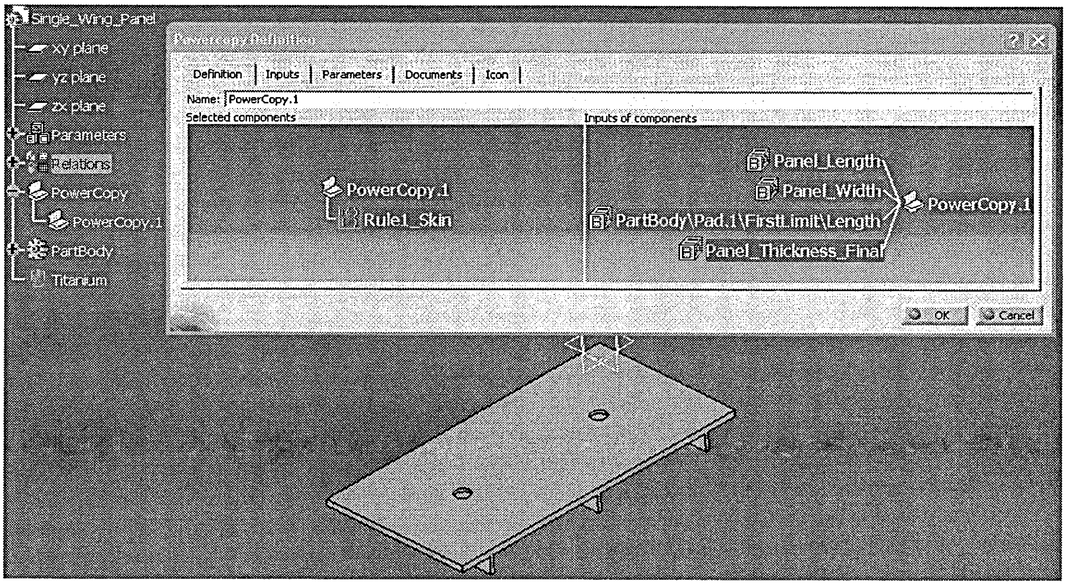


Figure 3.8 PowerCopy being created by a model
(to be reused by another model in Figure 3.9)

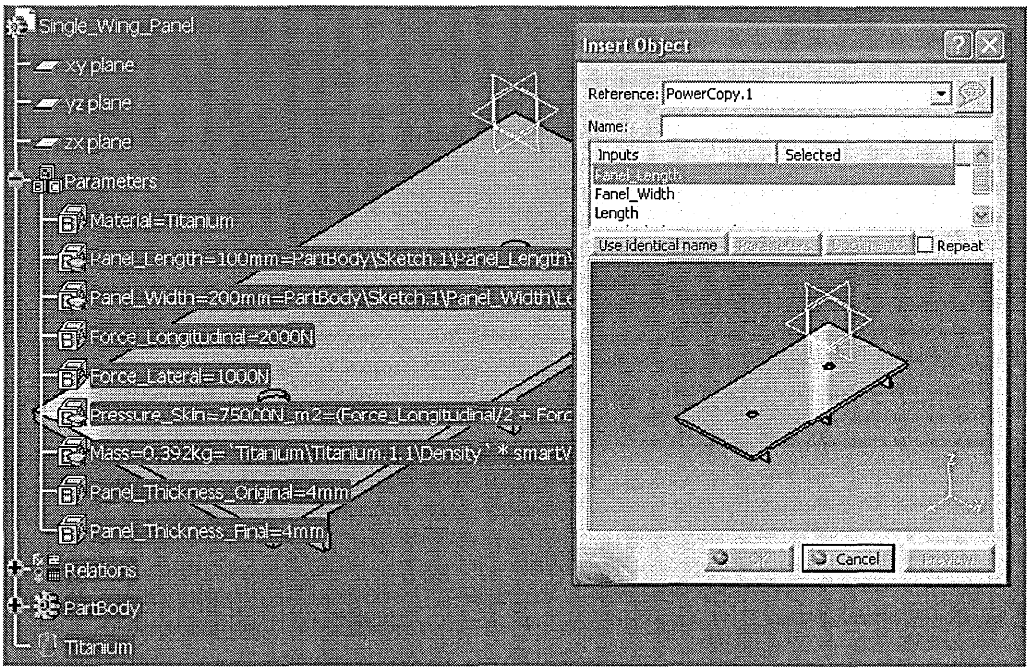


Figure 3.9 Model is about to insert a PowerCopy originally created in Figure 3.8 into its design

UDFs are similar to Catalogs and PowerCopies, which allow knowledge, information and data to be used and reused by other people. UDFs are in fact templates that work on the part level of the design, which means design parts may be defined as UDFs to contain geometries with rules, formulas, laws, etc. and can be used in the design of another model by simply inserting the UDFs into the model. UDFs are created in similar manner to PowerCopies and may also be saved in Catalogs.

3.2.4. Automation via APIs, Macros & Scripting Languages

Scripts may be written in Visual Basic (VBScript) or Knowledgeware's own programming language. Scripts may be recorded or written and stored in the required directory as script files or macros and made accessible to all CATIA users and models, so that they can be used and reused any time right across an enterprise. Rules, checks and laws may be run from script files or macros which have been coded in accordance with the CAA IDL API (the acronyms CAA, IDL and API stand for Component Application Architecture, Interface Definition Language and Application Programming Interface, respectively).

The CATIA software provides a full listing of its APIs, which means it is possible to automate the design and modelling processes with minimum effort by simply running the CATIA software via a script file that contains all the necessary commands and instructions to perform any design, modelling and analysis task.

The CAA C++ API and RADE (the acronym RADE stands for Rapid Application Development Environment) offer the possibility to run the CATIA software and manipulate all of its commands, instructions and functionalities outside the CATIA's environment such as from Microsoft Visual C++ Studio or UNIX platform. Extremely high level of integration and automation is achieved when using the CAA C++ API and RADE to access the CATIA's APIs because it means the CATIA software may be run from a central source which may also control and run other applications as well as CATIA.

Substantial benefits in terms of time, cost and speed are assured when the CATIA software is run from a script file, macro, CAA IDL API or CAA C++ API and RADE, because this means the modelling, design and analysis process may be integrated and automated.

Figures 3.10 and 3.11 show scripts and macros may be run from rules defined for the design automation by using the attributes and methods of the Knowledgeware language in CATIA. Figure 3.12 illustrates how macros may be recorded or written manually using the CAA IDL API and stored on the computer.

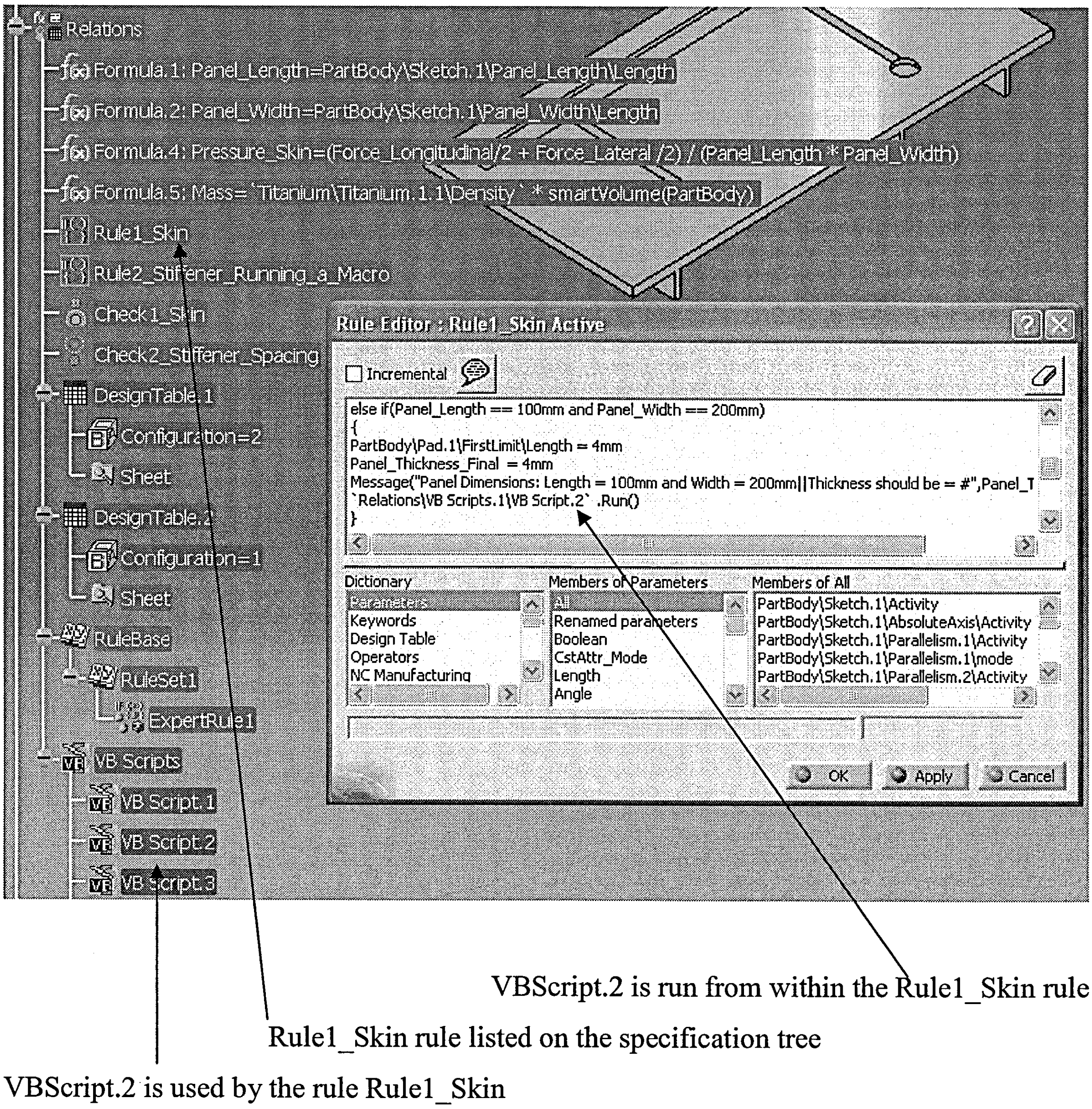
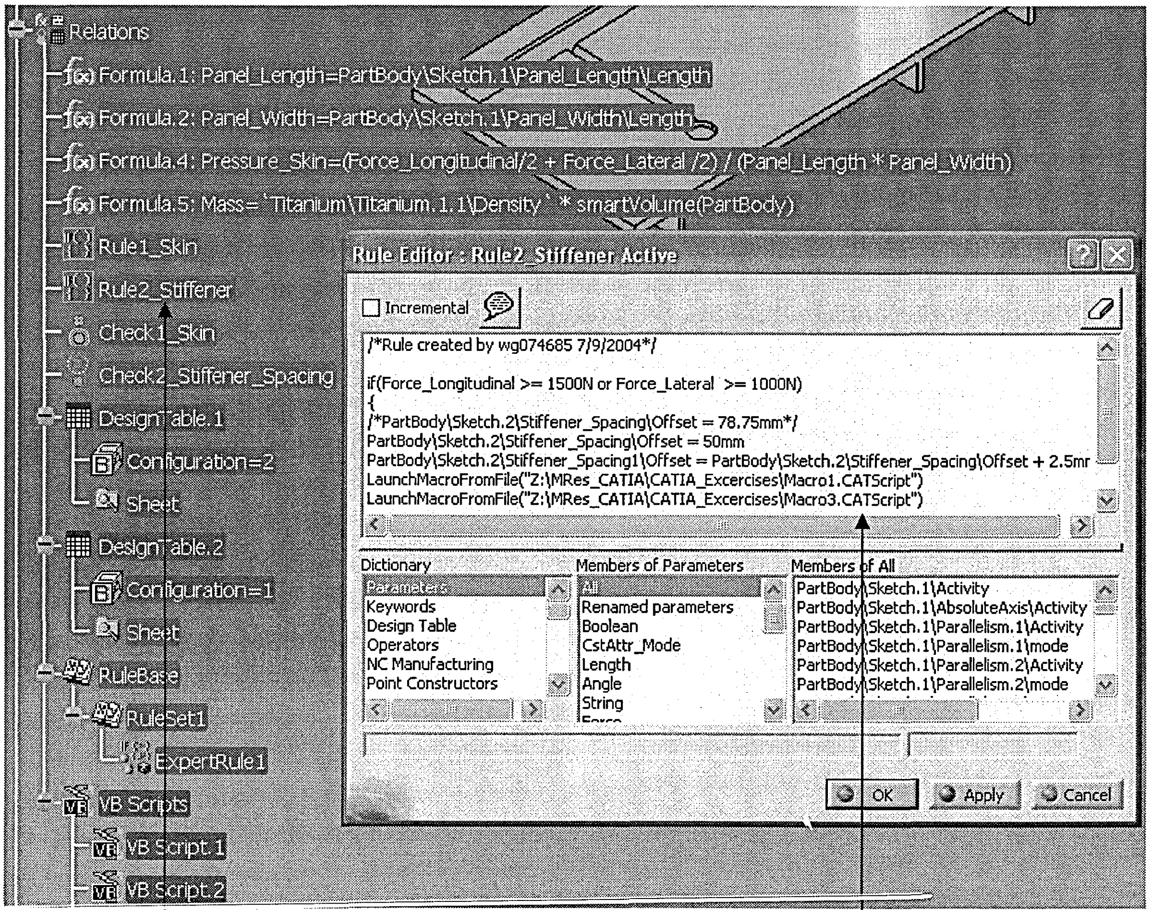


Figure 3.10 VBScripts may be run from a rule using attributes and methods of the Knowledgeware language in CATIA



Macro1 and Macro3 are launched from within the Rule2_Stiffener rule

Rule2_Stiffener rule listed on the specification tree

Figure 3.11 Macros may be launched from a rule using attributes and methods of the Knowledgeware language in CATIA

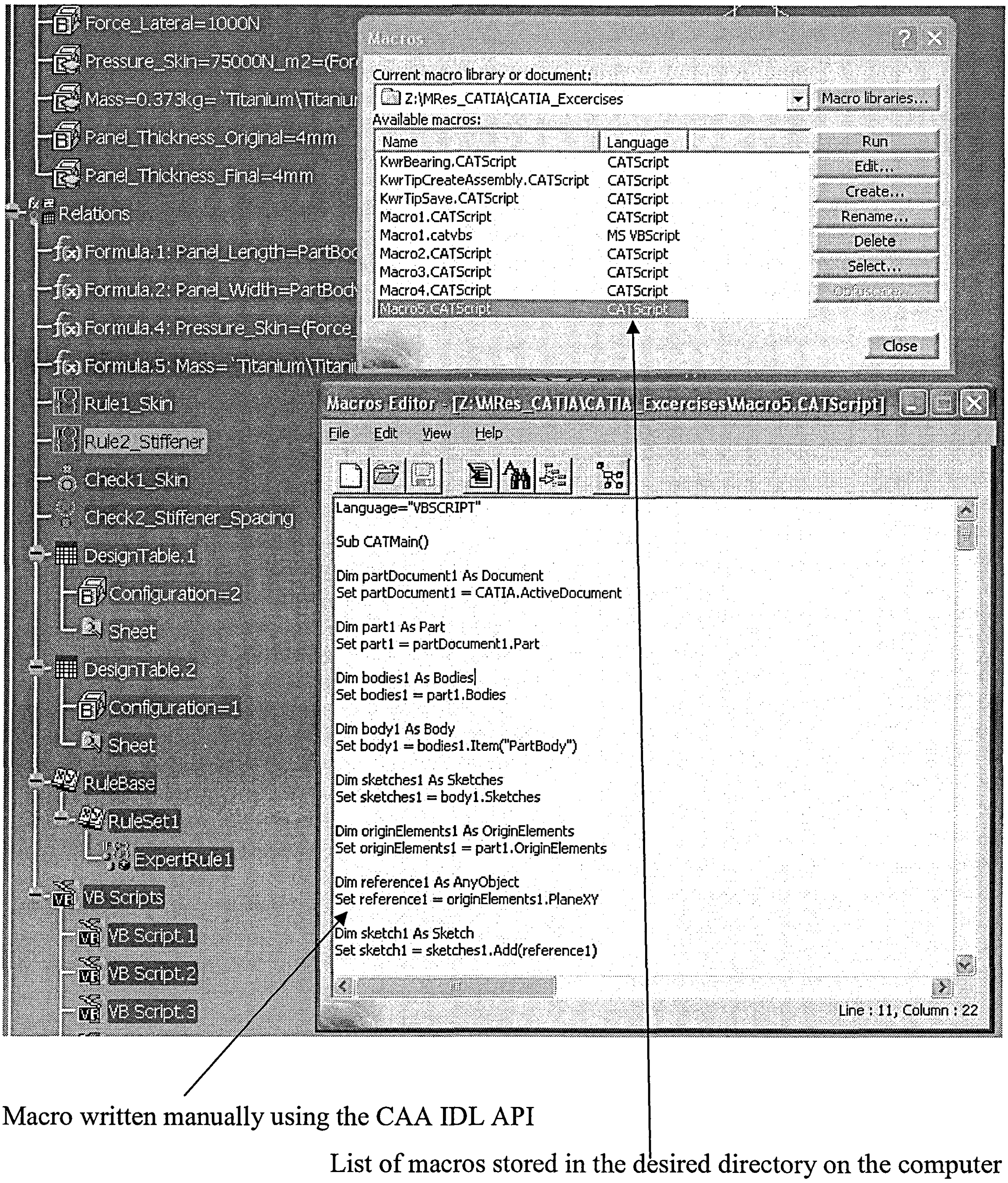


Figure 3.12 Macros may be recorded or written manually using the CAA IDL API and stored on the computer

CHAPTER 4: CASE STUDY – AIRCRAFT WING MODEL

4.1. Description of Work Involved for Case Study

The aim of the case study is to deliver a degree of comprehension and accountability when the CATIA (Knowledgeware) KBE application is used to perform a series of generative modelling and design tasks on an aircraft wing model used by stress and aerodynamics engineers at a wind tunnel and aerodynamics testing facility to calculate the pressure plots of wing structures.

The stainless steel design of the wing model has been obtained from the Aircraft Research Association (ARA) and Airbus UK. For the appreciation of the size of the model involved, see Figure 4.1 which shows some outline dimensions for the wing model.

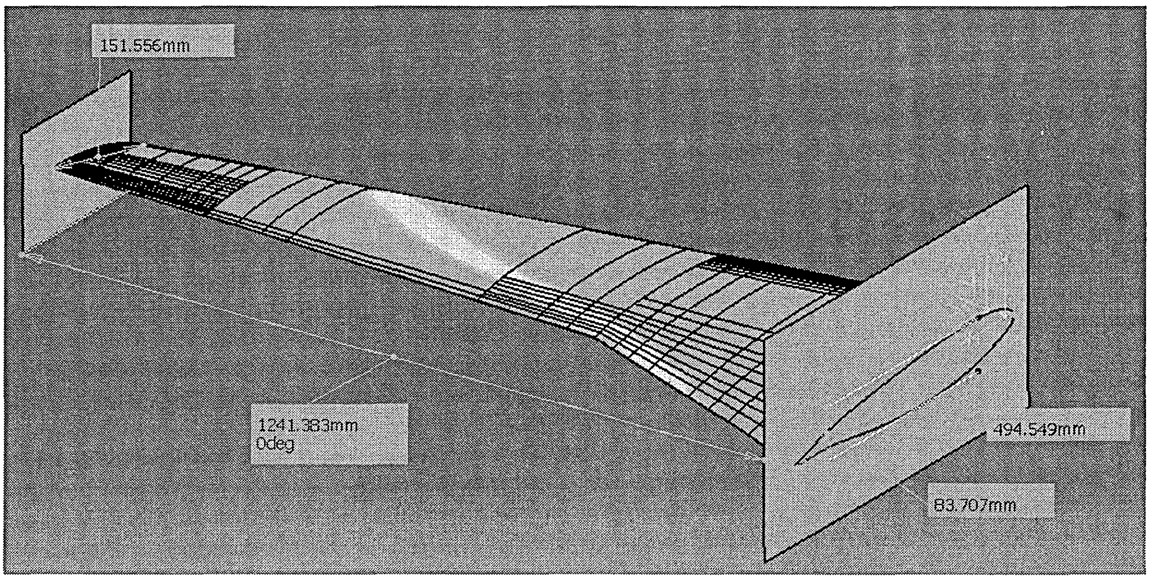


Figure 4.1Outline dimensions of aircraft wing model

The case study seeks to integrate the informal knowledge model (ICARE form) with the design automation model developed for the CATIA (Knowledgeware) KBE application, so that any engineering designs produced from the KBE application may be performed with a degree of comprehension and accountability because engineering rules and relationships used for defining the generative design function can be viewed conveniently from the human friendly ICARE forms.

The case study intends to demonstrate how KBE applications such as CATIA (Knowledgeware) may be used to perform a series of generative modelling and design tasks based on the inputted engineering rules and relationships.

Note that due to the time constraint placed on the research programme, the design automation model coded for the CATIA (Knowledgeware) KBE application and ICARE forms developed solely contain the most critical and fundamental engineering rules and relationships, i.e. less important and peripheral rules and relationships have not been included on the automation and knowledge models. Given additional time, more complex, intricate and comprehensive set of engineering rules, relationships, constraints, parameters, etc. may have been included.

Presently, engineers at ARA and Airbus obtain the pressure plots under wind tunnel test conditions by performing the following tasks and adhering to some fundamental rules:

- Specify the locations of the tappings, i.e. holes, on the aircraft wing model which house the ferrules of the pressure measurement devices (ferrule diameter and length circa 3-4mm and 7mm, respectively) to record the pressures at the specified locations, see Figure 4.2.
- Specify the locations of the primary and secondary channels, i.e. cuttings, on the aircraft wing model which house the tubings where the hollow tubes (tubings) are run throughout the wing model connecting the pressure measurement devices and calibration equipments, see Figure 4.2.
- Use CNC machines to drill/cut the tappings and channels at the specified locations on the wing model.

- Ferrules cannot be fitted at locations where the thickness of the wing is too thin and/or there is insufficient material to house the ferrules. In this case, small holes are drilled to locate the positions of the ferrules, so that a wire/cable bearing the pressure measurement device may be fitted through these holes.

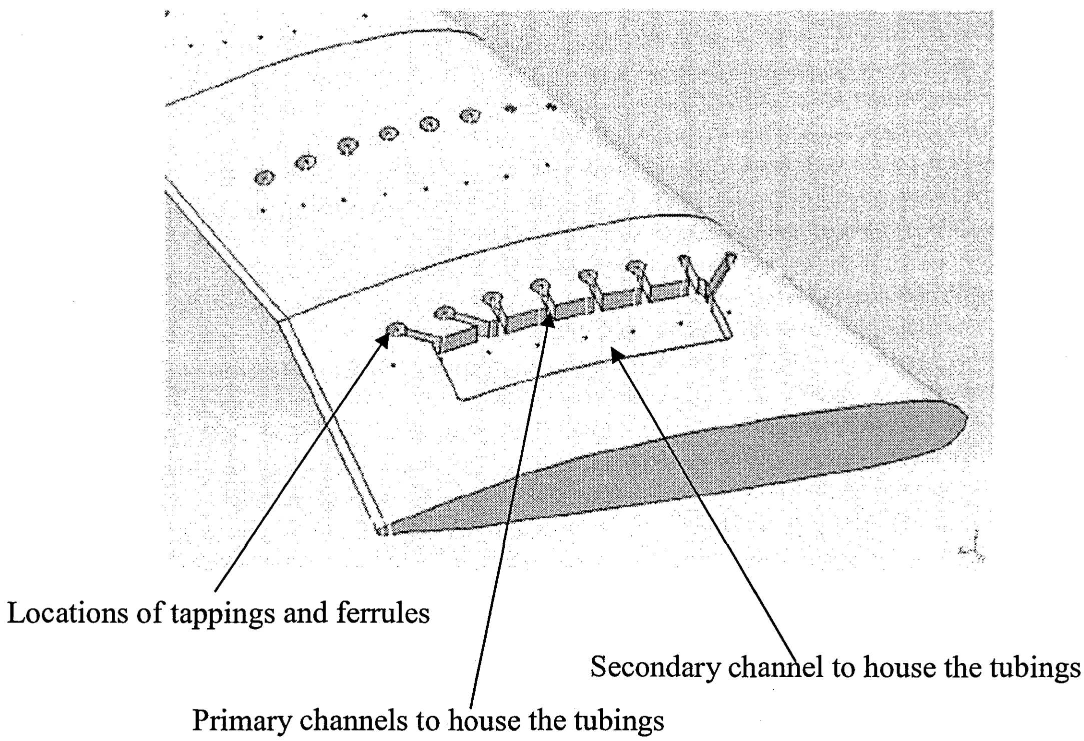


Figure 4.2 Locations of tappings, ferrules, tubings and channels (Courtesy of Fan and Bermel-Garcia) [1]

4.2. Automation Model of Aircraft Wing Model

The design automation model coded for the CATIA (Knowledgeware) KBE application demonstrates how the locations and sizes of the tappings (holes) and channels (cuttings) along the length and width of the wing may be adjusted automatically, based on the engineering rules and relationships defined by engineers at ARA and Airbus.

