RESEARCH TITLE:

THE APPLICATION OF KNOWLEDGE BASED SYSTEMS TO THE ABSTRACTION OF DESIGN AND COSTING RULES IN BESPOKE PIPE JOINTING SYSTEMS.

OLUWAKEMI VICTORIA FATONADE

ACKNOWLEDGEMENT

I gratefully acknowledge the funding and opportunity provided by the University of Glamorgan through the Knowledge Transfer Partnership Scheme that enabled me to undertake my MPhil study. I would like to thank the staff of Teddington Engineered Solutions Ltd for their contributions towards my research case studies and also thank Dr Emlyn Everitt and Mr Steve Gardner for encouraging me to undertake this research and guiding me through to completion.

Finally, I would like to thank my family and friends for their prayers and words of encouragement.

ABSTRACT

This thesis presents the work undertaken in the creation of a knowledge based system aimed at facilitating the design and cost estimation of bespoke pipe jointing systems. An overview of the problem domain is provided and the findings from a literature review on knowledge based systems and applications in manufacturing were used to provide initial guidance to the research. The overall investigation and development process involved the abstraction of design and costing rules from domain experts using a sub-set of the techniques reviewed and the development and implementation of the knowledge based system using an expert system approach, the soft systems methodology (SSM) and the system development lifecycle methodology. Based on the abstracted design and costing rules, the developed system automates the design of pipe jointing systems, and facilitates cost estimation process within third party configuration software. The developed system was validated using two case studies and was shown to provide the required outputs.

INTRODUCTION

This thesis details the research programme relating to the development of an expert system for the design and cost estimation of bellows and expansion joints. The research was undertaken as part of a collaborative industrial project between Teddington Engineered solution Ltd based in Llanelli, and the University of Glamorgan under the Knowledge Transfer Partnership scheme. Knowledge Transfer Partnerships is a Technology Strategy Board programme aimed at providing businesses with the opportunity to improve their competitiveness and productivity by employing the knowledge, technology and skills that exist within the UK knowledge base.

Teddington Engineered Solutions Ltd. (TES Ltd) design and manufacture bespoke bellows and expansion joints used in piping systems with diameters ranging from 10mm to 6m. These products are fabricated in stainless steel or various nickel alloys for harsh environments, using many different processes suited to each application. They are used in the aerospace, power, oil & gas, nuclear, steel, defence, rail & locomotive, ship building and general engineering industries requiring high integrity products.

The sales enquiries received by the company are generally for new, bespoke products that have to be designed to exacting standards, each requiring unique material, process and workflow specification. Based on these specifications, a design solution for the product is produced and an estimate of the overall manufacturing cost is sent along with a copy of the design solution as quotation which may be declined or accepted and returned as an order by a customer. The existing capability to produce designs and cost estimations is limited to three expert design engineers and two cost estimators in the company, each with specific accumulated knowledge and experience. The extant design process involved the use of Microsoft Access application software called [1]EJMA (see glossary) which provided design and production parameters based on customer specifications, (the underlying mathematical calculations within this software conform to the EJMA standards) as well as a 3D [5]CAD system called ***ProEngineer*** with capabilities such as Solid Modeling, Surfacing, Rendering, Data Interoperability, Routed Systems Design, Simulation, Tolerance Analysis, and NC and tooling Design. Cost estimations were based on cost data look up tables and performed using Microsoft Excel Spreadsheets.

The company's strategy for growth involved increase in sales through further export development and export market penetration. The opening of new sales offices and appointment of sales agents were required for the delivery of this strategy. However an internal analysis of sales versus quotation had shown a clear correlation between speed of response to enquiries by the issue of a formal quotation and acceptance of orders by customers. Therefore, a fundamental requirement for the delivery of the company's strategy was a major improvement in its capability to rapidly and efficiently process enquiries into quotations and thereafter new sales. The company's dependency on these individuals to translate enquiries into quotations limited its performance in acquiring sales hence creating a barrier to the implementation of its growth strategy.

The objective of this research programme was to systematically extract expertise knowledge in the areas of design and cost estimation, analyse the knowledge and employ appropriate expert system techniques in modelling a system that would integrate with the organisation's existing IT systems to provide support to experts and at the same time bring about the retention of expertise knowledge within the organisation to some extent. The envisaged approach was the development and implementation of a knowledge based expert system to facilitate product design and cost estimation processes with TES ltd. In order to ascertain the relevance of this approach and to determine a methodology for the undertaking of the work, a formal literature review was undertaken. This literature review provided an opportunity to thoroughly understand the theoretical and practical implications of routes taken by previous researchers in implementing knowledge based systems. Findings from the review are documented in the subsequent chapter.

CHAPTER 2– LITERATURE REVIEW

A critical review of published work in the fields of endeavour related to this research programme provided an essential insight into the background theory behind Knowledge Based Systems as well as the techniques and methodology employed by researchers and authors of relevant contemporary work that have been undertaken in both academic and industrial areas of manufacturing.

2.1 BACKGROUND

A Knowledge-based system can be defined as an artificial intelligence program that achieves expert-level competence in solving problems in task areas by referencing a body of knowledge about those specific tasks [1][5]. There are variations in the categories of KBS speculated by researchers, however, two commonly known categories are;

Expert Systems – These have been described as the most established and recognised knowledge-based technology [5]. They capture human problem-solving expertise and are useful for problems within narrow domains that require expertise and for which an expert is available to identify clear and complete problem-solving rules. [7]
Expert systems are typically used by less experienced members of a team for decision support, and also for training, as the reasoning processes and applications of knowledge in specific contexts can be observed. The decision support role of expert systems also allows experts to be available for the more challenging or unusual tasks [61] [62];

Artificial Neural Networks - Neural networks emulate biological neural networks. They share the common goal of enabling computers to capture and apply knowledge. Neural networks can improve their own performance, adapt, and discover relationships in data. They are nonlinear and pattern recognising in nature and they learn to solve problems by being shown examples of situations and associated solutions. The network learns a relationship that is valid between each of the sample situations and the associated solutions. Upon finding this relationship, the network is able to generalise and so provide an appropriate solution to a new situation. Fault diagnosis and robotic systems are some of the applications implemented by artificial neural networks. [2][61]

Many successes have been recorded in the implementation of expert systems in manufacturing areas ranging from high-level conceptual design and cost estimation of abstract entities through to the configuration of manufacturing processes on the factory floor. Further more, the capabilities of expert systems and their usefulness in solving problems difficult enough to require expertise, validates the distillation of this review to the analysis of the use of expert systems in manufacturing.

2.2 EXPERT-SYSTEMS IN MANUFACTURING

The need for ongoing and real-time support, process monitoring & control and product optimisation is common to all types of manufacturing. This need provides the objective for better use of knowledge, the best design and manufacturing expertise readily available; helping users identify relationships among design geometry, materials and production processes, and leveraging these relationships to provide the best combination of product features, quality and cost. [2] [58]

2.2.1 MANUFACTURING DESIGN

Design for manufacture (DFM) is a manufacturing design approach that integrates product and process design selection to ensure the best matching of needs and requirements.[8][70] The primary objective of DFM is to produce a design at a competitive cost by improving its manufacturability without affecting its functional and performance objectives [8][64]. A similar approach is adapted in the design to cost system developed by Shehab & Abdalla [41] to recommend the most economical assembly technique for a product and provide design improvement suggestions; and, Mohamed & Celik [67] to recommend alternative design and cost estimating. There are variations between the systems developed from these two approaches in terms of their overall structure and the level of detail of the design and cost estimation provided. However, both systems share the similarity of being integrated with computer-aided-design (CAD) systems and more significantly being developed in modular structure to provide an efficient way for modification or expansion of capability of the system. These similar characteristics form part of the crucial aspects of contemporary KBE systems. [84] Examples of which are the KBE applications developed and deployed using Genworks Generative Application Development system. [83] The Genworks development system runs on a proprietary GDL[7] platform and facilitates ease and speed of development as well as seamless web-based deployment for geometry-intensive, knowledge based engineering solutions. This is achieved through integration with a web server and the SMLib[8] geometry modeling kernel. [83][84]

The systems developed by Shehan & Abdalla and Mohammed & Celik consist of a number of modules which act as tools for choosing suitable product materials based on property requirements; performing process selection decisions based on a set of design and production parameters to achieve cost-effective manufacturing; and, estimating manufacturing cost based on the selected materials and processes processes. The following three modules for material selection, process analysis and selection, and cost estimation are common between both systems.

Material Selection Module - This module is activated in response to the user inputs of mechanical property requirements of the product. Its results affect the analysis of the other two modules. A range of materials used in industry is considered and represented in IF-THEN production rules. Whilst, the antecedent or conditional part of the rules represents the respective mechanical properties of the materials, the consequent part of the rules represents the corresponding materials. A user inputs a mechanical property of choice and if the input matches the conditions of a rule, the rule is triggered. When this happens, the material concerned in the triggered rule is selected as the recommended product material and its related data is used to activate operations of the other two modules in the expert system.

Process Analysis and Selection Module – This module is activated in response to user inputs of production requirements, product geometric features and information of the selected material from the previous material selection module.

Product Costs Estimation Module - The selected manufacturing processes and product materials from analysis of the two other modules are used as the basis for estimating the manufacturing and material costs. Other data required to complete the product costs estimation are based on tools required; product characteristics (product size, volume, and shape); production data (production volume, manufacturing time and labour rate); Material data; and, Overheads. The data and their effects are arranged as production rules in this module. When the module is executed, the overall product costs are output.

Whilst this sequential approach of material selection, process analysis / selection and cost estimation is very practical, the Cambridge Engineering Selector (CES) - a contemporary tool for the selection of material and design information offers the additional advantage of innovation and optimum use of engineering materials and manufacturing process. [85][86] The Ashby methods developed by Professor Mike Ashby – a co-founder of Granta designs, is a fundamental concept upon which the CES selector is based. [86]

The Ashby approach focuses on the ultimate design of a product and it is initiated by a response to the function of the component material in the design, the objectives which must be optimized, and the constraints which must be satisfied. For instance, an expansion joint (function) needs to be as flexible as possible (objective) to support a specified movement and acceptable resistance to contact with various environments (constraint). The Model-based selection concept of the Ashby method enables selection of property values of a material based on certain factors. As material performance for a specific application is often determined by multiple material properties, a mathematical analysis of the engineering problem is used to derive performance indices**.** Analysis and resolution of trade-offs between conflicting objectives - performance and cost for instance, is achieved through a quantitative representation of performance in relation to a combination of properties.

Ashby's selection charts provide a graphical environment in which to apply and analyze quantitative selection criteria, such as those captured in performance indices, and also to make trade-offs between conflicting objectives. The selection charts facilitate the derivation of alternative material selections. This ensures the optimisation of engineering materials and manufacturing processes and eliminates the possibility of getting null results on material selection - an occurrence which can be expected in a sequential material selection approach, if user responses on material requirements cannot be directly matched.

2.2.2 PRODUCT COST ESTIMATION IN MAKE-TO-ORDER MANUFACTURING

In the area of make-to-order or engineer-to-order manufacturing, technical expertise, delivery time and reliability have been established as factors relating to the basis on which companies compete for orders with other suppliers [37]. One important factor that is crucial at the customer enquiry stage is a fast enquiry to quotation process which involves an initial presentation of the product design and an estimated product cost (in form of a quotation) in response to an enquiry made by a customer. A great deal of flexibility is required to sustain the design and configuration of new or modified products whilst dealing with the uniqueness of each customer order. In addition, a constriction at this stage is the estimation of the overall production cost to be quoted [37]. According to Shehab & Abdalla [38], previous researchers have reported that although a product’s design phase accounts for only 6% of the total development cost [38][39], a significant percentage of the overall product cost is committed at the early stage of the design process [40]. As a result of this, making appropriate decisions concerning product cost is more crucial at the design stage than at the manufacturing stage [41].

Accuracy in cost estimation is crucial to the performance of a business in the sense that whilst underestimation may cause financial losses, overestimation may reduce a company’s competitiveness in the industry market consequently leading to loss of business and goodwill. This importance has led to extensive research into techniques and methods to achieve accuracy and consistency in producing cost estimation for the prompt delivery of high quality designs.

The final price quoted by TES ltd for the manufacture of any product is the sum of an estimated total production cost (which is made up of the material costs; labour costs; and, sundries e.g. work sub-contracted out) and a profit margin. Whilst the materials and sundries costs are based on pre-defined formulae and supplier costs, the labour costs are, to a great extent, dependent on historic estimation standards based on manufacturing activities / processes on product components. In addition to the fact that these standards are not validated in terms of an organised comparison of the estimated cost with the actual cost for the orders won, certain heuristics are applied as ‘rule of thumb’ by estimators/ experts in the different cost units mentioned. Kingsman and De Souza [37] noted that these heuristics involved the knowledge of the product / production system and market conditions in addition to economic and technological trends. Their applications by different estimators often yielded different estimations of costs for the same product and these limitations put together have a significant negative impact on the accuracy and consistency of estimations.

2.2.3 PRODUCT COST ESTIMATION TECHNIQUES

A number of product cost estimation techniques have been researched to address an extensive variety of issues encountered during initial investigations. Niazi and Dai [42] present a hierarchical classification of these techniques into Qualitative and Quantitative techniques. Quantitative techniques can be further categorized into parametric and analytical techniques and although they are capable of providing more accurate results, they require detailed analysis of product designs; features and corresponding manufacturing processes which are usually carried out at the final design stage after a quotation has been converted into an order [42][64][79]. This stage is outside the scope of this study.

Conversely, because Qualitative techniques rather make use of historic data in predicting the estimated costs for new products, they are more useful in deriving cost estimations in the early stages of product design [42] and serve as a good basis for decision making. As the company's current estimation process involves the use of historic data, the use of qualitative techniques is better suited to solving the cost estimation problems earlier discussed.

Niazi and Dai [42] further categorise Qualitative cost estimation techniques into Intuitive and Analogical techniques which are also mentioned by Chougule & Ravi [79].

(I) Analogical cost estimation techniques

This technique is demonstrated in Regression analysis models as adopted by Hundal [43] and Lewis [44] and Back-propagation Neural-Network models as adopted by Zhang and Fuh [45] [42]. Whilst the former is used to forecast the cost of new products by using historic cost data to create linear relationships between the cost for historic design cases and the value of selected variables, the latter adapts better to uncertain conditions and non-linearity through the use of neural networks based on a machine learning approach. However, both models share a common limitation which is the restrictiveness resulting from their employment of similarity criteria which is dependent on the cost data for historic design cases with known cost.

(II) Intuitive cost estimation techniques

These techniques are rather based on past experience i.e. the use of domain expert knowledge to methodically derive cost estimates for product parts and assemblies and are achieved using case based methodology or decision support systems [42][79]***.*** As in the case of the analogical techniques previously discussed, the use of case-based methodology is restricted to the availability of similar past designs. Past design cases are retrieved from a database using the attributes of a new design as search criteria and attempts are made to make necessary changes to parts and assemblies of the retrieved design or incorporate missing parts. The new design is stored and cost estimation for the new product is derived by combining the cost of the past design to that of the added components / assemblies. This approach greatly reduces the need to design or estimate cost from base data.

Decision support systems (DSS) on the other hand, act as decision-aid tools by representing domain expert experience (which could be in form of data and rules about processes and constraints, decision trees and other factors that could influence the process of decision making) in a manner well suited towards problem solving thus improving the judgments made by estimators at various levels of the estimation process. [37][42]

Three different techniques for the development of decision support systems have been established [41] [42]. These are: Rule based technique, fuzzy logic technique and expert system technique. Whilst advantages and disadvantages have been recorded for each of the approaches, the expert system approach possesses the strongest advantages of providing a quicker inference with more consistent and accurate results through its imitation of human expertise. It achieves this through automated logical reasoning often derived from rule based programming [42]. The use of a decision support expert system has been recorded in a system developed by Kingsman and De Souza [37] for cost estimation and pricing decisions in versatile manufacturing companies that implement make-to-order processes.

2.3 EXPERT SYSTEMS DESIGN

Like any software project, there are a number of considerations to be made prior to the commitment of people, resources and time to the development of a proposed expert system. In this section, the general guidelines and considerations for designing practical experts systems are presented and discussed.

2.3.1 SELECTING THE APPROPRIATE PARADIGM

It is crucial to select a problem domain or decide if an expert system is the appropriate paradigm for solving a problem. Suggestions have been made as to the factors on which this decision could depend on.

Expert systems are appropriate when an expert's knowledge is largely heuristic (experiential knowledge) and solutions to a problem can be derived only through reasoning. If the problem can be solved simply by logic and algorithm, a conventional program is best suited. In diagnosing some equipment for instance, if all the symptoms of malfunction are have been established in advance, then a look up table or decision tree of faults will be adequate [5].

An expert's knowledge is specific to one problem domain as opposed to general problem solving techniques. Like humans, experts systems are generally designed to be experts in one problem domain. It is therefore very important to have well-defined limitations on the capabilities of the expert system. The more domains there are, the expertise becomes relatively less and as a result the system eventually becomes more complex to compensate [6][69].

Justifying the expert system based on the reason of scarce human expertise is very difficult if there are already many experts. Implementing expert systems is pointless if experts or basic users are not willing to make use of it. It is critical for an organisation's management to support an expert system as deployment is sometimes viewed as a precursor to downsizing the workforce. Therefore, workers must be re-assured that the expert system will not lead to job losses but rather an opportunity to increase profitability, as expertise becomes available at a lower cost [34][6].

A human expert's enthusiasm about an expert systems project is crucial as not all experts are wiling to have their knowledge examined and fed into a computer. In cases where there are multiple experts, it might be advisable to limit the number of experts involved in the development as different experts may have different ways of solving a problem and sometimes may even reach different conclusions. This may create internal conflicts and incompatibilities [13][6][34] .

A human expert must be able to express expert knowledge in explicit terms. The expert's use of too many technical terms would mean that it would take much longer for the knowledge engineer to understand the expert, let alone translate the knowledge into explicit computer code [5][13] .

2.3.2 BENEFITS OF IMPLEMENTATION

Identifying the objectives of an expert systems project at the outset is a fundamental requirement and will relate closely to the expected benefits of deploying the system[5][68]. The benefits may include reduced cost of providing expertise, increased availability and permanence of expertise and increased efficiency. Increase efficiency can be defined as comprising of the following advantages of expert systems [69][6]:

- **Increased reliability** – By providing a second opinion to a human expert or a tie-breaker in disagreements among multiple human experts, such systems increase confidence that the correct decision was made.
- **Explanation** - An expert system can provide detailed explanation of the reasoning that led to a conclusion where a human expert may be unwilling or unable to do this at all times.
- **Fast response** - Depending on the software and hardware used, an expert system may respond faster than a human expert.
- **Reduced danger** - Expert systems can be used in environments that may be hazardous for a human expert.
- **Intelligent tutor and database –** An expert system can act as intelligent tutor to a novice / trainee by allowing sample programs to be run and providing explanations to the system's reasoning. Experts systems can also be used to access databases in an intelligent manner similar to data mining.

2.3.3 LANGUAGE, SHELLS AND TOOLS

A fundamental decision in defining a problem is deciding how best to model it. Besides the choices of the many languages available today, these terminologies are often misused in describing languages. Some vendors refer to their products as "tools", while others refer to "shells" and still others talk about integrated environments [5] [14] [33][58]. These terms can be defined as follows:

(I) Languages

A language can be defined as a translator of commands written in a specific syntax. An expert system language is a higher-order language than third-generation languages like LISP or C as it provides ease in achieving certain things. An expert system language will provide an inference engine which may provide forward or backward chaining or both depending on the implementation. Expert systems have been developed in conventional languages like C or Pascal; general artificial intelligence languages like LISP or PROLOG and their object oriented extensions - CLOS (Common Lisp Object System) and L&O (Logic and objects); and, in specialized production systems languages like CLIPS (C Language Integrated Production System) or COOL (object oriented language extension of CLIPS) [10][34]. Whilst CLIPS does not have all the features of other languages, it is simpler to learn and still maintains its original advantage of small program size and fast execution where real time response is critical. It also supports rule-based, object-oriented (COOL) and procedural programming paradigms.

Whilst conventional languages focus on providing flexible and robust techniques to represent data using specific data structures, data abstraction and encapsulation (through the use of objects, methods, packages), expert systems languages focus on providing flexible and robust ways to represent knowledge. The expert system paradigm allows two levels of abstraction: data abstraction and knowledge abstraction, and separates data from the methods of manipulating the data. An example of this kind of separation is that of facts (data abstraction) and rules (knowledge abstraction) in a rule based expert system language like CLIPS which provides objects and all the features of a true object-oriented language.

This difference in focus also leads to a difference in program design methodology. With procedural programming, programmers must carefully describe the sequence of execution because of the tight interweaving of data and knowledge. However, the explicit separation of data from knowledge in expert system languages requires less rigid control of the execution sequence. Typically, an entirely separate piece of code, the inference engine is used to apply the knowledge to the data. This allows a higher degree of parallelism and modularity [3][5].

(II)Shells

Shells are special purpose tools designed for certain types of applications in which the user must supply only the knowledge base. Ideally, it is a ready-made expert system, with the knowledge base missing [5] [12]. An expert system shell usually contains:

- A set of knowledge representation structures
- A built in inference engine
- Knowledge acquisition tools to help the knowledge engineer in the knowledge elicitation process
- A user interface and explanation facility
- Interfaces to other software systems which could be spreadsheets, databases, programming languages etc[3][10]

A classic example of this is EMYCIN (Empty MYCIN shell) which was made by removing the medical knowledge base of the MYCIN expert system. The idea of an expert system shell is as shown in Figure 2.1 below

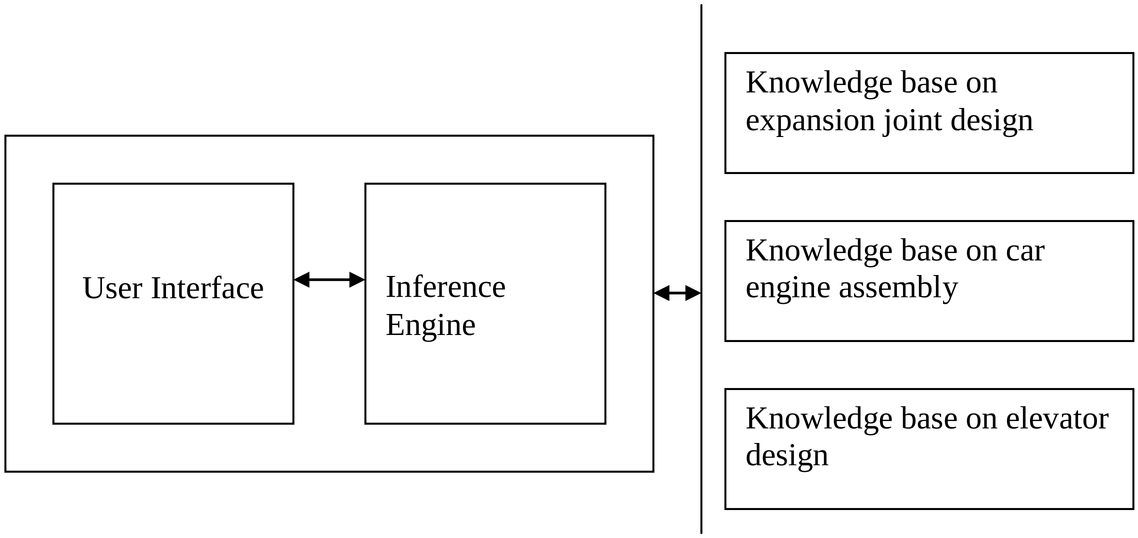


Figure 2.1*:* Expert system shell

(III)Tools

Tools can be defined as software development environments which support a language associated with utility programs to facilitate the development, debugging and delivery of application programs. Utility programs may include text, graphic or ontology and knowledge-base editors (e.g. Protégé), debuggers, file management and code generators [5][34]. The tools available for building expert systems can be classified into expert system Shells and programming languages / environments preferably artificial intelligence languages or specialized production systems as earlier discussed [11].

2.4 ELEMENTS OF AN EXPERT SYSTEM

Expert systems have been described as consisting of the following components
[3] [5][10]:

User Interface - presents questions and information to the user and supplies the user's responses to the inference engine. It receives and interpretes any values entered by the user and also checks all responses to ensure that they are of the correct data type. Any responses that are restricted to a legal set of answers are compared against these legal answers and the user is prompted whenever an illegal answer is entered.