Automated design examples are produced from the KBE application to illustrate how the tappings and channels along the length and width of the wing can be modelled to give the correct length, width, depth and diameter at the specified locations.

It can be seen from Figure 4.1 that the wing generally tapers and becoming thinner running from root to tip. Based on the rules defined by engineers at ARA and Airbus, large and deep holes and channels cannot be drilled and cut at certain locations along the length and width of the wing when there is insufficient material or the thickness of the wing is too thin.

The automation model has also been coded to set off warning flags whenever a proposed or requested design cannot be performed or complied with a specified rule or check defined for the KBE application, e.g. when there the thickness of the wing is too thin to permit a deep hole or channel.

VBScript and Knowledgeware languages are used for coding the design automation model on the CATIA (Knowledgeware) KBE application. Script files and macros are used with the intention that the modelling and design functions may be automated as much as possible based on the inputted engineering rules. The scripts and macros written/recorded for the KBE application used the CAA IDL API provided by the software vendor Dassault Systemes.

In CATIA (Knowledgeware), parameters, formulas, rules, checks, constraints, design tables, etc. are graphical based and inherently linked to the geometries of the graphical model. For this reason, Figures 4.3-4.18 are used to illustrate graphically the effort expended in creating the parameters, formulas, rules, checks, constraints and design tables for the design automation model of the aircraft wing model.

Figures 4.3-4.18 show that the design automation model has been developed in a well defined structure, giving three distinctive sets of parameters, formulas, rules, checks, constraints and design tables for the tappings, primary and secondary channels, which are responsible for controlling and regulating the generative design of the tappings, primary and secondary channels. The intention is that these three distinctive sets of parameters, formulas, rules, checks, constraints and design tables should become visible and apparent when viewed from the informal knowledge model (ICARE forms).

Graphical geometries and parameters, formulas, rules, checks, constraints, design tables, etc. developed for the design automation model in the case study are catalogued and stored for future uses within CATIA by using the Knowledgeware's catalog function.

Four macros have been produced using the CAA IDL API so that the modelling and design process of the tappings and secondary channel may be automated. Figures 4.17-4.18 show the macros produced and complete codes are given in Appendix 1. The macros may be manipulated in Knowledgeware with modifications and/or additional codes to achieve the desire results. The macros produced are:

- **RowOfTappings.CATScript:** produce a row of tappings at a specified location.
- **Tapping.CATScript:**: produce a single tapping at a specified location.
- **Secondary_Channel1.CATScript:** produce a sketch pad at a specified location for the secondary channel.
- **Secondary_Channel2.CATScript:** produce a pocket (cutting) at a specified location for the secondary channel.

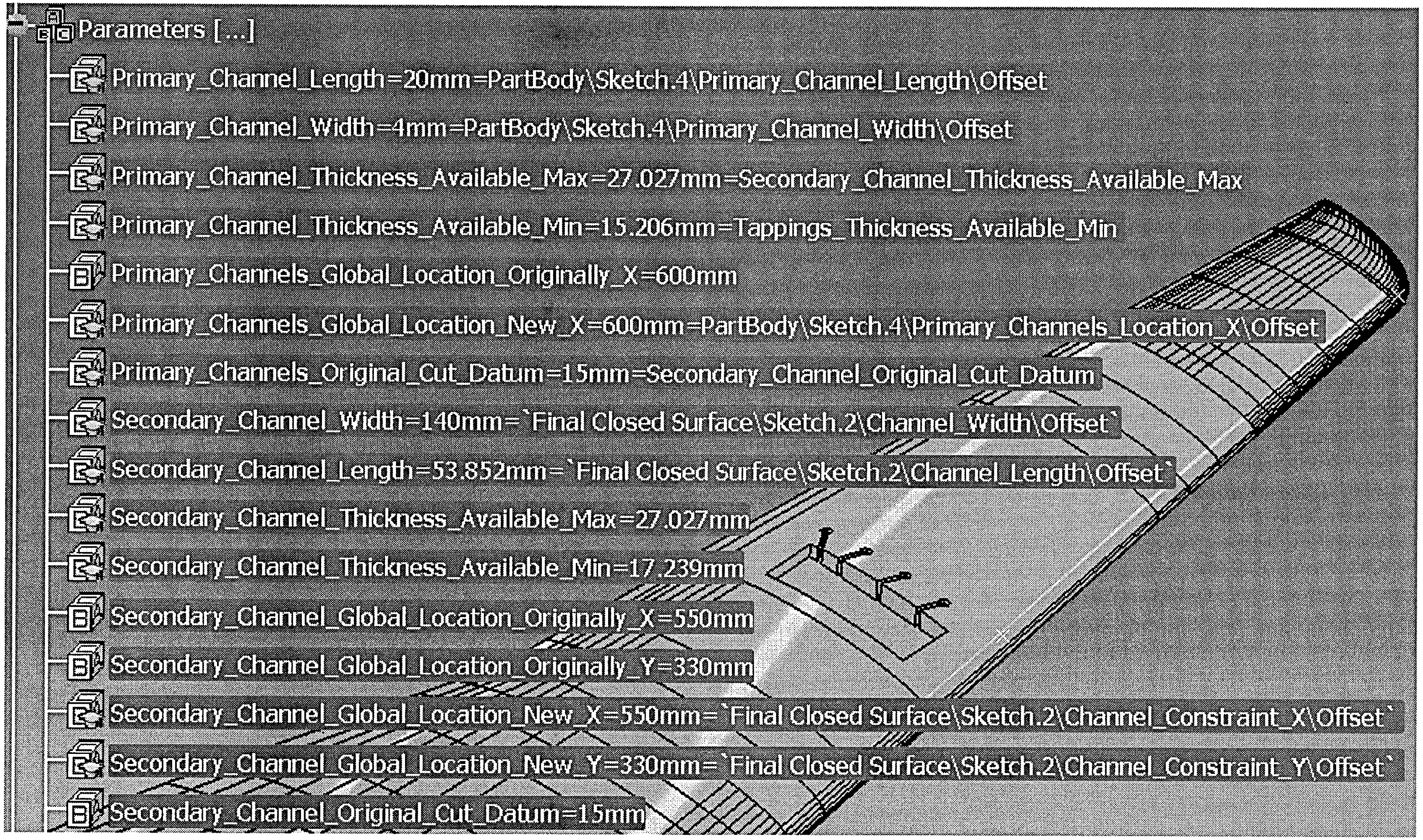


Figure 4.3 Parameters set for primary and secondary channels on the design automation model in CATIA (Knowledgeware) KBE application

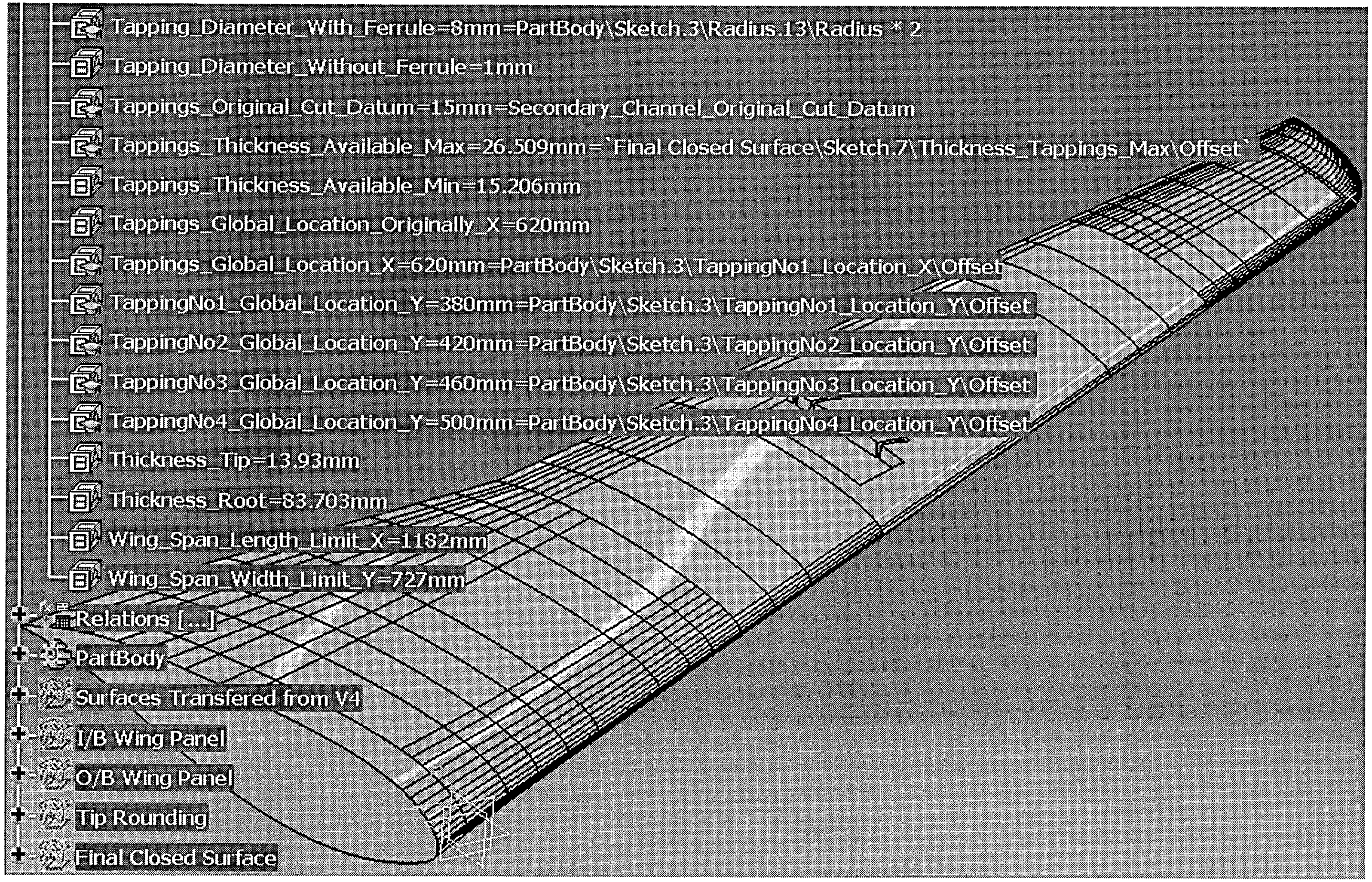


Figure 4.4 Parameters set for the tappings on the design automation model in CATIA (Knowledgeware) KBE application

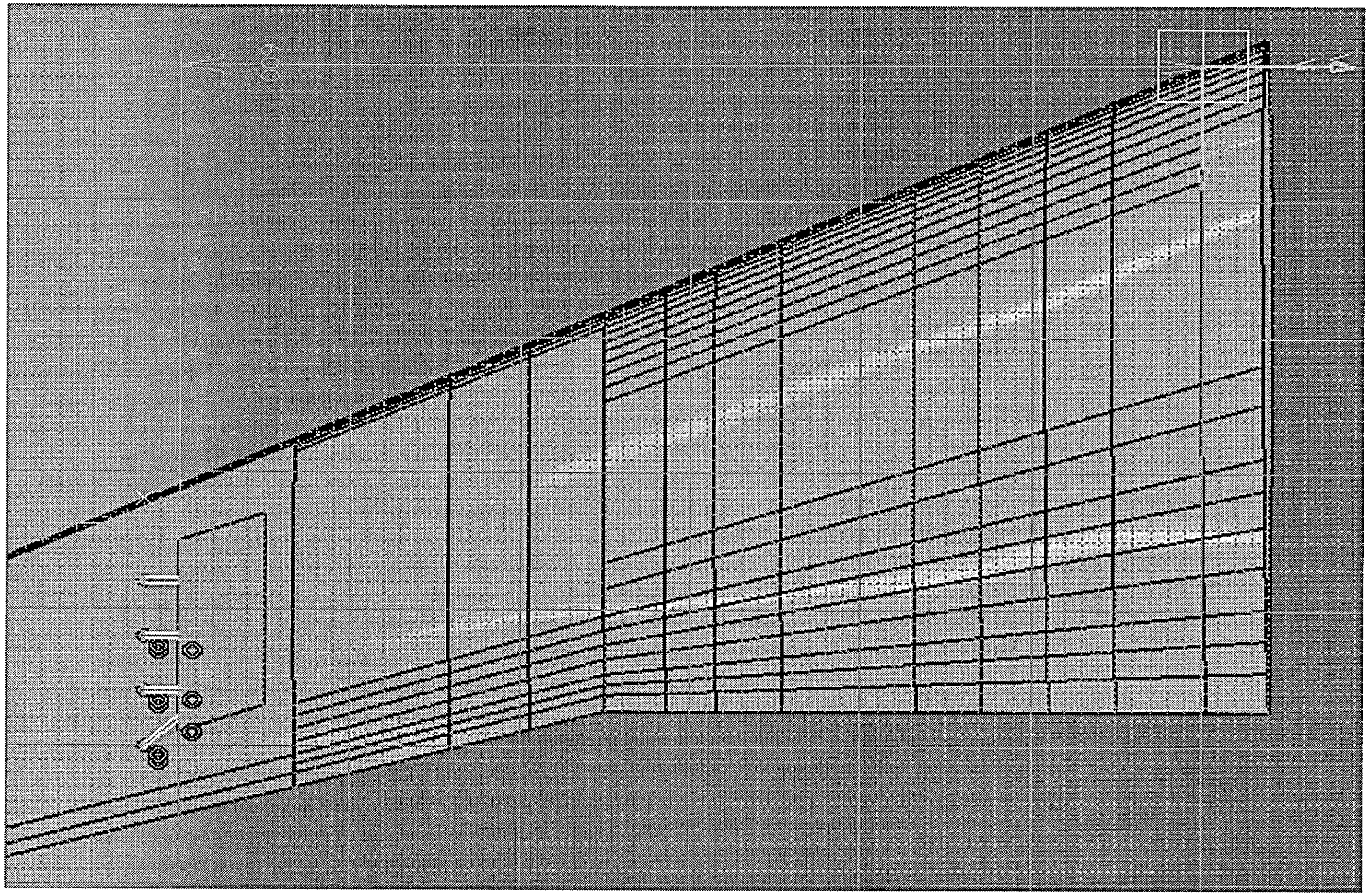


Figure 4.5 Constraints set for primary channels on the design automation model in CATIA (Knowledgeware) KBE application

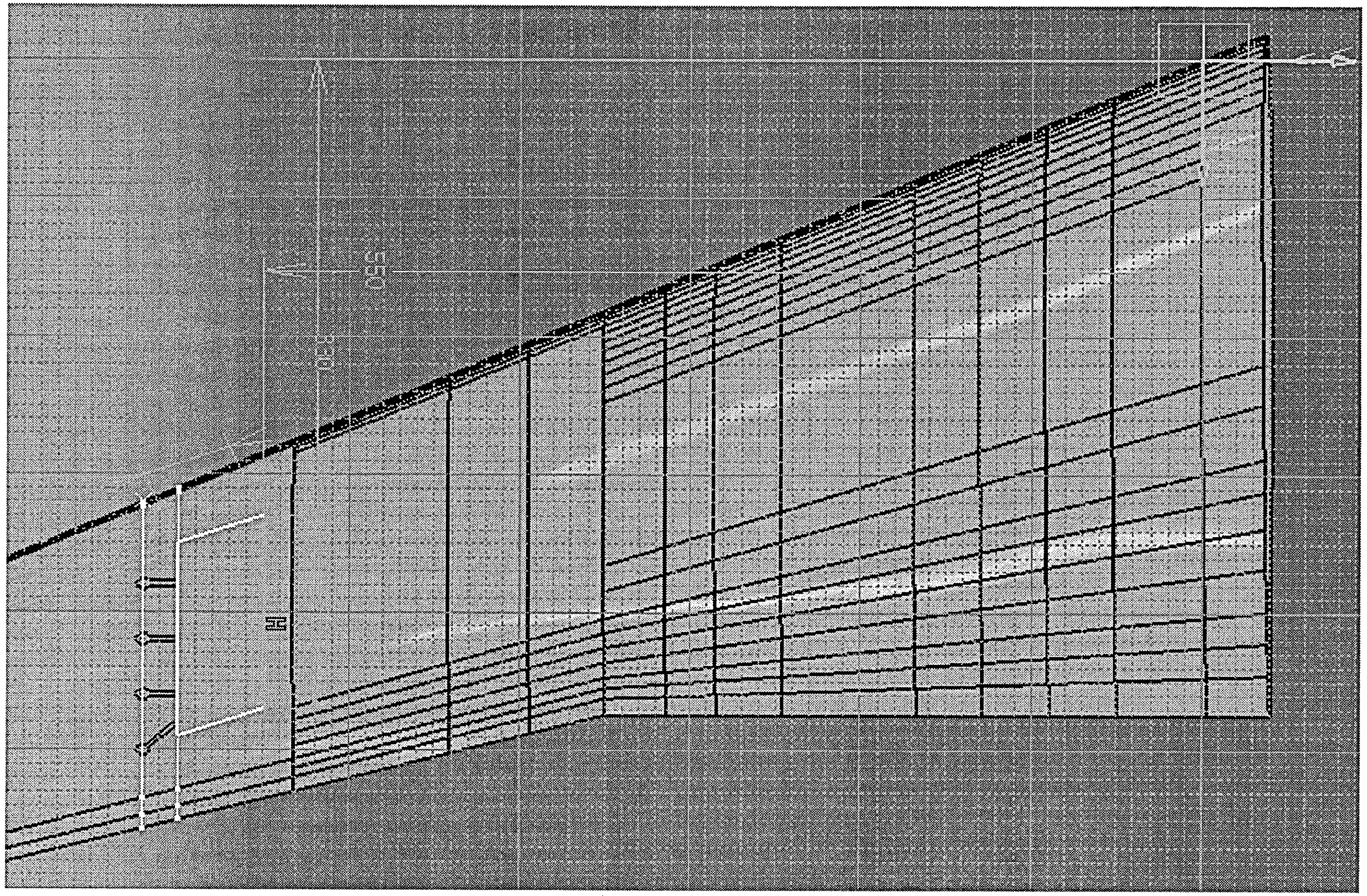


Figure 4.6 Constraints set for secondary channel on the design automation model in CATIA (Knowledgeware) KBE application

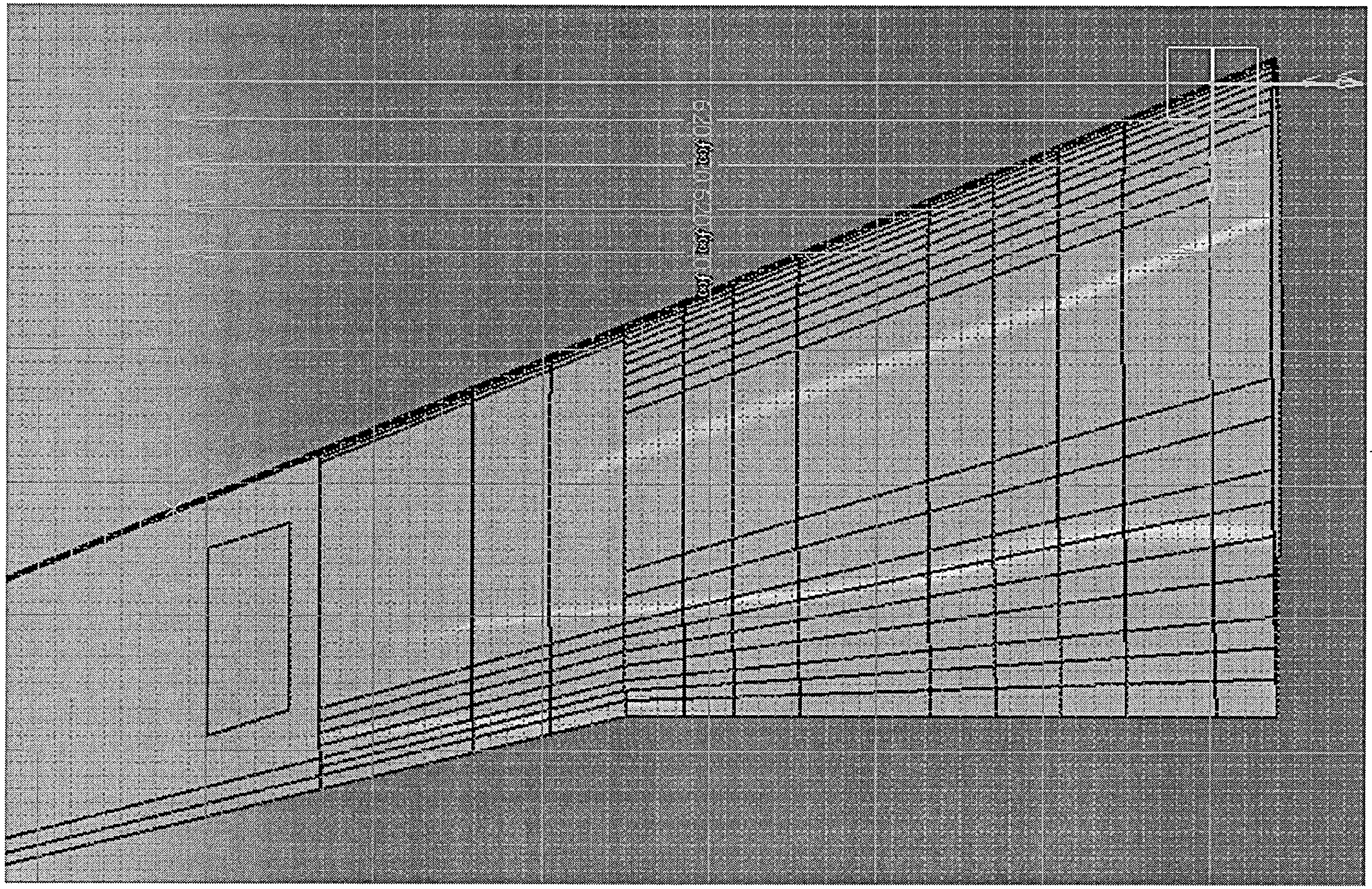


Figure 4.7 Constraints set for tappings on the design automation model in CATIA (Knowledgeware) KBE application

Figure 4.8 Formulas set for tappings, primary and secondary channels on the design automation model in CATIA (Knowledgeware) KBE application

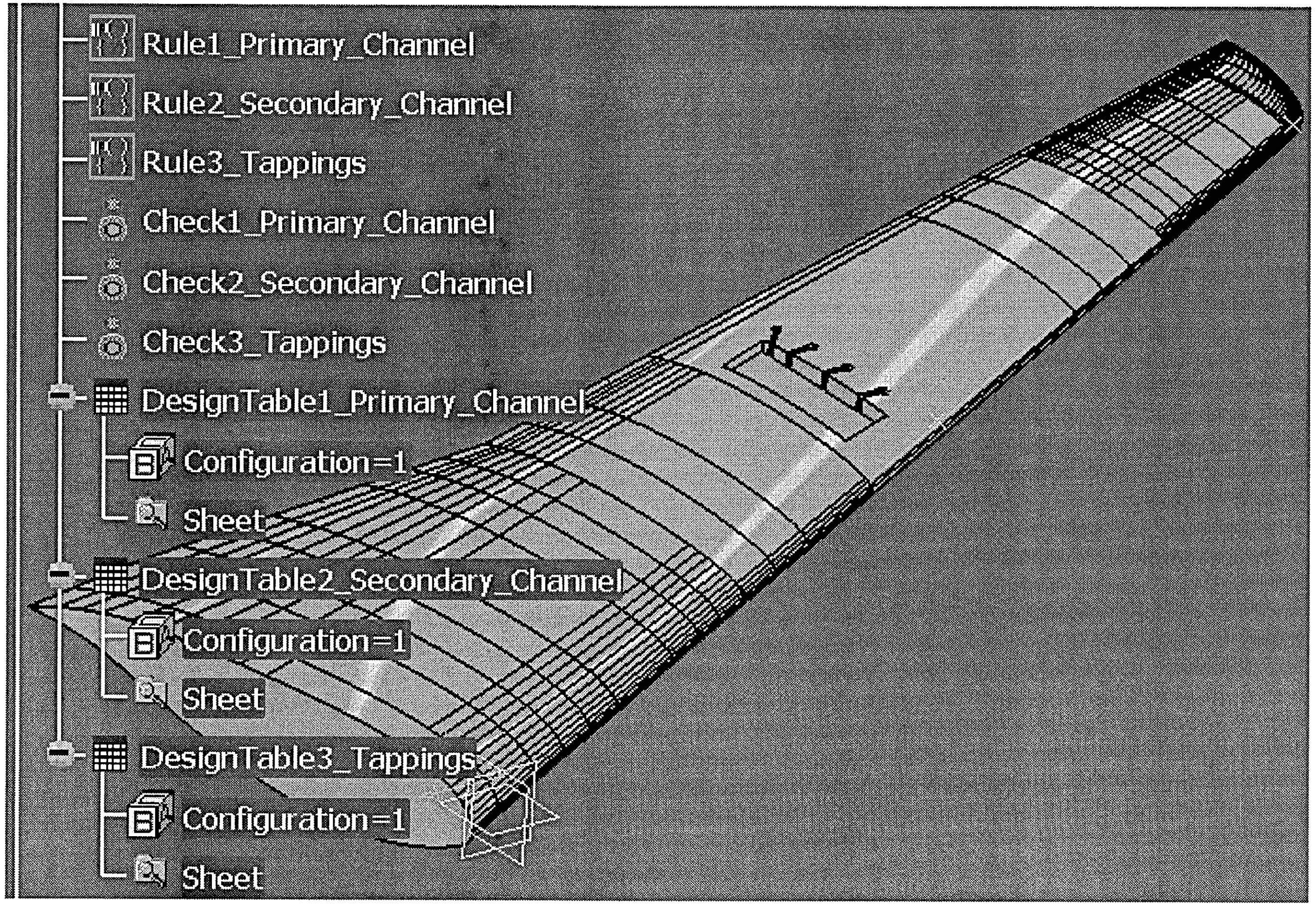


Figure 4.9 Rules, checks and design tables defined on the design automation model

in CATIA (Knowledgeware) KBE application

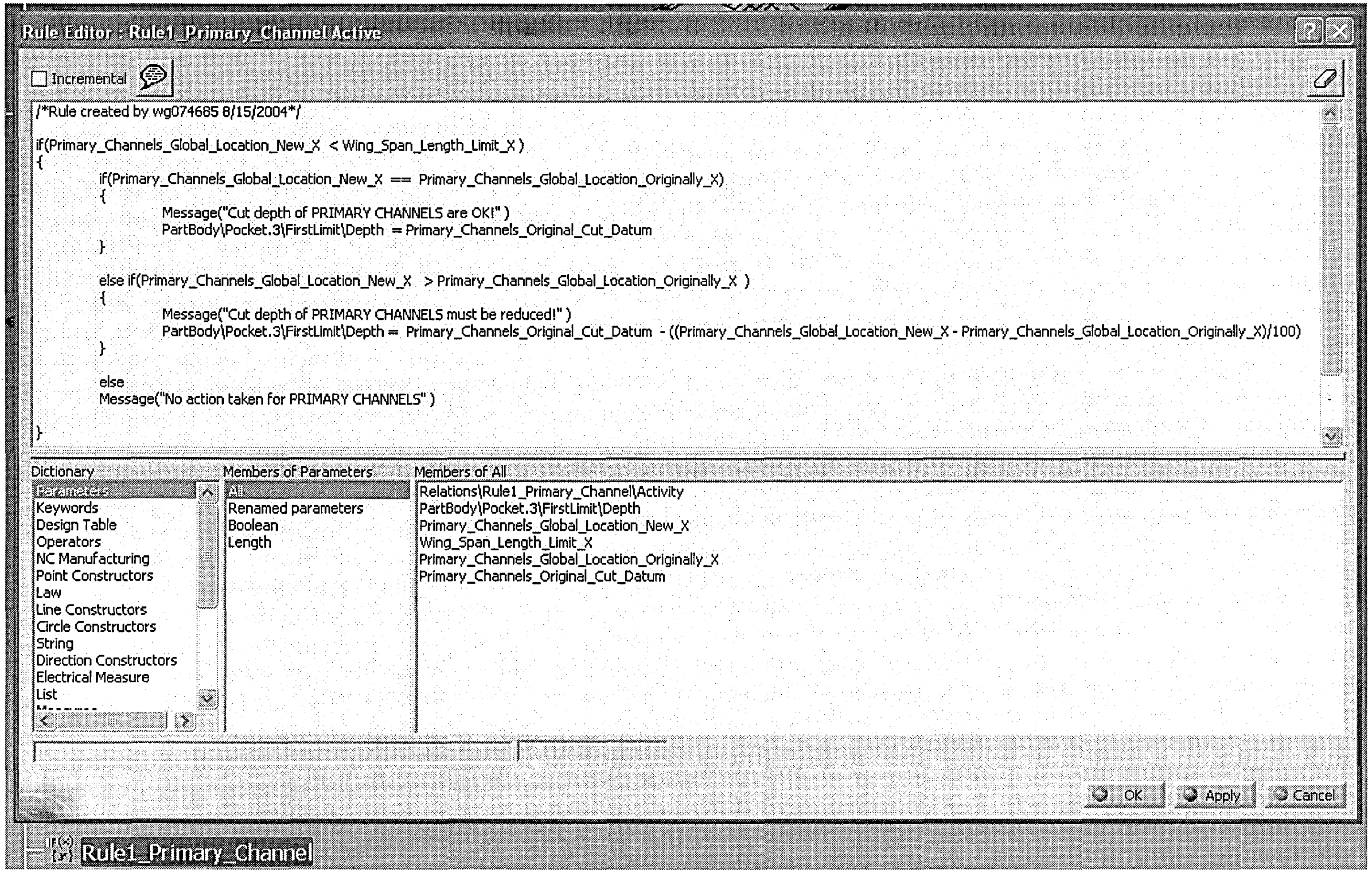


Figure 4.10 Detail of Rule 1 defined for primary channel on the design automation model

in CATIA (Knowledgeware) KBE application

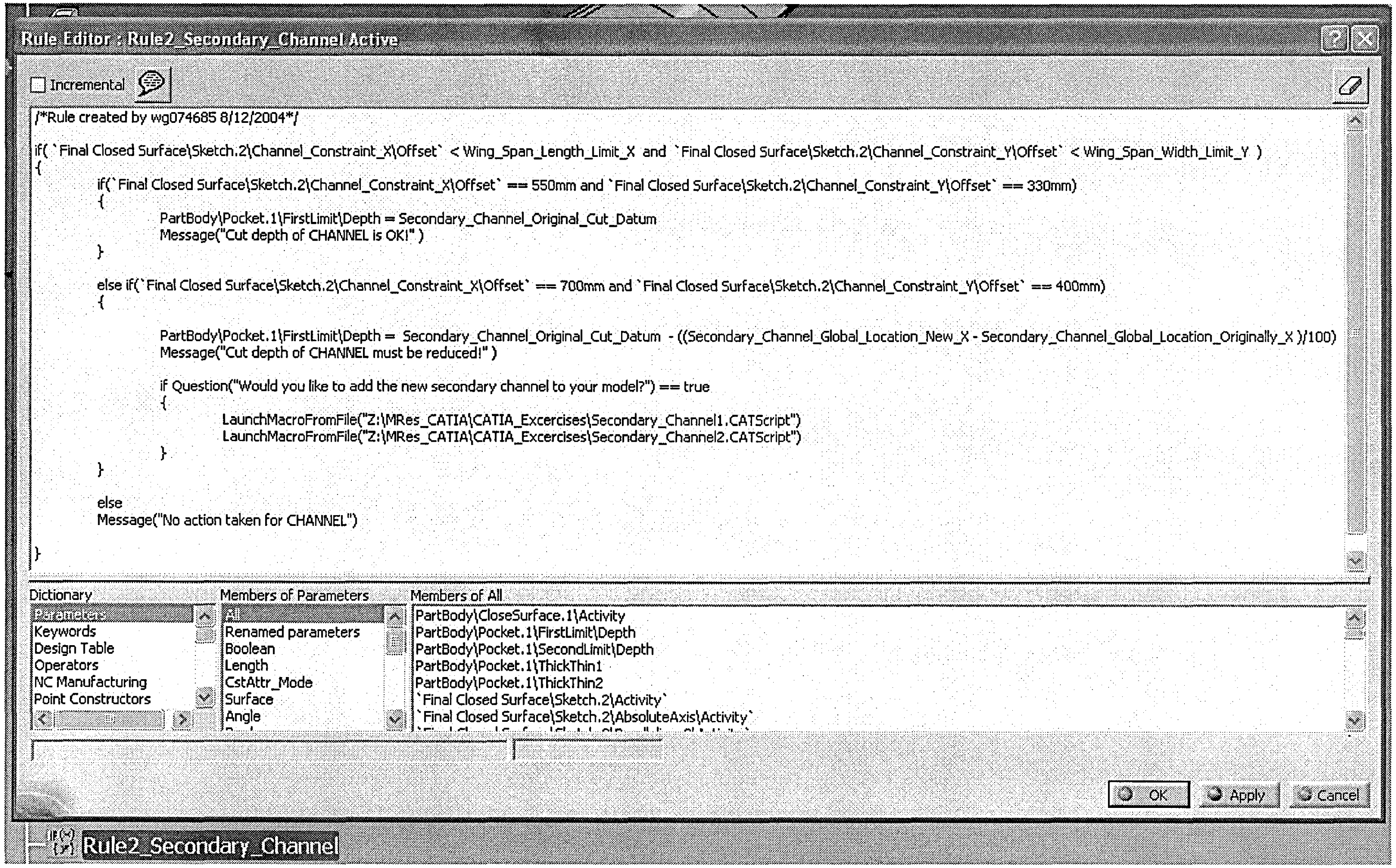


Figure 4.11 Detail of Rule 2 defined for secondary channel on the design automation model in CATIA (Knowledgeware) KBE application

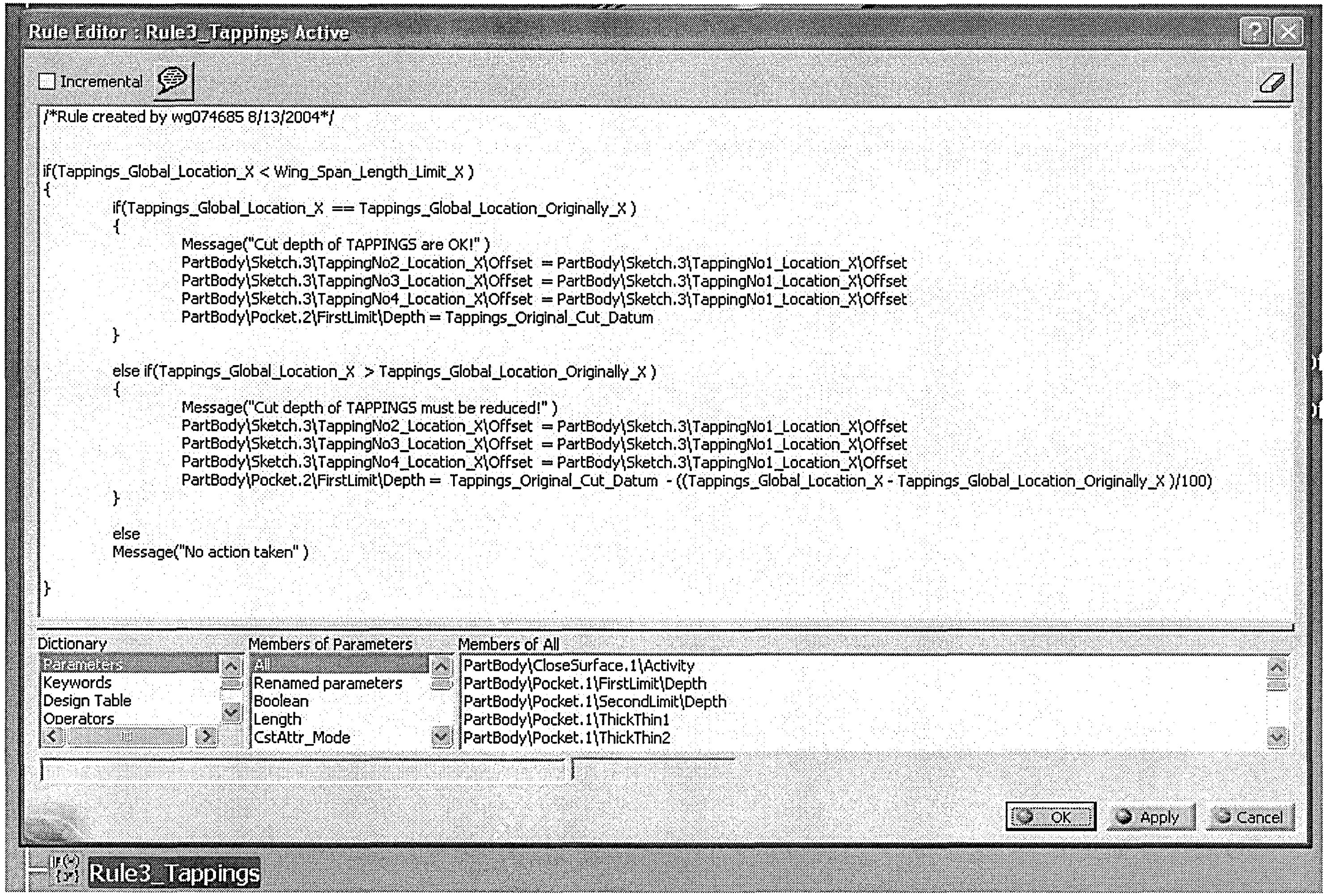


Figure 4.12 Detail of Rule 3 defined for tappings on the design automation model in CATIA (Knowledgeware) KBE application

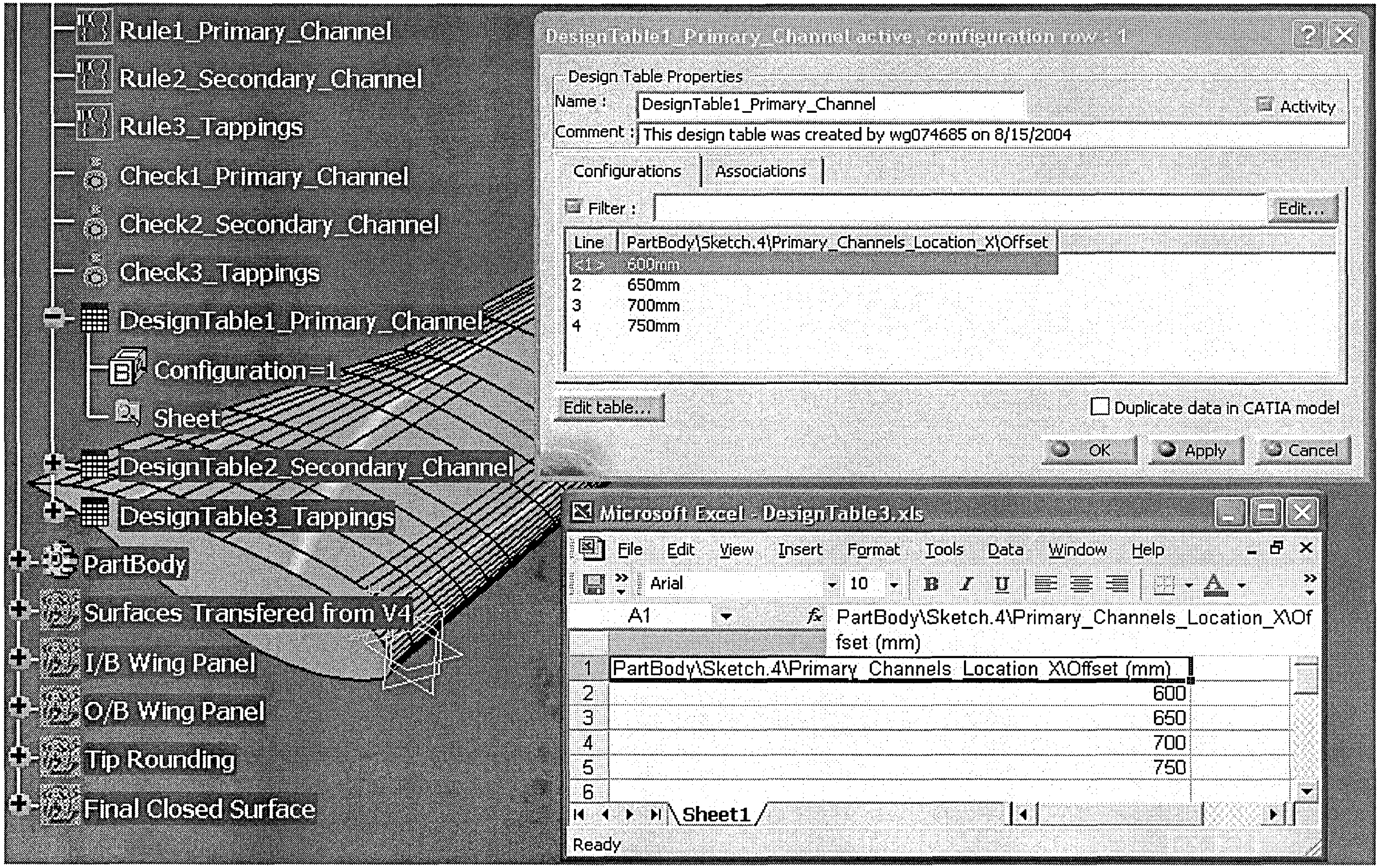


Figure 4.13 Detail of Design Table 1 defined in Excel spreadsheet for primary channel on the design automation model in CATIA (Knowledgeware) KBE application

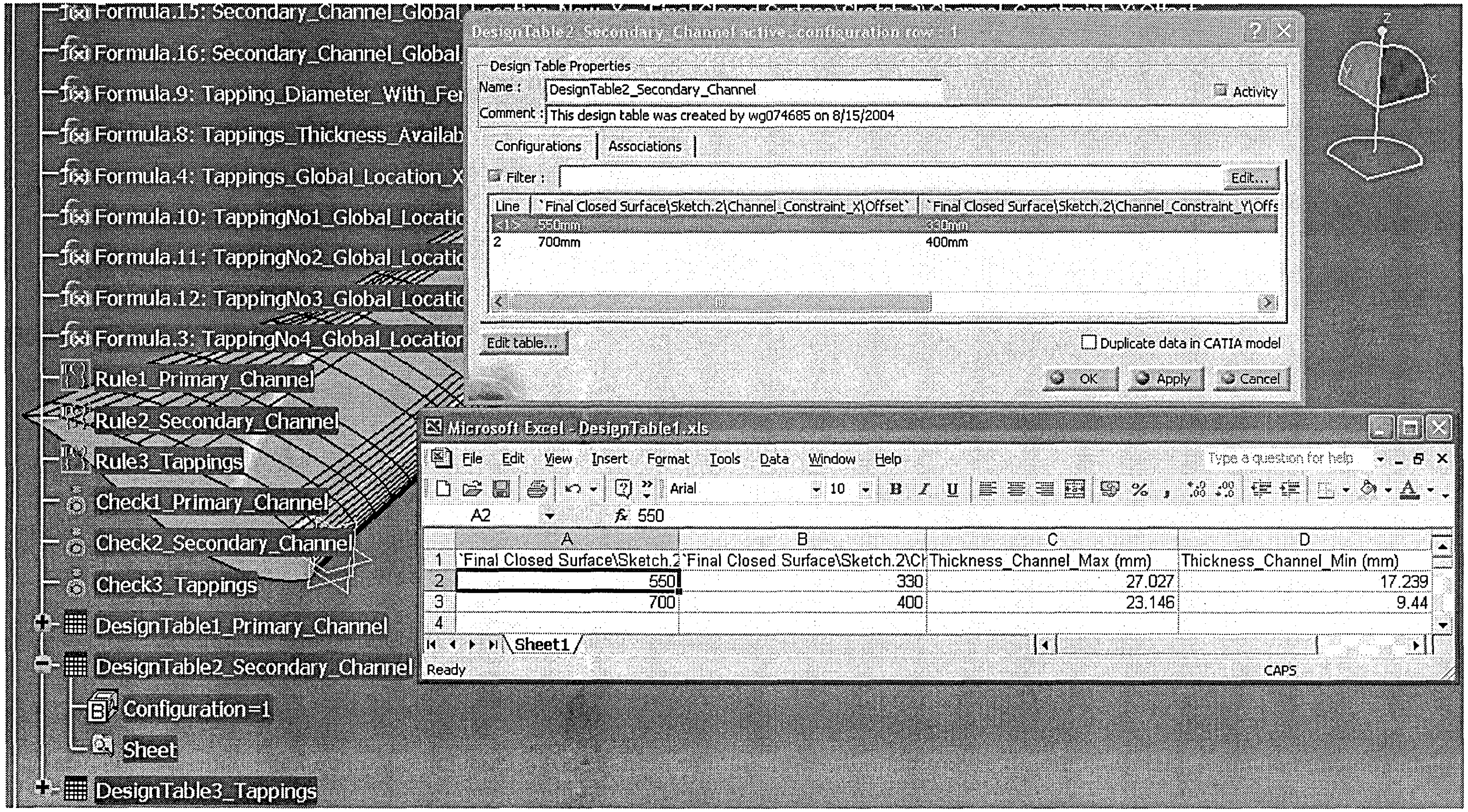


Figure 4.14 Detail of Design Table 2 defined in Excel spreadsheet for secondary channel on the design automation model in CATIA (Knowledgeware) KBE application

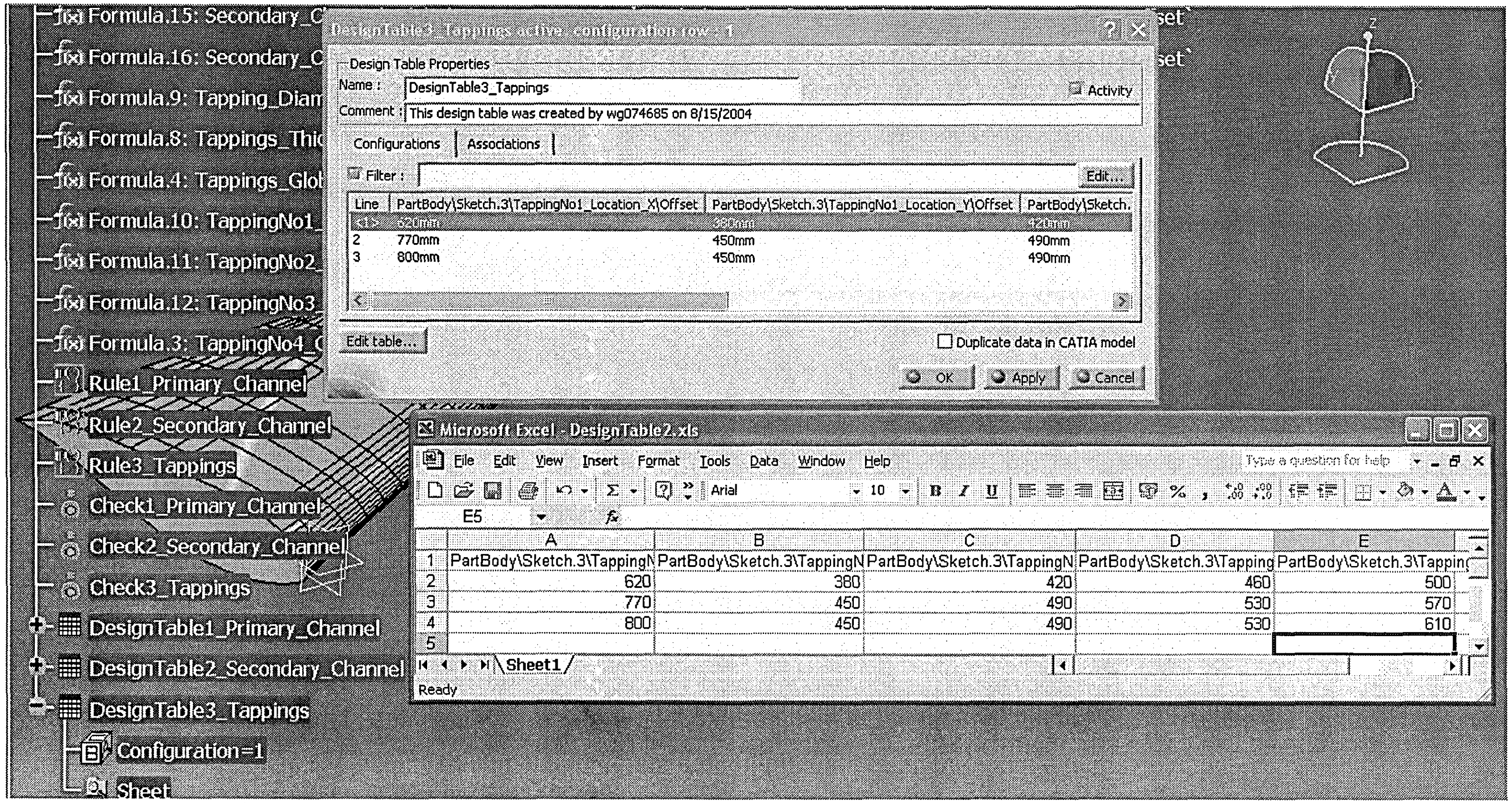


Figure 4.15 Detail of Design Table 3 defined in Excel spreadsheet for tappings on the design automation model in CATIA (Knowledgeware) KBE application