Explanation facility – explains the reasoning of the expert system to the user;

Working memory – a global database of facts used by the rules

Inference Engine – reasons with both the knowledge base and working memory. Expert system inferencing techniques are models of the process of human reasoning which involves the derivation of results or conclusion by combining facts with knowledge. In more intelligent forward chaining systems where rules are used to drive the derivation of conclusions or results from facts, rules are executed based on the context of the facts as opposed to a pre-defined order. One major consequence of this pattern of rule execution is multiple rule matching on facts. This occurrence is otherwise known as a conflict. One crucial requirement of expert systems is its ability to implement a control strategy to resolve rule conflicts. This process of achieving this is generally known as conflict resolution. Pakiarajah et al [80] mentioned a number of conflict resolution methodologies:

- **Recency Ordering -** The most recently used rule is prioritized and applied;
- **Prioritisation** - Dependent on priority information usually provided by an expert or knowledge engineer. The rule with the highest priority is selected and applied;
- **Context Limiting -** Rules are separated into groups to reduce the occurrence of conflict. A procedure is used to activate and deactivate groups and only one group of rules can active at any one time.
- **Fired Rules -** Otherwise referred *to as* Refractoriness***,*** It involves ignoring rules which have been previously executed.

Agenda - a prioritized list of rules created by the inference engine whose patterns are satisfied by the facts or objects in working memory.

Knowledge base – contains the knowledge with which the inference engine draws conclusion. The knowledge base of expert systems contains both factual and heuristic knowledge. *Factual knowledge* is that knowledge of the task domain that is widely shared, typically found in textbooks or journals, and commonly agreed upon by experts in the particular field while *Heuristic knowledge* is the less rigorous, more experiential, more judgmental knowledge of performance and it is largely individualistic [1].

Knowledge Acquisition facility – an automated tool which allows a user to enter knowledge into the system without having the knowledge engineer explicitly code the knowledge.

There are several inference techniques for expert systems. However, the common techniques are rule based techniques and case based techniques.

Rule-based techniques involve representation of knowledge in the IF-THEN pattern with the aim of proving a goal statement or achieving a goal state [7][34]. Two general methods of rule-based inferencing for expert systems are forward chaining (data driven) and backward chaining (goal driven). Forward chaining involves reasoning from facts to the conclusions resulting from those facts while backwards chaining involves reasoning from a potential conclusion to be proved to the facts that support the conclusion [5][34].

Case-based techniques however involve solving problems based on solutions for similar problems solved in the past (precedents). It requires storing, retrieving and adapting past solutions to similar problems [7][34].

The use of cased-based techniques in the subject matter of this research will be inappropriate because the company manufactures products based on specific customer descriptions therefore there are endless possibilities of receiving enquiries for designs which may not be similar to previous ones. Rule based inference techniques are more appropriate and will be discussed for the purpose of this research.

2.5 KNOWLEDGE ACQUISITION

2.5.1 KNOWLEDGE ACQUISITION IN EXPERT SYSTEM DEVELOPMENT

Knowledge acquisition is the process of eliciting, structuring and organizing elicited knowledge from domain experts and other sources for knowledge base representation in form of rules or other forms of representation such as frames [58][59][60]. It is imperative to the development and implementation of expert systems for it contains the information required to solve problems in expert system domain [57][59]. As a result of the challenges and difficulties faced in the transfer of expertise knowledge, knowledge acquisition has been described as the bottle neck of expert systems development [33][55][56].

There are several reasons for this challenge. For example, the logic or justification behind an expert's thought is not easily revealed and the omission of this tacit knowledge could cause a detrimental gap in knowledge required by an expert system to solve the problem for which it was built [13][59]. Another major challenge in knowledge acquisition is experts' lack of willingness to share knowledge. In an academic and research environment, an expert is part of an expert system development team and he/she is acknowledged in the resulting research paper or article. However, expert systems development within a company / organisation with the explicit intention of completely or partially replacing the domain expert hinders the co-operation and enthusiasm of domain experts as the threat of losing their jobs or prestige becomes perceptible [5][13][58].

The process of knowledge acquisition has been described to comprise of five stages namely the identification, conceptualization, formalisation, implementation and testing/ debugging stages [13][66].

In the identification stage, the goal and the use of the expert system must be specified and the basic aspects of the problem and the structure of the supporting knowledge are characterised. This structure may be fixed by the nature of the domain but also embodied by the outlines of an expert systems shell that may be used. In the context of the case study at TES Ltd, the proposed expert system involved the configuration and cost estimation of bellows and expansion joints. It is characterised by the selection of appropriate material and compatible product component parts required to manufacture a product as well as the estimation of material and labour costs required. The data required are present in the company's existing Enterprise Resource Planning, ProEngineer, spreadsheets and database systems. The terms associated with the identified problem domain can be derived from domain-specific knowledge and knowledge of the domain's problem solving methods. These could be informal knowledge from statements, behaviors, notes and sketches, or structured knowledge from verbal protocols, texts, diagrams, observations and arguments. The interrelations between these terms can be realised by mapping across the domain and problem solving ontologies as described by Swartout and Gil, who further suggested that this mapping could be useful in generating a knowledge acquisition tool that would allow a user to enter domain specific knowledge and provide an understanding of the how the knowledge will actually be used. Other important factors to be considered in this stage are the concepts i.e. rules or strategies used by the domain experts in deriving solutions; the extent of the relevant knowledge that underlies human solutions; and, any situations that are likely to affect the expert system.

In the conceptualisation stage, the key problem solving concepts and their relations are made explicit and the basis of the framework of the expert system is made. The problem solving knowledge and that which is used in justifying a solution must be identified and separated in addition to identifying the following:

- What data is given and what is inferred;
- Any partial hypotheses that are commonly used;
- The relationships between the domain related objects;
- The processes involved in the problem solution and the constraints on these processes; and,
- The information flow;

An outcome of the conceptualisation stage could be a hierarchy diagram showing causal and part-whole relations between objects and processes, set inclusions etc.

In the formalization stage, the concepts, rules and data realised from the conceptualisation stage are mapped into a formal framework using any of Entity-attribute grids, entity relationship diagrams, use case diagrams for data modelling and conceptual graphs.

Gaines and Shaw [14], describe a common framework which supports and illustrates the relations between various forms of knowledge gathered throughout the stages whilst identifying the concept of 'knowledge base' as a composite of informal, structured, formal and computational knowledge, all linked together through dependency relations providing mutual support in explanation and ongoing development of expert systems. The paradigms underlying the knowledge acquisition process are the use of hypertext and hypermedia tools to capture and structure informal knowledge, direct editing of knowledge in a semantic network, frame or rule representation indirect elicitation through repertory grids in which critical cases are described in terms of relevant attributes and inductive derivation of knowledge from data sets of varying quality.

Vlaanderen [13] identified three important aspects to be considered after formalisation, in specifying the contents of the data structures, the inference rules and the control strategies. These are:

- Linking of concepts to form hypotheses;
- Uncovering the underlying model of the processes used to generate solutions in the domain. Blythe et al [15] mention task models, interdependency models and knowledge acquisition scripts which enable knowledge acquisition tools to reason about the kind of knowledge they need to acquire from the user through interfaces, and how to add the knowledge to the existing knowledge base. Knowledge acquired through the user interfaces can be categorised as persistent data, object classes and choice constraints / preferences and are classified as computational knowledge in the framework described by Gaines and Shaw
- Understanding the characteristics of the data which helps to understand the structure of the problem space

In the implementation stage, the formalised knowledge is mapped into a representational framework which specifies the form of data gathered from previous stages. It is important to choose a representation which is compatible with the knowledge that the domain experts reveals whilst ensuring that the expert system works efficiently. At this stage a prototype knowledge acquisition system can be built.

The final stage involves testing, refining and debugging of the prototype knowledge acquisition system. In this stage, the prototype system is evaluated by a number of different challenging examples to find weak spots in the knowledge base and the inference structure. Incompleteness and inconsistency would mean that the knowledge base needs refining and reasoning errors will mean faults in the inference rules.

Over the years, researchers have proposed methods and techniques and even developed tools to aid the process of knowledge acquisition. These are further discussed in the following section.

2.5.2 KNOWLEDGE ACQUISITION METHODS, TECHNIQUES AND TOOLS

(I) Methods

Knowledge acquisition methods have been categorised into manual, and automatic methods [59][13][60]. The manual or conventional way of acquiring knowledge for expert systems development entails having the knowledge engineer repeat the cycle of interviewing and observing the domain expert, carrying out protocol analyses [55][34]. This process is usually extensive and time consuming and the domain expert has to be available and willing to reveal in-depth understanding of his field of expertise. Manual knowledge acquisition protocols are mostly unstructured and even where a knowledge engineer can go by certain guidelines during interviews, he can easily be distracted by details that the expert wants to mention [13][59]. In addition, knowledge bias can be induced if tacit information is excluded during interviews or the knowledge engineer lacks sufficient knowledge to comprehend or re-transcribe the expert's answers [58].

These problems are overcome in automated knowledge acquisition methods as they allow for construction of a prototype at a very early stage of the expert system development to sustain the expert's interest and also enable him to suggest improvements and extensions [13][55]. In addition, it allows the structuring of the knowledge acquisition in advance for the development of a better structured knowledge base.

A faster and more structured approach is provided by automatic knowledge acquisition methods employed in tools which support knowledge engineers (examples of which are SALT, ROGET and TERIESIAS [13][18][19]) and even experts who are non-programmers (e.g. EMeD [16][46]) to perform the Knowledge acquisition tasks of generating and refining knowledge more effectively. These tools can also be categorised based on their dependency or lack of it on certain expert systems. For instance, whilst SALT and TERIESIAS may use the same structure representation or inference as the expert systems for which they were developed i.e. VT and MYCIN respectively, EMeD which is based in the EXPECT framework is independent of any expert system and has its own method of acquiring knowledge. Automatic KA is also achieved in the use of machine-learning techniques to extract knowledge and generate rules. They require less or no participation by either knowledge engineers or domain experts. Therefore, they do not have the difficulties associated with human experts as there are with manual knowledge acquisition. Whilst rule induction is the most popular machine learning method, other machine learning methods which have been used to generate rules in machine learning include ID3, C4.5 and C5.1 [5][59].

It is important to mention that some form of manual acquisition, commonly interviewing, is involved in the process of developing automated knowledge acquisition tools. Automated knowledge acquisition approaches may be less laborious and simplify the acquisition of knowledge but they are not without limitations. Whilst machine learning could be very complex and requires a database of cases, limitations in other knowledge acquisition tools include possible errors in the generated rules, knowledge incompleteness and compatibility with selected problem solving strategies [13][58][65]. The latter however, is being addressed in recent publications [16] [24][28].

Manual knowledge acquisition methods were used in the course of this research as it was imperative to first acquire background domain knowledge and an in-depth understanding of the domain problem.

(II) Techniques

A number of knowledge acquisition techniques have been developed to aid and ensure structured manual acquisition of knowledge from an expert. Milton and Tecuci [31][32][35][36] describes the following techniques used for acquiring, analyzing and modeling knowledge.

Protocol generation technique –Usually applicable during the initial stage of the knowledge elicitation process, this technique produces a record (preferably electronic i.e. audio or video) of behaviors or protocol within a problem domain. Transcripts are later derived from this record. Records of behaviors are obtained during interviews sessions (unstructured, semi-structured and structured) or using techniques such as:

- **Reporting techniques**- where an expert provides a running commentary of their thought processes as they solve a problem; and,
- **Observational techniques**- where the knowledge engineer makes notes as the experts perform their daily activities, to acquire knowledge.

Protocol analysis techniques - Acting as a bridge between the use of protocol-based techniques and knowledge modeling techniques, protocol analysis techniques are used to identify basic knowledge objects or categories of fundamental knowledge by highlighting concepts, attributes, values, tasks and relationships within a protocol, usually transcripts of interviews or other text-based information. Project requirements could mean that more detailed categories are used for the identification of objects. An example given by Milton [35][36] is that of a transcript concerning the task of diagnosis being analysed using categories such as symptoms, hypotheses and error reporting techniques. Figure 2.2 shows Milton's description of the typologies adopted by knowledge engineers in analysing transcripts and constructing knowledge models.

Hierarchy-generation techniques- These techniques are synonymous with laddering techniques which are used to create review and modify knowledge in a hierarchical manner as with taxonomies, goal trees and decision networks. Various types of ladders include:

- ***Concept ladder***- Used to categorize concepts and their sub-types using the "*is a"* type relationship e.g. an apple *is a* fruit. Knowledge in almost all domains can be represented using concept ladders.
- ***Composition ladder***- Useful in understanding complex entities such as machines, organizations and documents, a composition ladder represents a knowledge object with a reflection of the constituent parts that make it up. All relationships in the ladder are of the "*has part"* or "*part-of"* relationship, e.g. a flange is *part of* a sub-assembly.
- ***Decision ladder***- A useful way of representing detailed process knowledge. A decision ladder is used to represent the possible choices or options available in making a decision whilst showing the rationale behind each cause of action as well as its advantages and disadvantages.
- ***Attribute ladder***- An effective way of representing knowledge of all the properties that can be associated with concepts in a domain. It shows attributes and their corresponding values (usually texts as opposed to numerical values) as sub-nodes. For example, the attribute *colour* would have as sub-nodes those colours appropriate in the domain as values, e.g. *red*, *blue*, *and green*.
- ***Process ladder*** – Used to represent processes which are made up of tasks and activities in relation to the sub-processes of which they are composed. Similar to the composition ladder, the relationships in a process ladder are of the "*part of" type*.

- **Declarative and Procedural Knowledge** — ***Knowledge on facts and how to do things. Often referred to as object knowledge and process or task knowledge respectively***
- **Tacit and Explicit Knowledge** — ***Tacit knowledge isn't easily expressed whilst Explicit knowledge is easily expressed***
- **Generic and Specific Knowledge** — ***Generic knowledge applies across many situations whilst specific knowledge applies to one or a few situations***
- **Knowledge Objects**
  - ***Concepts –*** Things that make up a domain e.g. physical objects, ideas, people etc. Each can be described by its attributes / properties as well as relation to other concepts in terms of a hierarchy.
  - ***Instances -*** An instantiated class. For example, "Hibiscus" is an instance of the concept/class "Flower". Instances only have the attributes of their class (including inherited attributes). They may override anv or all of the default values.
  - ***Processes -*** Processes (tasks, activities) are sets of actions performed to satisfy a goal or set of objectives. Processes are described using other knowledge objects, such as inputs, outputs, resources, roles and decision points.
  - ***Attributes / Values-***They describe the properties of other knowledge objects. **Attributes** are the generic properties, qualities or features belonging to a class of concepts whilst **Values** are the specific qualities of a concept
  - ***Rules -*** statements of the form "IF... THEN..." where the statement following the IF is known as the antecedent or the LHS (the condition to be satisfied) and the statement after the THEN is the action or RHS which will be executed if the LHS is satisfied.
  - ***Relationships -*** They represent the way knowledge objects (such as concepts and tasks) are related to one another. They are often represented as arrows on diagrams. Examples include the use of "*is a*" to show classification, "*part of*" to show composition, and those used in various

Figure 2.2: Typologies for analysing text and constructing knowledge models

Matrix-based techniques – These techniques involve the construction of matrices/grids [35][36], a type of tabular representation that comprises a two-dimensional representation of concept attributes or relationships using elements such as symbols, colour codes, numbers or text. Examples of the use of matrices/ grids as suggested by Milton include the representation of:

- Problems encountered against possible solutions as in a problem-solution matrix
- Knowledge objects against associating properties as in an attribute matrix
- Knowledge objects in relation to other knowledge objects as in a relationship matrix.

An important type of matrix-based technique is the use of repertory grid technique to elicit, rate, analyze and categorize the properties of concepts.

Sorting techniques are a well-known method for capturing the way experts compare and order concepts, and can lead to the revelation of knowledge about classes, properties and priorities.

With the card sorting technique which is the simplest form of the sorting techniques, a number of cards are used each displaying the name of a concept. These cards are repeatedly sorted into piles by the expert - the cards in each pile representing concepts that can be related. Where domain concepts can not be easily described using simple text, sorting objects or photographs can be used to replace cards.

Triadic elicitation or the ‘Three Card Trick’ technique is often used in conjunction with sorting techniques as a way of eliciting tacit attributes from the expert. The expert is prompted to generate new attributes by means of elicitation of the similarities and differences between three randomly chosen concepts.

Diagram-based techniques - Particularly important in capturing the "what, how, when, who and why" of tasks and events, these techniques facilitate the generation and use of network diagrams such as concept maps, state transition networks, event diagrams and process maps [31][32]. As experiments have demonstrated that people can better understand and relate to networks as compared to logic, the representation of knowledge using network diagrams ensures efficiency in the validation process as is the case with laddering techniques. Various types of knowledge have been elicited with the use of concept maps. However, in the area of knowledge acquisition for object oriented software development, the use of network diagrams has become a common technique. [6]UML (Unified Modeling Language), combines the use of concept maps and frames for the representation of object knowledge; state transition networks for dynamic modeling; and, process maps for functional modeling.

In figure 2.3 below, Milton [31][32] presents the various techniques described above in relation to the types of knowledge they are mainly aimed at eliciting. The vertical axis on the figure represents the dimension from concept knowledge to process knowledge, and the horizontal axis represents the dimension from explicit knowledge to tacit knowledge.

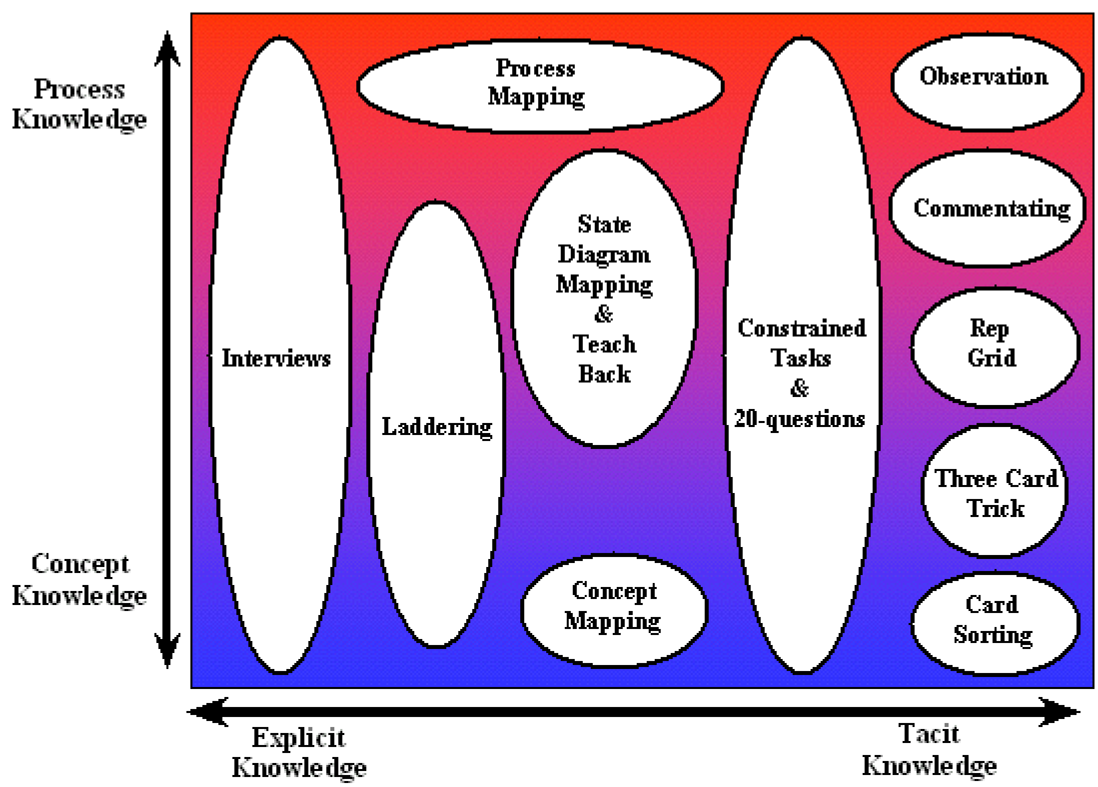


Figure 2.3: Comparison of KA Techniques

Following the description of the various techniques for acquiring, analyzing and modelling domain knowledge, Milton [31][32] goes on to illustrate how and when the techniques can be applied by describing a set of procedures which he summarises as:

- Conducting an initial informal interview with the expert in order to (a) scope what knowledge is to be acquired; (b) determine what purpose the knowledge is to be put; (c) gain some understanding of key terminology; and, (d) build a rapport with the expert;
- Transcribing the initial interview and analysing the resulting protocol, creating a concept ladder of the resulting knowledge to provide a broad representation of the knowledge in the domain and using the ladder to produce a set of questions which cover the crucial issues across the domain whilst serving the goals of the knowledge acquisition project.
- Conducting a more structured interview with the expert using the prepared questions to provide structure and focus; and, analysing the resulting protocol for the knowledge types present. These would usually be concepts, attributes, values, relationships, tasks and rules.
- Representing these knowledge elements using the most appropriate knowledge models, e.g. ladders, grids, network diagrams, hypertext, etc.
- Using the resulting knowledge models and structured text with techniques such as laddering, think aloud problem-solving and repertory grid to allow the expert to modify and expand on the knowledge already captured.
- Repeating the analysis, model building and acquisition sessions until the expert and knowledge engineer agree that the goals of the project have been realised.
- Validating the knowledge acquired with other experts, and making modifications where necessary.

(III) Tools

To address the knowledge acquisition bottleneck, researchers have investigated the development of tools to facilitate the knowledge acquisition process. Having stated that the construction of knowledge bases is easier and quicker with Knowledge acquisition tools / software, Milton [31][32] added that not only is the knowledge acquisition process made more efficient as a result of the tools' capability to represent knowledge in multiple ways, but they also improve the knowledge validation process ensuring that the knowledge acquired is accurate, complete, consistent and relevant.

Manual knowledge acquisition methods can be expedited through the use of tools which automate existing knowledge elicitation and domain modelling techniques [33][35]. An example of such application is PCPACK [36] which is a comprehensive suite of tools used to create, inspect and edit knowledge bases developed using XML technology. The versatility provided by the tools' various knowledge representation and capture techniques makes knowledge acquisition, storage and modelling more efficient and less prone to errors.

An increase in the complexity of knowledge acquisition problems has however led to the integration of a wide variety of different tools, techniques and methodologies into knowledge acquisition environments and architectures [14] [15]. Unlike the traditional knowledge elicitation techniques which are restricted by static concepts, the architecture proposed by Gaines and Shaw [14] provides an open framework which is the result of a synthesis of well founded and widely used techniques/approaches to knowledge acquisition. The tools and techniques designed to support the knowledge acquisition process in the Gaines & Shaw architecture is underlined by a number of concepts:

The concept of capturing and structuring informal knowledge through the use of hypertext and hypermedia tool involves the analysis of document text for associative clusters which may be grouped to indicate significant concepts to be refined by a domain expert. Using semantic networks, frame or rule representation domain experts are able to edit knowledge directly, through a graphic editing environment. They are able to interact with an underlying knowledge representation to elicit distinctions and relationships between domain entities. These elicitations subsequently make up procedural, decision making rules in the domain. Where experts cannot enter a knowledge structure directly, the concept of indirect elicitation through the use of repertory grids is applied. Experts are prompted for distinctions relevant to the problem domain as well as critical cases that exhibit significant occurrences in the domain. Where experts are not able to enter critical cases directly, the concept of Inductive derivation of knowledge from data sets of varying quality is applied. Empirical induction techniques may be used to derive knowledge structures underlying the decisions made in expertise case histories which are described in terms of relevant attributes and correct decisions

2.5.3 KNOWLEDGE REPRESENTATION

Knowledge representation is important in the expert systems development as it affects the development, efficiency, speed and maintenance of an expert system [65][16]. The knowledge acquisition process should fit in with knowledge representation paradigm and the knowledge representation should also be adequate for the problem. Knowledge representation can be described in contexts of both the inference and problem solving strategies of a proposed expert system. In the context of inference strategy, a number of different knowledge representation techniques have been devised which include production rules, semantic nets, frames, scripts, logic and conceptual graphs [1][5]. However, production rules are more commonly used as they possess the advantage of more efficient and modular storage capability ease in building explanation facilities and, similarity to the human cognitive process [1][5]. In rule based systems, production rules are usually expressed in IF – THEN format as:

Rule: Rolling Form

IF

Inner Diameter of Bellows is > 324mm (antecedent)

THEN

Use Solar machine (conditional element)

The amount of knowledge about rules in an expert system is an important factor as the lack of it causes difficulty in understanding a rule without reference to others whereas an excess of it causes lack of structure in the case of large numbers of rules. An expert system becomes difficult to modify in the case of the latter. To avoid any of these occurrences, ontology of the problem domain should be formally constructed before an expert system is developed, to identify any potential inconsistencies and inadequacies [34] [53].