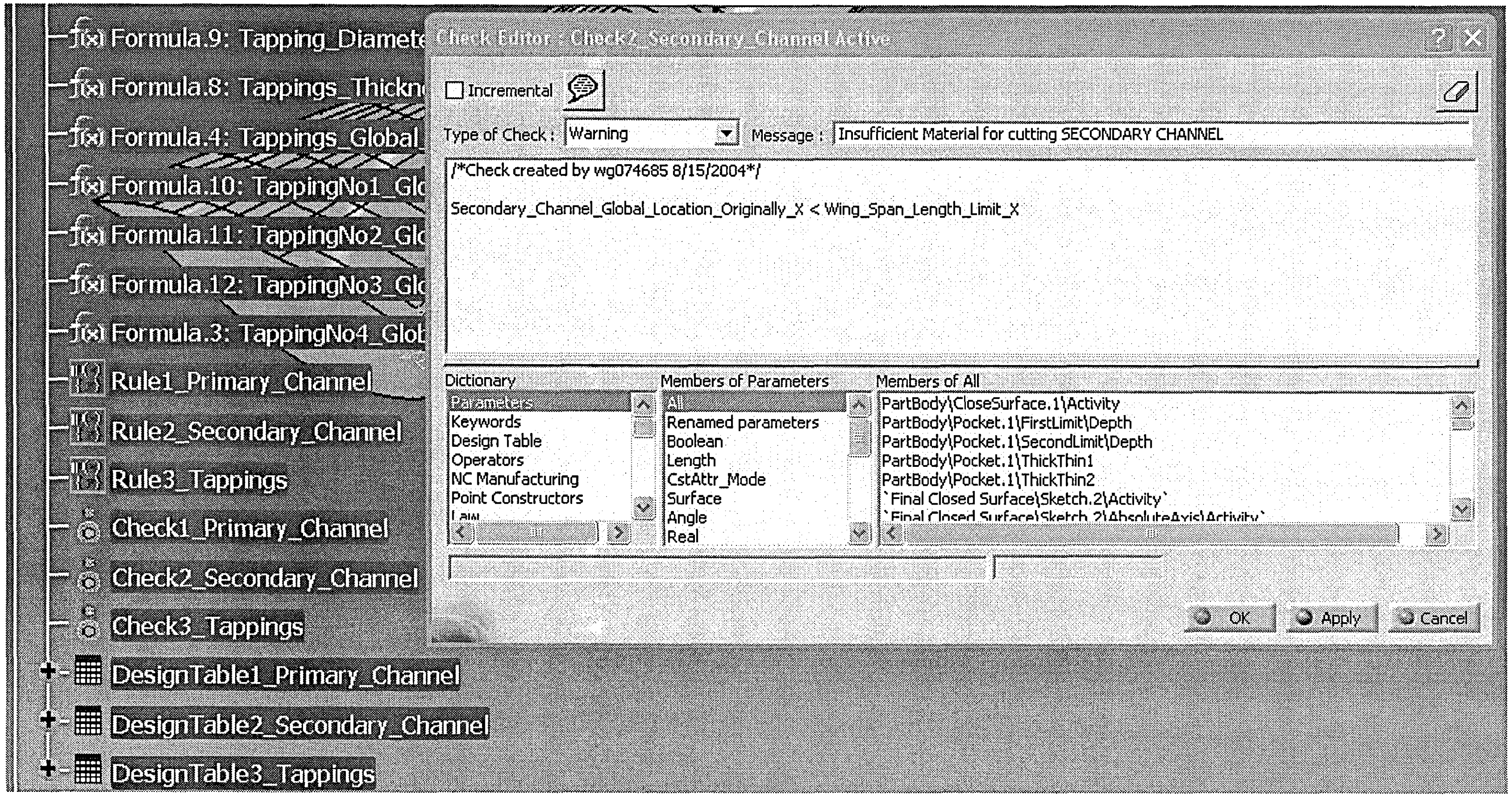


Figure 4.16 Detail of a simple check defined for the secondary channel on the design automation model in CATIA (Knowledgeware) KBE application

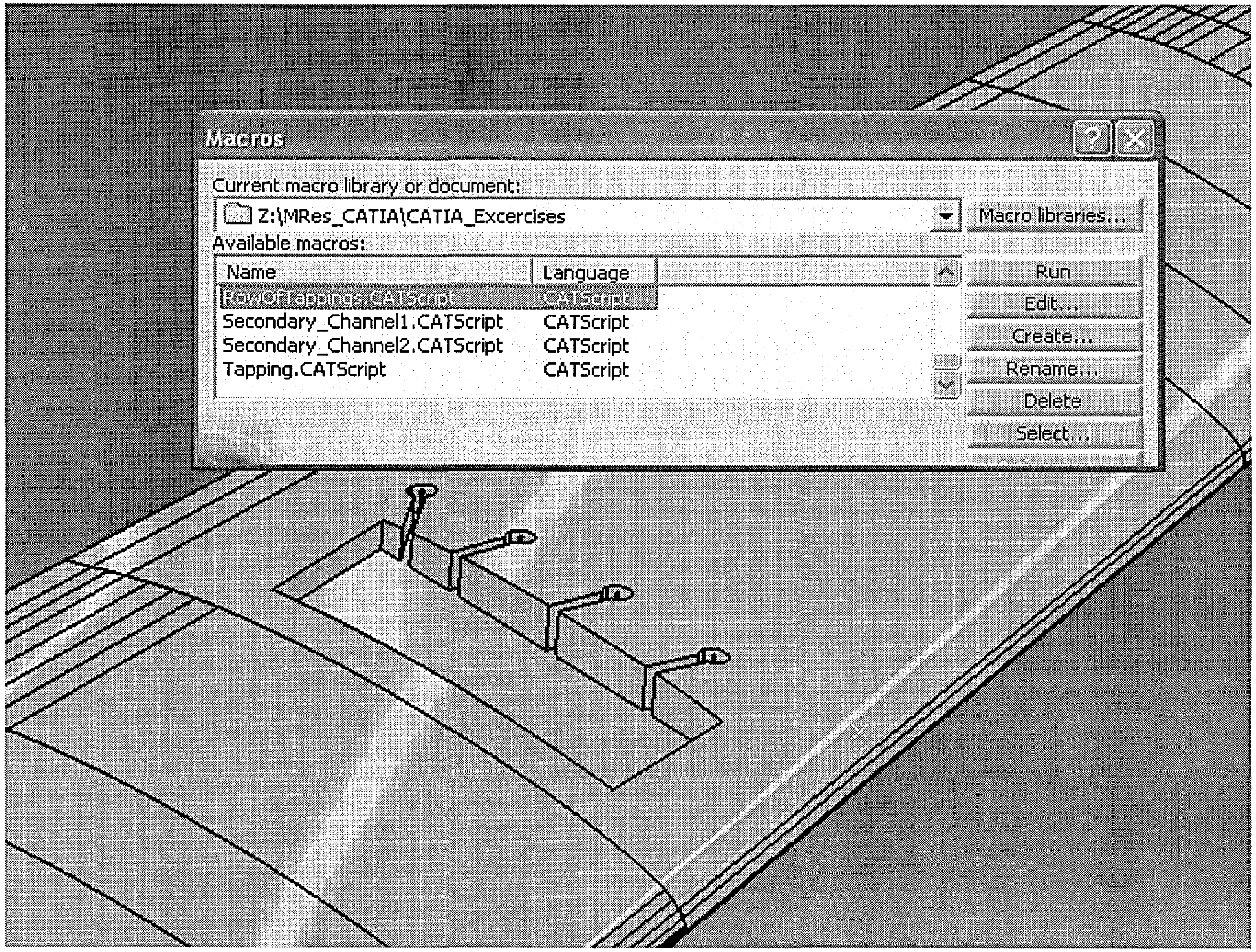


Figure 4.17 Macros produced for the automation of modelling and design of tappings and secondary channel on the design automation model in CATIA (Knowledgeware) KBE application

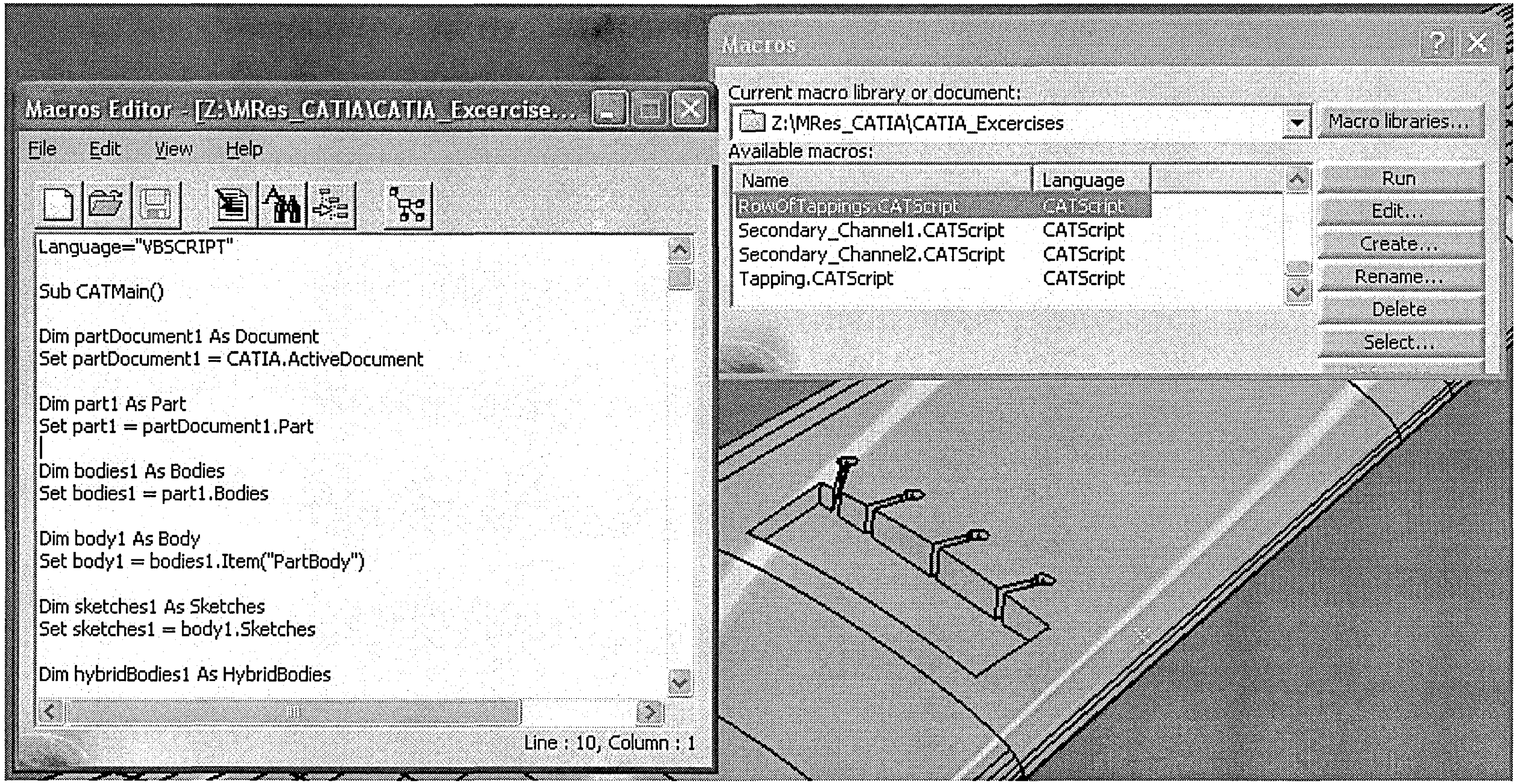


Figure 4.18 Sample of codes written using the CAA IDL API for one of the macros on the design automation model in CATIA (Knowledgeware) KBE application

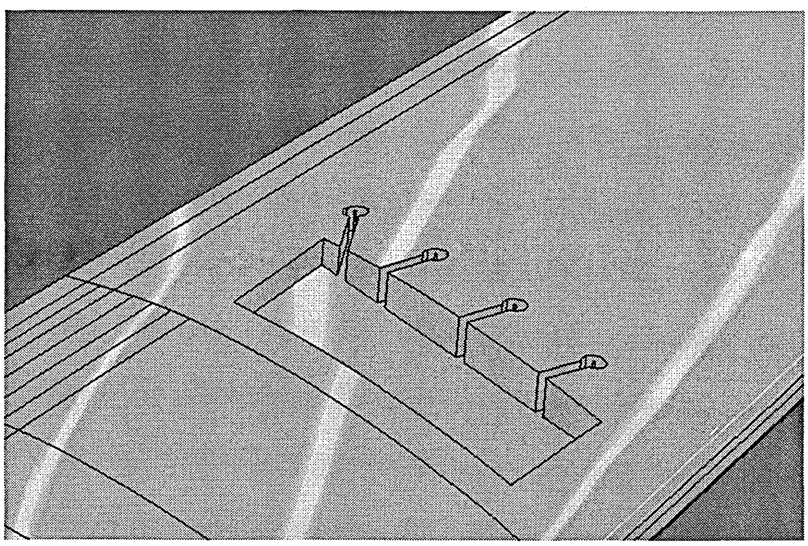
4.3. Knowledge Model (ICARE Forms) of Aircraft Wing Model

Engineering rules, relationships, constraints, formulas and parameters defined for the design automation model on the CATIA (Knowledgeware) KBE application are now translated into illustrations, constraints, activities, rules and entities on the informal knowledge model using the MOKA's ICARE forms.

The ICARE forms are built in such a way that they reflects the inner working of the generative modelling and design function defined on the CATIA (Knowledgeware) KBE application. Thus, any engineering rule and relationship used by the generative modelling and design function of the KBE application may be seen and understood by viewing the human friendly ICARE forms which are responsible for mapping and linking related illustrations, constraints, activities, rules and entities.

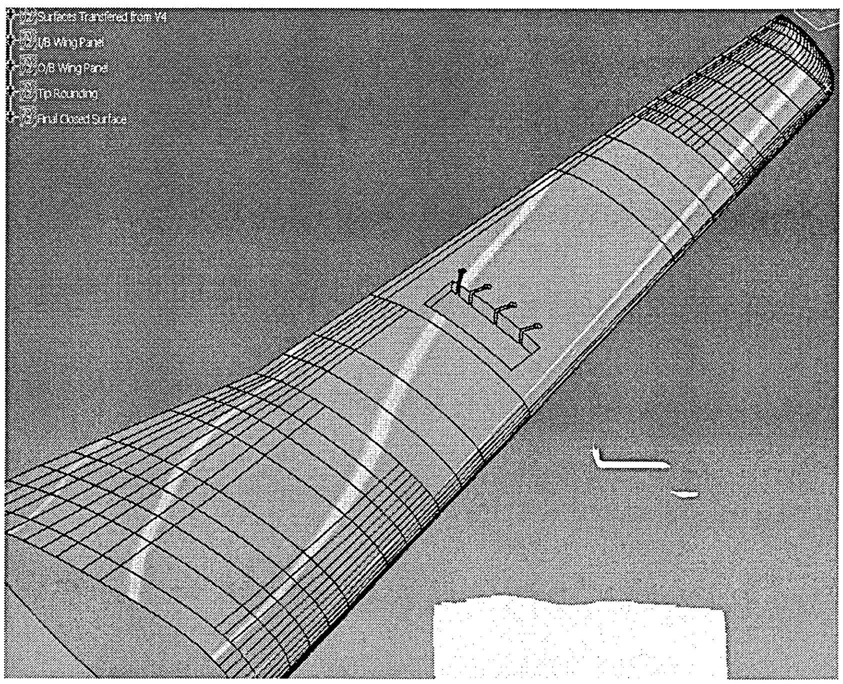
4.3.1. I-Forms

I-forms serve the purpose of providing any additional or relevant information which may make it easier for other people to understand what information and knowledge are being captured and stored on the database.

I-Form	
Name	Tappings_Channel_Illustration
Reference	I_Tappings_Channels001
Context, info, validity	Illustration may be applied in cases where pressure plots are required for models in wind tunnel testing involving tappings, primary channels and secondary channel.
Description	Illustration showing what the tappings, primary channels and secondary channel look like and how they fit together. 
Related Constraints	C_Wing_Span_Length_Limit_X001, C_Wing_Span_Width_Limit_Y001, C_Primary_Channel_Length001, C_Primary_Channel_Width001, C_Primary_Channel_Thickness_Available_Max001, C_Primary_Channel_Thickness_Available_Min001,

	C_Secondary_Channel_Length001, C_Secondary_Channel_Width001, C_Secondary_Channel_Thickness_Available_Max001, C_Secondary_Channel_Thickness_Available_Min001, C_Tapping_Diameter_With_Ferrule001, C_Tapping_Diameter_Without_Ferrule001, C_Tapping_Thickness_Available_Max001, C_Tapping_Thickness_Available_Min001
Related Activities	A_Cut_Primary_Channel001, A_Cut_Secondary_Channel001, A_Drill_Tapping001
Related Rules	R_Primary_Channel001, R_Secondary_Channel001, R_Tapping001
Related Entities	E_PrimaryChannel001, E_ SecondaryChannel001, E_Tapping001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.1 I-form of tappings, primary and secondary channels together

<table>
<tr><th colspan="2">I-Form</th></tr>
<tr><td>Name</td><td>Wing_Span_Illustration</td></tr>
<tr><td>Reference</td><td>I_Wing_Span001</td></tr>
<tr><td>Context, info, validity</td><td>Illustration may be applied in cases where pressure plots are required for models in wind tunnel testing involving tappings, primary channels and secondary channel.</td></tr>
<tr><td>Description</td><td>Illustration showing the tappings, primary channels and secondary channel are located within the length and width spans of the wing.

</td></tr>
<tr><td>Related Constraints</td><td>C_Wing_Span_Length_Limit_X001,
C_Wing_Span_Width_Limit_Y001,
C_Primary_Channel_Thickness_Available_Max001,
C_Primary_Channel_Thickness_Available_Min001,
C_Secondary_Channel_Thickness_Available_Max001,
C_Secondary_Channel_Thickness_Available_Min001,
C_Tapping_Diameter_With_Ferrule001,
C_Tapping_Diameter_Without_Ferrule001,
C_Tapping_Thickness_Available_Max001,
C_Tapping_Thickness_Available_Min001</td></tr>
</table>

Related Activities	A_Cut_Primary_Channel001, A_Cut_Secondary_Channel001, A_Drill_Tapping001
Related Rules	R_Primary_Channel001, R_Secondary_Channel001, R_Tapping001
Related Entities	E_PrimaryChannel001, E_ SecondaryChannel001, E_Tapping001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.2 I-form of tappings, primary and secondary channels within the length and width of the wing

4.3.2. C-Forms

C-forms describe any constraint that exists on the informal knowledge model. Constraints are linked to the appropriate entities (Constraint→Entity) and serve the purpose of placing limitation on the entities.

C-Form	
Name	Wing_Span_Length_Limit_X
Reference	C_Wing_Span_Length_Limit_X001
Objective	Set the max span limit along the length of the wing (x-axis)
Context, info, validity	Use by tappings, primary and secondary channels to check their location along the length of the wing, i.e. the x-axis
Description	Use by the tappings, primary and secondary channels to check that they are actually located within the span of the wing.
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001
Related Rules	R_Primary_Channel001, R_ Secondary_Channel001, R_Tapping001
Related Entities	E_Primary_Channel001, E_ Secondary_Channel001, E_Tapping001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.3 C-form for span along the length (along x-axis) of the wing

C-Form	
Name	Wing_Span_Width_Limit_Y
Reference	C_Wing_Span_Width_Limit_Y001
Objective	Set the max width limit along the width the wing (y-axis)
Context, info, validity	Use by tappings, primary and secondary channels to check their location along the width of the wing, i.e. the y-axis.
Description	Use by the tappings, primary and secondary channels to check that they are actually located within the width span of the wing.
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001
Related Rules	R_Primary_Channel001, R_ Secondary_Channel001, R_Tapping001
Related Entities	E_Primary_Channel001, E_ Secondary_Channel001, E_Tapping001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.4 C-form for the width (along y-axis) of the wing

C-Form	
Name	Primary_Channel_Length
Reference	C_Primary_Channel_Length001
Objective	Set the limit for the length of the primary channel which runs along the length of the wing span.
Context, info, validity	Apply to primary channels used in wind tunnel testing to obtain the pressure plots.
Description	Primary channels may have a range of fixed sizes of length and width and constraints such as this one may be used to limit the size of particular primary channel being used.
Related Illustrations	I_Tappings_Channels001
Related Rules	R_Primary_Channel001
Related Entities	E_Primary_Channel001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.5 C-form for the length of primary channel

C-Form	
Name	Primary_Channel_Width
Reference	C_Primary_Channel_Width001
Objective	Set the limit for the width of the primary channel which runs perpendicular to the length span of the wing.
Context, info, validity	Apply to primary channels used in wind tunnel testing to obtain the pressure plots.
Description	Primary channels may have a range of fixed sizes of length and width and constraints such as this one may be used to limit the size of particular primary channel being used.
Related Illustrations	I_Tappings_Channels001
Related Rules	R_Primary_Channel001
Related Entities	E_Primary_Channel001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.6 C-form for the width of primary channel

C-Form	
Name	Primary_Channel_Thickness_Available_Max
Reference	C_Primary_Channel_Thickness_Available_Max001
Objective	Max. thickness of material available for the primary channel at the specified location along the length of the wing.
Context, info, validity	Can only be applied to primary channels in areas where the thickness of the material is sufficiently thick to allow the max. depth of the cut.
Description	Due to the curved shape of the wing the thickness of the wing does not remain constant along the length and width of the wing, therefore the thickness of material available for the primary channel varies along the length and width of the channel giving a max and min thickness.
Related Illustrations	I_TappingsChannels001, I_WingSpan001
Related Rules	R_Primary_Channel001
Related Entities	E_Primary_Channel001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.7 C-form for thickness (max) of primary channel

C-Form	
Name	Primary_Channel_Thickness_Available_Min
Reference	C_Primary_Channel_Thickness_Available_Min001
Objective	Min. thickness of material available for the primary channel at the specified location along the length of the wing.
Context, info, validity	Normally, it is the min. thickness that determines the how deep the cut of the primary channel should be at the specified location along the length of the wing.
Description	Due to the curved shape of the wing the thickness of the wing does not remain constant along the length and width of the wing, therefore the thickness of material available for the primary channel varies along the length and width of the channel giving a max and min thickness.
Related Illustrations	I_TappingsChannels001, I_WingSpan001
Related Rules	R_Primary_Channel001
Related Entities	E_Primary_Channel001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.8 C-form for the thickness (min) of primary channel

C-Form	
Name	Secondary_Channel_Length
Reference	C_Secondary_Channel_Length001
Objective	Set the limit for the length of the secondary channel which runs along the length of the wing span.
Context, info, validity	Apply to secondary channels used in wind tunnel testing to obtain the pressure plots.
Description	Secondary channels may have a range of fixed sizes of length and width and constraints such as this one may be used to limit the size of particular secondary channel being used.
Related Illustrations	I_Tappings_Channels001
Related Rules	R_Secondary_Channel001
Related Entities	E_Secondary_Channel001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.9 C-form for the length of secondary channel

C-Form	
Name	Secondary_Channel_Width
Reference	C_Secondary_Channel_Width001
Objective	Set the limit for the width of the secondary channel which runs perpendicular to the length span of the wing.
Context, info, validity	Apply to secondary channels used in wind tunnel testing to obtain the pressure plots.
Description	Secondary channels may have a range of fixed sizes of length and width and constraints such as this one may be used to limit the size of particular secondary channel being used.
Related Illustrations	I_Tappings_Channels001
Related Rules	R_Secondary_Channel001
Related Entities	E_Secondary_Channel001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.10 C-form for the width of secondary channel

C-Form	
Name	Secondary_Channel_Thickness_Available_Max
Reference	C_Secondary_Channel_Thickness_Available_Max001
Objective	Max. thickness of material available for the secondary channel at the specified location along the length of the wing.
Context, info, validity	Can only be applied to secondary channels in areas where the thickness of the material is sufficiently thick to allow the max. depth of the cut.
Description	Due to the curved shape of the wing the thickness of the wing does not remain constant along the length and width of the wing, therefore the thickness of material available for the secondary channel varies along the length and width of the channel giving a max and min thickness.
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001
Related Rules	R_Secondary_Channel001
Related Entities	E_Secondary_Channel001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.11 C-form for thickness (max) of secondary channel

C-Form	
Name	Secondary_Channel_Thickness_Available_Min
Reference	C_Secondary_Channel_Thickness_Available_Min001
Objective	Min. thickness of material available for the secondary channel at the specified location along the length of the wing.
Context, info, validity	Normally, it is the min. thickness that determines the how deep the cut of the secondary channel should be at the specified location along the length of the wing.
Description	Due to the curved shape of the wing the thickness of the wing does not remain constant along the length and width of the wing, therefore the thickness of material available for the secondary channel varies along the length and width of the channel giving a max and min thickness.
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001
Related Rules	R_Secondary_Channel001
Related Entities	E_Secondary_Channel001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.12 C-form for the thickness (min) of secondary channel

C-Form	
Name	Tapping_Diameter_With_Ferrule
Reference	C_Tapping_Diameter_With_Ferrule001
Objective	Set the diameter of tappings with ferrules.
Context, info, validity	Only apply to locations of tappings where the thickness of the wing is sufficiently thick to hold a ferrule.
Description	The holes are drilled at specified locations for the tappings where the pressure measurement valves held inside ferrules.
Related Illustrations	I_Tappings_Channels001
Related Rules	R_Tapping001
Related Entities	E_Tapping001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.13 C-form for diameter of tapping with ferrule

C-Form	
Name	Tapping_Diameter_Without_Ferrule
Reference	C_Tapping_Diameter_Without_Ferrule001
Objective	Set the diameter of tappings without ferrules.
Context, info, validity	Only apply to locations of tappings where the thickness of the wing is insufficiently thick to hold a ferrule.
Description	The holes are drilled at specified locations for the tappings where the pressure measurement valves are slotted through.
Related Illustrations	I_Tappings_Channels001
Related Rules	R_Tapping001
Related Entities	E_Tapping001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.14 C-form for diameter of tapping without ferrule

C-Form	
Name	Tapping_Thickness_Available_Max
Reference	C_Tapping_Thickness_Available_Max001
Objective	Max. thickness of material available for the tapping at the specified location along the length of the wing.
Context, info, validity	Max thickness can only be applied to tappings in areas where the thickness of the material is sufficiently thick to allow the max. depth of the drilling.
Description	Due to the curved shape of the wing the thickness of the wing does not remain constant along the length and width of the wing, therefore the thickness of material available for the tappings vary along the length and width of the wing giving a max and min thickness.
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001
Related Rules	R_Tapping001
Related Entities	E_Tapping001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.15 C-form for the thickness (max) of tapping

C-Form	
Name	Tapping_Thickness_Available_Min
Reference	C_Tapping_Thickness_Available_Min001
Objective	Min. thickness of material available for the tapping at the specified location along the length of the wing.
Context, info, validity	Normally, it is the min. thickness that determines the how deep the drilling of the tapping should be at the specified location along the length of the wing.
Description	Due to the curved shape of the wing the thickness of the wing does not remain constant along the length and width of the wing, therefore the thickness of material available for the tappings vary along the length and width of the wing giving a max and min thickness.
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001
Related Rules	R_Tapping001
Related Entities	E_Tapping001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.16 C-form for the thickness (min) of tapping

4.3.3. A-Forms

A-forms are used for describing the activities that are required in the design process, thus recording the WHAT to-do in the design process (the HOW to-do in the design process is recorded on the R-form). Activities are linked to the appropriate rules (Activity→Rule).