The knowledge acquired from experts or other sources must be expressed in the knowledge base in a manner compatible with the problem solving strategies of an expert system. This concept is fundamental in Swartout & Gil's "Role limiting approach" [24] to explicit representation of knowledge roles which is a key factor in the development of automated knowledge acquisition tools. This approach was based on the observation that the role that a particular kind of knowledge plays in problem solving strongly constrains how that knowledge should be expressed.

2.6 CRITICAL REVIEW

The literature survey has provided a wide range of benefits to this research. It has confirmed the relevance and potential benefits of the knowledge based systems in the area of design and cost estimation in make to order manufacturing. Techniques have been explored by previous researchers with the aim of achieving competitive cost and the improving manufacturability of products whilst maintaining their functional and performance objectives. Where detailed information on the attributes of new products is provided, design and cost estimation can be achieved by relating these variables to historic cases. As this detailed information is often not available at quotation stages, the methodical derivation of design and cost estimation based on domain experts' knowledge or past experience eliminates potential restriction. This technique is demonstrated in decision support systems which are basically tools that appropriately represent the domain expert experience and allow for the making of improved judgments levels at various levels by users.

Of the various approaches to the development of decision support systems, the expert system approach provides the strongest advantage of providing quicker inference and more consistent and accurate results through its imitation of human expertise derived through rule based logical reasoning.

During the period in which the literature survey was conducted, the management team at TES Ltd decided on the implementation of Configur8or - a browser based configuration software; to facilitate configuration and cost estimation of bellows from historic data, at the enquiry to quotation stage of their order process. *Configur8or* can be described as a vague expert system shell in that it possesses the following functionalities which to some extent can be compared to that of a real expert system shell:

- A user interface;
- A built in forward chaining inference method with support for rule based representation;
- Knowledge acquisition tool for the development and maintenance of a knowledge base i.e. domain and problem solving knowledge; and,
- Interfaces to other software systems i.e. spreadsheets and databases;

Like the knowledge acquisition tools reviewed, the knowledge acquisition tool within *Configur8or* is impacted by the needs of a knowledge engineer who would often possess substantial knowledge about programming. Consequently, an end user without any programming knowledge is not able to easily update or make any changes to the knowledge base through interaction with this knowledge acquisition tool.

With this point of view, the rationale for this research was to develop a semi-automatic expert system, a wizard in effect which will utilise some of the techniques and methods reviewed so far, in guiding non-programmers through information manipulation and relevant data input into the design and cost estimation models within configur8or in the appropriate format or syntax.

The Design for manufacture (DFM) approach described in the literature review has been successfully implemented in a number of manufacturing design expert systems. This approach is however not applicable in the context of this research because the expert system wizard is not intended to output product designs, a selection of material and manufacturing processes, and subsequent cost estimation of products. Rather it is intended to input data objects / parameters into extant design and cost estimation models within *configur8or*. This objective would be best achieved through the implementation of a decision support system to provide guidance to its users based on an explicit representation of expert knowledge which is a combination of the domain knowledge and an understanding of *configur8or*'s problem solving strategies.

In addition to the expert knowledge which forms the knowledge base of the intended expert system wizard, the system would possess a user interface; an explanation facility for the description of expert system reasoning; an inference engine with a forward chaining rule based reasoning technique and a working memory i.e. a global database of facts used by the rules.

A significant part of the expert knowledge is to be manually acquired from the company's design and cost estimation experts as well as the expert trainers of the *configur8or* software. To overcome the knowledge acquisition bottleneck discussed in the review and achieve appropriate representation of knowledge acquired, the structured techniques and procedure described in section 2.5.2 will be implemented. The implementation of these techniques and procedure are discussed in detail in later chapters of this thesis.

CHAPTER 3 - METHODOLOGY

3.1 RESEARCH METHODOLOGY

Generally, research has been classified into two classes namely: pure or basic research and applied research [81] [82]. This classification has been interpreted to imply that while basic research supplies or improves original theories, applied research seeks to test out these accepted theories and principles by applying them in solving real world problems. However, other classes of research have been identified. Estelle and Pugh [81] criticise the rigidity of the traditional classifications and consider an alternative classification into exploratory, testing-out and problem solving research methodologies. Using any of the classifications mentioned, quantitative or qualitative research methods are applicable to any type of research methodology. Analyses preceding this research revealed certain shortcomings within the business processes at TES ltd and the need for these shortcomings to be resolved. Based on these analyses, the problem domain has been identified and the problems have been pre-defined and formulated. It was therefore most appropriate to employ a problem-solving research approach in discovering the methods of solution. Because the research was not aimed at tackling issues about which little is known or trying to discover limits of previously proposed theories or generalizations, the application of the exploratory or testing out approaches would be inappropriate. The strength of the problem solving approach employed lies in the application of an open system of thought in reviewing a wide variety of well established theories and techniques as well as the works of previous researchers in the field. However, the weakness of this approach is drawn from the fact that solving real world problems often involve a variety of theories and techniques from more than one discipline. E.g. knowledge based systems, systems analysis and soft systems methodology, systems development lifecycle.

Using some of the techniques described in section 2.5.2, a qualitative method was employed in researching the existing business and information processes within TES ltd with the aim of identifying the drawbacks within the processes; understanding and also justifying the relevance of this research in mitigating the drawbacks. An analysis of the business process and information systems was carried out and this included observations of the processes, interviewing and discussing relevant factors with key individuals in the manufacturing, sales and design and IT departments.

The findings from the analysis undertaken indicated that the company's dependency on a few key individuals (design engineers and cost estimators) to process enquiries into quotations, was the major cause of the decrease in the number of orders accepted by customers resulting from the slow response to enquiries from customers. In addition, the use of obsolete and possibly incorrect data in the cost estimation process could result in financial losses and consequently act as a barrier to the company's growth if for example, the labour time quoted and costed for a job happened to be less than the actual duration of the job. More importantly, as cost estimation data is envisaged to be an important input to the proposed knowledge based system, the purpose of implementing an effective system would be defeated. These findings helped to justify the need for knowledge based system and also identify the need for the implementation of a shop floor data capture system to capture accurate operational information relating to product manufacturing.

There were numerous hardware and software considerations relating to the implementation of the data capture system so decisions had to be made on the appropriate hardware / architecture to be implemented (taking into consideration factors like the shop floor environment where the data capture terminals are to be installed) and whether to purchase a commercial off-the-shelf solution or develop bespoke software.

The information gathering process involved carrying out a literature review on the application of bespoke and commercial software in organisations whilst inviting suppliers / vendors to suggest potential options for a solution and to provide implementation costs. Part of the decision making process involved arranged demonstrations of proposed software by the suppliers and visits to other sites where similar software and hardware are in use.

Based on the information gathered, recommendations were made for the implementation of thick client-server architecture with the use of touch screen capture terminals, bar-code scanners and a bespoke development of a browser –based shop floor data capture system using open source technology. The strengths, risks, process changes (introduction of bar-coded works orders for example) and resources required to implement the recommendations were highlighted in a commercially confidential report which was accepted by the company directors.

3.2 KNOWLEDGE ACQUISITION METHODOLOGY

The knowledge acquisition process for expert systems development requires elicitation of expertise knowledge from at least one human expert. However, the literature survey revealed that a single expert's opinion involves some form of uncertainty therefore interviewing and observing multiple experts would give broader results and verify the completeness and accuracy of expertise knowledge acquired [13][71]. Having mentioned the benefits of the involvement of multiple experts, there are also limits drawn from the fact that different experts may solve problems in different ways or even reach different conclusions or solutions to a particular problem [5][55]. Varying expert opinions could hinder the knowledge engineering process as conflict and incompatibilities may be created within the knowledge base [5][55][71]. For instance, whilst acquiring knowledge on the derivation of parametric properties of parts, two experts were interviewed. Whilst Expert A would calculate the length of a bellow sleeve as a certain percentage of the tube length, Expert B would arbitrarily use a length lower than the tube length albeit within a sensible range based on his own discretion. As this variance is not dependent on any context or process, it becomes impossible to match them against any rules and would therefore result in overlapping knowledge which cannot be meaningfully represented. Some of the methods suggested by other researchers to manage knowledge acquisition from multiple experts include the consultation of experts on an individual basis, the designation of a primary expert and integration of multiple opinions through brainstorming, consensus decision taking and nominal group forming techniques [55] [71]. Selection of a primary expert with superior knowledge may tend to be difficult and lack group creativity whilst knowledge acquired from groups may be inferior to that of an individual expert due to personal or professional conflicts, politics or varying mental models. [71] These limitations can be eliminated by Individual expert consultation. Although this technique may also lack group creativity, it allows the knowledge engineer to integrate different forms of relevant knowledge and lines of reasoning to achieve a reliable knowledge base.

Peter Checkland's Soft Systems Methodology (hitherto referred to as SSM) provides an organized way of resolving problematic social situations caused by the existence of different and conflicting world view so that appropriate action can be taken to bring about improvement. [75]

In the field of knowledge acquisition, researchers have adapted stages of the soft systems methodology in shaping and simplifying interventions from the various world views of subject matter experts to abstract relevant conceptual models from the subject matter [71][72][73]. Besides the obvious benefits of supporting the process to knowledge elicitation and abstraction of conceptual models, the implementation of the soft systems methodology promotes transparency within the problem domain by revealing other objectives which otherwise would not have been apparent. Having mentioned these, it is important to add that they can only be completely achieved if the SSM processes are not strongly constrained by organizational or other external structures. In this case, the SSM would only proffer a temporary solution without resolving the root cause of the problem situation [76].

The soft systems approach is a seven-stage process which can be further group into the 4 main activities [72][75][76]. The application of these activities is discussed in the context of a case study in the cost estimation of product assembly parts later on in this thesis.

3.3 EXPERT SYSTEM DEVELOPMENT METHODOLOGY

Rapid prototyping using special purpose hardware and software such as LISP machines and expert system shells was once the prevailing paradigm for building knowledge based systems. However, many developers have realised that a structured development approach is just as necessary in the KBS development as it is in conventional software development projects. Guus et al [4] describe the CommonKADS methodology as one with the aim of filling the need for this structured approach by constructing a set of engineering models of problem solving behaviour which takes into consideration, the application of the KBS as well as the organisation in which it will be implemented. The models together explore the software development lifecycle (one very useful concept which views expert system development as a series of stages from initial concept to system evaluation and maintenance) and project management aspects of KBS development.

The CommonKADS methodology provides four development models; the organization, task, agent and communication models, specifically aimed at modelling the organizational environment of a KBS, and a central model; the expertise model, geared at modelling the problem solving behaviour of an agent in terms of knowledge that is applied to perform a certain task [4][34].
This methodology also proposes a project management activity model which interacts with development work through model states attached to the development models. The management process is executed in a cyclic, risk-driven manner. At the start of a management cycle, objectives for the cycle are defined and associated risks are identified. Within the cycle, a set of model states is realised from these objectives and risks. These model states are then projected into development activities that result into elements of the development models. At the end of each development cycle, a check is performed on the quality of the results through reviews based on the overall objectives and tasks. A CommonKADS project may consist of many cycles depending on the identification of new objectives and risks and all

development models do not have to be fully developed in a project. Only those having a bearing on the objectives and risks are selected [4].
Giarratano and Riley [5] describe an Expert System Development Life Cycle Methodology which like the CommonKADS methodology, explores the software development lifecycle and to a lesser extent, project management or planning activities. However, unlike the cyclic CommonKADS project management model which could vary in varying scenarios, the planning aspect of this methodology represents a more stable and more reliable approach which involves feasibility assessment; resource management; task phasing schedules; functional layouts; and high level requirements [5].

Many KBS failures have resulted from the lack of concern for organizational factors. Yet many system development methods focus on the technical aspects only and provide little support for the analysis of the organizational elements that determine the success or failure. Both methodologies described above, support the analysis of organisational elements. The Expert System Development Life Cycle Methodology was used for the purpose of this research.

CHAPTER 4 - KNOWLEDGE ACQUISITION TECHNIQUE & EXPERT SYSTEM IMPLEMENTATION

4.1 KNOWLEDGE ACQUISITION TECHNIQUE

As discussed in chapter 2, the first step towards acquiring knowledge for expert systems development involves the specification of the objective of the proposed expert system and identification of the scope of knowledge to be acquired. To that end a series of consultations were held with key stakeholders to gather opinions and ideas on how to facilitate the use of *configur8or* within the company and the soft systems methodology discussed in chapter 2 was applied in resolving conflicting ideas to achieve a consensus on an effective and efficient solution. Following from this agreement, the objective of the proposed expert system was to provide user interfaces that will guide expert users who are non-programmers in developing configurable product models within *configur8or.* Based on this objective, the scope of knowledge to be acquired for the development of the expert system ranges from the knowledge required by *configur8or* as well as the underlying structures that control the presentation and use of this knowledge, to knowledge about the design and cost estimation processes (i.e. vocabulary, statements/facts, reasoning and rules, data) and how these fit into the underlying structures within *configur8or.*

Manual knowledge acquisition techniques were employed in acquiring domain knowledge for this research. Whilst basic knowledge i.e. data and vocabulary were elicited from databases, spreadsheets and reference materials. A significant amount of knowledge about facts, reasoning and rules where acquired directly from respective subject matter experts. The knowledge acquisition process involved protocol generation and analysis.

(I) PROTOCOL GENERATION

Informal conversations were held with experts to gather basic knowledge about the domain. Thereafter more structured interviews were conducted to gain in-depth understanding of the structure of the knowledge to be acquired and any rules or strategies used by the domain experts in deriving solutions.

The interviews involved the use of protocol generation techniques of observation and reporting mentioned in Chapter 2.5.2. Notes were taken as the experts were observed performing their daily activities and audio reports were gathered from the running commentary of their thought processes as they solved problems. Transcripts were later on derived from these reports as well as other sources of relevant knowledge which include spreadsheets containing estimation data and other reference materials.

(II) PROTOCOL ANALYSIS

The transcripts, notes and other documents derived from the protocol generation were analysed by highlighting concepts, attributes, values, tasks and rules which represent knowledge objects. Examples of the concepts identified from the protocol analysis were: Final assemblies, sub-assemblies, component parts, materials and manufacturing operations. These concepts were modelled using matrix and hierarchy based techniques. The taxonomy generated from the use of hierarchy based techniques such as *concept maps* and *composition trees* provided better understanding of the knowledge objects and the matrix based techniques was used in mapping out the relationships between the concepts. Below are examples of how these techniques were used.

-Hierarchy-generation techniques:

The final assembly of a bellow / expansion joint comprises of various component parts and or sub-assemblies which are also made up of a selection of the various component parts. These parts are either made from a variety of raw materials or are bought in as factored or pre-manufactured materials. An understanding of the *final assembly* object is achieved using a ladder technique. Figure 4.1 below shows a concept map which is a representation of how the concepts relate together in a design process.

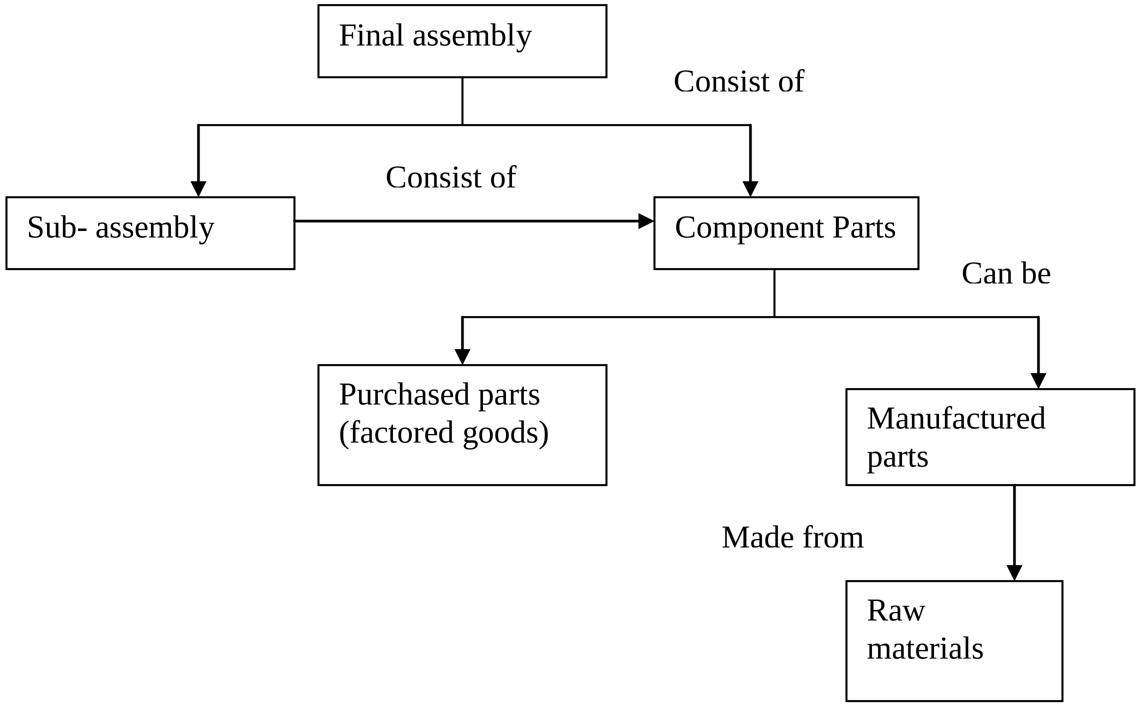


Figure 4.1 Example of a concept map

Some component parts have sub-types. For instance, a ‘flange part’ could be rectangular, round, oval etc in shape whilst a ‘cuff part’ could be expanded or with holes. Using a similar structure as the concept map, an understanding of these component parts is achieved by representing them as objects within a composition tree in relation to their various sub-types as shown in Figure 4.2 below and Figure 4.6 in Appendix A

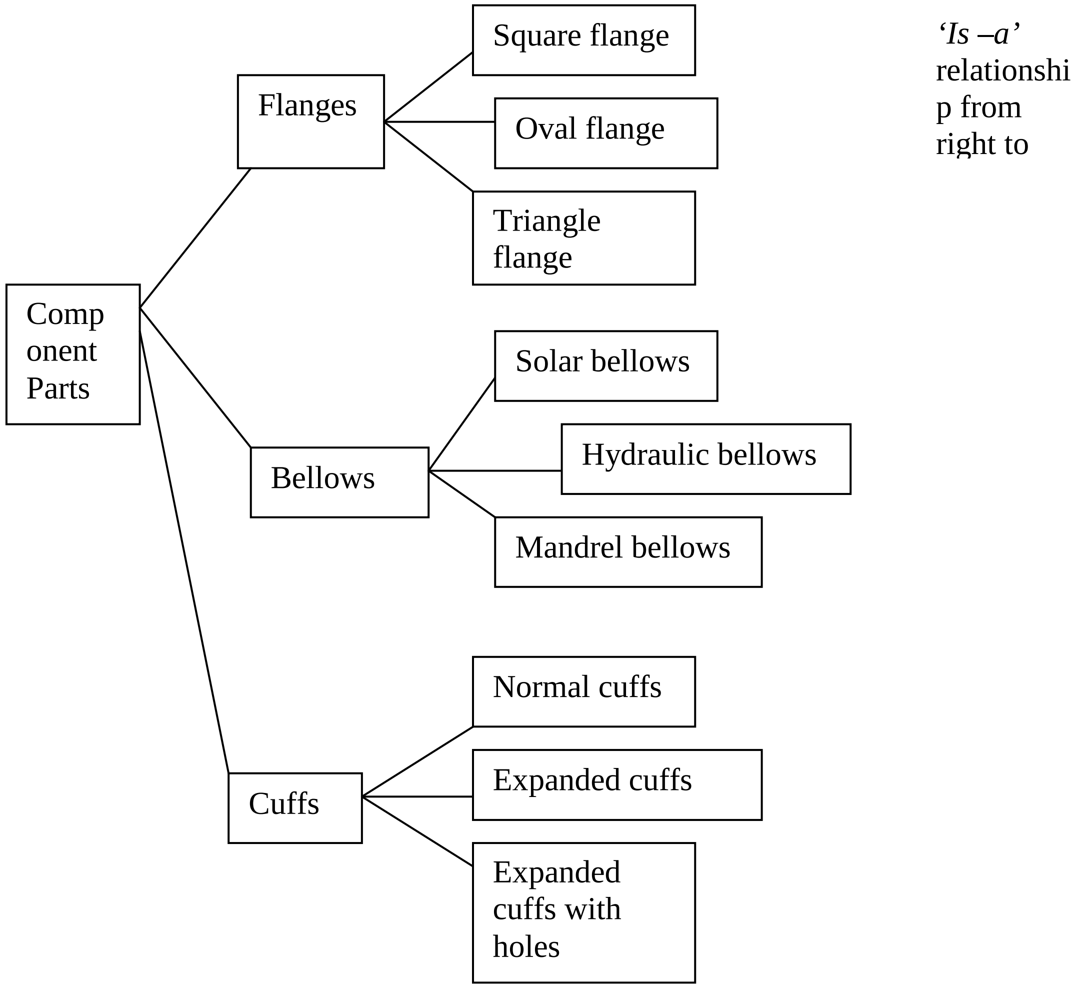


Figure 4.2 Composition tree

-Matrix-based techniques:

The estimated cost of a configured / finished assembly is the sum of the material cost of its constituent component parts and the cost of labour employed to manufacture the finished assembly. The manufacturing process involves a number of operations which are carried out on individual components or a sub-assembly of components. The basis of a cost model that would generate accurate estimates would be a correlation between component parts and materials as well as manufacturing operations. As historic labour estimates on the various operations are currently held within separate tables, it also became imperative to map every operation to its corresponding table. These mappings would ensure more organised searches through relevant knowledge bases/ rule bases. The relationship between component parts and their corresponding manufacturing operations is represented in a 2-dimensional matrix / grid shown in Figure 4.3 in Appendix A.

The knowledge elicited was validated by the experts. Interviews, analysis and modelling were repeated until all relevant knowledge had been acquired. The structured knowledge models created allowed for easier validation and modification of the knowledge acquired ensuring that it is accurate, complete and consistent. These knowledge models act as the basis for the creation of the expert systems knowledge base and also guide the definition and classification of the rules relating to each concept that was identified.

As in real world scenarios where people have different ideas and opinions, the knowledge elicitation processes of interviewing and observation revealed that experts would sometimes have conflicting but purposeful opinions about solving the same problem or achieving the same outcome. The soft systems methodology earlier discussed in chapter 2 was employed in resolving such conflicts. One of such scenarios occurred in the process of eliciting knowledge about cost estimation of product assembly parts. The SSM activities were applied as follows:

(I) Identification of a problematic situation – The problematic situation in this case isn't the problem domain which in the context of this research is the design and cost estimation process but rather the acquisition of knowledge about this process. The identification process was carried out in 3 stages. The first stage involves providing a brief description of the problem situation. In the course of this research for instance, a knowledge acquisition problem was encountered in eliciting knowledge on the estimation of labour cost for the ***sleeve*** component part of an assembly. The actual length of component part is usually not provided on the assembly design sketch issued out for estimation at the enquiry stage of the order process. This parameter is however needed to derive the estimated time for carrying out the required operations – a function of the cost of manufacturing this part. Two cost estimation experts where interviewed who had different views about the size of the part in reference to the length of bellows tube. Although both agreed that the length of sleeve would be less than the length of the bellows tube**,** they had different discretions about what the differences should be. For instance, given a bellows tube length of 321 mm, one estimator would estimate the guillotine operation on a sleeve at 2.2 minutes based on tube length of 250mm from the table 4.1 below whilst the other would estimate at 3.0 minutes based on the 300mm tube length. This could cause inconsistencies in the estimated cost of manufacturing a particular product at different times even when other factors such as labour rate, cost price of materials remain constant.

Cuff, Ring, Sleeve, Center Tube & Spinning								Length of Tube in mm.						
				Batch										
				Set up	25	30	115	225	250	300	350	400	450	500
Guillotine				5	0.5	1.0	1.0	1.4	2.2	3.0	3.5	3.5	4.0	4.0

Table 4.1*:* Labour estimation for guillotine operation

The second stage of the problem identification activity involves the use of Rich pictures with the aim of expressing the problem situation. The rich picture informally captures the main entities, structures and view points in the situation, the processes being undertaken, current and potential issues. [75]. Figure 4.4 shows a rich picture representation of the problem situation in question.