A-Form		
Name	Cut_Primary_Channel	
Reference	A_Cut_Primary_Channel001	
Objective	Cut slot for primary channel	
Trigger	Need to store and route wires and cables of tubings.	
Input	E_PrimaryChannel001	
Output	Cutting for primary channel	
Input requirements	Coordinates with max and min thicknesses of material available at the specified location where the primary channel is to be cut.	
Potential failure modes	• Sufficiently deep to store and route any wires and cables • Cutting may remove too much material beyond a set limit • Must not cut right through the wing	
Context, info, validity	Activity can be applied whenever a primary channel is required.	
Description	Before this activity can be carried out, sub-activities may be required to determine the exact position of the primary channel and its depth, providing the following data: • Coordinates, e.g. x, y and/or z, marking the position of secondary channel • Length of primary channel • Width of primary channel • Depth of primary channel	
Related Activities	Parent activity: Sub-activities: Preceding activities: Following activities:	A_Cut_Secondary_Channel001, A_Drill_Tapping001
Rules involved	R_Primary_Channel001	
Entities required	E_PrimaryChannel001	
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001	
Information Origin	Diem Lam, Cranfield University, 17 August 2004	
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete	

Table 4.17 A-form for primary channel

A-Form		
Name	Cut_Secondary_Channel	
Reference	A_Cut_Secondary_Channel001	
Objective	Cut slot for secondary channel	
Trigger	Need to store wires and cables of tubings.	
Input	E_ SecondaryChannel001	
Output	Cutting for secondary channel	
Input requirements	Coordinates with max and min thicknesses of material available at the specified location where the secondary channel is to be cut.	
Potential failure modes	• Sufficiently deep to store and route any wires and cables • Cutting may remove too much material beyond a set limit • Must not cut right through the wing	
Context, info, validity	Activity can be applied whenever a secondary channel is required.	
Description	Before this activity can be carried out, sub-activities may be required to determine the exact position of the secondary channel and its depth, providing the following data: • Coordinates, e.g. x, y and/or z, marking the position of secondary channel • Length of secondary channel • Width of secondary channel • Depth of secondary channel	
Related Activities	Parent activity: Sub-activities: Preceding activities: Following activities:	A_Cut_Primary_Channel001, A_Drill_Tapping001
Rules involved	R_Primary_Channel001	
Entities required	E_PrimaryChannel001	
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001	
Information Origin	Diem Lam, Cranfield University, 17 August 2004	
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete	

Table 4.18 A-form for secondary channel

A-Form		
Name	Drill_Tapping	
Reference	A_Drill_Tapping001	
Objective	Drill hole for tapping	
Trigger	Chosen location to obtain the pressure plot	
Input	E_Tapping001	
Output	Hole for the tapping	
Input requirements	Coordinates with max and min thicknesses of material available at the specified location where the tapping is to be drilled.	
Potential failure modes	• Sufficiently deep to hold a ferrule for tapping with ferrule • Hole may remove too much material beyond a set limit • Hole must not be drilled right through the wing	
Context, info, validity	Activity can be applied whenever a tapping is required.	
Description	Before this activity can be carried out, sub-activities may be required to determine the exact position of the tapping and its depth, providing the following data: • Coordinates, e.g. x, y and z, marking the position of tapping • Diameter of tapping • Depth of tapping	
Related Activities	Parent activity: Sub-activities: Preceding activities: Following activities:	A_Cut_Primary_Channel001, A_Cut_Secondary_Channel001
Rules involved	R_Tapping001	
Entities required	E_Tapping001	
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001	
Information Origin	Diem Lam, Cranfield University, 17 August 2004	
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete	

Table 4.19 A-form for tapping

4.3.4. R-Forms

R-forms are used to define the rules that exist in the design process, thus recording the HOW to-do in the design process (the WHAT to-do in the design process is recorded on the A-form). Rules are linked to by the appropriate activities (Activity→Rule).

R-Form	
Name	Rule1_Primary_Channel
Reference	R_Primary_Channel001
Objective	Rule to determine the depth of the cutting for the primary channel.
Context, info, validity	Apply whenever primary channels are used.
Description	The rule is based on a pro rata equation that the depth of the cutting for the primary channels is reduced as they move towards the tip of the wing, because the thickness of the wing becomes thinner from root to tip. Equation: cutting depth = original cut datum[i] – ((new location along the length span of the wing – old location along the length span of the wing)/100) [i]*Note: original cut datum is determined taking into consideration the desired depths of secondary channel and tapping.*
Related Activities	List of activities where the rule is used
Related Entities	E_PrimaryChannel001
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001
Linked Constraints	C_Wing_Span_Length_Limit_X001
Linked Rules	R_Secondary_Channel001, R_Tapping001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.20 R-form for primary channels

R-Form	
Name	Rule2_Secondary_Channel
Reference	R_Secondary_Channel001
Objective	Rule to determine the depth of the cutting for the secondary channel.
Context, info, validity	Apply whenever secondary channels are used.
Description	The rule is based on a pro rata equation that the depth of the cutting for the secondary channels is reduced as they move towards the tip of the wing, because the thickness of the wing becomes thinner from root to tip. Equation: cutting depth = original cut datum[i] – ((new location along the length span of the wing – old location along the length span of the wing)/100) [i]*Note: original cut datum is determined taking into consideration the desired depths of secondary channel and tapping.*
Related Activities	List of activities where the rule is used
Related Entities	E_SecondaryChannel001
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001
Linked Constraints	C_Wing_Span_Length_Limit_X001
Linked Rules	R_Primary_Channel001, R_Tapping001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.21 R-form for secondary channels

R-Form	
Name	Rule3_Tapping
Reference	R_Tapping001
Objective	Rule to determine the depth of the hole for the tapping.
Context, info, validity	Apply whenever tappings are used.
Description	The rule is in two parts. First, the group of tappings is aligned along the length span of the wing based on the x-coordinate of one of the tappings. Second, there is a pro rata equation that calculates the depth of the tappings. This is because the thickness of the wing becomes thinner from root to tip, therefore, the depth of the tappings must be reduced when they are located closer to the tip of the wing. Equation: x-coordinate of Tapping No 2 = x-coordinate Tapping of No 1 x-coordinate of Tapping No 3 = x-coordinate Tapping of No 1 x-coordinate of Tapping No 4 = x-coordinate Tapping of No 1 cutting depth = original cut datum[i] – ((new location along the length span of the wing – old location along the length span of the wing)/100) [i]*Note: original cut datum is determined taking into consideration the desired depths of secondary channel and tapping.*
Related Activities	List of activities where the rule is used
Related Entities	E_Tapping001
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001
Linked Constraints	C_Wing_Span_Length_Limit_X001
Linked Rules	R_Primary_Channel001, R_Secondary_Channel001
Information Origin	Diem Lam, Cranfield University, 17 August 2004
Management	Author: DHL Date: 17 August 2004 Version No: 001 Status: Complete

Table 4.22 R-form for tappings

4.3.5. E-Forms

E-forms serve to describe all the objects, i.e. parts, components, assemblies, sub-assemblies and/or products, which are required in the design process, along with their structure, behaviour (transitional state) and functional aspects. Entities have constraints applied to them (Constraint→Entity) which define their boundaries and limitations.

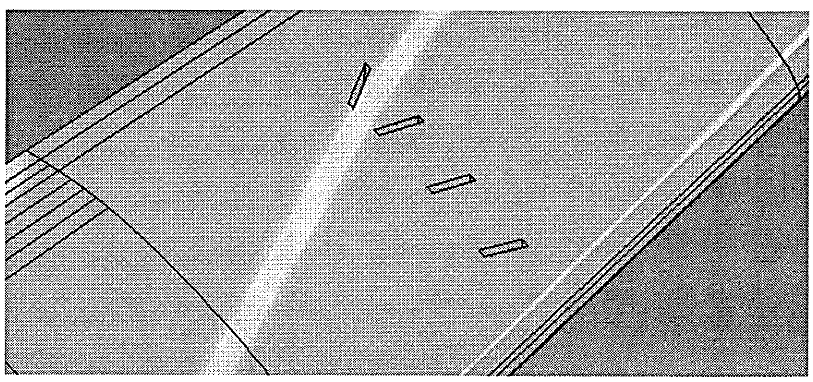
E-Form	
Name	Primary_Channel
Reference	E_PrimaryChannel001
Entity type	Structure
Function	Store cables and wires of tubings of pressure measurement valves running between the tappings and secondary channel on the aircraft wing model.
Behaviour	Must be sufficiently deep to house cables and wires of tubings.
Context, info, validity	Cannot be located on parts of the wing where there is insufficient material or the thickness is too thin.
Description	On the stainless steel aircraft wing model, the primary channels provide areas where the cables and wires may be stored and run between the tappings and secondary channel. 
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001
Related Constraints	C_Wing_Span_Length_Limit_X001, C_Wing_Span_Width_Limit_Y001, C_Primary_Channel_Thickness_Available_Max001, C_Primary_Channel_Thickness_Available_Min001, C_Primary_Channel_Length001, C_Primary_Channel_Width001
Related Activities	A_Cut_Primary_Channel001
Related Rules	R_Primary_Channel001
Information Origin	Diem Lam, Cranfield University, 1 August 2004
Management	Author: DHL Date: 16 August 2004 Version No: 001 Status: Complete

Table 4.23 E-form for primary channel

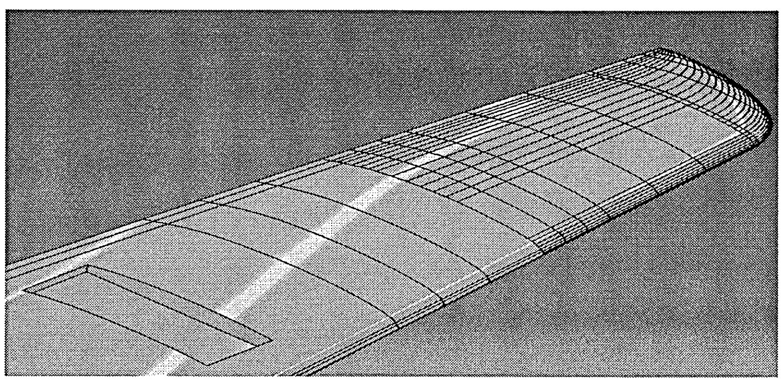
E-Form	
Name	Secondary_Channel
Reference	E_SecondaryChannel001
Entity type	Structure
Function	Large cutting forming the main storage area of cables and wires at specified location on the aircraft wing model.
Behaviour	Must be sufficiently deep to house cables and wires.
Context, info, validity	Cannot be located on parts of the wing where there is insufficient material or the thickness is too thin.
Description	On the stainless steel aircraft wing model, the secondary channel is the largest cutting providing an area where wires and cables may be stored and hidden away. 
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001
Related Constraints	C_Wing_Span_Length_Limit_X001, C_Wing_Span_Width_Limit_Y001, C_Secondary_Channel_Thickness_Available_Max001, C_Secondary_Channel_Thickness_Available_Min001, C_Secondary_Channel_Length001, C_Secondary_Channel_Width001
Related Activities	A_Cut_Secondary_Channel001
Related Rules	R_Secondary_Channel001
Information Origin	Diem Lam, Cranfield University, 1 August 2004
Management	Author: DHL Date: 16 August 2004 Version No: 001 Status: Complete

Table 4.24 E-form for secondary channel

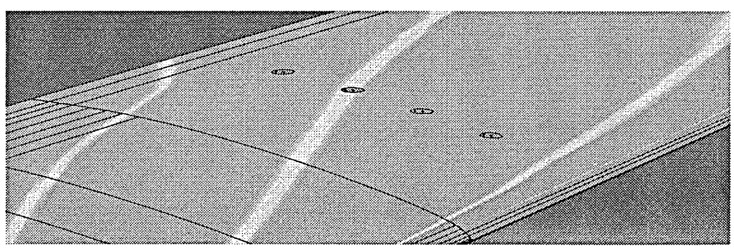
E-Form	
Name	Tapping
Reference	E_Tapping001
Entity type	Structure
Function	Hole marking the specified location on the wing model where pressure plots are obtained via the pressure measurement valve.
Behaviour	The diameter of the hole must be sufficiently large to hold the ferrule containing the pressure measurement valve.
Context, info, validity	Hole for tapping with ferrule cannot be drilled on parts of the wing when there is insufficient material or the thickness is too thin. In this case, a small hole is drilled to mark the proposed location of the pressure measurement valve.
Description	On the stainless steel aircraft wing model, the tappings provide the locations where pressure measurement valves should be located, with or without the ferrule, to obtain the desired pressure plots. 
Related Illustrations	I_Tappings_Channels001, I_Wing_Span001
Related Constraints	C_Wing_Span_Length_Limit_X001, C_Wing_Span_Width_Limit_Y001, C_Tapping_Diameter_With_Ferrule001, C_Tapping_Diameter_Without_Ferrule001, C_Tapping_Thickness_Available_Max001, C_Tapping_Thickness_Available_Min001
Related Activities	A_Drill_Tapping001
Related Rules	R_Tapping001
Information Origin	Diem Lam, Cranfield University, 1 August 2004
Management	Author: DHL Date: 16 August 2004 Version No: 001 Status: Complete

Table 4.25 E-form for tapping

CHAPTER 5: RESULTS & DISCUSSION

For the case study, the engineering rules, relationships, constraints, parameters, formulas, etc. defined on the design automation for the CATIA (Knowledgeware) KBE application are shown graphically in Figures 4.3-4.18. The automation model has been developed to model and design the sizes and locations of the tappings, primary and secondary channels based on the engineering rules and relationships defined by engineers from Airbus and ARA.

The corresponding informal knowledge model for the design automation model produced in the study is represented by the ICARE forms shown in Tables 4.1-4.25. The informal knowledge model (ICARE forms) has been developed specifically to translate and represent all the engineering rules, constraints, parameters, formulas, etc. used on the design automation model in the CATIA (Knowledgeware) KBE software.

The ICARE forms presented in Tables 4.1-4.25 aim to enlighten the esoteric rules and relationships that have been coded on the design automation model shown in Figures 4.3-4.18, so that the inner working and design of the generative function of the CATIA (Knowledgeware) KBE application may be seen and understood by viewing the human friendly ICARE forms.

Figures 5.1A-5.1D show a sequence of automated modelling and design tasks from the generative function in the CATIA (Knowledgeware) KBE application to design and generate new secondary channels based on coordinate inputs for their locations from the Excel design table. This particular sequence of events is interesting because in the background the design automation model relies on a couple of macros (Secondary_Channel1.CATScript and Secondary_Channel2.CATScript) coded in VBScript using the CAA IDL API and a rule (Rule 2 shown in Figure 4.11) to regulate the cutting depth of secondary channels based on the location inputted via an Excel design table (Design Table 2 shown in Figure 4.14). Note that the macros

Secondary_Channel1.CATScript and Secondary_Channel2.CATScript are only run to generate an additional secondary channel when the user has clicked the "Yes" button, see Figure 5.1B.

Figures 5.2A and 5.2B show how the drilling depth of the tappings are first modified as defined in a rule (Rule 3 shown in Figure 4.12) before moving to the new specified locations based on a set of coordinates given from an Excel design table (Design Table 3 shown in Figure 4.15). This automated example of modifying and moving the tappings to another location is befitting in practice because this type of design changes is common and highly representative of the problems faced by industry, which can consume considerable amount of time, effort and money.

The example described in the above paragraph and shown in Figures 5.2A and 5.2B is highly auspicious and pertinent during the course of the case study, as shown by the comparison below:

- It took under three seconds to relocate four tappings for the example shown in Figure 5.2B from the case study.
- It took one of the collaboration partners one man week to relocate 300 tappings.

From the above comparison, it is reasonable to say that such a task of modifying and moving 300 tappings may be accomplished within seconds or minutes by a KBE tool. (The study recognises that time and effort is required to design and code the initial design automation model). As far as industry is concerned, the above comparison shows that KBE tools can be used to gain substantial savings in terms of time, speed and money.

Figures 5.3A and 5.3B briefly demonstrates how the primary channels are shifted to another location based on the chosen coordinate along the span of the wing from an Excel Design Table 1 shown in Figure 4.13 with Rule 1 illustrated in Figure 4.10.

KBE softwares such as ICAD have previously been criticised as user unfriendly, arcane and occult and nothing can be seen and understood by the common engineers and team members who work along side the KBE engineers/programmers. This is because KBE softwares in the past are usually: (1) not graphical based; (2) do not have user friendly GUIs (graphic user interface); (3) emphasis is on programming and writing highly intricate computer codes and; (4) have been designed specifically to serve highly technical individuals such as KBE engineers and computer programmers, while ignoring the ordinary CAD/CAM/CAE operators, designers and engineers.

From the present study, it seems that software developers from Dassault Systemes, software vendor of CATIA, have taken note of the criticism aimed at previous KBE softwares, see list of criticism in above paragraph, and have endeavoured to address the drawbacks of using KBE softwares by ensuring that the design of the automation model is graphical based via user friendly GUIs. This means the graphically defined automation model with its rules and relationships may be designed, coded, seen and understood swiftly, without the need for writing abstruse and obtuse computer codes/programs or worse still forcing people to stare at lines of incomprehensible computer codes/programs. Therefore, it is justified to reason that the CATIA (Knowledgeware) software has partly removed the mysterious, unknown, hidden and "lost of control" elements of using KBE applications and made this kind of technology more open, comprehensible and accessible to practicing engineers, albeit the cost of software is another issue!

Experience from the case study shows that the ICARE forms may be used as a form of documentation for the design automation model, assisting and forcing KBE engineers and programmers to think and code logically, which prevents them from writing wantonly and unstructured codes without any form of documentation, explanation or accountability. This is important because it means engineering rules and relationships defined for the design automation model are designed and coded in a structured and logical manner, which makes it easier to translate and transfer these rules and relationships onto the ICARE forms.

The structured and logical manner of the design automation model is seen in Figures 4.3-4.4, 4.8-4.9 and 4.13-4.16, which show there is a definite naming system for the relationships, rules, checks, parameters, formulas and design tables of the tappings, primary and secondary channels. This structured and logical manner from the design automation model makes it easier for the author to develop the corresponding knowledge model (ICARE forms). In the case study, the author was forced to think and present the design automation in a logical manner, so that this structured and orderly manner can also be seen on the ICARE forms, e.g. Figures 4.3-4.4 show the parameters defined on the design automation model which may be seen and understood clearly in the corresponding C-forms from Tables 4.3-4.16. Therefore, anyone viewing the design automation model from the CATIA (Knowledgeware) software may intuitively understand the defined rules, parameters, formulas, etc. by reading the human friendly ICARE forms.

For the design automation model in the case study, the main decision maker that determines and controls the logics and predicates for the design of the tappings, primary and secondary channels is contained in the three rules Rule1_Primary_Channel, Rule2_Secondary_Channel and Rule3_Tappings, see the top left corner of Figure 4.9 and in details from Figures 4.10-4.12. In all three rules, before any design changes are considered, checks are performed to confirm that any modification to the tappings, primary and secondary channels must actually happen on the aircraft wing model itself, i.e. any new coordinates given must be within the x and y limits of the wing model itself. All three rules take no action when the checks confirm that all the coordinates inputted matched the original set.

The if-else statements in rules Rule1_Primary_Channel, Rule2_Secondary_Channel and Rule3_Tappings perform the appropriate tasks according to the conditions of the model at a particular moment in time and the responses from the users. For example, in rule Rule2_Secondary_Channel from Figure 4.11, the cutting depth of the secondary channel remains unaltered if the channel has not been moved but is reduced if the channel is moved closer towards the tip of the wing. Appropriate message, dialogue and decision boxes are generated either to inform users of any necessary information or requesting them to make a decision, see examples in Figures 5.1B and 5.1C. Also, note that in rule Rule2_Secondary_Channel, the macros are only launched to create a new secondary channel when the response from the decision box is “Yes”.

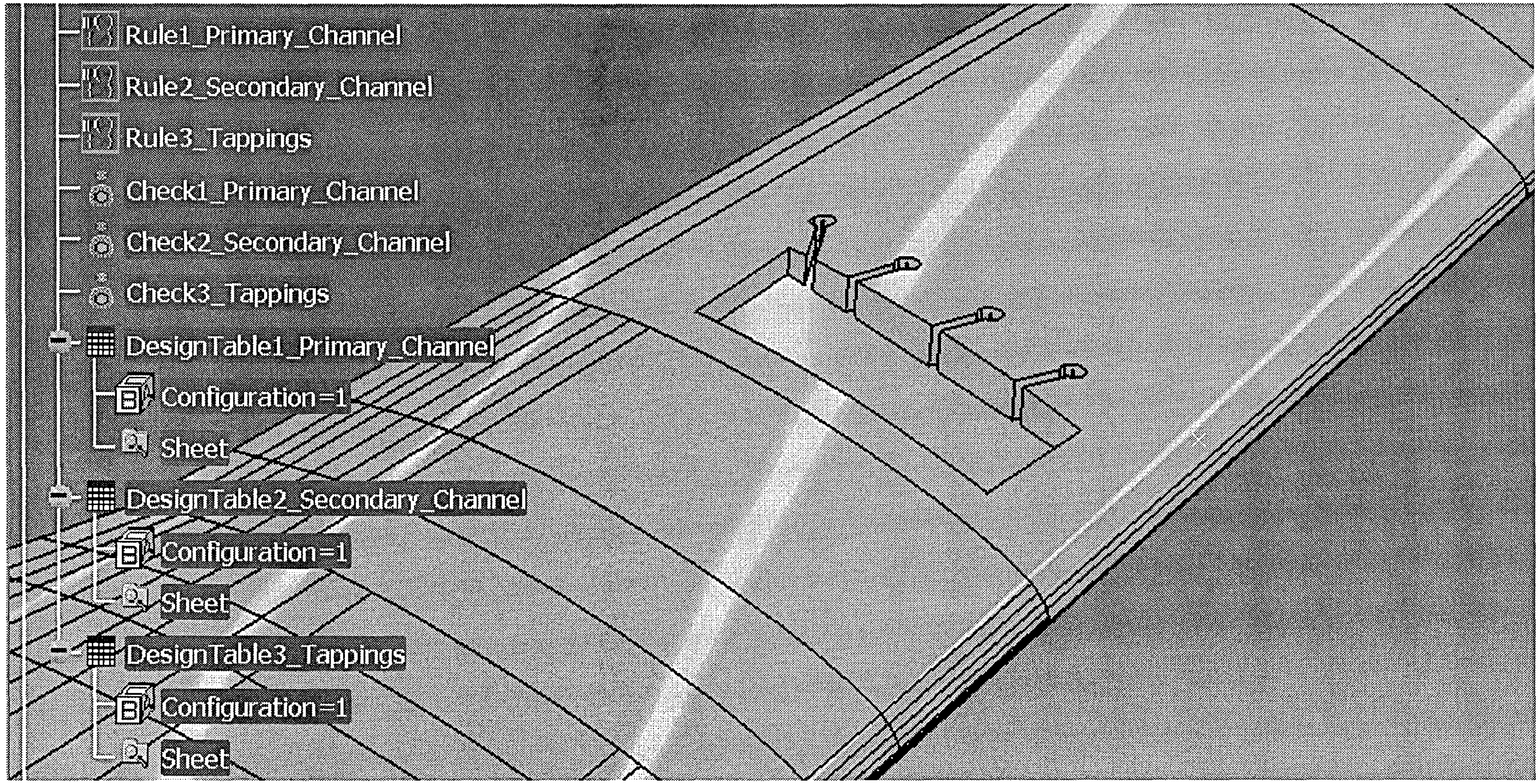


Figure 5.1A Initial model for the tappings, primary and secondary channels of the aircraft wing model

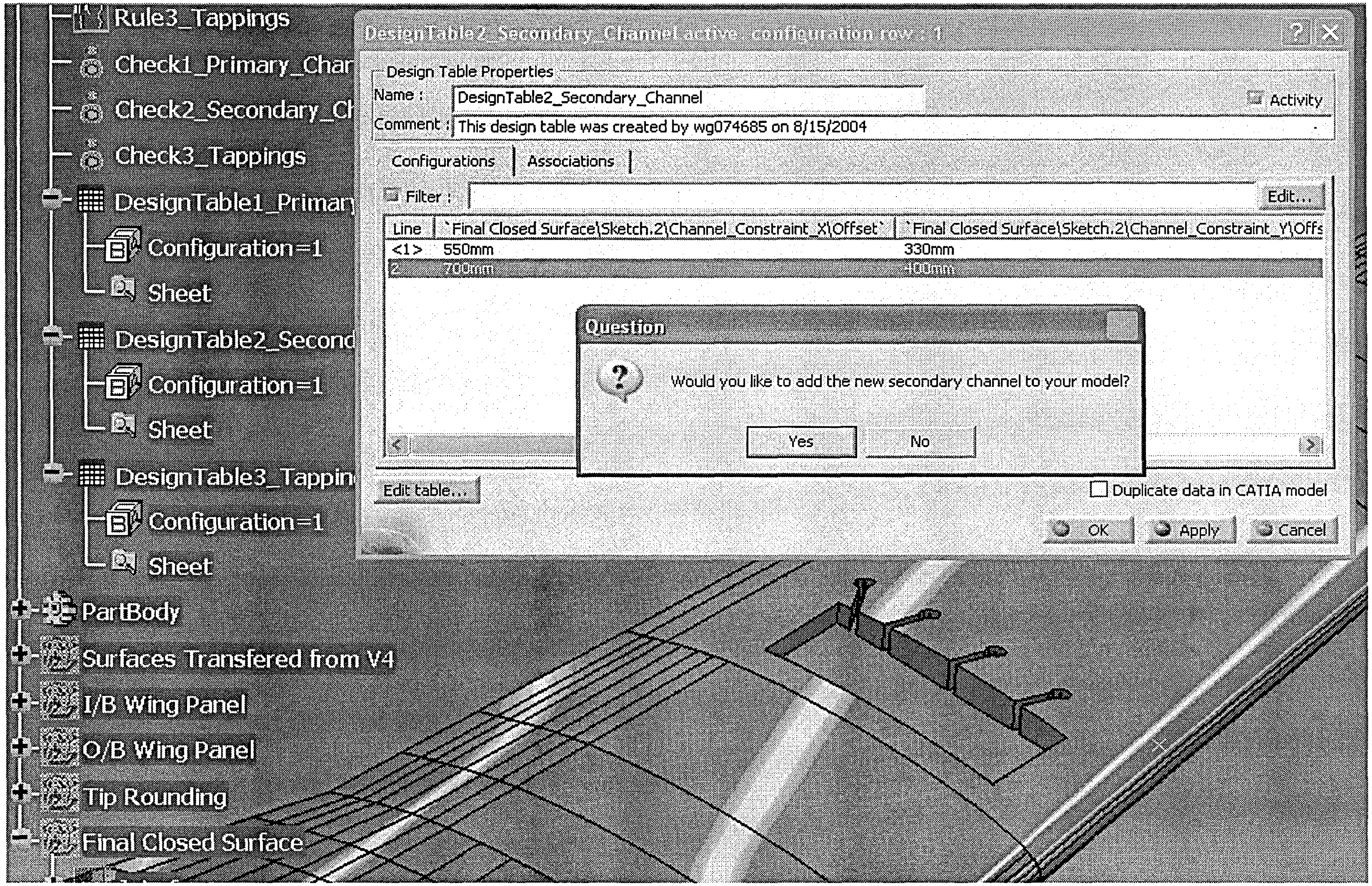


Figure 5.1B Dialogue box generated from the rule governing the design of secondary channels confirming whether the new secondary channel should be added to the model based on new coordinate inputs from an Excel design table

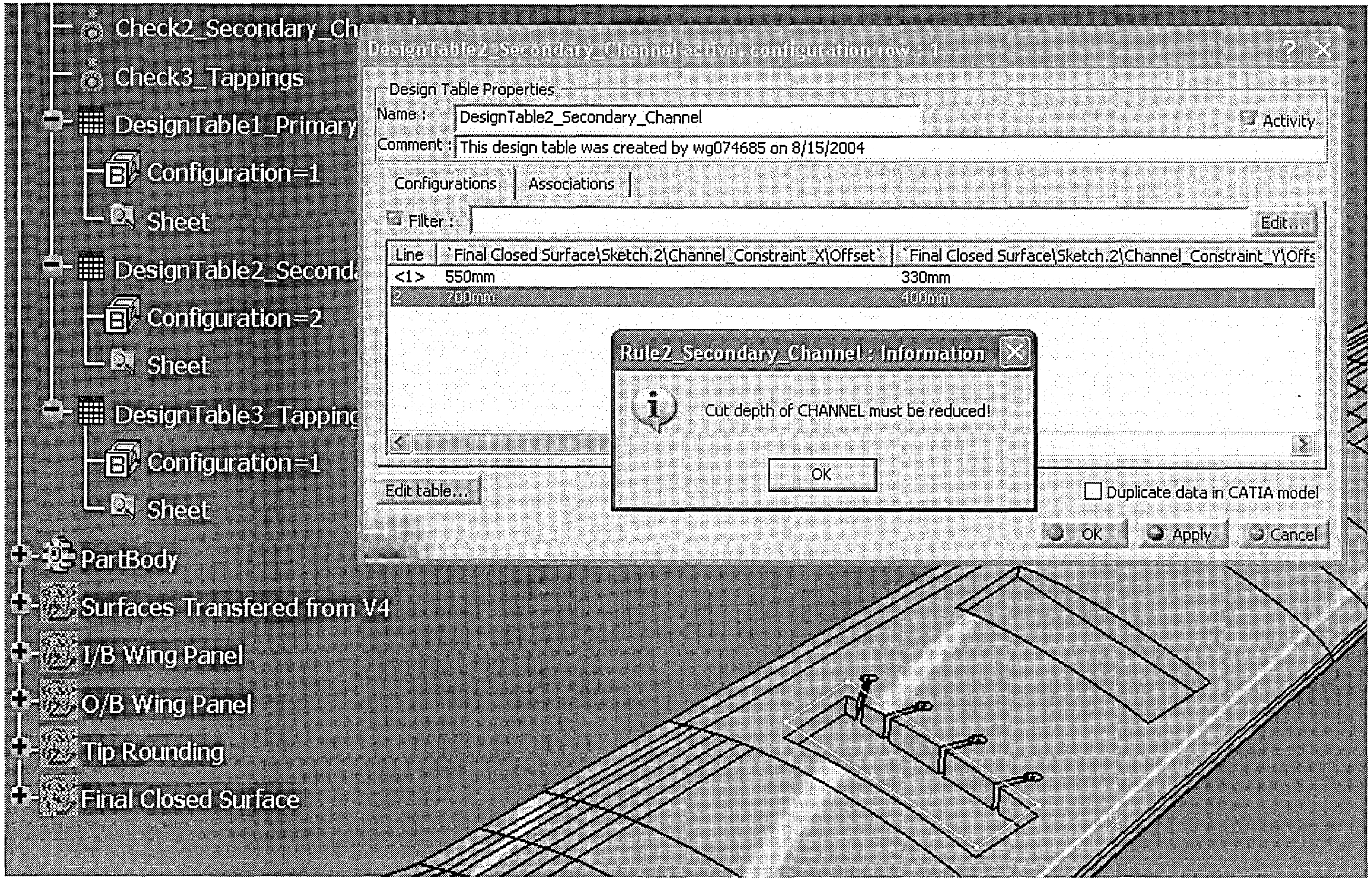


Figure 5.1C Dialogue box generated from the rule governing the design of secondary channels notifying users that the cutting depth must be reduced for the new location of the secondary channel

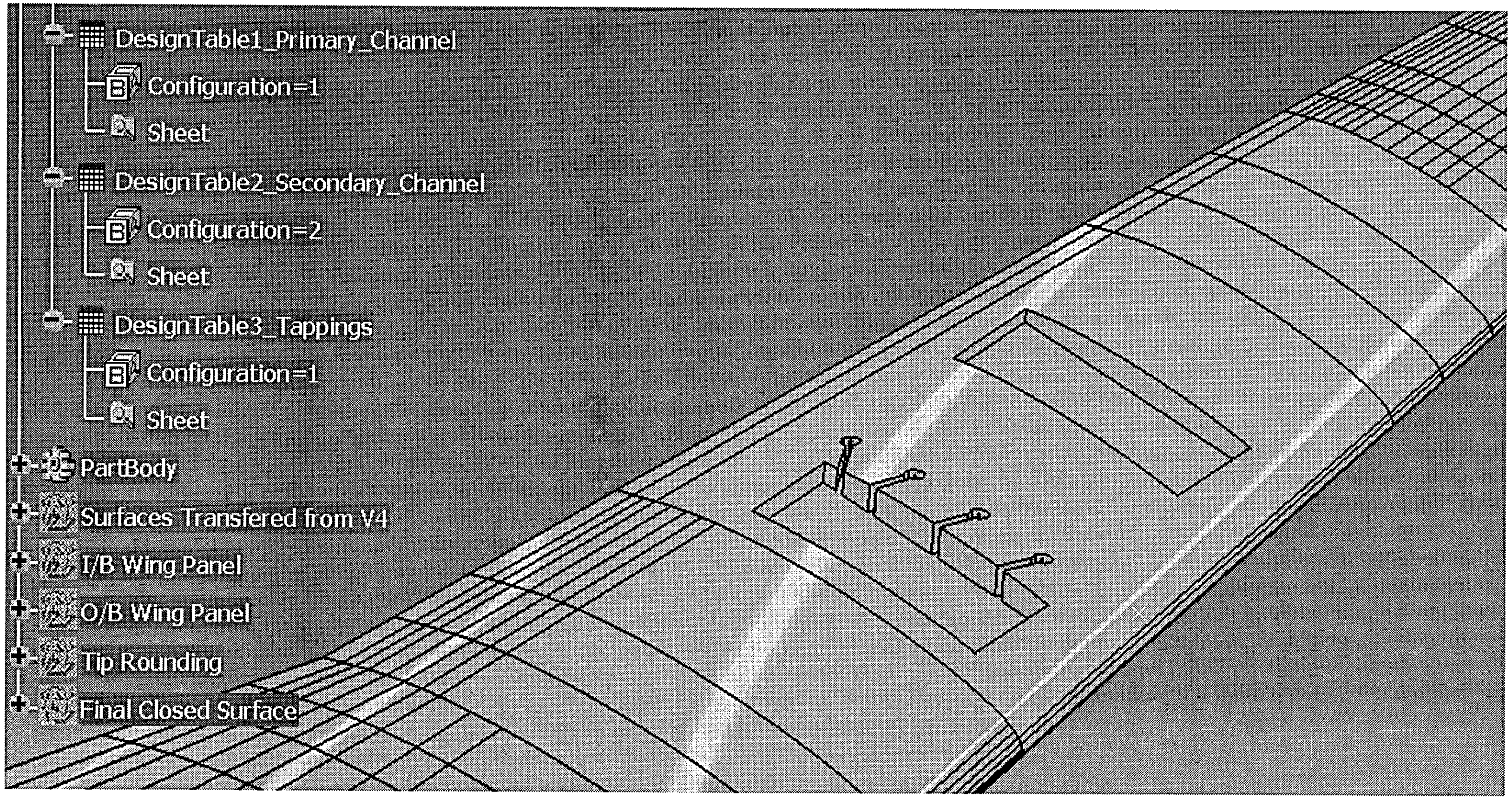


Figure 5.1D New design for the aircraft wing model after the new secondary channel has been added to the model

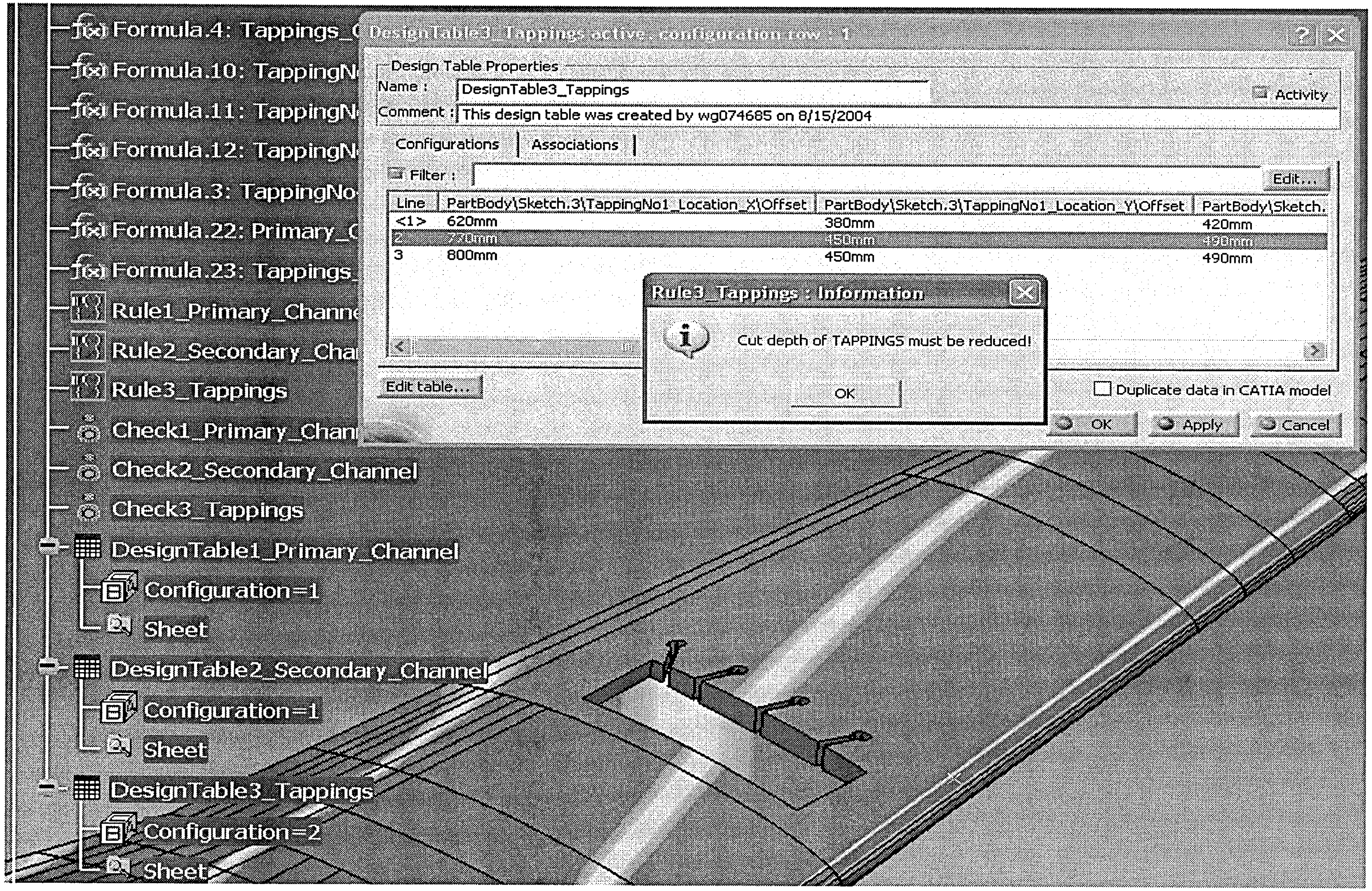


Figure 5.2A Engineering rule governing the tappings is automatically activated, informing users that the drilling depth must be reduced based on the new set of coordinates given for the tappings from an Excel design table

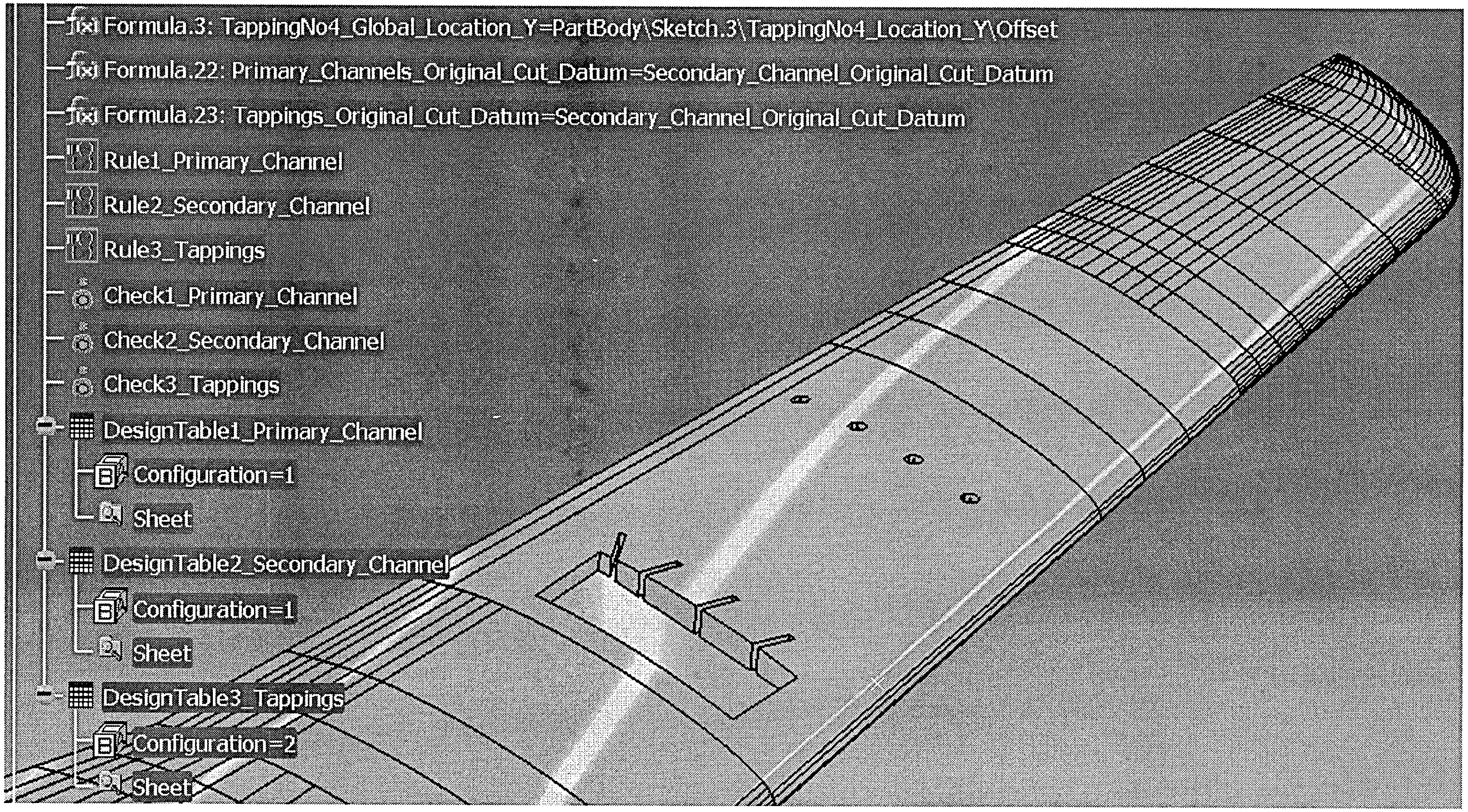


Figure 5.2B New design for the aircraft wing model after the tappings have been moved to the new set of coordinates

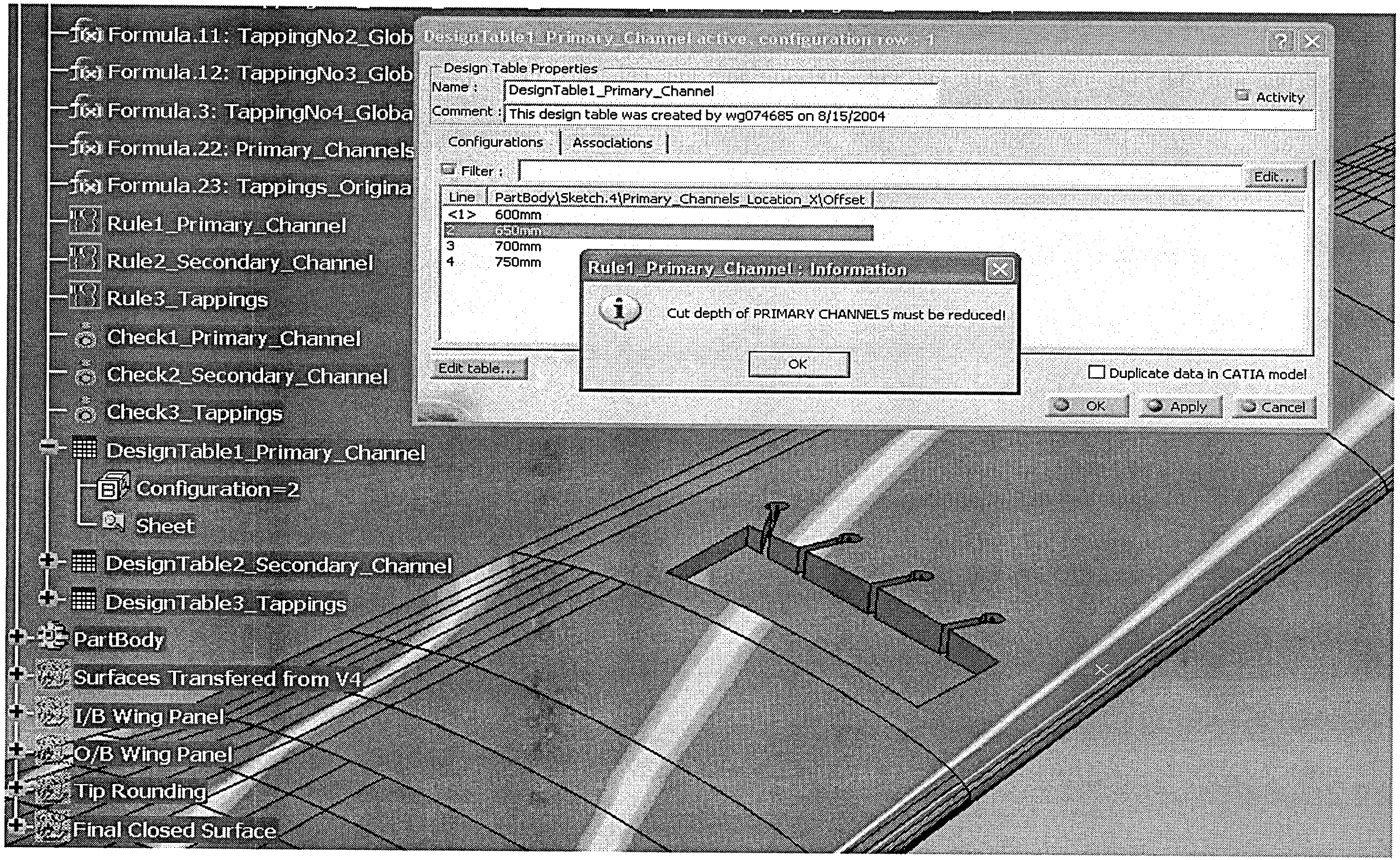


Figure 5.3A Engineering rule governing the design of primary channels is automatically activated, informing users that the cutting depth must be reduced based on the new coordinate chosen from an Excel design table