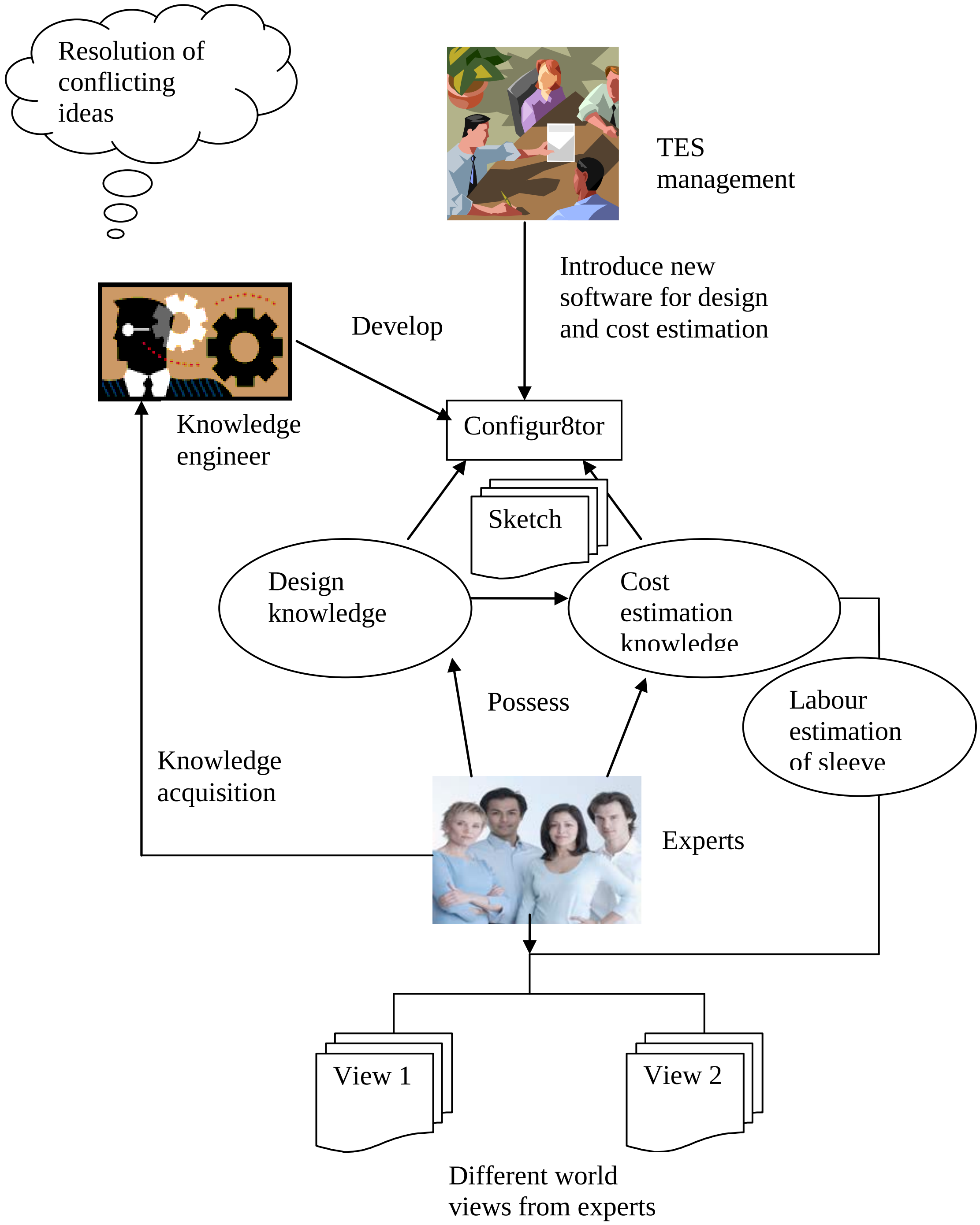


Figure 4.4: Rich picture representation of the problem situation

The third stage involves analyses of the Roles, Social system and political system of the problem situation.

In the Role analysis, 3 key players are identified – ‘the client’ who caused the problem interventions to happen, ‘the practitioner’ who conducts an investigation using the SSM, and ‘the issue owner(s)’ who are affected by the situation or outcome of the effort to improve the problem situation. In the context of this research, the client and practitioner is the same person i.e. the knowledge engineer and the issue owners are the estimators.

The social system analysis identifies the roles, norms and values that shape the situation. For instance, the design experts can provide advice on the variance between the length of tube and length of sleeve (the problem situation). However, the norm is such that the precise length of sleeve is provided along with other detailed parametric values on a full set drawing of a product which is only drawn up when the customer enquiry has been converted to an order for production.

The political system analysis identifies the effects of politics or power on the problem situation. There aren’t any effects of politics or power on the problem situation in questions. However, one of the estimators manages the sales team and would often be consulted by the other estimator.

(II) Creating purposeful activity models: This activity involves the description of transformation processes from the perspectives of the various world views that would achieve the desired interventions. This description is known as the ‘Root Definition’ of and leads to the activity system(s) to be modeled. Soft Systems Methodology provides a ‘PQR’ formula which is useful for shaping the root definitions. The formula answers the What, How, and Why Questions in the manner: do P, by Q, in order to help achieve R. The CATWOE analysis which could also be used as guidance for developing the ‘*Rich Picture*’ of the problem situation enriches the root definitions. CATWOE is an acronym for:

- Customers (and other stakeholders), i.e. people who are affected by the transformation

- Actors, i.e. the people who perform the activities in the transformation

- Transformation process i.e. stating what is changed and to what
- World-view or perspective from which the transformation is meaningful
- Owner(s), i.e. the person or people who control the transformation
- Environmental / external factors, i.e. anything that constrains the transformation.

Both cost estimators suggested transformation processes based on their different perspectives on the problem situation i.e. Resolving variances between length of tube and length of sleeve in achieving consistency in the time estimated for carrying out a guillotine operation on a sleeve part**.**

World view 1 – The first estimator suggested that the variance should be set by the design engineers to achieve consistent estimations

World view 2 – The second estimator suggested the norm should be adjusted such that a precise value for the length of sleeve is assigned by the design engineers at the enquiry stage to eliminate the need for setting variance and consequently achieve consistent estimations.

These transformation processes / world views were monitored for feasibility and suitability by ensuring efficiency and effectiveness. Figure 4.5 shows a purposeful activity model for one of the world views.

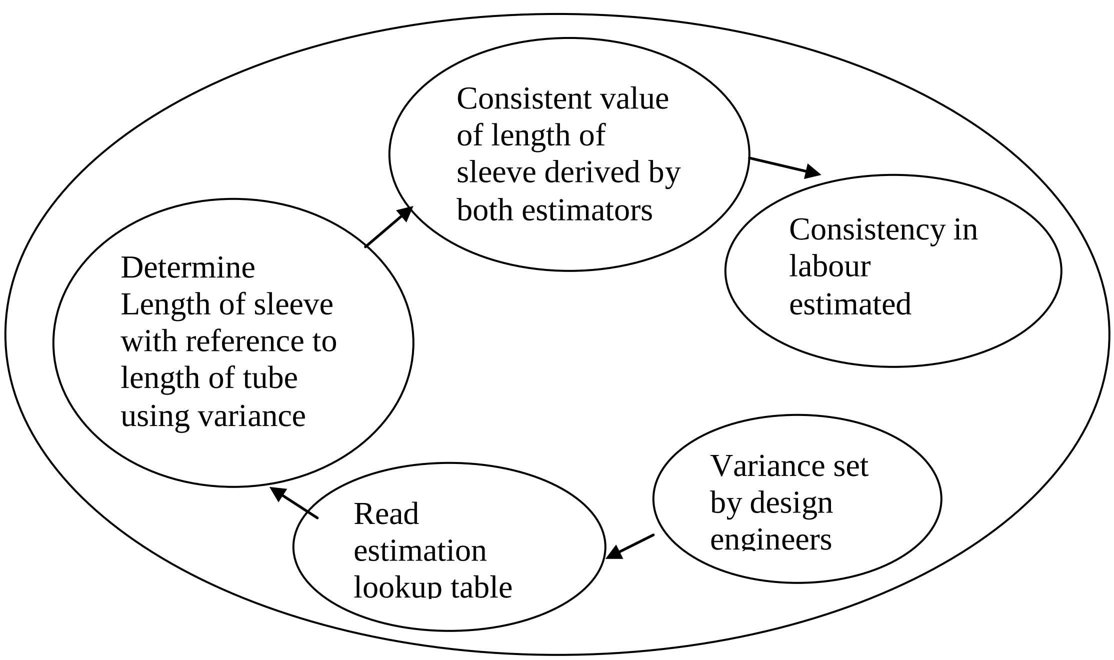


Figure 4.5: Purposeful activity model

(III) Debating the problem situation – This activity involves analysing the purposeful activity models from the different world views against the real world situation as earlier defined. The aim is to ensure that the suggested changes are achievable and desirable within the organisation. As the activity models are still in various perspectives at this stage, dialogues or debates are undertaken to achieve an acceptable consensus between both world views by exploring assumptions to achieve a change model without compromising on any relevant perspective. This dialogue may result in modifications to the change models and the 'Rich Picture' definition of the real world situation.

In the context of the subject matter under analysis, a consensus was reached between both estimators and the design engineers that it would be more appropriate for the design engineers to set a variance which is to be highlighted on the sketches issued out for estimation. This change can hardly be represented on the 'Rich Picture' as it requires a change in the content of an entity which already exits.

(IV) Defining and taking action – The final activity involves the development of plan for the agreed change model and taking action(s) to implement it. A structured e.g. project management or unstructured approach could be used depending on the change model. Where a project is involved, the effects of the project should be closely monitored in order to identify and manage any other problem situation that may arise. Implementing the change model in the context of the subject matter involved a consensual approach based on agreements between the cost estimators and design experts.

4.2 EXPERT SYSTEM IMPLEMENTATION

TES ltd implemented *Configur8or*, a browser based configuration software, to facilitate the design configuration and cost estimation of bellows at the enquiry to quotation stage of their order process. *Configur8or acts* an expert system shell in that it possesses the following tools and capabilities with which a knowledge engineer can develop a knowledge base:

- A user interface;
- A built in forward chaining inference method with support for rule based representation;
- Knowledge acquisition tool for the development and maintenance of a knowledge base i.e. domain and problem solving knowledge; and,
- Interfaces to other software systems i.e. spreadsheets and databases;

Configur8or's knowledge acquisition tool includes matrices; rule editors; query editors; and, templates for uploading graphics. By entering relevant knowledge into these structures, the knowledge engineer creates a configuration interface for each configurable product. This interface would display default dialogue questions and subsequent ones based on answers provided by an end user. A graphical model of the product is also displayed which is then manipulated based on answers from the dialogue, to produce a configured product. Changes in product model are simultaneously reflected as user interaction takes place

As with most knowledge acquisition tools, Configur8or's knowledge acquisition tool is impacted by the needs of a knowledge engineer who would possess substantial knowledge about programming. Consequently, an end user without any programming knowledge is not able to easily update or make any changes to the knowledge base through interaction with this knowledge acquisition tool.

With this point of view, the rationale for the implementation of this research was to develop an expert system – a wizard in effect; which would utilise some of the techniques and methods reviewed so far, in guiding non-programmers to input relevant knowledge into the knowledge representation structures in the appropriate format or syntax whilst maintaining any dependencies.

Configur8or also had the capability to simultaneously derive the cost for a configured product using historic cost estimation data which could be accessed from external databases using queries and a cost model which would usually be predefined and programmed into the software by the knowledge engineer.

CHAPTER 5 - KNOWLEDGE REPRESENTATION

It was concluded in chapter 2, that knowledge acquired from experts or other sources make up the knowledge which must be expressed in a manner compatible with the problem solving strategies of an expert system. The proposed expert system wizard is aimed at providing a browser-based graphical user interface through which a user would automatically create a product model to be exported for use within the configur8or software. Configur8or stores knowledge about product models a number of XML files. XML provides meaningful storage of information about concepts by using tags within a text file. A tag is used to represent a concept or an instance of a concept. Whilst the attributes and values of concepts are represented within the tags, the relationship between concepts is usually expressed by embedding a tag within another in a manner that reflects the hierarchy between them. For instance, the XML information below describes the flanges and cuffs as *Parts* concepts but instances of a higher concept *Component.* Both concepts also have a common attribute *type* which has been assigned values.

<Component>
<part type='standard'>flange</part>
<part type='drilled'>cuffs</part>
</Component>

The XML files store information about the following:

-Design of a product generic model i.e. the parts that make up the product;

-Configuration i.e. variables which store information about positioning of these parts to form the model, as well as the material and labour cost for manufacturing the parts;

-Display of questions which will be answered by users to configure a product for the generic model; and,

-Database or look-up table queries whose result sets serve as pre-defined answers to the display questions.

As this information is significantly dependent on the parts selected to constitute the generic model, the expert system user would make selections from a list of single parts or sub-assemblies via a user interface and contents for each of the XML files mentioned above will be generated to make up a generic model. Figure 5.1 below depicts the tasks goals of the proposed expert system. These tack goals are achieved by problem solving methods which invoke inferences that refer to domain knowledge.

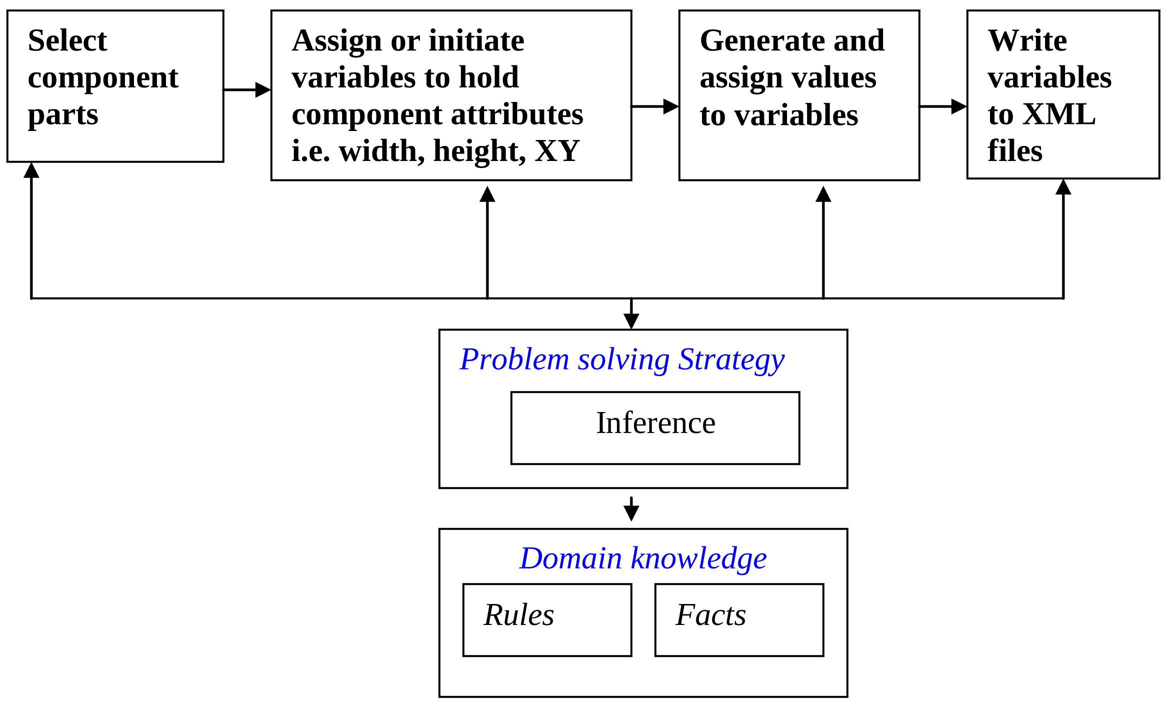


Figure 5.1: Expert system tasks

5.1 DOMAIN KNOWLEDGE

The knowledge models achieved from the knowledge acquisition and analysis processes constitute the domain knowledge which also represents the knowledge base of the expert system. As shown in figure 5.1 above, this domain knowledge can be further categorised into Rules i.e. IF THEN production rules which make up the knowledge base and into Facts. The Facts represents the modelled knowledge concepts and any attributes or values that they might have. Rather than representing reasoning knowledge in a declarative way, the Rules are a natural language representation of the understanding or reasoning required for problem solving using the Facts**.** As the Facts are used by the Rules**,** it is important that they are represented in a manner that emphasises their relationships. According to Milton [34], XML, conventional databases and text files are common technologies used in creating expert system knowledge bases. Although XML allows for customised encoding, navigation of the knowledge base and storage of information [4], relational databases will be better suited for the maintenance of the Facts and Rules as well as their relationships as the knowledge base grows. The use of relational database technology for the development of the proposed expert system's knowledge base is therefore justified by its fulfilment of this important requirement. SQL server database management system was used.

Figure 5.2 shows the meta-knowledge i.e. the relationship between the various categories of domain knowledge upon which the proposed expert system was to be developed. It was represented as a schema of various objects i.e. sets of facts and rules and the relationship between them.

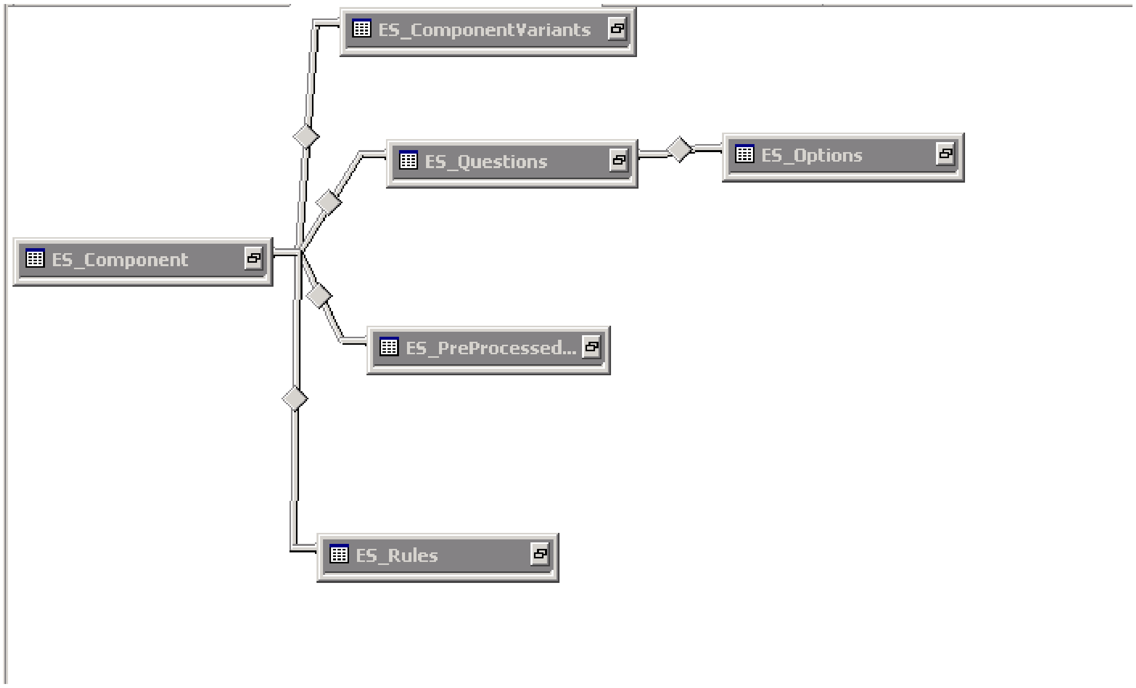


Figure 5.2: Domain Meta-knowledge

The diagram above illustrates the meta-knowledge in the product design domain.

Every product design or model is made up of component parts or in some cases sub-assemblies of parts. Each component part has the following:

- One or more types or variants
- Variables representing the parameters or attributes which can be varied to correspond to design needs. These attributes in this case are the width, height, and the positioning co-ordinates X and Y
- Dialogue questions and corresponding options -These are displayed on the user interface and are the basis for achieving dialogue and user interaction with the system

The process of configuring a product model is categorised by a number of tasks or problem solving strategies. These tasks are significantly dependent on domain knowledge, some of which are represented as rules. These rules are triggered in response to user input. For example, the assembly and configuration of the product models are achieved through the execution of product part compatibility rules and subsequently part attribute variation rules. These rules are further discussed in chapter 6.

An order of priority is reflected in the rule representation to facilitate conflict resolution when more two or more Rules are satisfied. Also, to ensure that the expert system functions with efficiency and speed, the representation of Rules reflects an appropriate level of granularity i.e. the amount of knowledge fused into one rule. A Rule becomes more difficult to modify if it consists of too many pieces of knowledge / facts. On the other hand, if the pieces of knowledge in a rule are too few, the rule will hardly be understood without reference to other rules.

5.2 EXPERT SYSTEM INFERENCE

The inference technique implemented for the inference of the proposed expert system was forward chaining i.e. a conclusion or the consequent part of a rule is only achieved when the antecedent or condition of the rule is satisfied. By matching the Rule conditions to Facts*,* the Inference Engine of the proposed expert system makes the decision on the Rules that are satisfied and fired. Rule conflicts would occur in a scenario where conditions are satisfied in more than one rule. These conflicts are resolved by executing the rules in order of their priority.

Some of the most common languages for the development of expert system inferences have been mentioned in chapter 2. However, expert system inferences [77] [78] have been developed using object oriented paradigms commonly used for conventional programming. The adaptation of an object oriented approach to the development of the expert system user interface and consequently, inference mechanism was influenced by capabilities such as modularity and inheritance of values amongst class objects. The expert system was developed using PHP – an object oriented scripting language with the capability of interfacing with common database systems. The choice of object oriented language allows for the expert system wizard to be executed via a web-browser for easier distribution to selected users across the company's network. The development was carried out within an integrated development environment with tools for display building**.** The inference engine is represented as a separate piece of code and the data and knowledge required by the inference engine for reasoning are explicitly separated as shown in appendix B.

CHAPTER 6 - EXPERT SYSTEMS DEVELOPMENT LIFE CYCLE

The linear life cycle model can be described as an amalgamation of project management activities and tasks from conventional software engineering approaches to the development of expert systems. It consists of a sequence of stages from planning to system evaluation which is repeated until an expert system is completed and subsequently used for the maintenance and evolution of the developed system [5][53][54] . There are slight variations in the number and stages of lifecycle models proposed by various authors. However, there seems to be a consistency with the objectives and the linear sequence of the basic phases of requirement analysis, system design, and system implementation.The stages in the linear life cycle model as described by Giarratano & Riley [5] further discussed.

6.1 Planning

In this stage, a work plan is produced to guide and evaluate the development of the expert system. This basically is a set of documents detailing Feasibility assessment; Resource management; Task phasing / scheduling; Preliminary functional layout; and, High-level requirements.

The first 3 tasks constitute the project management aspect of this stage and are imperative as they ensure that the objectives of the project are achieved on time and within specified budget. The need for the development of an expert system was justified based on factors such as the selection of an appropriate problem domain and development tools, identification of potential benefits and an analysis of development and implementation costs. Like any other project, the resources acquired for the development of the proposed systems i.e. people resources (academic and company personnel), time, money, were managed accordingly. Development tasks were also specified and scheduled within the lifecycle stages.

The last 2 tasks can be respectively compared to the functional specification and design phase architecture tasks of the waterfall lifecycle model commonly used during the development of conventional software. These tasks define what the expert system should accomplish by specifying the high level functions of the system and how these will work together to accomplish the purpose of the system.

6.2 Knowledge definition

In this stage the knowledge requirements of the expert system is defined by identifying and selecting the appropriate source from which knowledge will be acquired for analysis. Below are some of the tasks involved in this stage of the lifecycle model.

(I)Requirement specification- The first step in this phase is the specification of the requirements for the expert system i.e. defining what the system is supposed to achieve. The proposed expert system will guide domain experts who are also non-programmers, in building bellows design and cost models using the configur8or software. The design aspect will involve interaction with user friendly interfaces. Users will be prompted to provide answers to key questions and additional questions may be triggered by the users' response. Based on these responses, certain rules will be triggered to create output which will be converted into XML format which will be fed into the corresponding underlying structures within Configur8or.

(II)Preliminary control layout – Following the specification of requirements, a general description of phases to be executed by the expert system is provided to correspond with the agenda or group of rules that are triggered to control the execution of flow. For instance, Figure 5.1 depicts a preliminary control flow for the proposed expert systems

(III)Detailed functional layout - This task provides a more technical and detailed specification of the system's functionalities (i.e. procedural functions, knowledge bases and databases) based on the preliminary control layout. The functional layout for the proposed expert system is represented using a database Schema and class diagram as shown in Appendix D.

(IV)Acquisition strategy – The knowledge required to achieve the specified capabilities of the system is achieved using a combination of the techniques discussed in chapter 2.5.2. In the course of this research, knowledge was acquired by interviewing key staff; observing certain activities as they were carried out; analysing information from certain key databases / spreadsheets; and, drawing up process/concept maps and information grids/matrices. These knowledge acquisition techniques are discussed in chapter 3*.* The outcome of the acquisition task is the identification and structuring of key knowledge elements to aid understanding and verification of knowledge by the knowledge engineer / programmer.