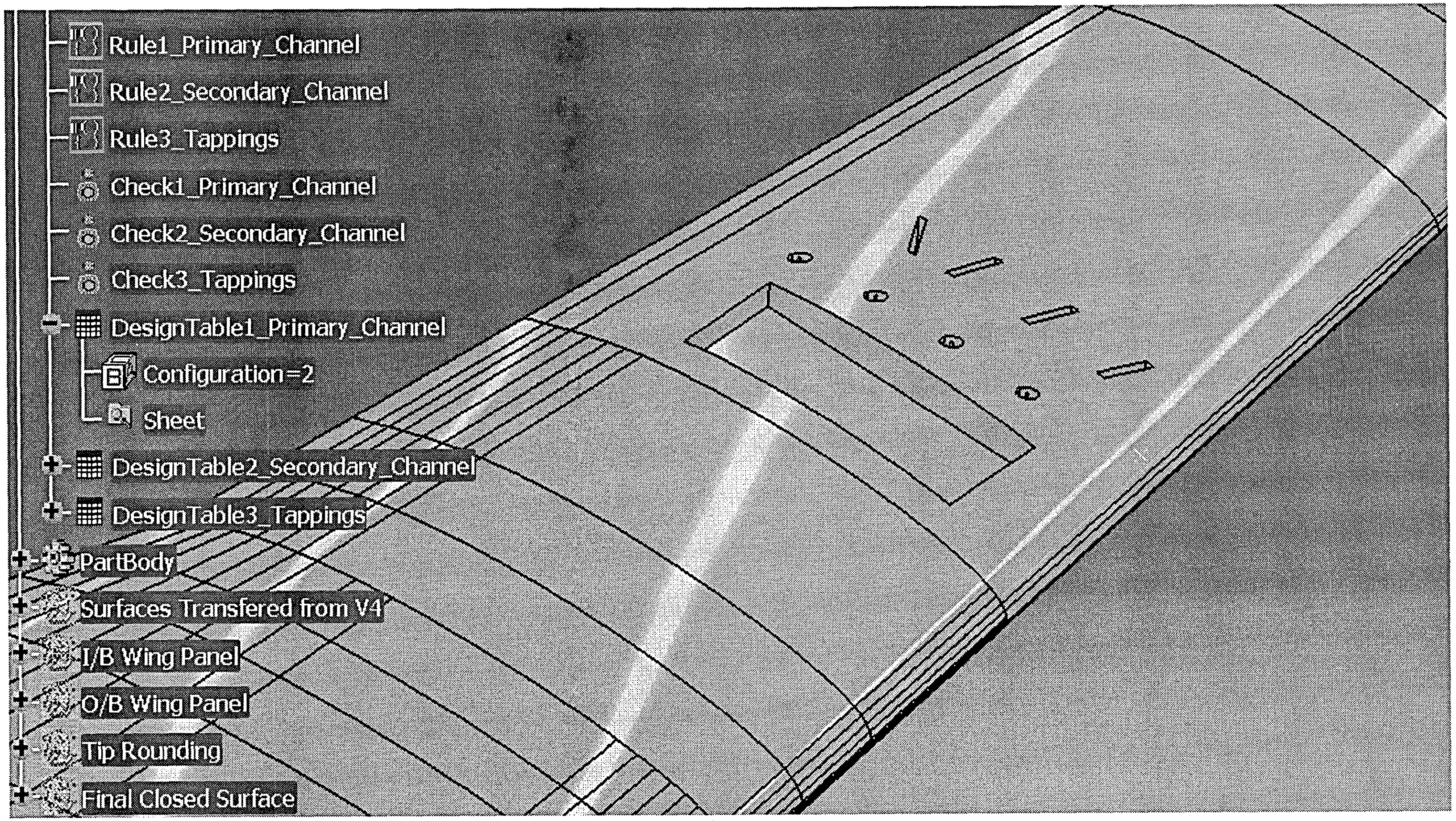


Figure 5.3B New design for the aircraft wing model after the primary channels have been moved to the locations

As the research programme progresses in the study, it becomes evidently clear that the informal knowledge model (ICARE forms) can be used to serve two very different functions depending on how it is used and the intended purpose of using it:

- ICARE forms can be used to capture knowledge for building KBE applications.
- ICARE forms can be used to translate and illustrate engineering rules and relationships that have been coded on the design automation model, so that people may understand the inner working and design of the generative function of KBE applications.

Work from the present study covers both of the functions described above, although it may be argued that the focus of the case study is more aligned with the latter function. Equally important and in addition to the case study, site visits to Airbus show that in practice the former function is more critical in large enterprises for reasons such as large numbers of transient workforces involved, which means there is the need to capture knowledge and experience of the employees before they depart or move to another project.

For the case study, it is important to state that the design automation model was developed first and the informal knowledge model (ICARE forms) was developed subsequently to illustrate the engineering rules and relationships that have been coded on the automation model. In reverse to this modelling sequence, engineers at Airbus first develop the ICARE forms to capture knowledge and KBE applications are subsequently developed based on the ICARE forms developed.

In one way, the ICARE forms from the case study have undoubtedly proven that they can be used to provide a degree of transparency between the automation and knowledge models. For example, the rule that determines the locations of the tappings on the design automation model in the CATIA (Knowledgeware) software shown in Figure 4.12 can be visualised on the corresponding R-form of the ICARE forms in Table 4.22.

However, the real question is how well, easy and complete do the ICARE forms manage to translate and portrait the engineering rules and relationships that have been coded on the design automation model in the KBE application. This question is pertinently illustrated in Knowledgeware by the example involving the three parameters Primary_Channels_Original_Cut_Datum, Secondary_Channel_Original_Cut_Datum and Tappings_Original_Cut_Datum shown in Figures 4.3-4.4 and Formulas 22-23 in Figure 4.8, where there is an implicit rule/relationship represented by a formula linking the three parameters for the cutting datum, i.e. the formula is set so that the initial cutting datum for the tappings and primary channels equals that of the secondary channel's. But, this rule/relationship is not reflected in any ICARE forms of Tables 4.1-4.25, for which there are several explanations and arguments that may be accounted for this omission. For example, it is debatable whether the author should have created an appropriate ICARE form, e.g. another R-form or C-form, to represent this rule/relationship between the mentioned parameters. The counter argument is that the rule/relationship between the three parameters is in fact an implicit one and strictly speaking it is not a rule as far as Knowledgeware is concerned.

Following on from the arguments raised in the above paragraph, it is equally valid to say that perhaps the author should have re-written the relationship between the parameters Primary_Channels_Original_Cut_Datum, Secondary_Channel_Original_ Cut_Datum and Tappings_Original_Cut_Datum defined by Formulas 22 and 23 in Figure 4.8, so that it appears explicitly as a rule in Knowledgeware. By doing this, total transparency and transferable of the rule between the automation and knowledge models is achieved by having a corresponding R-form of the ICARE forms to represent the explicit rule defined in Knowledgeware. The counter-argument for this overture is that strictly speaking this relationship between the three mentioned parameters is not a rule as discussed in the above paragraph and in actuality it is more natural in Knowledgeware to define this relationship as it is, i.e. using formulas to define a relationship in lieu of a rule. Another argument against re-writing and modifying the design automation model is that additional time and effort are required, resulting in inefficiency of work with added costs because of unnecessary changes to the automation model.

The previous two paragraphs provide a mooting point and dilemma how a simple relationship/rule/formula may be modelled and coded in the design automation and knowledge models. Imagine what it would be like if this relationship/rule/formula had been a highly complex one. This then adds validity and poignancy to the original question of how well, easy and complete do the ICARE forms manage to translate and represent the engineering rules, constraints, parameters, formulas, relationships, etc. that have been coded on the design automation model of KBE applications.

Most KBE softwares, e.g. ICAD, Knowledgeware, etc., have their particular characteristics, peculiarities and quirkiness with their own distinctive design structure for defining, coding and linking rules and relationships on the design automation model. It is highly probable that the design structure of the ICARE form proposed by the MOKA consortium will differ from that of the KBE softwares, resulting in loss of translation and incompatibility of representation of the engineering rules and relationships that have been coded on the design automation model of KBE applications. In other words, there is the issue of compatibility between the methodologies and design structures of the ICARE forms and KBE softwares. This is precisely what the present study found between the MOKA's ICARE forms and Knowledgeware.

Another example of difference in language, design structure, methodology and compatibility is shown by the constraints shown in Figures 4.5-4.7. Constraints set in Knowledgeware are in fact additional user defined parameters set to the geometries of the model, which allows geometries to be manipulated and modified in a very specific manner as chosen by the KBE modeller and the manipulations/modifications are carried out by simply changing the value(s) of the constraints. Whereas, the word constraint has a wholly different meaning in the MOKA's world of ICARE forms, because constraints are used to place limitations on entities and have nothing to do with setting additional parameters for manipulation and modification purposes. In fact, the constraints shown in Figures 4.5-4.7 cannot be translated, transferred or represented on any of the ICARE forms produced.

Work from the case study shows that there is a compromise between: (1) achieving complete transparency between the automation and knowledge models and (2) carrying out repetitive tasks of filling out semi-duplicating or very similar ICARE forms. The former requires significant input of time and effort and the latter wastes time and effort by filling out very similar ICARE forms that may be written and presented in another manner. By trying to attain completeness in the translation between the automation and knowledge models, this can introduce confusion, misunderstanding and much tedium when reading the ICARE forms because overly large amount of ICARE forms have been produced unnecessarily in the translation between the automation and knowledge models. An example of this can be seen in the three rules shown in Tables 4.20-4.22, where the rules are very similar and could have been presented in one convenient form instead of three.

In practice, enterprises who want to adopt KBE technology and MOKA's methodology and guidelines in earnest must consider establishing a system, specification or code of practice for how engineering rules and relationships should be designed, written and coded for KBE applications systematically right across the enterprise. This is so that high levels of visibility, transparency, traceability and compatibility between the automation and knowledge models may be attained. With a well defined system in place, this should also prevent KBE engineers and programmers from writing and coding the design automation models in their own peculiar styles, which can introduce difficulty in understanding the automation model at best and misunderstanding and utter confusion at worse.

As stated previously, Airbus are considering to use the ICARE forms for capturing and storing knowledge and KBE applications are developed subsequently based on the captured knowledge on the initial ICARE forms. However, this consideration raises two issues that need to be addressed. The first issue is consistency between the ICARE forms stored on the database and KBE applications which have been developed based on those ICARE forms. There must be a mechanism for ensuring consistency between the ICARE forms and KBE applications when changes are made to the ICARE forms. The second issue that needs to be addressed in practice is the maintenance, consistency and validity checks of the overtly large amount of convoluted ICARE forms required for mapping, linking and correlating all the illustrations, constraints, activities, rules and entities, particularly when changes are involved. This issue may be overcome by using a sophisticated database or product data management (PDM) system, so that consistency, validity, cost and other checks are automatically performed to resolve and highlight any effects caused by modifying an illustration, constraint, activity, rule or entity of the ICARE forms. In terms of size, complexity and project management, the engineering design chosen for the case study is a very simple standalone engineering problem and yet it takes 25 ICARE forms to present this simplified and minor problem. Imagine how many ICARE forms may be required in practice when confronted with a realistic engineering problem that is large in size, high in complexity and involving a vast number of people and teams.

It is worth noting that the author is not the ideal person to interpret and comment on how well, easy and successful the ICARE forms can be used to illustrate the engineering rules and relationships that have been coded on the design automation model of the KBE application because (1) this interpretation is based solely on the views of one individual, i.e. the author's in this study and (2) the author has also been responsible for coding and developing both automation and knowledge models, which means he is bound to see and comprehend the rules, relationships and logics illustrated on the ICARE forms more readily than anyone else. Therefore, future studies should seek to quantify how well, easy and complete the ICARE forms can be used to illustrate the engineering rules and relationships defined for the design automation model of KBE applications by developing a questionnaire to ask a group of users for their views other than the KBE engineers/programmers who developed and coded the automation and knowledge models.

The ICARE forms produced for the case study have not been stored on the database as a consequence of time constraint and software configuration problems encountered with the knowledge capturing and mapping SophX-Pack software. This task will now be performed by another researcher at a later date.

Parts of the work presented in this study have been demonstrated to the industrial collaboration partners and Cranfield University is developing additional industrial programmes to support further studies in the use of KBE technology specifically in the aerospace industry. Also, parts of the work documented in the present study have been included in a consortium confidential DTI's research project grant report (DTI's ref. no. CHAD/002/00008) [1] on the use of KBE systems.

CHAPTER 6: CONCLUSIONS

6.1. Conclusion

Presently, concerns have been expressed from academia and industry that there is a lack of visibility, transparency, traceability and accountability when KBE applications are used to perform generative engineering designs. The concerns are centred on the fact that engineering rules, relationships, constraints, parameters, etc. defined on the design automation model of KBE applications are not easily seen and understood by most people, other than the KBE engineers and programmers who developed and coded the automation models. Consequently, there are tentative elements of mysterious, hidden, unknown and "lost of control" fear when KBE technology is deployed.

The present study seeks to remove the mysterious, hidden, unknown and "lost of control" elements of using KBE applications and overcome the perceived lack of visibility, transparency, traceability and accountability by developing and integrating knowledge models with the design automation models of KBE applications. By doing this, it means the inner working and design of the generative design function defined for the KBE application via its rules and relationships may be seen and understood readily by viewing the knowledge model.

The knowledge model developed for the present study is based on the European MOKA's methodology and it is in fact known as an informal knowledge model represented by the ICARE forms. The informal knowledge model (ICARE forms) residing on the database provides the ontology and serves the purpose of defining how products/assemblies/parts should be designed, processed and manufactured within a prescribed set of illustrations, constraints, activities, rules and entities.

The CATIA (Knowledgeware) KBE application has been used in the present study and the design automation model residing on the KBE application has been developed using the

Knowledgeware programming language, VBScript, macros and CAA IDL API. The automation model coded for the present study permits a series of modelling and design tasks to be automated within the CATIA software.

Work from the present study is validated by a case study involving an aircraft wing model representative of the Airbus A380, which is used by stress and aerodynamic engineers at a wind tunnel testing facility to calculate pressure plots of the wing structure. Based on the engineering rules defined by practicing engineers from industry, a design automation model from the CATIA (Knowledgeware) KBE application has been developed specifically for the case study, which enables a series of automated and integrated generative modelling and design tasks to be performed to obtain the correct sizes and locations of the tappings, primary and secondary channels. The corresponding informal knowledge model (ICARE forms) has been developed to translate and reflect the engineering rules, constraints, parameters, formulas, etc. that have been defined on the design automation model of the KBE application, so that the inner working and design of the KBE application may be readily seen and understood by other personnel besides the KBE engineer/programmer himself.

Results from the present study have highlighted several significant findings based on the author's experience about the MOKA's methodology for building informal knowledge models (ICARE forms) and the benefits offered by KBE technology. The case study shows that KBE tools may be used to gain substantial commercial benefits in terms of time, cost and speed by integrating and automating the modelling, design, analysis and manufacturing processes to obtain the required pressure plots on the aircraft wing model. Comparison from the study shows that it takes under three seconds to relocate four tappings but it takes one of the industrial collaboration partners one man week to relocate 300 tappings.

Based on the author's experience, it is evidently clear from the study that the MOKA's methodology and ICARE forms may be used to capture knowledge for KBE applications. The author believes that the ICARE forms can be used to provide a satisfactory means for depicting the inner working and design of the generative function of KBE applications, which means a degree of visibility, transparency, traceability and accountability can be achieved and any hidden, unknown and "lost of control" fear is allayed when KBE applications are used to perform generative engineering designs.

However, the study believes that there is a limit to how well, easy and complete the ICARE forms can be used to translate and represent all the engineering rules and relationships that have been coded on the design automation model of KBE applications.

The study proposes that a PDM system should be exploited for performing consistency and validation checks of the ICARE forms and between automation-knowledge models, particularly when the size of the knowledge model is large, highly complex rules and relationships are used and intermittent changes occur for the knowledge model.

Experience gained from the study suggests that it is unnecessary and undesirable to have complete transparency between the automation and knowledge models for the following reasons:

- Require excessive amount of time and effort in achieving this completeness which serves little purpose and has minor added value.
- Excessive quantity of ICARE forms generated which wastes time, effort and resources in designing and producing the ICARE forms, as well as introducing confusion, misunderstanding and much tedium when reading the ICARE forms.

The study believes that the ICARE forms cannot provide a fully comprehensive representation of the design and inner working of the KBE application with all the intrinsic and complex rules and relationships involved because of reasons such as:

- There are many ways how engineering rules, relationships, parameters, constraints, etc. may be coded for design automation models inside most KBE applications. Moreover, many KBE softwares have their particular characteristics, peculiarities and quirkiness with their own distinctive design structure for how the engineering rules and relationships should be written and linked together, which may differ vastly from the MOKA's ICARE design structure. Therefore, it is reasonable to say that differences in the design structure and incompatibility of methodologies between the MOKA guidelines and diverse KBE applications may result in the MOKA's ICARE forms not able to capture and represent adequately some engineering rules and relationships that have been coded on the design automation models inside some KBE applications.

- To achieve a high level of transparency between the automation and knowledge models with relative ease, both models must be designed and developed by the same individual, but this may not be feasible or practical in practice. Hence, difficulties such as misunderstanding, misrepresentation and confusion are likely to be encountered when the automation and knowledge models have been developed by different people.

Finally, parts of the work presented in this study have been demonstrated to the industrial collaboration partners and Cranfield University is developing additional industrial programmes to support further studies in the use of KBE technology specifically in the aerospace industry. Also, parts of the work documented in the present study have been included in a consortium confidential DTI's research project grant report (DTI's ref. no. CHAD/002/00008) [1] on the use of KBE systems.

6.2. Limitations of Present Study

As a consequence of time constraint and software configuration problems encountered in the course of the study, the informal knowledge model (ICARE forms) has not been transferred for storage on the database using the specialised knowledge capturing and mapping software called SophX-Pack. Also, there was insufficient learning time for the author to master the SophX-Pack software thoroughly. Without deploying the SophX-Pack software, the present study cannot comment on the level of sophistication and complexity that may be handled by the ICARE forms for capturing, linking and mapping knowledge for KBE uses.

There was a tight time constraint set for the case study and overall research programme. This constraint places a limitation on the study in the sense that the size of the case study and the engineering rules and relationships used have been kept deliberately small and simple to ensure the completion of the case study. Therefore, engineering rules and relationships defined for the design automation model and corresponding knowledge model developed in the case study lack the appropriate level of complexity, intricacy and size, which would have given the MOKA's methodology and ICARE forms a sterner test in translating and representing much more complex and intrinsic rules and relationships.

The present study is a public study which means all findings are made open to the public. However, the actual engineering rules, relationships, parameters, formulas, etc. used by the industrial collaboration partners for the design of the tappings, primary and secondary channels are deemed as commercially sensitive and confidential material. Therefore, engineering rules and relationships defined for the design automation model and the corresponding knowledge model built in the case study are not reflective of the actual ones used by practicing engineers in industry which are more involved and complicated.

6.3. Recommendations for Future Studies

It is recommended that the ICARE forms developed for the present study should be transferred for storage on the database using the SophX-Pack software.

The size and complexity of the rules and relationships used for future studies should be expanded, so that the MOKA's methodology and ICARE forms may be assessed and evaluated for cases when the knowledge models are large and involved many cross functional engineers and teams and the rules and relationships used are highly complex and intricate. A PDM system may be deployed to perform consistency and validation checks, particularly when the size of the knowledge model is large and highly complex rules and relationships are involved.

For the present study, the automation and knowledge models have been designed and built by the same engineer, which has produced two positive effects:

- Easier for the engineer to interpret and transfer the engineering rules and relationships that have been defined on the automation model to the knowledge model (ICARE forms).
- Higher level of visibility and transparency between the automation and knowledge models is achieved, which means it is easier for viewers/readers to see, understand and interpret the engineering rules and relationships that have been defined on the automation model when viewing from the knowledge model (ICARE forms).

Conversely to the positive effects described above, future studies should investigate and evaluate any adverse effects that may exist when the automation and knowledge models are designed and built by different KBE engineers, e.g.

- From the KBE engineers' point of view, how difficult it is to translate, transfer and present the rules, relationships, constraints, parameters, etc. from the automation model onto the informal knowledge model (ICARE forms).
- From the viewers'/readers' point of view, how accurate is the translation between the automation and knowledge models and how simple it is to see and understand the inner working of the KBE application from viewing/reading the knowledge model.

The modelling cycle for the present study is that the automation model was coded first and the knowledge model was developed subsequently to illustrate the engineering rules and relationships that have been coded on the automation model. Future studies should reverse this modelling cycle to see how easy or difficult it is to build KBE applications based on given knowledge models.

The design automation and informal knowledge models developed for the present study have not followed the modelling cycle proposed by the MOKA researchers, i.e.
Informal knowledge model? Formal knowledge model? Codes for automation model.
Future studies should follow this modelling cycle so that it is possible to assess its validity

and practicality for building KBE applications.

Finally, the author is not the ideal person to interpret and comment on how well, easy and successful the ICARE forms can be used to illustrate the engineering rules and relationships that have been coded on the design automation model of the KBE application because:

- Interpretation is based solely on the views of one individual, i.e. the author's in this study.
- The author has also been responsible for coding and developing both automation and knowledge models, which means he is bound to see and comprehend the rules, relationships and logics illustrated on the ICARE forms more readily than anyone else.

Therefore, future studies should seek to quantify how useful, easy and successful the ICARE forms can be used to illustrate the engineering rules and relationships defined for the design automation model of KBE applications by developing a questionnaire to ask a group of non-biased users, i.e. besides the KBE engineers/programmers who developed and coded the application, for their views.

REFERENCES

[1] Fan, I S and Bermel-Garcia, Pablo. 2004. *Wind Tunnel Model Design and Manufacture Research (DTI Ref No.CHAD/002/00008), KBE Work Package, Final Report*, Cranfield University, Bedfordshire.

[2] Cooper, S, Fan, I S and Li, G. 1999. *DTI's A Best Practice Guide: Achieving Competitive Advantage Through Knowledge-Based Engineering*. IFS, Letchworth, Bedfordshire.

[3] Toupin, L A. 2001. *Knowledgeware Makes Reusing Existing Designs a Snap*. Design News, Oct 15, 56 (20), pg 41.

[4] Dassault Systemes: Product & Solutions. http://www.catia.com (accessed 6th April 2004).

[5] Clarke, Charles. 2003. *Design Engineering: CATIA-ing for an Easier Life*. The Engineer, London, Jul 11, pg 47.

[6] Kochan, A. 1999. *Jaguar Uses Knowledge-Based Tools to Reduce Model Development Times*. Assembly Automation, Jun, Vol 19(2), pg 114.

[7] Callot, M, Kneebone, S and Oldham, K. 1998. *MOKA – A User Driven Project to Develop Methodology and Tools Oriented to Knowledge Based Engineering Applications*. Proceedings of European PDT Days 1998, QMS, Watford, pages 19-24.

[8] Stokes, M, (ed). 2001. *Managing Engineering Knowledge – MOKA Methodology for Knowledge-Based Engineering Applications*. Professional Engineering Publishing, London.

[9] ARA. *Home Page of Aircraft Research Association Ltd.* http://www.ara.co.uk (accessed 6th April 2004).

[10] Stilo. *Stilo: content engineering - content transformation - content integration - content delivery*. http://www.stilo.com/ (accessed 6th April 2004).

[11] KTI. *Knowledge Technologies International: KTI Home Page.* www.ktiworld.com (accessed 6th April 2004).

[12] Boothroyd, Dewhurst and Knight. 1994. *Product Design for Manufacturing and Assembly.* Marcel Dekker, NY.

[13] Cavanagh, John et al. 2002. *Alternatives to Economic Globalization.* Berrett-Koehler Pub, USA.

[14] Klein, Naomi. 2002. *No Logo: No Space, No Choice, No Jobs.* Picador, USA.

[15] Ghemawat, Pankaj. 2003. *The Forgotten Strategy.* Harvard Business Review, Nov, Vol 81 (11), pg 76.

[16] Grandl, Reinhard, 2001, *Virtual Process Week in the Experimental Vehicle Build at BMW AG*, Robotics and Computer-Integrated Manufacturing, Feb/Apr, Vol 17 (1-2), pg 65.

[17] Hollingum, Jack. 1995. *Jaguar Seeks Evolution in Robotics.* The Industrial Robot, Vol 22 (4), pg 3.

[18] Whitward, L. 1995. *Computer Gets a Grip on Fastener Selection.* Design Engineering, 26 (July/Aug), pg 31.

[19] Penoyer, J A, Burnett, G, et al. 2000. *Knowledge-Based Product Life Cycle Systems: Principles of Integration of KBE and C3P.* Computer-Aided Design, May, 32(5-6), pg 311.

[20] Jackson, Colin. 1989. *Building a Competitive Advantage Through Information Technology.* Long Range Planning, Aug, Vol 22 (116), pg 29.

[21] Ohno, Taiichi. 1998. *Toyota Production System, Beyond Large-Scale Production.* Productivity Press, Cambridge, Massachusetts.

[**22**] Womack, J P, Jones, D T and Roos, D. 1990. *The Machine that Changed the World.* Rawson Associates/Macmillan, NY.

[**23**] Naylor, J B, Naim, M M and Berry, D. 1999. *Leagility: Integrating the Lean and Agile Manufacturing Paradigms in the Total Supply Chain.* International Journal of Production Economics, Vol 62, pg 107.

[**24**] Maskell, B. 2001. *The Age of Agile Manufacturing: Supply Chain Management.* International Journal, Vol 6 (1), pg 5.

[**25**] Goldman, S L, Nagel, R N and Preiss, K. 1995. *Agile Competitors and Virtual Organisations*. Van Nosstrum Reinhold, New York.

[**26**] Monden, Y. 1993. *Toyota Production System: An Integrated Approach to Just-In-Time.* Industrial Engineering and Management Press, Institute of Industrial Engineers, Norcross, Georgia.

[**27**] Mortimer, John. 2001. *Robots at the Heart of Fully Automated Mini Bodyshop.* The Industrial Robot, Vol 28 (6), pg 466.

[**28**] Vonderembse, M A, Raghunathan, T S and Subba Rao S. 1997. *A Post-Industrial Paradigm: to Integrate and Automate Manufacturing*. International Journal of Production Research, Vol 35 (9), pg 2579.

[**29**] Ranky, Paul G. 1983. *The design and operation of FMS, flexible manufacturing systems*. IFS Publications, Kempston, Bedford.

[**30**] Raouf, A and Ahmad, S I. 1985. *Flexible manufacturing: recent developments in FMS, robotics, CAD/CAM, CIM.* Elsevier, Amsterdam.

[**31**] Kruth, J P. 1991. *Material Incress Manufacturing Rapid Prototyping Technques.* Annals of the CIRP, Keynote Papers, Vol 40 (2), pg 603.

[32] Miles, J C and Moore, C J. 1994. *Practical Knowledge Based Systems for Conceptual Desig*. Springer-Verlag, New York.

[33] Lovett, P J, Ingram, A and Bancroft, C N. 1999. *Knowledge-Based Engineering: A Little Knowledge is a Wondrous Thing*. Manufacturing Engineer, Vol 78 (3), pg 125.

[34] Jackson, P. 1999. *Introduction to Expert Systems*, 3rd ed. Addison-Wesley, Harlow.

[35] EDS. *UGS PLM Solutions*. http://www.eds.com/products/plm/index.shtml (accessed 6th April 2004).

[36] Windchill. *Windchill*. http://www.ptc.com/products/windchill (accessed 13th April 2004).

[37] Dassault Systemes: Product & Solutions / Enovia. http://www.enovia.com (accessed 13th April 2004).

[38] SAP. *SAP*. http://www.sap.com (accessed 6th April 2004).

[39] Baan. *SSA Global*. http://www.baan.com (accessed 13th April 2004).

[40] IGES. *IGES Project*. http://www.nist.gov/iges (accessed 19th April 2004).

[41] ISO. *ISO - International Organization for Standardization*. http://www.iso.ch/iso/en (accessed 19th April 2004).

[42] EPM Technology. *EPM Technology*. http://www.epmtech.jotne.com (accessed 19th April 2004).

[43] Collins. 1994. *Collins English Dictionary*, 3rd ed. HarperCollins Publishers, Glasgow.

[44] Gammons, R. 1992. *ESKIMO: An Expert System for Kodak Injection Moulding Operations.* Artificial Intelligence in Engineering Design, Volume 2, pg 81-104.

[45] Jovic, F. 1992. *Expert Systems in Process Control.* Chapman & Hall, London.

[46] Arzen, K. 1987. *Realization of Expert System Based Feedback Control.* Studentlitteratur, Lund.

[47] Edwards, J S. 1991. *Building Knowledge-Based Systems: Towards a New Methodology.* Pitman Publishing. London.

[48] McGhee, J, Grimble, M J and Mowforth, P (ed). 1990. *Knowledge Based Systems for Industrial Control.* Peter Pereginus, London.

[49] Bollart, L, Krijgsman, A and Vingerhoeds R A (ed). 1992. *Application of Artificial Intelligence in Process Control.* Pergamon Press, Oxford.

[50] Stephanopoulos, G and Han, C (ed). 1996. *Intelligence Systems in Process Engineering.* Academic Press, San Diego.

[51] Boullart, L. 1992. Object Orientation and Object Oriented Programming. In: *Application of Artificial Intelligence in Process Control,* edited by Bollart, L, Krijgsman, A and Vingerhoeds R A. Pergamon Press, Oxford.

APPENDIX 1 – COMPUTER CODES

Secondary_Channel1.CATScript

```
Language="VBSCRIPT"

Sub CATMain()

Dim partDocument1 As Document
Set partDocument1 = CATIA.ActiveDocument

Dim part1 As Part
Set part1 = partDocument1.Part

Dim bodies1 As Bodies
Set bodies1 = part1.Bodies

Dim body1 As Body
Set body1 = bodies1.Item("PartBody")

Dim sketches1 As Sketches
Set sketches1 = body1.Sketches

Dim hybridBodies1 As HybridBodies
Set hybridBodies1 = part1.HybridBodies

Dim hybridBody1 As HybridBody
Set hybridBody1 = hybridBodies1.Item("Final Closed Surface")

Dim hybridShapes1 As HybridShapes
Set hybridShapes1 = hybridBody1.HybridShapes

Dim reference1 As HybridShape
Set reference1 = hybridShapes1.Item("Plane.7")

Dim sketch1 As Sketch
Set sketch1 = sketches1.Add(reference1)

Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0.333324
arrayOfVariantOfDouble1(1) = 0.000000
arrayOfVariantOfDouble1(2) = 24.996000
arrayOfVariantOfDouble1(3) = 0.000000
arrayOfVariantOfDouble1(4) = 1.000000
arrayOfVariantOfDouble1(5) = 0.000000
arrayOfVariantOfDouble1(6) = -0.999911
arrayOfVariantOfDouble1(7) = 0.000000
```

```
arrayOfVariantOfDouble1(8) = 0.013334
sketch1.SetAbsoluteAxisData arrayOfVariantOfDouble1

Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()

Dim geometricElements1 As GeometricElements
Set geometricElements1 = sketch1.GeometricElements

Dim axis2D1 As GeometricElement
Set axis2D1 = geometricElements1.Item("AbsoluteAxis")

Dim line2D1 As CATBaseDispatch
Set line2D1 = axis2D1.GetItem("HDirection")

line2D1.ReportName = 1

Dim line2D2 As CATBaseDispatch
Set line2D2 = axis2D1.GetItem("VDirection")

line2D2.ReportName = 2

Dim point2D1 As Point2D
Set point2D1 = factory2D1.CreatePoint(330.000000, -550.000000)

point2D1.ReportName = 3

Dim point2D2 As Point2D
Set point2D2 = factory2D1.CreatePoint(470.000000, -550.000000)

point2D2.ReportName = 4

Dim line2D3 As Line2D
Set line2D3 = factory2D1.CreateLine(330.000000, -550.000000, 470.000000, -
550.000000)

line2D3.ReportName = 5

line2D3.StartPoint = point2D1

line2D3.EndPoint = point2D2

Dim constraints1 As Constraints
Set constraints1 = sketch1.Constraints

Dim reference2 As Reference
Set reference2 = part1.CreateReferenceFromObject(line2D3)
```

```
Dim reference3 As Reference
Set reference3 = part1.CreateReferenceFromObject(line2D1)

Dim constraint1 As Constraint
Set constraint1 = constraints1.AddBiEltCst(catCstTypeHorizontality, reference2,
reference3)

constraint1.Mode = catCstModeDrivingDimension

Dim point2D3 As Point2D
Set point2D3 = factory2D1.CreatePoint(490.000000, -600.000000)

point2D3.ReportName = 6

Dim line2D4 As Line2D
Set line2D4 = factory2D1.CreateLine(470.000000, -550.000000, 490.000000, -
600.000000)

line2D4.ReportName = 7

line2D4.StartPoint = point2D2

line2D4.EndPoint = point2D3

Dim point2D4 As Point2D
Set point2D4 = factory2D1.CreatePoint(350.000000, -600.000000)

point2D4.ReportName = 8

Dim line2D5 As Line2D
Set line2D5 = factory2D1.CreateLine(490.000000, -600.000000, 350.000000, -
600.000000)

line2D5.ReportName = 9

line2D5.StartPoint = point2D3

line2D5.EndPoint = point2D4

Dim line2D6 As Line2D
Set line2D6 = factory2D1.CreateLine(350.000000, -600.000000, 330.000000, -
550.000000)

line2D6.ReportName = 10

line2D6.StartPoint = point2D4

line2D6.EndPoint = point2D1
```

```
Dim reference4 As Reference
Set reference4 = part1.CreateReferenceFromObject(line2D6)

Dim reference5 As Reference
Set reference5 = part1.CreateReferenceFromObject(line2D4)

Dim constraint2 As Constraint
Set constraint2 = constraints1.AddBiEltCst(catCstTypeParallelism, reference4,
reference5)

constraint2.Mode = catCstModeDrivingDimension

Dim reference6 As Reference
Set reference6 = part1.CreateReferenceFromObject(line2D3)

Dim reference7 As Reference
Set reference7 = part1.CreateReferenceFromObject(line2D5)

Dim constraint3 As Constraint
Set constraint3 = constraints1.AddBiEltCst(catCstTypeParallelism, reference6,
reference7)

constraint3.Mode = catCstModeDrivingDimension

sketch1.CloseEdition

part1.Update

End Sub
```

Secondary_Channel2.CATScript

```
Language="VBSCRIPT"

Sub CATMain()

Dim partDocument1 As Document
Set partDocument1 = CATIA.ActiveDocument

Dim part1 As Part
Set part1 = partDocument1.Part

Dim shapeFactory1 As Factory
Set shapeFactory1 = part1.ShapeFactory

Dim bodies1 As Bodies
Set bodies1 = part1.Bodies

Dim body1 As Body
Set body1 = bodies1.Item("PartBody")

Dim sketches1 As Sketches
Set sketches1 = body1.Sketches

Dim sketch1 As Sketch
Set sketch1 = sketches1.Item("Sketch.8")

Dim pocket1 As Pocket
Set pocket1 = shapeFactory1.AddNewPocket(sketch1, 15.000000)

part1.Update

End Sub
```

Tapping.CATScript

```
Language="VBSCRIPT"

Sub CATMain()

Dim specsAndGeomWindow1 As Window
Set specsAndGeomWindow1 = CATIA.ActiveWindow

Dim viewer3D1 As Viewer
Set viewer3D1 = specsAndGeomWindow1.ActiveViewer

viewer3D1.ZoomOut

Dim viewpoint3D1 As Viewpoint3D
Set viewpoint3D1 = viewer3D1.Viewpoint3D

viewer3D1.ZoomOut

Set viewpoint3D1 = viewer3D1.Viewpoint3D

Dim partDocument1 As Document
Set partDocument1 = CATIA.ActiveDocument

Dim part1 As Part
Set part1 = partDocument1.Part

Dim bodies1 As Bodies
Set bodies1 = part1.Bodies

Dim body1 As Body
Set body1 = bodies1.Item("PartBody")

Dim sketches1 As Sketches
Set sketches1 = body1.Sketches

Dim hybridBodies1 As HybridBodies
Set hybridBodies1 = part1.HybridBodies

Dim hybridBody1 As HybridBody
Set hybridBody1 = hybridBodies1.Item("Final Closed Surface")

Dim hybridShapes1 As HybridShapes
Set hybridShapes1 = hybridBody1.HybridShapes

Dim reference1 As HybridShape
Set reference1 = hybridShapes1.Item("Plane.7")
```

```
Dim sketch1 As Sketch
Set sketch1 = sketches1.Add(reference1)

Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0.333324
arrayOfVariantOfDouble1(1) = 0.000000
arrayOfVariantOfDouble1(2) = 24.996000
arrayOfVariantOfDouble1(3) = 0.000000
arrayOfVariantOfDouble1(4) = 1.000000
arrayOfVariantOfDouble1(5) = 0.000000
arrayOfVariantOfDouble1(6) = -0.999911
arrayOfVariantOfDouble1(7) = 0.000000
arrayOfVariantOfDouble1(8) = 0.013334
sketch1.SetAbsoluteAxisData arrayOfVariantOfDouble1

Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()

Dim geometricElements1 As GeometricElements
Set geometricElements1 = sketch1.GeometricElements

Dim axis2D1 As GeometricElement
Set axis2D1 = geometricElements1.Item("AbsoluteAxis")

Dim line2D1 As CATBaseDispatch
Set line2D1 = axis2D1.GetItem("HDirection")

line2D1.ReportName = 1

Dim line2D2 As CATBaseDispatch
Set line2D2 = axis2D1.GetItem("VDirection")

line2D2.ReportName = 2

Dim point2D1 As Point2D
Set point2D1 = factory2D1.CreatePoint(480.000000, -850.000000)

point2D1.ReportName = 3

Dim circle2D1 As Circle2D
Set circle2D1 = factory2D1.CreateClosedCircle(480.000000, -850.000000, 4.000000)

circle2D1.CenterPoint = point2D1

circle2D1.ReportName = 4

Dim constraints1 As Constraints
Set constraints1 = sketch1.Constraints
```

```
Dim reference2 As Reference
Set reference2 = part1.CreateReferenceFromObject(point2D1)

Dim reference3 As Reference
Set reference3 = part1.CreateReferenceFromObject(line2D1)

Dim constraint1 As Constraint
Set constraint1 = constraints1.AddBiEltCst(catCstTypeDistance, reference2,
reference3)

constraint1.Mode = catCstModeDrivingDimension

Dim length1 As Dimension
Set length1 = constraint1.Dimension

length1.Value = 850.000000

Dim reference4 As Reference
Set reference4 = part1.CreateReferenceFromObject(point2D1)

Dim reference5 As Reference
Set reference5 = part1.CreateReferenceFromObject(line2D2)

Dim constraint2 As Constraint
Set constraint2 = constraints1.AddBiEltCst(catCstTypeDistance, reference4,
reference5)

constraint2.Mode = catCstModeDrivingDimension

Dim length2 As Dimension
Set length2 = constraint2.Dimension

length2.Value = 480.000000

Dim reference6 As Reference
Set reference6 = part1.CreateReferenceFromObject(circle2D1)

Dim constraint3 As Constraint
Set constraint3 = constraints1.AddMonoEltCst(catCstTypeRadius, reference6)

constraint3.Mode = catCstModeDrivingDimension

Dim length3 As Dimension
Set length3 = constraint3.Dimension

length3.Value = 4.000000
```

```
sketch1.CloseEdition

part1.Update

Dim shapeFactory1 As Factory
Set shapeFactory1 = part1.ShapeFactory

Dim pocket1 As Pocket
Set pocket1 = shapeFactory1.AddNewPocket(sketch1, 15.000000)

pocket1.IsThin = True

part1.Update

End Sub
```

RowOfTappings.CATScript

```
Language="VBSCRIPT"

Sub CATMain()

Dim partDocument1 As Document
Set partDocument1 = CATIA.ActiveDocument

Dim part1 As Part
Set part1 = partDocument1.Part

Dim bodies1 As Bodies
Set bodies1 = part1.Bodies

Dim body1 As Body
Set body1 = bodies1.Item("PartBody")

Dim sketches1 As Sketches
Set sketches1 = body1.Sketches

Dim hybridBodies1 As HybridBodies
Set hybridBodies1 = part1.HybridBodies

Dim hybridBody1 As HybridBody
Set hybridBody1 = hybridBodies1.Item("Final Closed Surface")

Dim hybridShapes1 As HybridShapes
Set hybridShapes1 = hybridBody1.HybridShapes

Dim reference1 As HybridShape
Set reference1 = hybridShapes1.Item("Plane.7")

Dim sketch1 As Sketch
Set sketch1 = sketches1.Add(reference1)

Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0.333324
arrayOfVariantOfDouble1(1) = 0.000000
arrayOfVariantOfDouble1(2) = 24.996000
arrayOfVariantOfDouble1(3) = 0.000000
arrayOfVariantOfDouble1(4) = 1.000000
arrayOfVariantOfDouble1(5) = 0.000000
arrayOfVariantOfDouble1(6) = -0.999911
arrayOfVariantOfDouble1(7) = 0.000000
arrayOfVariantOfDouble1(8) = 0.013334
sketch1.SetAbsoluteAxisData arrayOfVariantOfDouble1
```

```
Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()

Dim geometricElements1 As GeometricElements
Set geometricElements1 = sketch1.GeometricElements

Dim axis2D1 As GeometricElement
Set axis2D1 = geometricElements1.Item("AbsoluteAxis")

Dim line2D1 As CATBaseDispatch
Set line2D1 = axis2D1.GetItem("HDirection")

line2D1.ReportName = 1

Dim line2D2 As CATBaseDispatch
Set line2D2 = axis2D1.GetItem("VDirection")

line2D2.ReportName = 2

Dim point2D1 As Point2D
Set point2D1 = factory2D1.CreatePoint(500.000000, -850.000000)

point2D1.ReportName = 3

Dim circle2D1 As Circle2D
Set circle2D1 = factory2D1.CreateClosedCircle(500.000000, -850.000000, 4.000000)

circle2D1.CenterPoint = point2D1

circle2D1.ReportName = 4

Dim constraints1 As Constraints
Set constraints1 = sketch1.Constraints

Dim reference2 As Reference
Set reference2 = part1.CreateReferenceFromObject(point2D1)

Dim reference3 As Reference
Set reference3 = part1.CreateReferenceFromObject(line2D1)

Dim constraint1 As Constraint
Set constraint1 = constraints1.AddBiEltCst(catCstTypeDistance, reference2,
reference3)

constraint1.Mode = catCstModeDrivingDimension

Dim length1 As Dimension
Set length1 = constraint1.Dimension
```

```
length1.Value = 850.000000

Dim reference4 As Reference
Set reference4 = part1.CreateReferenceFromObject(point2D1)

Dim reference5 As Reference
Set reference5 = part1.CreateReferenceFromObject(line2D2)

Dim constraint2 As Constraint
Set constraint2 = constraints1.AddBiEltCst(catCstTypeDistance, reference4,
reference5)

constraint2.Mode = catCstModeDrivingDimension

Dim length2 As Dimension
Set length2 = constraint2.Dimension

length2.Value = 500.000000

Dim reference6 As Reference
Set reference6 = part1.CreateReferenceFromObject(circle2D1)

Dim constraint3 As Constraint
Set constraint3 = constraints1.AddMonoEltCst(catCstTypeRadius, reference6)

constraint3.Mode = catCstModeDrivingDimension

Dim length3 As Dimension
Set length3 = constraint3.Dimension

length3.Value = 4.000000

Dim point2D2 As Point2D
Set point2D2 = factory2D1.CreatePoint(540.000000, -850.000000)

point2D2.ReportName = 5

Dim circle2D2 As Circle2D
Set circle2D2 = factory2D1.CreateClosedCircle(540.000000, -850.000000, 4.000000)

circle2D2.CenterPoint = point2D2

circle2D2.ReportName = 6

Dim reference7 As Reference
Set reference7 = part1.CreateReferenceFromObject(point2D2)
```

```
Dim reference8 As Reference
Set reference8 = part1.CreateReferenceFromObject(line2D1)

Dim constraint4 As Constraint
Set constraint4 = constraints1.AddBiEltCst(catCstTypeDistance, reference7,
reference8)

constraint4.Mode = catCstModeDrivingDimension

Dim length4 As Dimension
Set length4 = constraint4.Dimension

length4.Value = 850.000000

Dim reference9 As Reference
Set reference9 = part1.CreateReferenceFromObject(point2D2)

Dim reference10 As Reference
Set reference10 = part1.CreateReferenceFromObject(line2D2)

Dim constraint5 As Constraint
Set constraint5 = constraints1.AddBiEltCst(catCstTypeDistance, reference9,
reference10)

constraint5.Mode = catCstModeDrivingDimension

Dim length5 As Dimension
Set length5 = constraint5.Dimension

length5.Value = 540.000000

Dim reference11 As Reference
Set reference11 = part1.CreateReferenceFromObject(circle2D2)

Dim constraint6 As Constraint
Set constraint6 = constraints1.AddMonoEltCst(catCstTypeRadius, reference11)

constraint6.Mode = catCstModeDrivingDimension

Dim length6 As Dimension
Set length6 = constraint6.Dimension

length6.Value = 4.000000

Dim point2D3 As Point2D
Set point2D3 = factory2D1.CreatePoint(580.000000, -850.000000)

point2D3.ReportName = 7
```

```
Dim circle2D3 As Circle2D
Set circle2D3 = factory2D1.CreateClosedCircle(580.000000, -850.000000, 4.000000)

circle2D3.CenterPoint = point2D3

circle2D3.ReportName = 8

Dim reference12 As Reference
Set reference12 = part1.CreateReferenceFromObject(point2D3)

Dim reference13 As Reference
Set reference13 = part1.CreateReferenceFromObject(line2D1)

Dim constraint7 As Constraint
Set constraint7 = constraints1.AddBiEltCst(catCstTypeDistance, reference12,
reference13)

constraint7.Mode = catCstModeDrivingDimension

Dim length7 As Dimension
Set length7 = constraint7.Dimension

length7.Value = 850.000000

Dim reference14 As Reference
Set reference14 = part1.CreateReferenceFromObject(point2D3)

Dim reference15 As Reference
Set reference15 = part1.CreateReferenceFromObject(line2D2)

Dim constraint8 As Constraint
Set constraint8 = constraints1.AddBiEltCst(catCstTypeDistance, reference14,
reference15)

constraint8.Mode = catCstModeDrivingDimension

Dim length8 As Dimension
Set length8 = constraint8.Dimension

length8.Value = 580.000000

Dim reference16 As Reference
Set reference16 = part1.CreateReferenceFromObject(circle2D3)

Dim constraint9 As Constraint
Set constraint9 = constraints1.AddMonoEltCst(catCstTypeRadius, reference16)
```

```
constraint9.Mode = catCstModeDrivingDimension

Dim length9 As Dimension
Set length9 = constraint9.Dimension

length9.Value = 4.000000

Dim point2D4 As Point2D
Set point2D4 = factory2D1.CreatePoint(460.000000, -850.000000)

point2D4.ReportName = 9

Dim circle2D4 As Circle2D
Set circle2D4 = factory2D1.CreateClosedCircle(460.000000, -850.000000, 4.000000)

circle2D4.CenterPoint = point2D4

circle2D4.ReportName = 10

Dim reference17 As Reference
Set reference17 = part1.CreateReferenceFromObject(point2D4)

Dim reference18 As Reference
Set reference18 = part1.CreateReferenceFromObject(line2D1)

Dim constraint10 As Constraint
Set constraint10 = constraints1.AddBiEltCst(catCstTypeDistance, reference17,
reference18)

constraint10.Mode = catCstModeDrivingDimension

Dim length10 As Dimension
Set length10 = constraint10.Dimension

length10.Value = 850.000000

Dim reference19 As Reference
Set reference19 = part1.CreateReferenceFromObject(point2D4)

Dim reference20 As Reference
Set reference20 = part1.CreateReferenceFromObject(line2D2)

Dim constraint11 As Constraint
Set constraint11 = constraints1.AddBiEltCst(catCstTypeDistance, reference19,
reference20)

constraint11.Mode = catCstModeDrivingDimension
```

```
Dim length11 As Dimension
Set length11 = constraint11.Dimension

length11.Value = 460.000000

Dim reference21 As Reference
Set reference21 = part1.CreateReferenceFromObject(circle2D4)

Dim constraint12 As Constraint
Set constraint12 = constraints1.AddMonoEltCst(catCstTypeRadius, reference21)

constraint12.Mode = catCstModeDrivingDimension

Dim length12 As Dimension
Set length12 = constraint12.Dimension

length12.Value = 4.000000

sketch1.CloseEdition

part1.Update

Dim shapeFactory1 As Factory
Set shapeFactory1 = part1.ShapeFactory

Dim pocket1 As Pocket
Set pocket1 = shapeFactory1.AddNewPocket(sketch1, 15.000000)

part1.Update

End Sub
```