(v)Knowledge baseline – As in project management, it is important to baseline the knowledge acquired so that any changes must be made by formal change request. This is a means of ensuring that required changes are reflected throughout the development lifecycle.

6.3 Knowledge design

In this stage, a detailed design of the expert system is produced. This involves the detailed specification of the system's control structures, detailed specification of user interfaces, and the organization and representation of the knowledge acquired in the knowledge base as discussed in chapter 2.5.3. In the course of this research, facts and rules gathered from knowledge acquisition is represented in database tables. The inference mechanism is developed using an object oriented paradigm and represented as a separate piece of code within the software that make up the rest of the expert system. The concluding tasks in this stage are the specification of code testing / verification methods and the baseline documentation of the design;

6.4 Coding

This stage involves the actual coding / programming of the expert system, code testing and the use of comments or code documentation as well as the preparation of user manuals, system description documents and installation guides to assist experts / users in using the system and providing necessary feedback.

6.5 Knowledge verification

The developed system was verified against the base-lined requirement specifications and functional layouts to determine the correctness, accuracy and consistency of the answers / results provided by the system as well as the source of problems if any. The commons sources are the rules, inference chains or uncertainty.

6.6 System evaluation

This was the final phase of the lifecycle. In this phase, the result of the testing and verification was summarized and recommendations were made for changes to the system. The final system was also validated against the user requirements earlier agreed. A final or interim report was issued based on results from the testing, validation or verification carried out.

6.7 Conclusion

Development lifecycles are important in the management and development of quality systems. The lifecycle model discussed in this chapter explores the managerial (i.e. project management) and technical aspects of system developments carved out from conventional software engineering approaches such as the waterfall and spiral models. The implementation of the lifecycle model was successfully.

CHAPTER 7 – EXPERT SYSTEM ARCHITECTURE

The objective of the proposed expert system wizard was to guide expert users who were non-programmers in developing configurable product models within *configur8or.*

Configur8or provides a graphic representation of configurable product models so that changes to these models are simultaneously visualised as the user is engaged in dialogue through a user interface. The browser based software employs a rule-based problem solving strategy and an underlying structure which contains the rules, problem solving knowledge and other relevant data which are all represented in XML file format. The XML files are generated for each product model designed and are stored in a specified root folder on the server / host machine from where the software is centrally accessed by users. They represent the design and cost estimation models on which configurable product models are developed and are derived from a combination of rules, knowledge and data which are coded into the system through a design interface made up of a number of grids and matrices. Although users are not expected to interact with the underlying XML, any alterations to these are reflected in the design interface. This possibility is the basis upon which the anticipated integration between the proposed expert system and configur8or was to be achieved.

In comparison to the design model, the cost estimation model was more generic and could as such, be easily tailored to any design model. It was made up of a series of pre-defined static formulae and queries that made up the estimated material and labour costs for a number of pre-defined product parts, product materials and manufacturing operations. Although these pre-defined parts, materials and operations were essential to the configuration and cost estimation of product models, the development, configuration and graphical representation of the configured product model resulting from user interaction was to a greater extent dependent upon the product parts as compared to the materials or operations. That is to say that the selection of materials or operations had no effect on the graphical representation of the configured model.

Each component part that made up a product model was characterised by attributes i.e. width, height, and X/Y co-ordinates which drive the graphical positioning of individual parts in relation to another resulting in a product model. Within Configur8or**,** the values of these attributes were held in variables which could be varied to correspond with design or configuration needs. User interaction with Configur8or was achieved through a user interface which provided a graphical representation of a product model which was configured using a series of dialogue questions (and in some cases, corresponding options which relate to the product parts), and the execution of certain rules. These rules were used to derive the values of part attribute variables and controlled the display of relevant dialogue questions. Configur8or's admin tool provided a set of grids through which a design / configuration expert would create dialogue questions, variables and relevant rules. For any particular product model designed within Configur8or, any inputs made into these grids were represented in XML file format within the model titled folder which was located in the Configur8or software root folder. These XML files were automatically generated when a new product model was initiated within Configur8or.

The developed system allowed users to select from a number of pre-defined product parts without the requirement of any product design expertise and it automated the design of configurable product models within Configur8or. It comprised of product part selection and rule editing modules each having a user interface, an inference module, a pre-processor, an output/error reporting module, a knowledge base of rules which represent the domain knowledge, a database of design rules and a database of parts and graphical images. These various components are further discussed in following sections. The system outputted a set of variables which represented geometric parameters for the user selected parts and the constraints or conditions which influenced the values of these parameters. The system integrated with configur8or software by spinning the output into designated XML files which were located in the root folder of the configur8or software.

Appendix B shows the overall structure of the developed system and its integration with configur8or*.*

7.1 Product Part Selection Module

In operation, the first task carried out by a user on the developed system is product part selection. As a result, the part selection module is the index module of the developed system. It possesses a user interface for user interaction. The interface presents a selection form and buttons which enable the user to submit a selection, make changes to the a selection, spin the output results into XML format for use within configur8or, and navigate to the rule editing module**.** The selection module is shown in Figure 7.1 below**.**

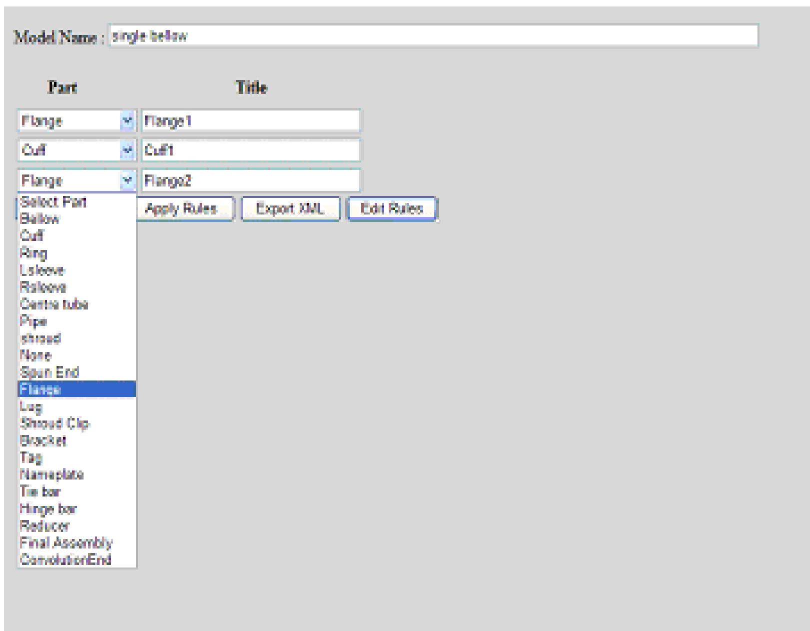


Figure 7.1: Part Selection Module

A name is assigned to the product model via the input field at the top of the form and the selections are made from a drop-down list of parts pre-loaded from the database of parts and graphical images.

7.2 Rule Editing Module

This module allows users to make changes or new additions to the design rules which are held in a database. Like the product part selection module, this module possesses a user interface which displays a form through which existing rules are updated and new ones are added; and buttons to submit changes and navigate to the index page of the system. The Rule editing module is shown in Figure 7.2(a) & (b) below.

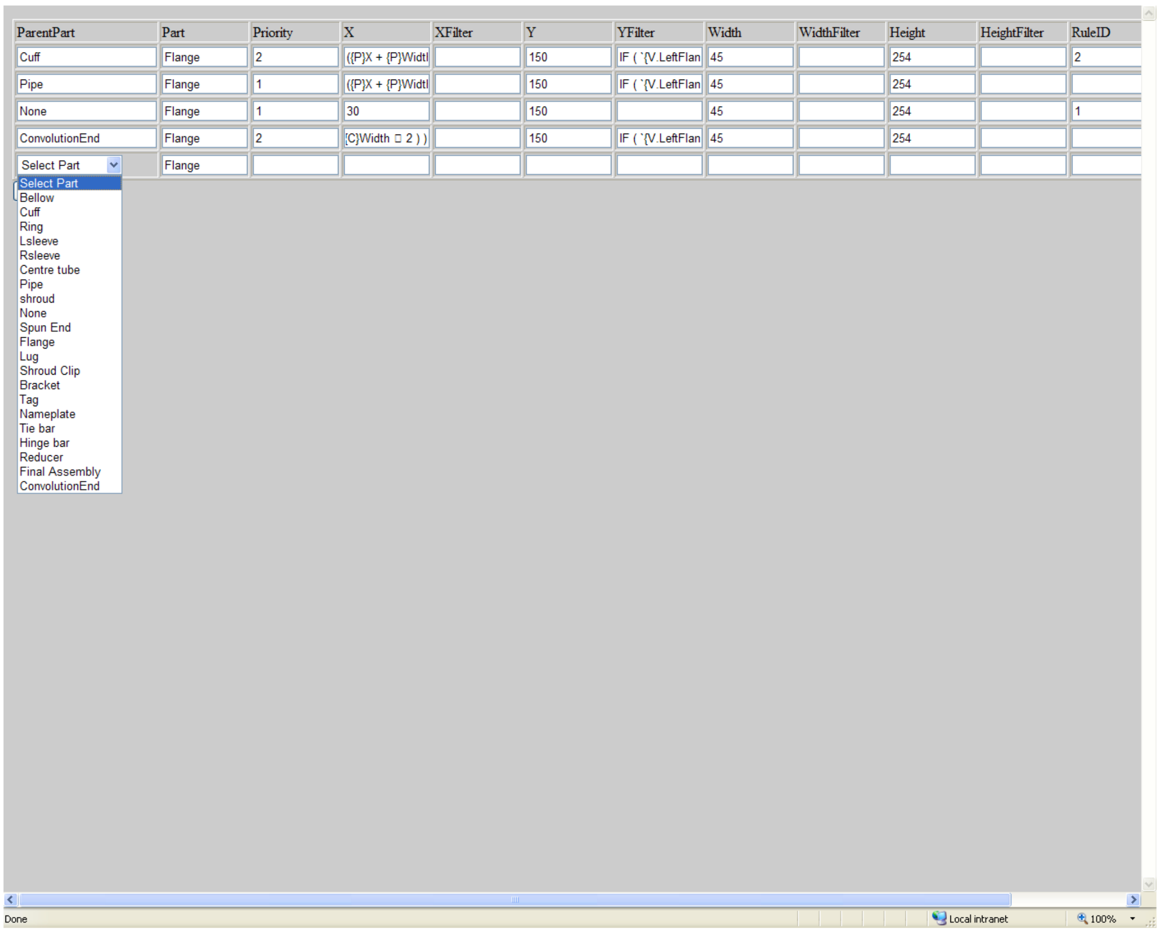


Figure 7.2(a): Rule editing module

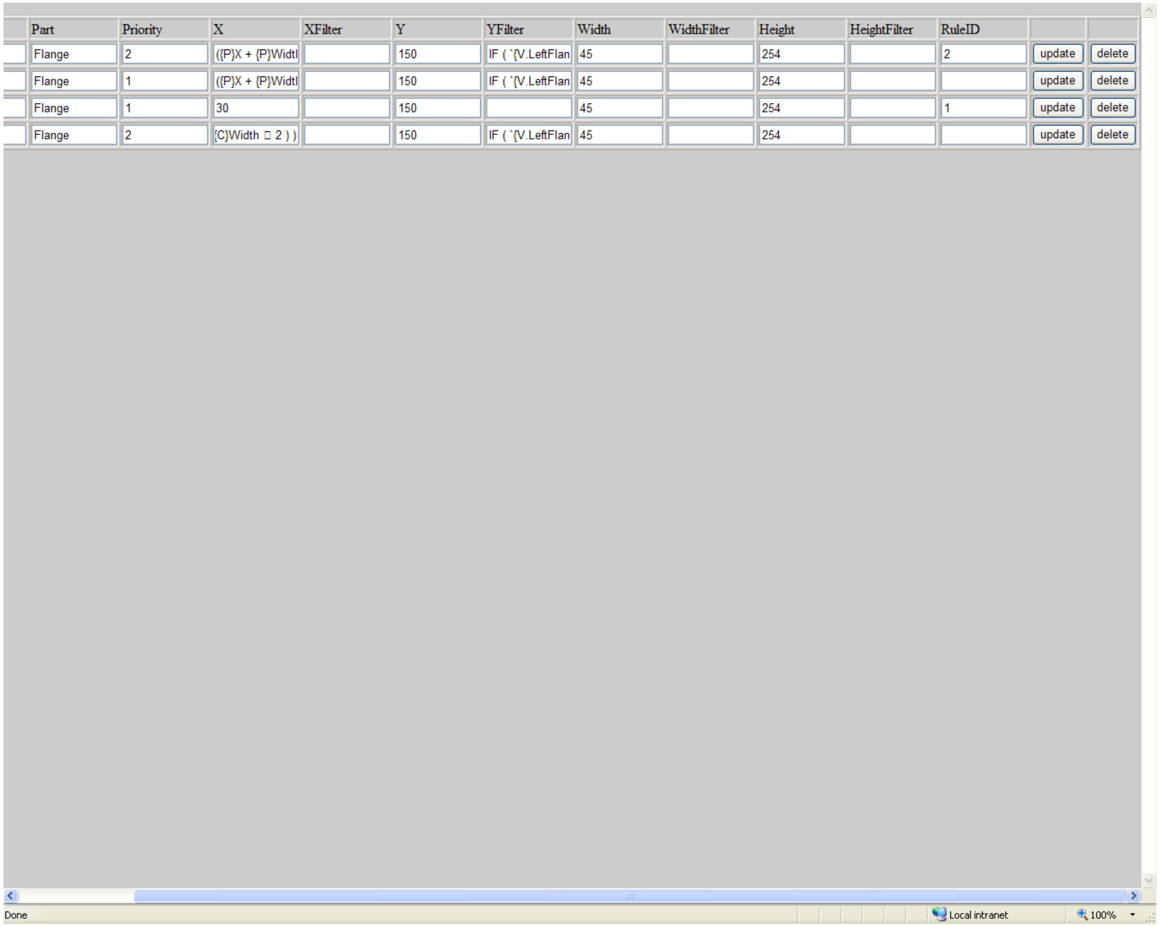


Figure 7.2(b): Rule editing module

The user is able to select a product part from a pre-loaded dropdown list of parts and existing constraint rules, which match the selection, are loaded from the database and displayed. Within the design rules database, the products parts are represented using numeric values as they are better suited to the systems problem solving strategy. These values are however mapped out to a text format to facilitate user interaction. Within the user interface and the database, the rules and constraints are represented in a configur8or compatible format. When editing existing rules/constraints or adding new ones, the user is presented with a popup form (another simple wizard in effect) through which the user can build rules / constraints in a natural language. The popup form as shown in Figure 8.3 provides buttons which are labelled in natural language and represent the product part parameters. Relevant arithmetic operators i.e. +,-,*, % can be input via the keyboard where necessary. The finished rule (in natural language representation) is submitted and automatically converted to the configur8or compatible format within the rule editing module.

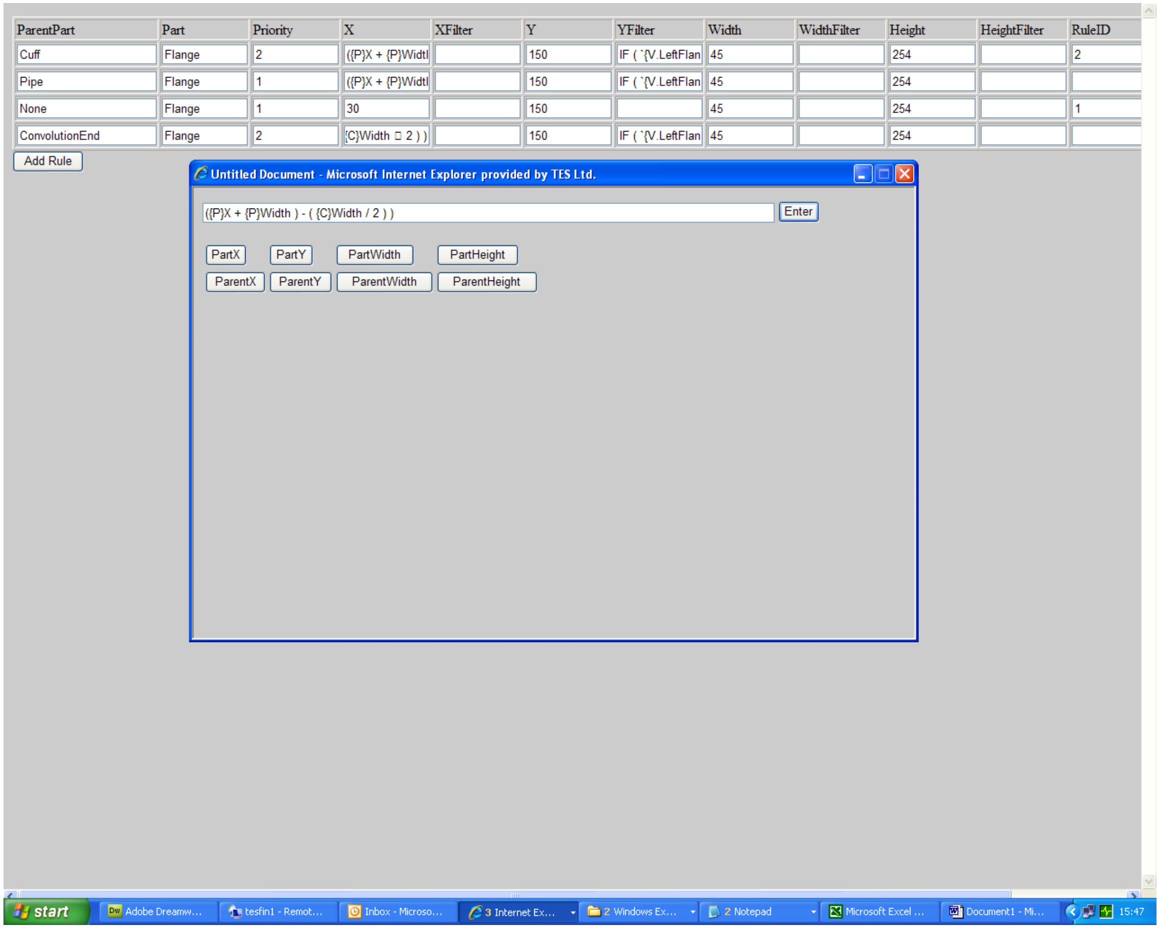


Figure 7.3: Pop-up wizard

7.3 Pre-Processor Module

The parts selected from the product part selection module are submitted to the pre-processor where they are stored in variables. The pre-processor is a series of code written in object oriented paradigm. It contains classes of functions through which the process of inference is initiated. The pre-processor functions work with the user input and facts from the working memory to return parameters which are qualified by certain keywords. These parameters are then passed unto the inference module to be matched against the antecedents of the rules which represent the domain knowledge. Using the accompanying keyword or value of parameter as the antecedent, the inference module triggers the consequent procedures which are again executed via the pre-processor to deliver the final output. It does not possess a user interface and its mechanism is not visible to the user.

7.4 Inference Module

The developed system employs a rule based forward chaining inference mechanism which has been designed and implemented using an object oriented programming paradigm. Using the processed parameters and keywords from the pre-processor module, the inference module triggers procedures to match the antecedents in the knowledge base production rules with the data/instantiations from the working memory which have been processed via the pre-processor. The rule consequents are a set of logic procedures used to derive the final output or solution. The inference module triggers procedures which can subsequently trigger only one rule at any given time. In the scenario whereby more than one rule is matched, the conflict resolution strategies of *refractoriness* and *prioritisation* are implemented. This involves filtering out the rules that have been previously executed and executing the remaining rules in the order of priority.

Every rule in the knowledge base holds an integer which represents the order of priority – 1 being the highest order. This order of priority is determined by an expert at the point of entering a new rule or editing an existing one. For every part selected in the part selection module, the inference engine with the help of the pre-processor module, triggers the selection of rule(s) - including the priority index integer, associated with that part's dependency on any other parts which have also been selected in the selections module. This rule(s) are stored in a multidimensional array. Where more than one rule is matched at any given time, the pre-processor module searches through the 'rule index' field of the array and executes the rule with the lowest integer value i.e. the highest priority. For instance, the rule associated with the following dependencies would be triggered for the 'bellow' part selection if the 'pipe' and 'flange' parts are also selected via the selection module. The priority suggested by the indexing is such that if a pipe and flange are selected, the pipe-bellow dependency rule is executed first because ideally, a bellow would ideally be welded unto a pipe rather than a flange, if a pipe has been decided as a component part of the assembly. However, if only one of pipe and flange part is selected alongside the bellow part, then only one single rule will be selected.

Pipe – Bellow (index 1)
Flange – Bellow (index 2)

Like the pre-processor, this module does not have a user interface and its mechanism is not visible to the user.

7.5 Output/Error reporting module

This module does not support user interaction. As shown in Figure 7.4, It displays the results / conclusion generated by the developed system as well as a step by step account of the reasoning applied in deriving the result thereby acting as a knowledge explanation facility. This module could also be used for the purpose of error reporting by the knowledge engineer. The output array for each selected part is separated by the part headers displayed in bold red font. The output of the developed system is an array of values representing attributes in relation to sizing and positioning co-ordinates that define the assembly of the selected parts to form the required product model.

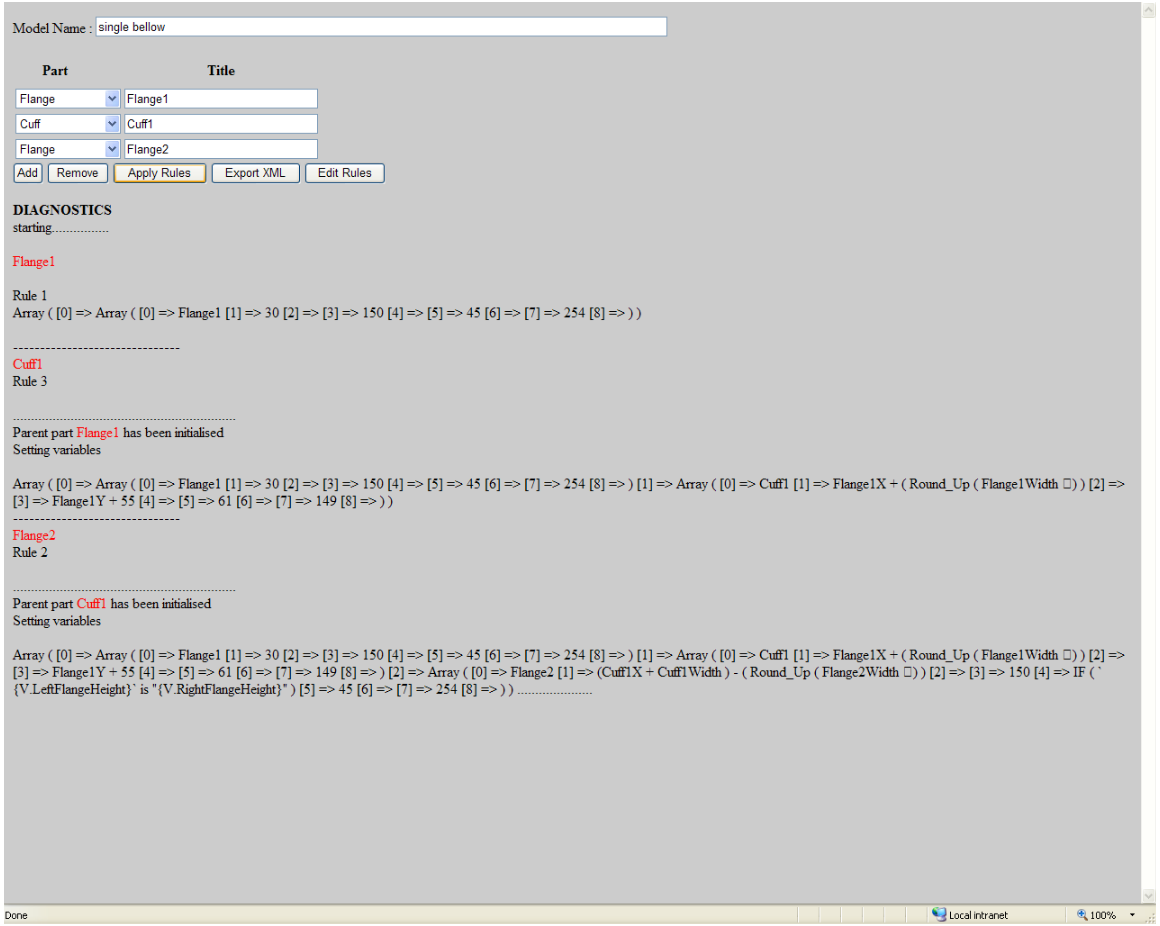


Figure 7.4: Output / error reporting Module

7.6 Knowledge Base

The knowledge base of the developed expert system is a collection of rules which make up the expert / expertise knowledge of the domain. The knowledge base held 30 rules at the point of test and was structured within a database table in a manner similar to a frame representation. This was to ensure that knowledge was well organized and easily updateable. Each rule was represented by a database row or record. The first two columns or fields represent the antecedent part of the rules whilst the remaining fields represent the consequent parts of the rules. Figure 7.5 shows the database representation of the rules. The rules are accessed via a multidimensional array. Each cell /value in the array is read as **array[a][b]** where **a** represents the row/record index starting from 0 and **b** represents the field/column index also starting from 0. Based on this representation, a matching rule will read as follows:

If (array[rowindex][0] == {workingmemory input} && array[rowindex][1] == {workingmemory input})

Then

{

***antecedent* procedure;**

}

Field indexes [0] and [1] represent values from the first two columns of the table.

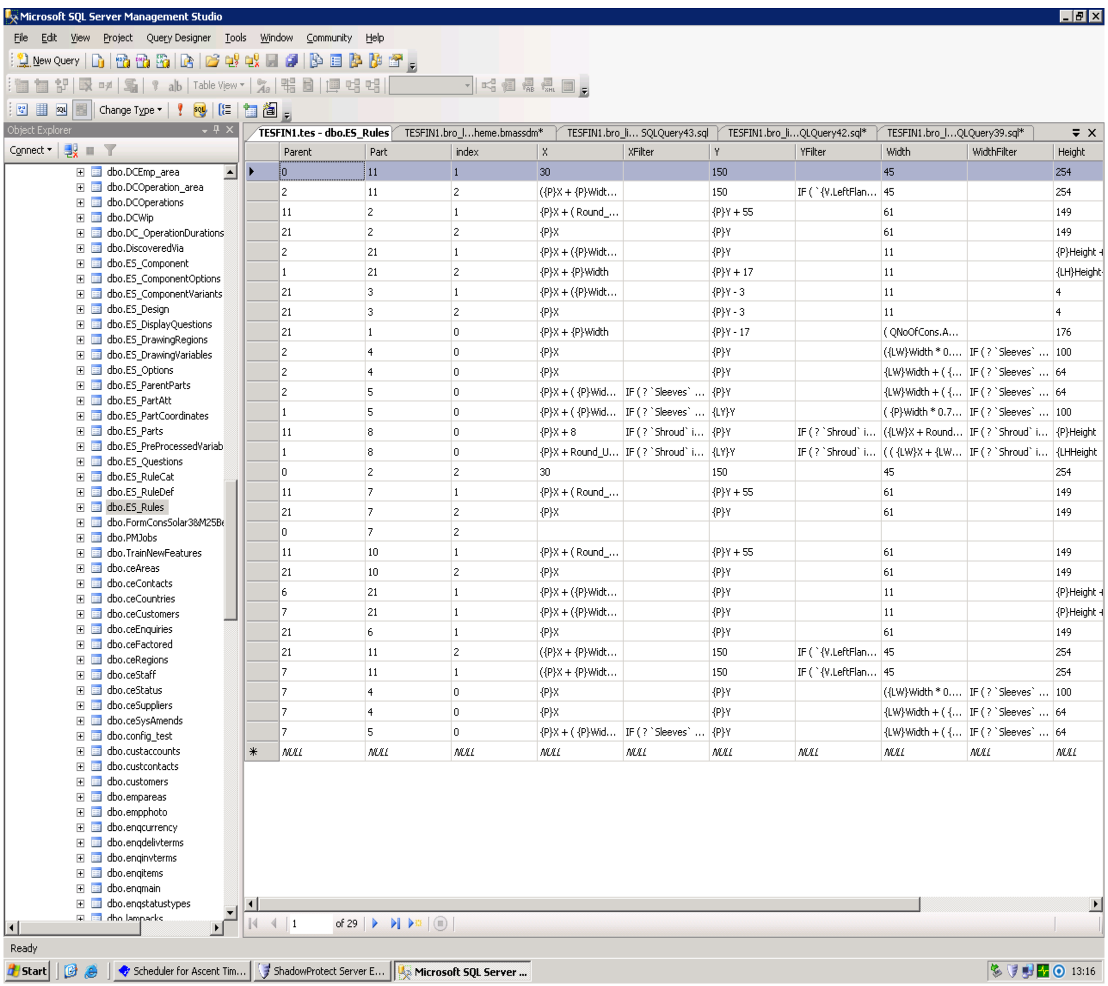


Figure 7.5: Knowledge base

7.7 CONCLUSION

With reference to the findings from the prior literature review undertaken, the developed system possesses the major functionalities and capabilities which characterise an expert system. The overall structure demonstrates an order of rule execution which is not pre-determined, but dependent on user input and the separation of domain expertise knowledge which is represented in the rules from problem solving strategy i.e. the knowledge of the use of rules to derive solutions. This ensures modularity which encourages re-use of code with little or no modification. It also ensures interaction with domain rules in a structured and easy manner. Whilst the domain rules can be updated by product design experts who are not necessarily knowledge engineers, the expertise of a knowledge engineer is still required to make changes to the system inference and thus the problem solving strategy as it involves programming procedures.

CHAPTER 8– OPERATION AND PERFORMANCE VALIDATION

This chapter describes the physical implementation of the knowledge based system that has been designed and described in the previous chapter, into the existing structures within the host company. The goal of the system was to guide experts and non-experts in designing product models within configur8or. The strategy employed was to have the output of the developed system translated into XML format and exported into configur8or***.***

8.1 OPERATION

The outputs of the developed system were arrays of component part names and corresponding variables (which represented part attributes and values) from which the graphical representation of product models were derived. To have a set of output arrays translated into the required XML format and exported into configur8or, the anticipated product model had to be initiated within configur8or***.*** Domain expertise knowledge was not required to carry out this task as it was a straightforward task achieved by clicking on the 'Add' button on the admin tool user interface within Configur8or. As shown in Figure 8.1, a dialogue box was provided to assign a name to the new product model. Initiating a new model ensured that the required XML files were intelligently created and located with the software root folder.

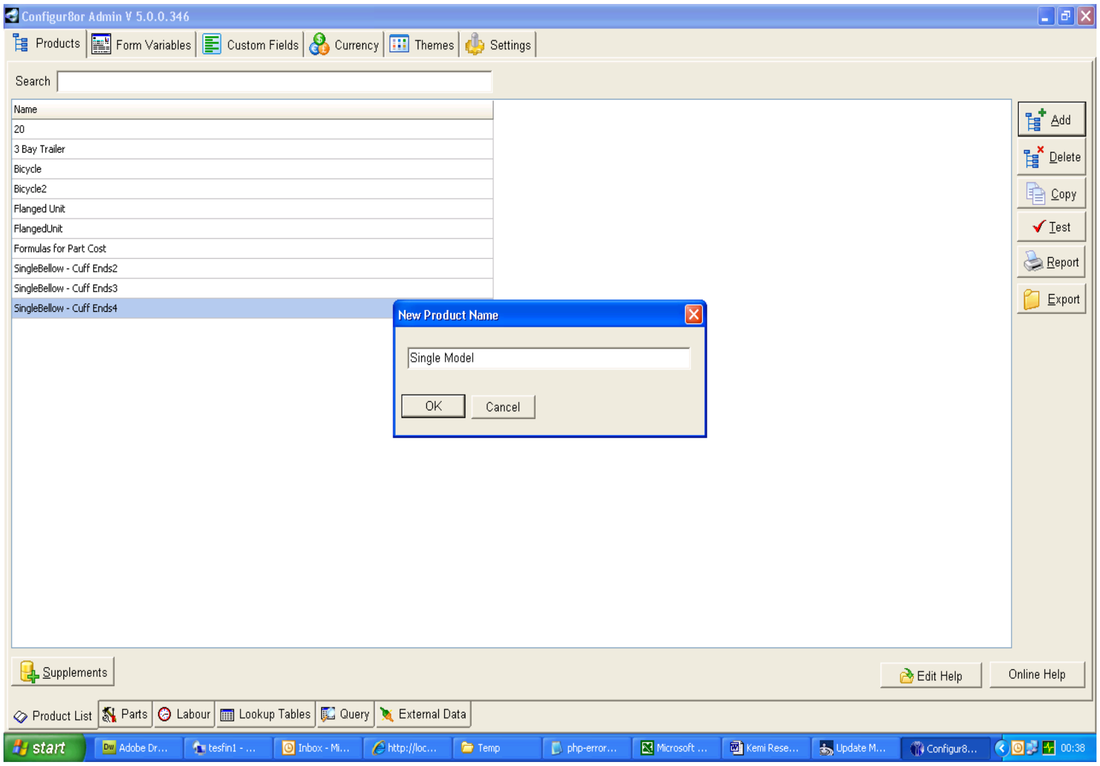


Figure 8.1: Initiating product model

A number of blank XML files were created and located within the product model folder. However, only two of them were designated to hold product design and construction knowledge. These are the '*construction.xm*l' and '*design.xml'* files. On the product selection interface, the *'Export XML'* button when clicked exported the output array of part attributes to the named XML files. The structure and content of each file was different. Whilst '*construction.xm*l' represented the set of variables which held values of the part attributes X, Y, Width and Height, 'design.xml' represented the alignment of the corresponding graphical images of the selected parts using the variables declared in '*construction.xm*l'. In Figure 8.2 below, a confirmation alert is displayed to assert that the expert system resulting output array had been successfully exported to the designated XML files.

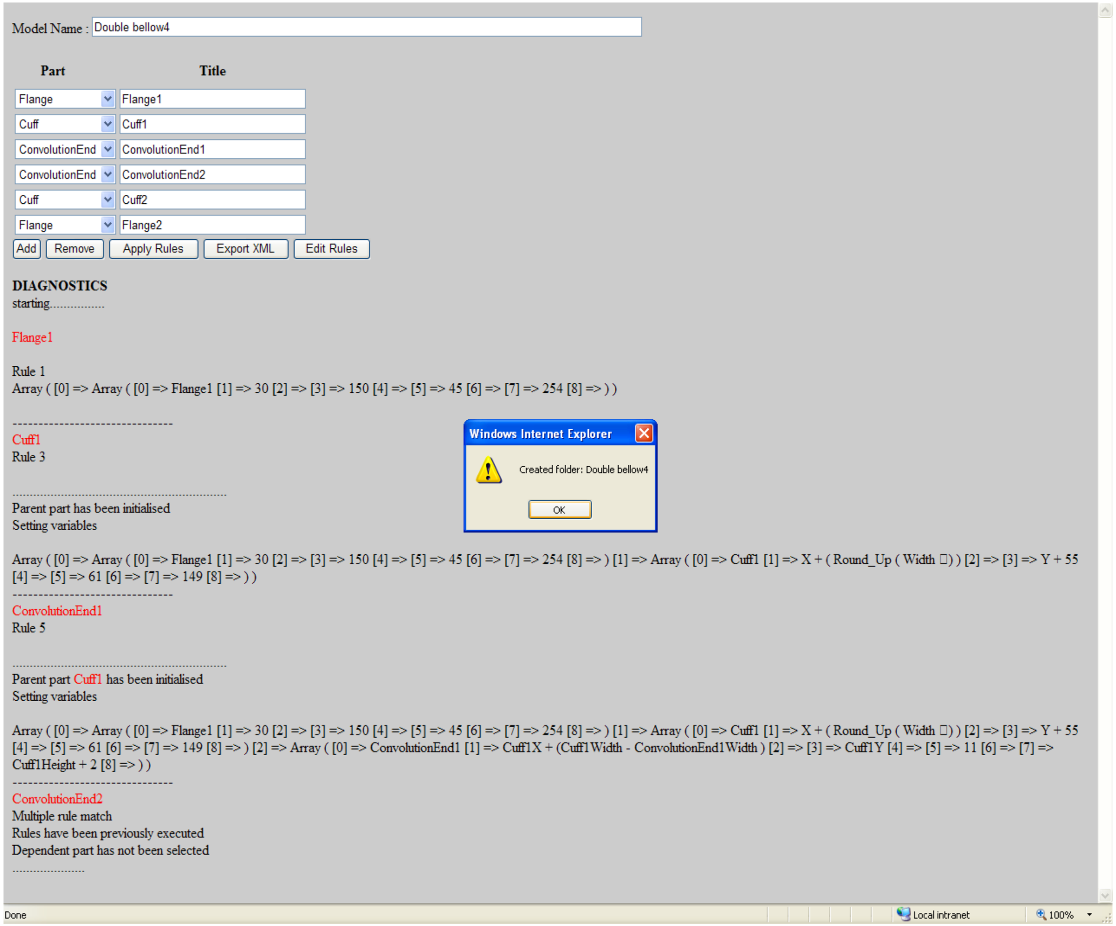


Figure 8.2: Output array exported to XML files within Configur8or

Figure 8.3 shows the representation of the XML information within Configur8or's 'variables' admin tool and Figure 8.4 shows the graphical representation of the designed product model via a web browser.

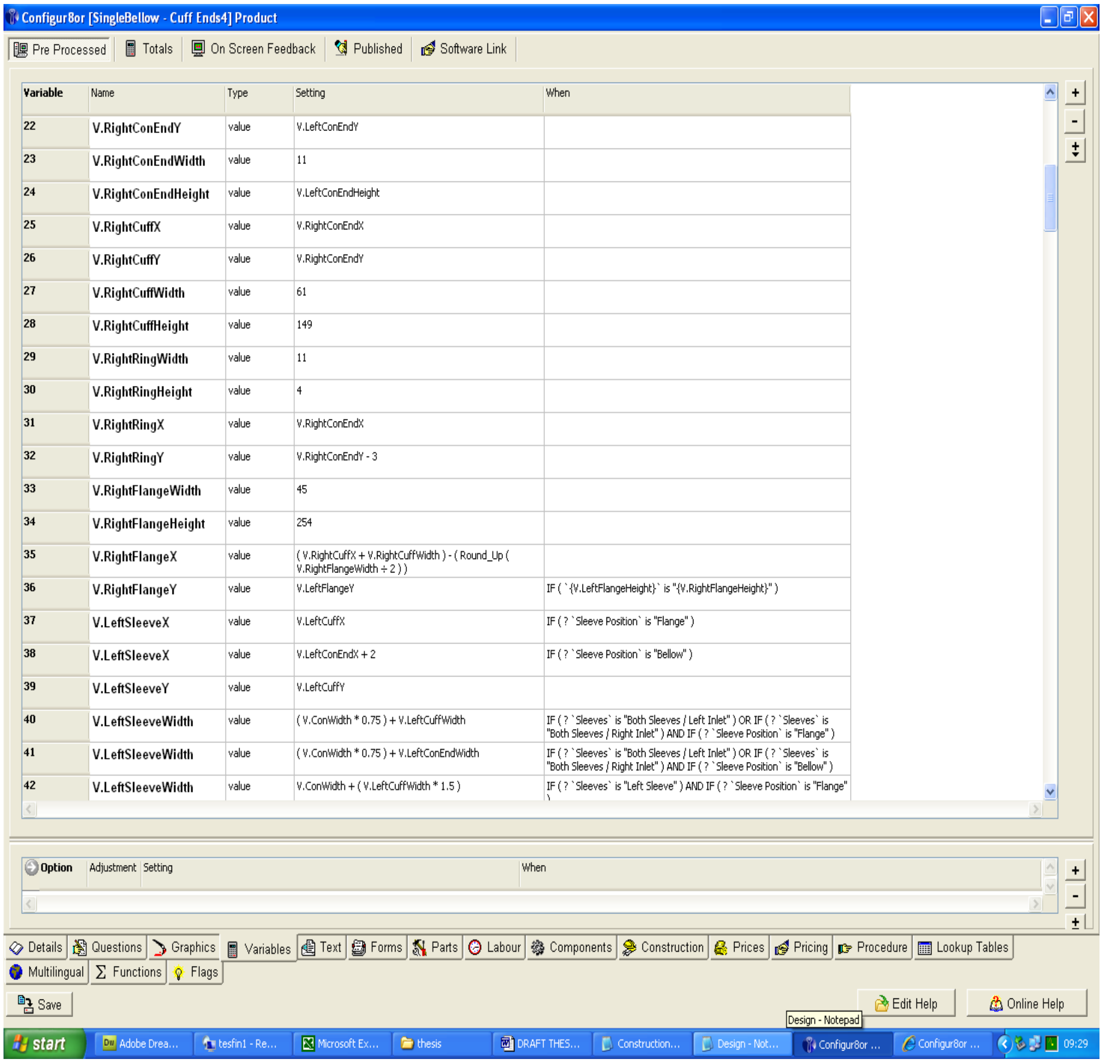


Figure 8.3: Representation of *construction.xml* within configur8or's admin tool

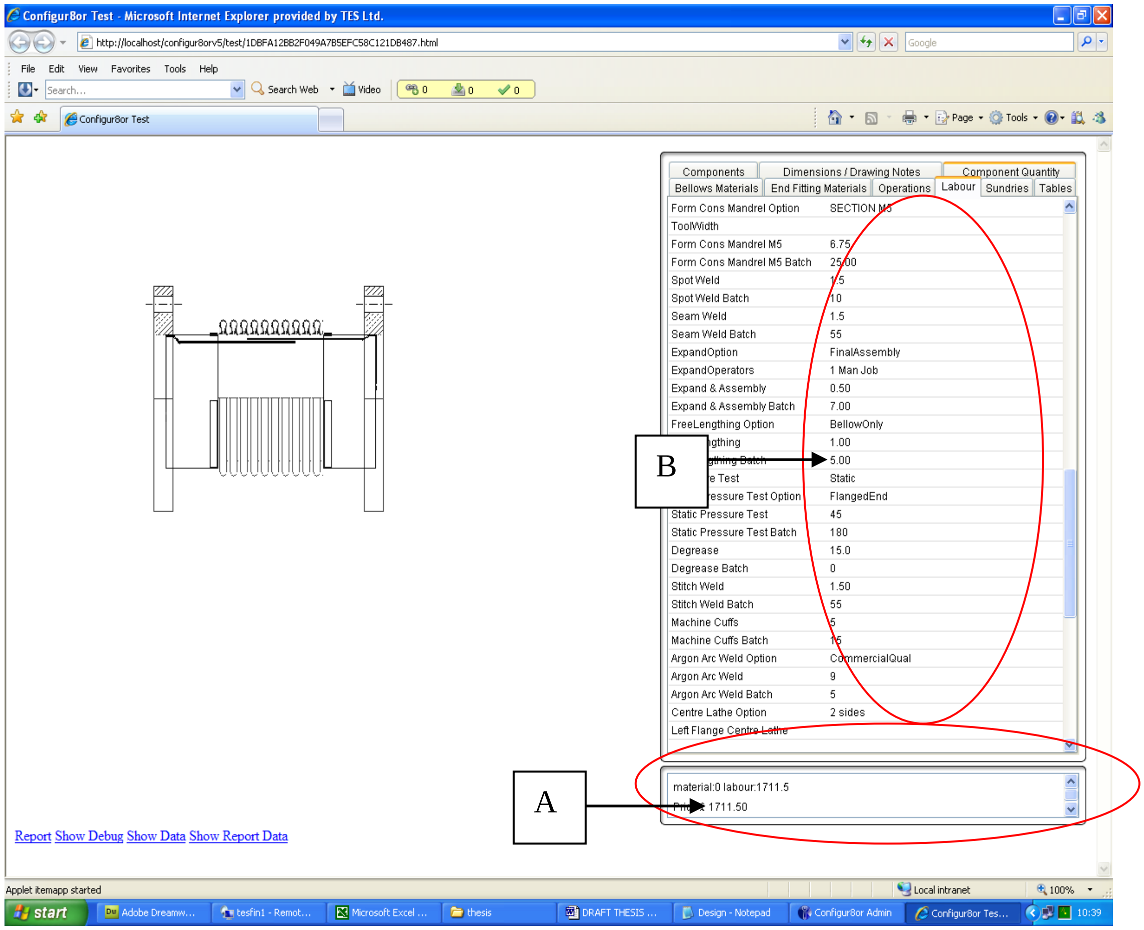


Figure 8.4: Graphical representation of configurable product

A generic cost estimation model had been developed within configur8or. The model comprises of sub-models for materials and labour cost estimation which are based upon a pre-defined formulae derivations and queries to external cost estimation data. Within these models, the material and labour cost for manufacturing every product part manufactured within the company has been associated with at least one query and a formula. The labour and materials cost of a part can only be calculated if that part has been selected via the user interface during product configuration. This is achieved by assigning pre-set rules against each part and labour operation costing as shown in Figure 8.5 below. The user could make material selections via the user interface, and also select or deselect labour operations where expert intuition was needed. The estimated costs for all the selected parts are summed up to give a final estimated cost for a configured product as highlighted in Figure 8.4(A) above. Figure 8.6 shows an illustration of the cost estimation model.

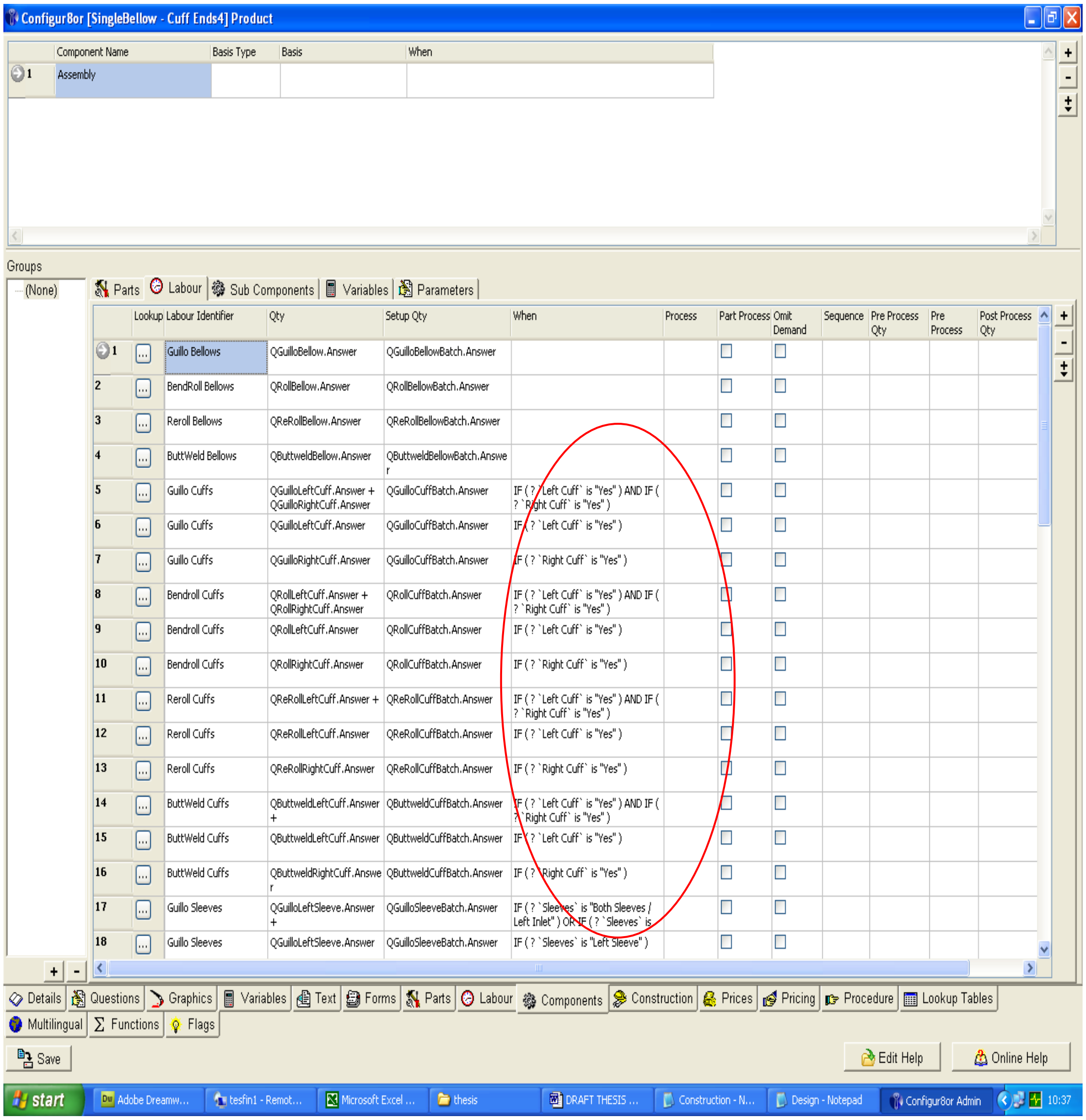


Figure 8.5: Cost estimation constraints within configur8or

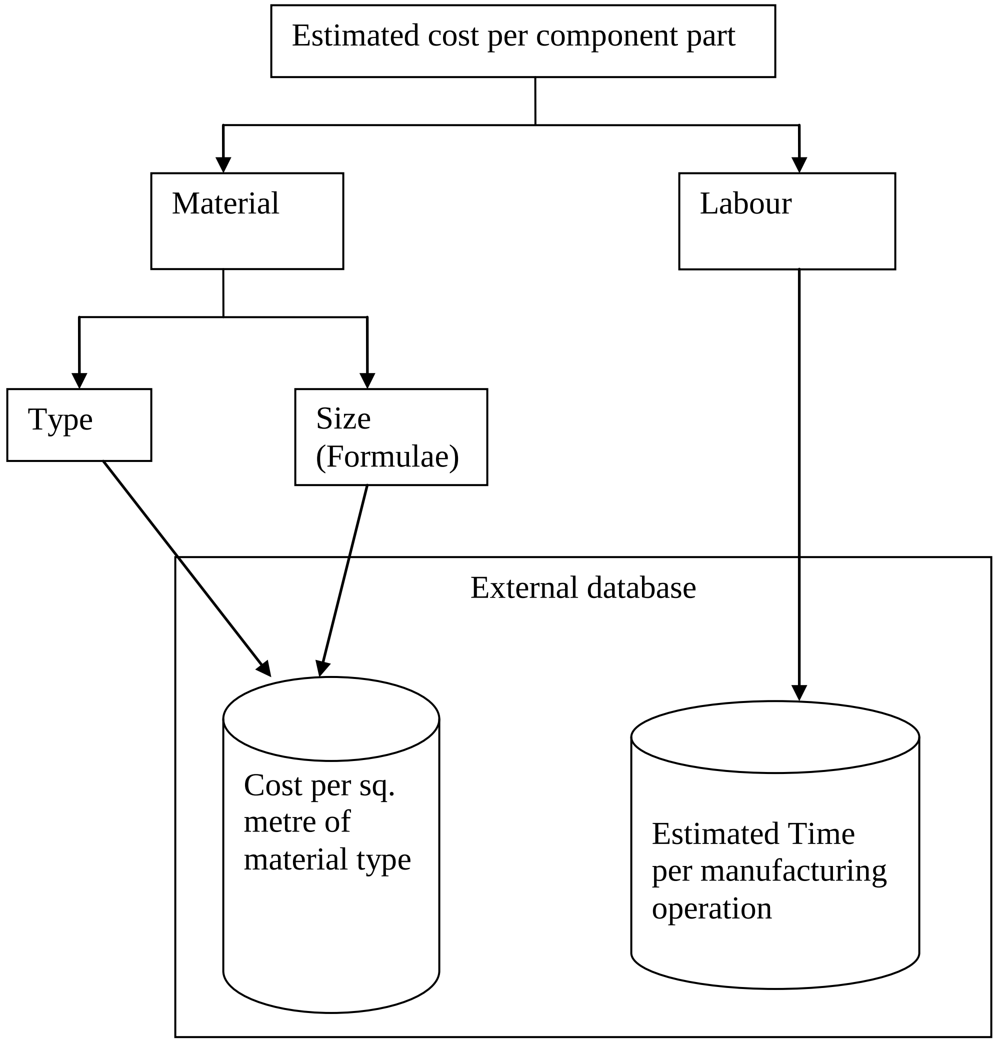


Figure 8.6: Cost estimation model

Although cost estimation is not being achieved using an expert system approach or within the developed expert system wizard, it employs the *intuitive cost estimation technique* explored in the prior literature review. Configur8or's user interface acts as a decision aid tool which uses domain expert knowledge (queries, formulae and constraints) to methodically derive cost estimates for product parts and consequently configured assemblies whilst allowing estimators to make judgments at various levels of the estimation process based on their past experiences.

8.2 PERFORMANCE VALIDATION

Validation was carried out through a case study to compare the times taken to design a product model with and without use of the developed expert system. The results of the tests undertaken indicated increased speed in the process of designing product models using the developed system***.*** Whereas it took a design expert 2 hours on an average to model a product with pre-determined parts using the admin tool within ***Configur8or,*** the same task was carried out by an expert and a non-expert at separate times using the developed expert system, and was completed in an average of 20 minutes. In addition to improvement of the design process, the testers commended the system's application of uniformity in the invocation of rules and assignment of values to parameters of the parts which made up the model. A potential increase in the productivity of the experts was also mentioned as the use of the developed experts system would enable key staff to spend the time saved from developing models, on the research and development of other processes.

The developed expert system wizard runs via a web-browser for easier access across the company's network. It has been tested on Internet Explorer 7.0 and Mozilla firefox 3.0 browsers on the windows platform. It is centrally distributed to selected users via an internet information service (IIS) web-server which is located on the company's central windows server. The configur8or software is also hosted on the same server and executed via the same web-server.

The empirical evidence clearly demonstrated the ability of the system to meet the initial goals of this research and further anecdotal evidence highlighted the extended gains that had been made.

CHAPTER 9 – CONCLUSION & FURTHER WORK

9.1 CONCLUSION

Knowledge remains a fundamental aspect of expert systems development and the concerns about knowledge acquisition and representation is still widely researched. The experiences gained from the process of acquiring and representing knowledge throughout this project can be related to findings from the review of relevant theories in chapter 2. In accordance to the theories of knowledge acquisition, manual knowledge acquisition proved very challenging and time consuming. Interviewing and observing multiple experts gave broader results and verified the completeness and accuracy of expertise knowledge acquired. However, varying expert opinions in a few cases posed a threat to the knowledge engineering process. This threat was eliminated through individual consultation of experts, designation of a primary expert and integration of multiple opinions using techniques from the Peter Checkland's Soft Systems methodology. One challenge which could not be exactly solved using any of the reviewed techniques was that of the regular availability of the product design / cost estimation experts during the knowledge elicitation phase. The experts were still responsible for their day-to-day business and usual roles within the company. At times when workload increased or design jobs had to be completed as a matter or urgency, priority was not given to scheduled meetings or knowledge elicitation sessions. This, however, in no way detracts from the methodology, design or implementation of the programme, but simply the estimated time line to the completion.

The theory of knowledge representation suggests that knowledge acquired from experts or other sources must be expressed in a manner compatible with the problem solving strategies of an expert system. In accordance to this theory, the appropriate representation of the knowledge base and inference engine using the relational database technology and production rules respectively facilitated the development, efficiency, speed and maintenance of an expert system.

The Expert System Development Life Cycle Methodology was used for the purpose of this research. This linear methodology involved the amalgamation of organisational analysis activities, project management activities and technical activities from conventional software development approaches such as the waterfall model. This ensured that a functional system was delivered and eliminated the chance of system failure which has been recorded in the development of many KBS as a result of lack of concern for organizational factors.

The expert system described is an object oriented and rule-based system which encapsulates the company's existing knowledge relating to product design rules to facilitate the development of product models within the company's existing product configuration and cost estimation software – configur8or. The developed system comprised a user interface, a rule-based forward chaining inference engine developed using the Object Oriented Programming paradigm, a pre-processor which executed the logic associated with the rule antecedents and consequents, an explanation facility, database of design rules which served as the knowledge base, and database of facts. The output of the system was spun into XML format which were read and interpreted by configur8or to create configurable product models. The user interface allowed for users to input and view analysis of results in a structured and easily interpretable manner.

The objective of this research programme has been to facilitate product design and cost estimation processes and at the same time bring about the retention of expertise knowledge within the company. The initial literature review undertaken provided a thorough understanding of the theoretical and practical implications of applying the expert systems approach and its associated relevant methodologies / techniques. The developed system has been based on the expert system approach as demonstrated by its characteristics and capabilities and employs with sound rationale some of the methodologies and techniques explored to extract, analyse and represent expertise knowledge.

In the earlier stages of the research project, it was anticipated that the proposed expert systems wizard would facilitate the design of product models and provide decision support in the derivation of cost estimations. However, as TES ltd made commercial decision on the implementation of a configuration software, the course of the research was slightly altered in order to integrate with other strategic imperatives within the company. Further research into the capabilities and functionalities of this third party software was undertaken. The research revealed the possibility of achieving generic one-size-fits-all cost estimation model involving the use of pre-defined queries and formulae which have remained the same over many years and there is no immediate plan for this to be changed. This model can be tailored to any combination of parts. On the other hand, the procedure for designing configurable product models involves the derivation of product part attribute values using dynamic formulae. This process of manually creating the product models within configur8or not only requires design expertise but could also be very time consuming and tedious for design experts who might lack knowledge engineering expertise and even a knowledge engineer who might lack design expertise. Following from this finding, the rationale from this point onwards was to apply the expert system approach in providing a practical solution for facilitating the process of designing product models within the configuration software to integrate with the in-built cost estimation model. The ultimate objective was still to ensure that only optimal time is spent on the quotation stage i.e. the design and cost estimation stages of the company's business process. It became unnecessary to constantly re-invent the wheel by implementing an expert system approach to cost estimation modelling.

The knowledge acquired from the cost estimation experts was therefore invested in the development of the cost estimation model within configur8or in a manner which reflects the technique explored in the course of the literature review.

From the outcome of the system validation undertaken, it can be concluded that the implementation of the developed system has achieved the planned objective.

9.2 FURTHER WORK

As part of the strategic review undertaken by the directors of the company in the light of the success of the implementation of this programme of work, it was decided that the automated shop floor data capture system relating to the product progress and completeness at each stage of manufacture, should be integrated into the system.

This facet of the work was partially undertaken concurrently to the implementation of the Knowledge Based System and consisted of a network of embedded systems, with touch screen and barcode interfaces. This facilitated real time shop floor information to be directly integrated into the Knowledge Based System which allowed costing to be validated by means of an organised comparison of the estimated cost with the actual cost for the orders won, to achieve more accurate and consistent cost estimations

The system implemented is shown in Appendix C and development of rules for comparison analysis within the Knowledge Based System will be the next phase of work.

GLOSSARY

[1] **EJMA** stands for the Expansion Joints Manufacturers Association. It also describes a software program in the context of this report.

[2] **CLIPS** stands for C Language Integrated Production System.

[3]**Ontology** has been defined as the meta-knowledge that describes everything known about the problem domain

[4]**XML** stands for Extensible Mark-up Language. It is a web standard designed to transport and store data.

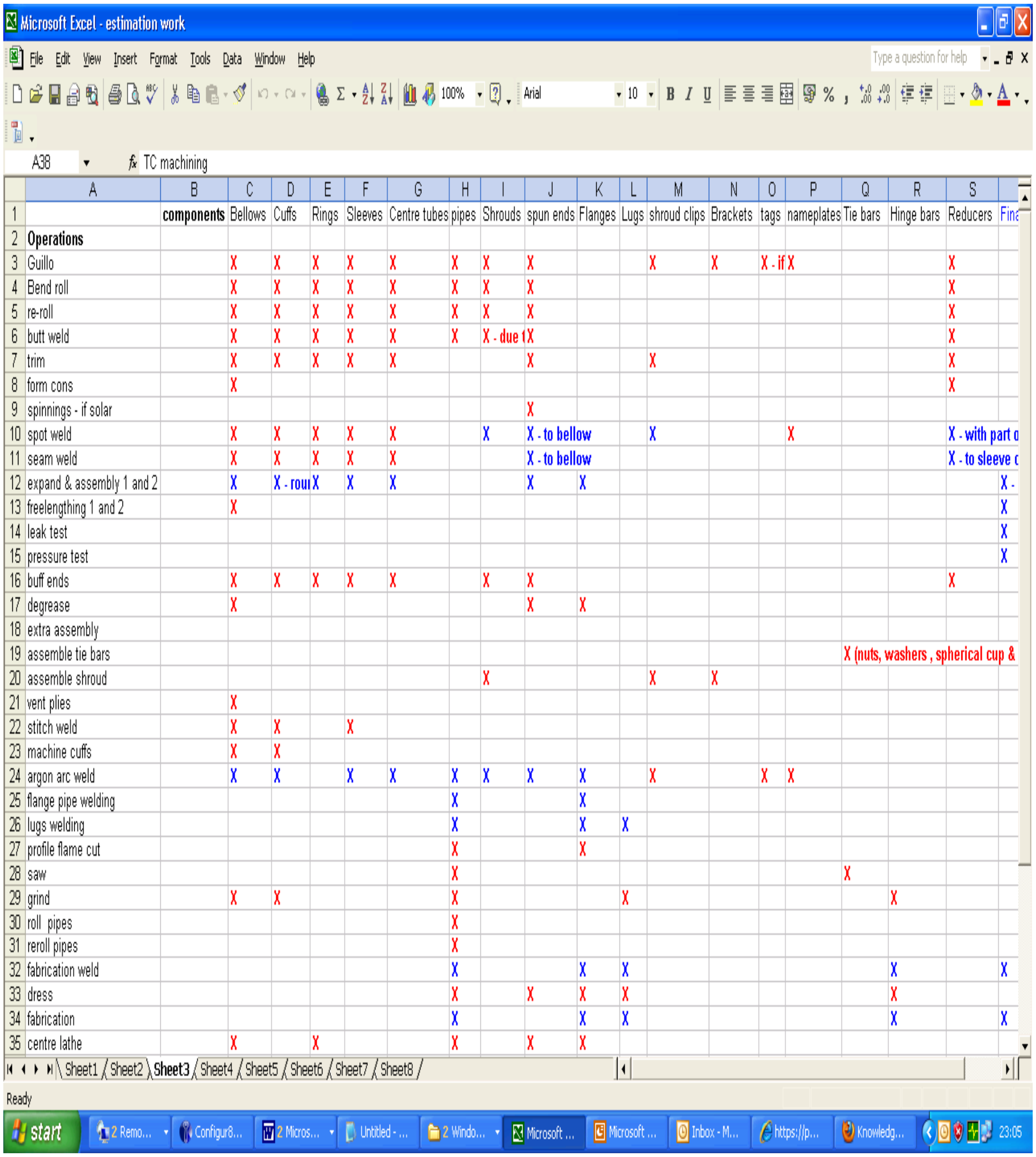
[5]**CAD** stands for Computer Aided Design

[6]**UML** stands for Unified Modelling Language**.** It is the industry standard used in the analysis and design stages of software development

[7]**GDL** stands for General-purpose Declarative Language. A KBE systems development platform from Genworks.

[8] **SMLib** is an advanced geometric modeling kernel.

APPENDIX A



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1		**components**	Bellows	Cuffs	Rings	Sleeves	Centre tubes	pipes	Shrouds	spun ends	Flanges	Lugs	shroud clips	Brackets	tags	nameplates	Tie bars	Hinge bars	Reducers	Fina
2	**Operations**																			
3	Guillo		X	X	X	X	X	X	X	X			X	X	X - if	X			X	
4	Bend roll		X	X	X	X	X	X	X	X									X	
5	re-roll		X	X	X	X	X	X	X	X									X	
6	butt weld		X	X	X	X	X	X	X - due t	X									X	
7	trim		X	X	X	X	X			X			X						X	
8	form cons		X																X	
9	spinnings - if solar									X										
10	spot weld		X	X	X	X	X		X	X - to bellow			X			X			X - with part o	
11	seam weld		X	X	X	X	X			X - to bellow									X - to sleeve c	
12	expand & assembly 1 and 2		X	X - rou	X	X	X			X	X									X -
13	freelengthing 1 and 2		X																	X
14	leak test																			X
15	pressure test																			X
16	buff ends		X	X	X	X	X		X	X									X	
17	degrease		X							X	X									
18	extra assembly																			
19	assemble tie bars																X (nuts, washers , spherical cup &			
20	assemble shroud								X				X	X						
21	vent plies		X																	
22	stitch weld		X	X		X														
23	machine cuffs		X	X																
24	argon arc weld		X	X		X	X	X	X	X	X		X		X	X				
25	flange pipe welding							X			X									
26	lugs welding							X			X	X								
27	profile flame cut							X			X									
28	saw							X									X			
29	grind		X	X				X				X						X		
30	roll pipes							X												
31	reroll pipes							X												
32	fabrication weld							X			X	X						X		X
33	dress							X		X	X	X						X		
34	fabrication							X			X	X						X		X
35	centre lathe		X		X			X		X	X									

Figure 4.3: *Component part-Operations* matrix

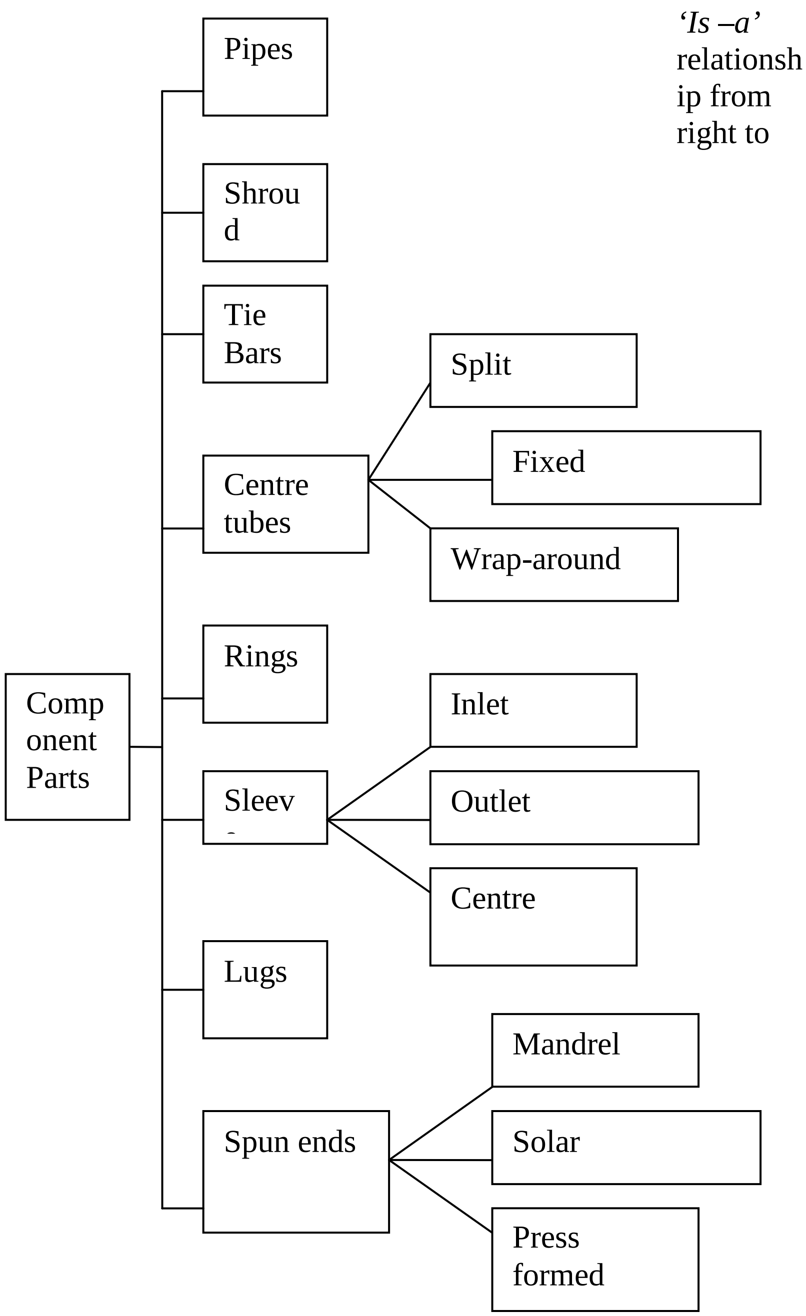
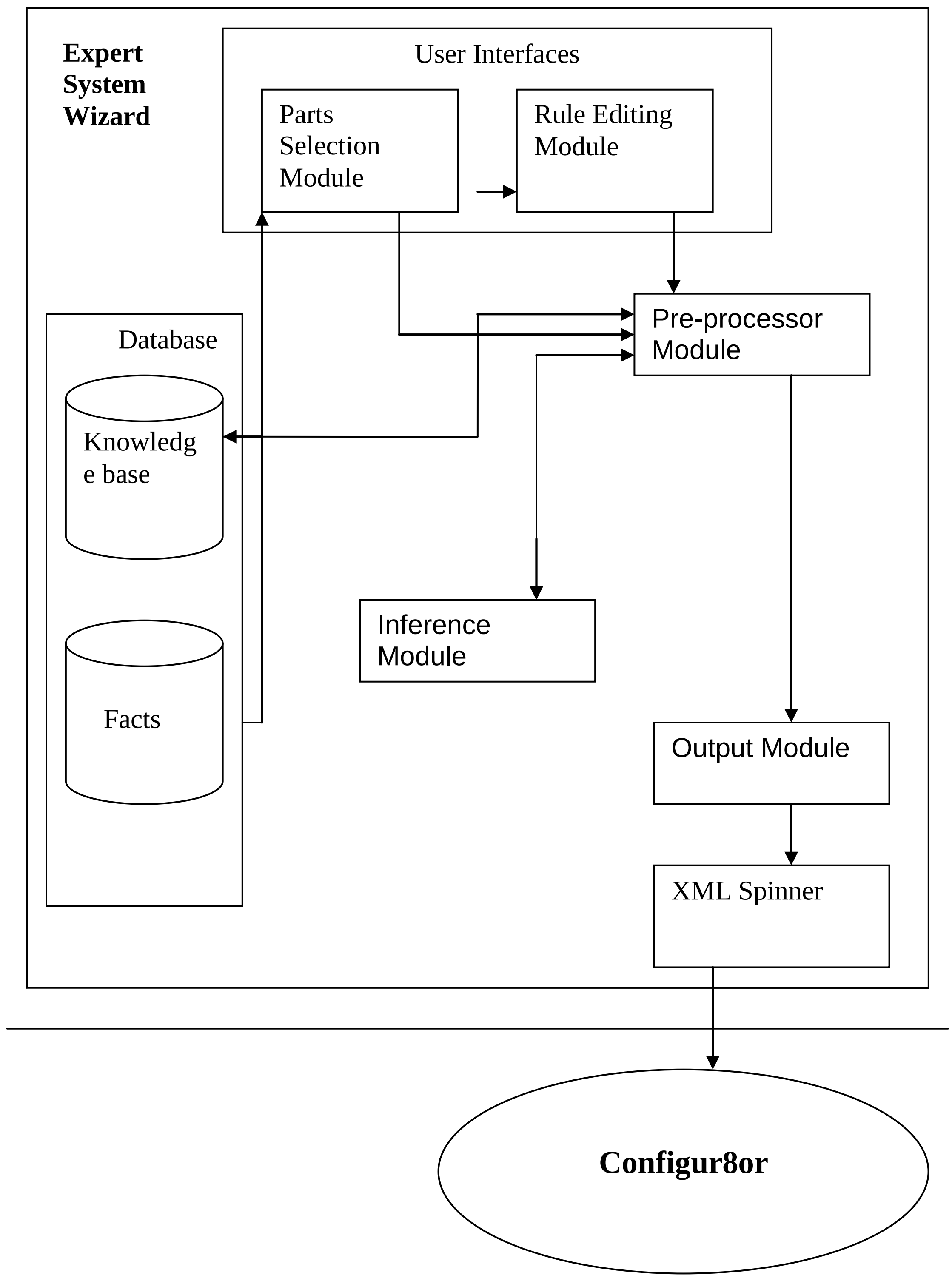
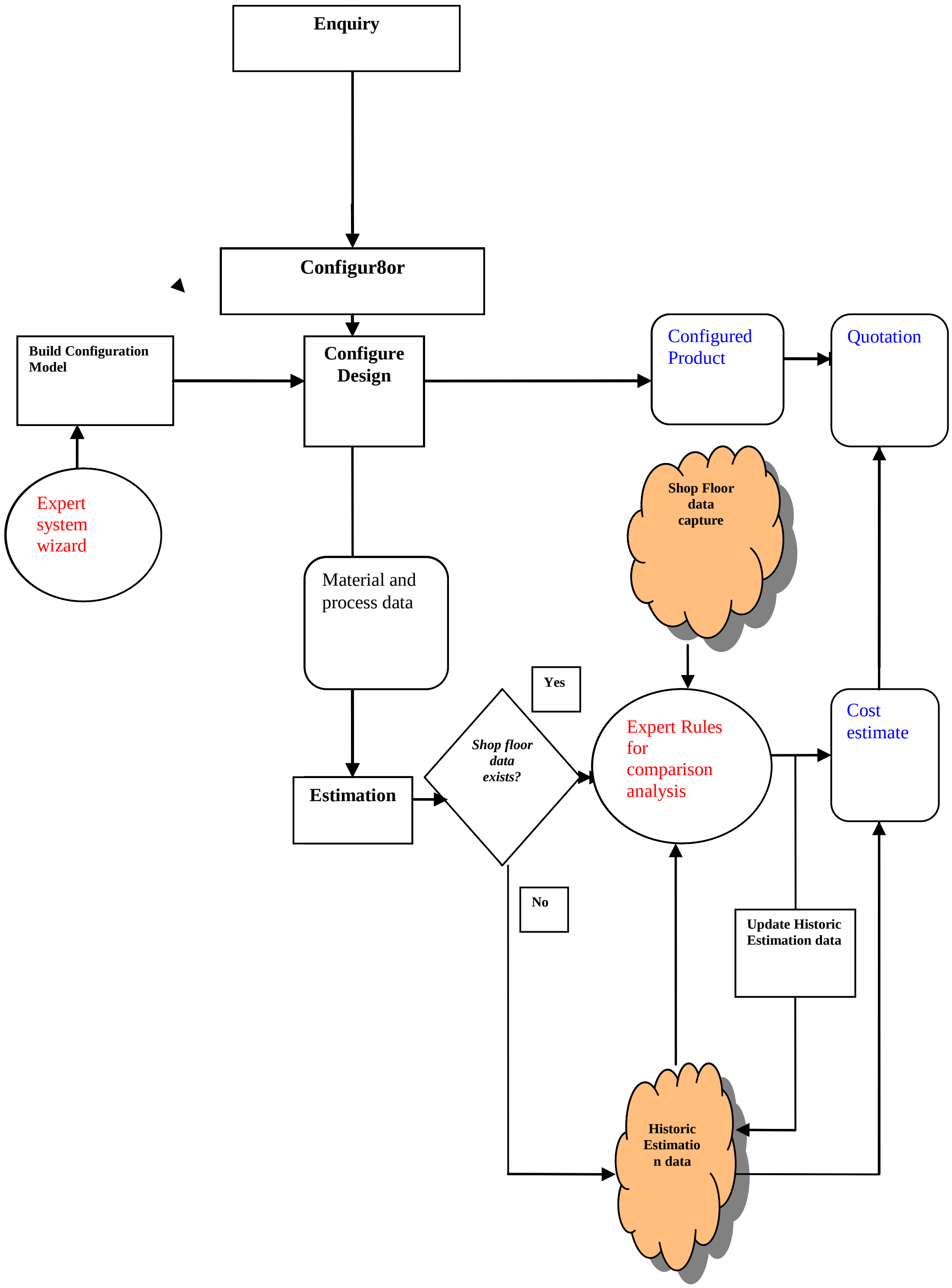


Figure 4.6 Composition tree for component parts

APPENDIX B - EXPERT SYSTEM ARCHITECTURE



APPENDIX C - REPRESENTATION OF EMBEDDED SYSTEM



APPENDIX D – DETAILED FUNCTIONAL LAYOUTS

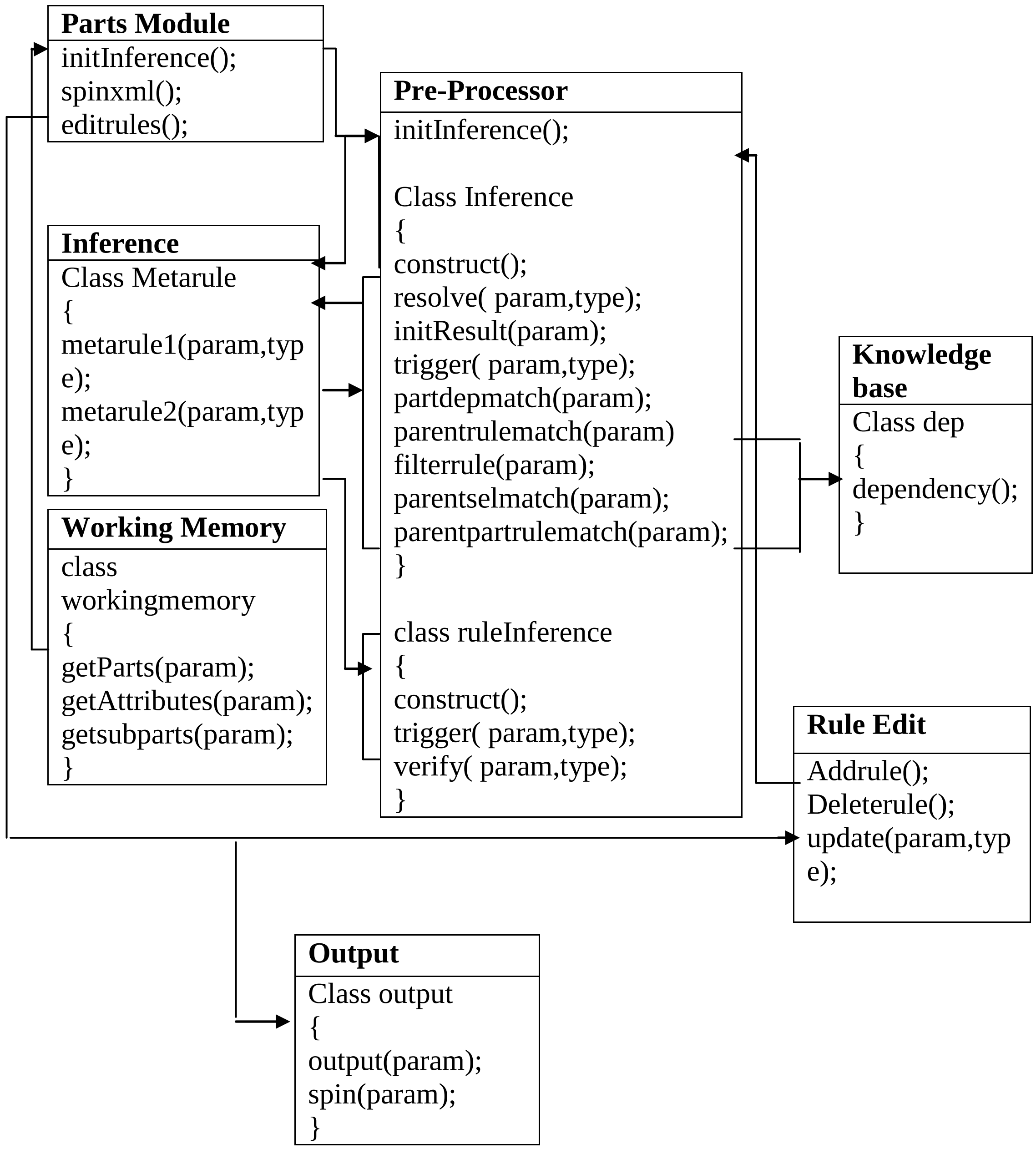


Figure 6.1: UML Class diagram

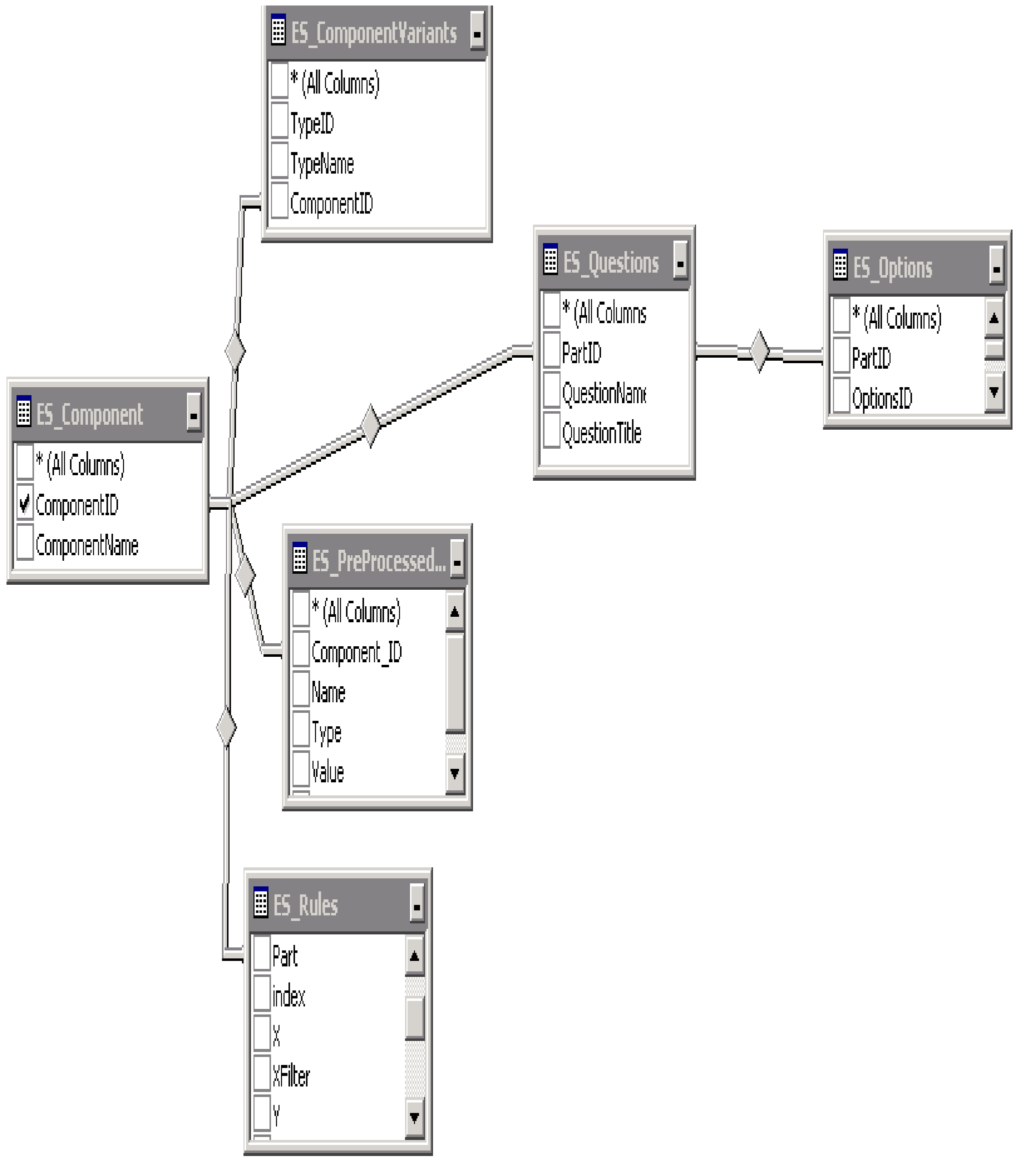


Figure 6.2 Database Schema

REFERENCES

[1] Engelmore, R. and Feigenbaum, E. (1993, May) *Expert Systems and Artificial Intelligence.* Available at: http://www.wtec.org/loyola/kb/c1_s1.htm (Accessed: 3 February 2011).

[2]VerDuin, W. (2001) *Knowledge based systems for manufacturing, Intelligent Manufacturing*. Available at: http://www.lionhrtpub.com/IM/IMsubs/IM-2-95/knowledge.html (Accessed 3 February 2011)

[3] Buchanan, B. and Smith, R. (1988) *Fundamentals of Expert Systems.* Available at : http://media.wiley.com/product_data/excerpt/18/04712933/0471293318.pdf (Accessed 3 February 2011)

[4] Schreiber, G., Wielinga, B., de Hoog, R., Akkermans, H. and Van de Velde, W. "CommonKADS: A Comprehensive methodology for KBS development", 1994 Volume 9, Issue 6, pp 28-37

[5] Giarratano, J. and Riley, G. "Design of expert systems", in *Expert Systems principles and Programming*, 4th ed. Thomson course technology, 2005

[6] Alison, A. (1994) *Expert system architecture*. Available at: http://cinuresearch.tripod.com/ai/www-cee-hw-ac-uk/_alison/ai3notes/subsection2_5_2_1.html. (Accessed 3 February 2011)

[7] Lakner, R. (no date) *Knowledge Based Systems*. Available at http://acg6415.wikispaces.com/file/view/Knowledge-based+systems.pptx (Accessed 3 February 2011)

[8] Chan, D. and Lewis, W. "Application of expert systems in manufacturing processes selection", in *6th International Conference on Manufacturing Engineering, Melbourne,* 1995

[9] Boothroyd, G., Dewhurst P., Knight, W.A. "Design for manufacture and computer aided design", in *Product Design for Manufacture and Assembly*, 2nd ed revised. CRC Press, 2001

[10] Nikolopoulos, C. "Expert system shells and programming environments: Expert systems", in *Introduction to first and second generation and hybrid knowledge based systems*, CRC Press, 1997

[11] Moutinho, L. "Tools for expert systems development", in *Expert systems in tourism marketing,* Routledge, 1996

[12] John, A. (no date) *Expert system shells.* Available at: http://www.cwa.mdx.ac.uk/bis2040/JohnLects/01bkb08p.ppt.

(Accessed 3 February 2011)

[13] Vlaanderen, M.J. "Automated knowledge acquisition for expert systems: An overview"*.* Unpublished PhD thesis. Erasmus University Rotterdam, 1990.

[14] Gaines, B. and Shaw, M. Integrated knowledge Acquisition Architectures. *Journal of Intelligent Information Systems, 1 (1) 1992*

[15] Blythe, J., Kim, J., Ramachandran, S. and Gil, Y. "An integrated environment for Knowledge Acquisition", *In Proceedings of International Conference on Intelligent user Interfaces,* 2001

[16] Swartout, B. and Gil, Y. "EXPECT: Explicit Representations for Flexible Acquisition", *In Proceedings of the Ninth Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'95)* Banff, Canada.

[17] Rafea, A. et al "Automatic knowledge acquisition tool for irrigation and fertilization expert systems", *Expert Systems with Applications 24,* 2003, pp. 49-57

[18] Marcus, S. "SALT: A knowledge-acquisition tool for propose-and-revise systems," in *Automating Knowledge acquisition for expert systems* S.Marcus (ed), 1988, pp. 81-121.

[19] Bennett, J.S. "ROGET: A knowledge based system for acquiring the conceptual structure of an error reporting expert system", in *Journal of Automated Reasoning* (1), 1985 pp. 49-74

[20] Davis, R. "Applications of meta-level knowledge to the construction, maintenance, and use of large knowledge bases", PhD thesis, Stanford University, 1976

[21] Trinkunas, J., Vasilecas, O. "Building ontologies from relational databases using reverse engineering methods", *International conference on computer systems and technologies*, 2007

[22] Musen, M. A., and Tu, S. W. "Problem-solving models for generation of task-specific knowledge acquisition tools", In J. Cuena (Ed.), *Knowledge-Oriented Software Design*, 1993, Elsevier, Amsterdam

[23] Musen, M. A."Overcoming the limitations of role-limiting methods," *Knowledge Acquisition*, 4(2), 1992, pp165-170.

[24] Swartout, B. and Gil, Y. “EXPECT: Explicit Representations for Flexible Acquisition” *In Proceedings of the Ninth Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW’95)* Banff, Canada, 1995

[25] Gil, Y., and Melz, E. “Explicit representations of problem solving strategies to support knowledge acquisition”, *ISI Technical report ISI/RR-96-436,* 1996

[26] Buchanan, B.G., and Shortliffe, E.H. “Rule based expert systems”, *The MYCIN experiments of the Stanford heuristics programming project*, 1984

[27] Marcus, S., Stout, J., McDermott, J. “VT: An expert system elevator design that uses knowledge based back tracking”, *In Artificial intelligence in Engineering design*,1992, vol. 1, pp 317-55.

[28] Puerta, A.R, Egar, J.W., Tu, S.W. and Musen. M “A multiple method knowledge-acquisition shell for automatic generation of knowledge-acquisition tools”, 1992

[29] MacGregor, R. “A deductive pattern matcher”, *In proceedings of the 1988 National Conference on artificial intelligence*, St Paul, MN, 1988

[30] MacGregor, R. “The evolving technology of classification-based knowledge representation systems”, *Principles of semantic networks: Explorations in the representation of knowledge*. Morgan Kaufmann, San Mateo, CA, 1991

[31] Miton, N.R (2003) .*Knowledge Acquisition.* Available at: http://www.epistemics.co.uk/Notes/63-0-0.htm", (Accessed 3 February 2011)

[32] Miton, N.R (2003) .Knowledge Modelling. Available at: http://www.epistemics.co.uk/Notes/87-0-0.htm”, (Accessed 3 February 2011)

[33] Wagner et al, “The impact of problem domains and knowledge acquisition techniques: a content analysis of Production/Operations Management expert system case studies”, in *Expert systems with Applications 24, 2003, pp 79-86*

[34] Milton, N.R. “Knowledge Technologies”, volume 3 (2008)

[35] Tecuci, G. ”Steps towards automating knowledge acquisition for expert systems”, *In proceedings of the AAAI-91 Workshop “Knowledge Acquisition: From Science to Technology to Tools”, Anaheim, CA,*1991

[36] Milton N.R. (2008, December) “PCPACK5: Technical Overview”. Available at: http://www.epistemics.co.uk/Notes/207-0-0.htm (Accessed July 20th 2010)

[37] Kingsman, B. G. and De Souza, A. “A Knowledge based decision support systems for cost estimation and pricing decisions in versatile manufacturing companies”,. *Int. J. Production Economics 1997,53 119-139*

[38] Shehab, E. M. and Abdalla, H.S. “Manufacturing cost modelling for concurrent product development”, *Int. J. Robotics and computer integrated manufacturing, 2001, 17(4), 341-353*

[39] Nevins, J. L. and Whitney, D. L. Concurrent Design of products and processes, 1989 (McGraw-Hill, New York)

[40] Pham, D. T. and Ji, C. “A concurrent design system for machined parts”, *Proc. Instn, Mech. Engrs, Part B, Journal of Engineering Manufacture*, 1999,213 (B8), 841-846

[41] Shehab, E. M. and Abdalla, H.S. “A design to cost system for innovative product development”, *Proc. Instn. Mech. Engrs. Vol 216 part B: J Engineering Manufacture*

[42] Niazi, A. and Dai, J.S. “Product cost estimation: Technique classification and methodology review”, *Journal of manufacturing science and engineering, May 2006, vol. 128*

[43] Hundal, M. S., “Design to cost,” *Concurrent Engineering: Contemporary Issues and Modern Design Tools, H. R. Parsei and W.G. Sullivan, eds., Chapman and Hall, London,* 1993, *pp. 330-351*

[44]Lewis, J. ”Metrics Mapping Cuts Estimating time,” Des. News, 55(18), pp. 107-110, 2000

[45] Zhang, Y.F., and Fuh, J. Y. H. “A neural network approach for early cost estimation of packaging products,” Comput. Ind. Eng., 34(2), 1998, pp. 433-450.

[46] Kim, J. and Gil, Y. "Acquiring problem-solving knowledge from end users: Putting Interdependency Models to the test", Proceedings of the 7th National Conference on Artificial Intelligence (AAAI-2000)

[47] Eriksson, H. et al. "Task modeling with reusable problem solving methods". *Artificial Intelligence* 79,1995, pp 293–326.

[48] Marcus, S., and McDermott, J. SALT: A knowledge acquisition language for propose-and-revise systems. *Artificial Intelligence* 39(1),1989, pp 1–37.

[49] Kim, J., and Gil, Y. Deriving expectations to guide knowledge base creation. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 1999, pp 235–241

[50] Tallis, M. and Gil, Y. Designing scripts to guide users in modifying knowledge-based systems. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence* (AAAI-99) July 1999, Orlando, Florida

[51] Gil, Y., and Tallis, M. "A Script-based Approach to Modifying Knowledge-based Systems", *In Proceedings of the Fourteenth National Conference on Artificial Intelligence,* 1997

[52] Blythe, J. "Extending role limiting approach: Supporting end users to acquire problem solving knowledge". *In ECAI 2000 Workshop on Applications of Ontologies and problem solving methods, 2000*

[53] Brewster et al. **"**Knowledge Acquisition for Knowledge Management: Position Paper", Unpublished

[54] Stefanou C. J. "System Development Life Cylce", *In Encyclopedia of Information Systems, Volume 4, 2003*

[55] Yang, H. "Information / Knowledge Acquisition methods for decision support systems and expert systems". *In Information Prcoessing & Management, Vol 31, No 1, 1995, pp 47-58,*

[56] Walczak, S. Knowledge acquisition and knowledge representation with class: the object – oriented paradigm. *In Expert Systems with Applications 15, 1998, pp 235 - 244*

[57]Chen, Z. "Knowledge acquisition assisted by CLIPS programming", *In Engineering Applications of Artificial Intelligence 12, 1999, pp 379-387*

[58] Gebus. S, and Leiviska, K. "Knowledge acquisition for decision support systems on an electronic assembly line",. *In Expert Systems with Applications 36 (2009) 93–101*

[59] Jeng, B. et al. "Interactive Induction of Expert knowledge", *In Expert Systems With Applications, Vol. 10, No. 3/4, pp. 393--401, 1996*

[60] Wagner, W.P. "Knowledge Acquisition", in *Encyclopedia of Information Systems.* 2004

[61] Liao, S. "Expert system methodologies and applications—a decade review from 1995 to 2004", *In Expert Systems with Applications 28, 2005, pp 93–103*

[62] Akerkar, R and Sajja, P. *Knowledge Based Systems*. Jones & Bartlett Learning, 2009

[64] Schreve, K. et al. "Manufacturing cost estimation during design of fabricated parts", *In Proceeding from the Institution of Mechanical Engineers Vol 213 Part B, 1999*

[65] Wu, C. and Srihari, K. "Automated knowledge acquisition for the surface-mount PWB assembly domain", *In Journal of Intelligent* manufacturing (1996) 7, pp 393-398

[66] Ayala, A.P. "Acquisition, representation and management of user knowledge", *In expert systems with Applications 37, 2010 pp 2255 - 2264*

[67] Mohamed, A. and Celik, T. "Knowledge based-system for alternative design, cost estimating and scheduling", *In Knowledge –based systems 15*, 2002, pp177-188

[68] Martin, B. "Benefits from expert systems: An exploratory investigation", *In Expert systems with Applications Vol. 11, No. 1, pp.53-58, 1996*

[69] Nurminen, J.K et al. "What makes expert systems survive over 10 years – empirial evaluation of several engineering applications", *In Expert Systems with applications 24 (2003) 199-211.*

[70] Poli, C. Design for manufacturing: a structured approach, Vol 1. *Butterworth-Heinemann, 2001*

[71] Medsker, L. et al. "Knowledge Acquisition from Multiple Experts: Problems and Issues", *In Expert Systems With Applications. Vol. 9, No. 1, pp.35-40, 1995*

[72] Kotiadis, K. “Conceptual modelling: Knowledge acquisition and model abstraction”, *In proceedings of the 2008 winter simulation conference. S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.*

[73] Biggam, J. “Exploiting Soft Systems Methodology (SSM) and knowledge types to facilitate knowledge capture issues in a web site environment”, *In proceedings of the 35th Hawaii International Conference on System sciences - 2002*

[74] Kotiadis, K. “Using Soft Systems methodology in conceptual modelling: A case study in intermediate health care”, Unpublished

[75] Checkland, P. and Poulter, J. *Learning for Action: A short definitive account of soft systems methodology and its use for practitioners, teachers and students,* John Wiley, 2006.

[76] Lester, S (2008) “Soft systems methodology,” Available at: http://www.sld.demon.co.uk/ssm.pdf, (Accessed July 20th 2010)

[77] Valdez, J. et al. “Object Oriented Design and Implementation of Inference Engine for Fuzzy systems”, Unpublished.

[78] Lanovaz, D. and Szafron, D. “An object-oriented inference engine for PROLOG”, *In Journal of systems and software, Volume 19, Issue 1, September 1992, pp13-25*

[79] Chougule, R. and Ravi, B. Casting cost estimation in an integrated product and process design environment. *In international journal of computer integrated manufacturing, Vol. 19, No. 7, Oct-Nov 2006, pp 676 - 688*

[80] Pakiarajah, V. et al. Conflict Resolution Techniques for Expert Systems Used to Classify Remotely Sensed Satellite Images,

[81] Phillips, E.M., Pugh, D. “How to do research” in How to get a PhD: A handbook for students and their supervisors, 4th edition, Open University Press 2006

[82] Rajasekar, S. et al. *Reseach Methodology.* Available at: http://www.scribd.com/doc/6949151/Research-Methodology. (Accessed May 30 2011)

[83] Genworks Intenational (2011) *Genworks GDL software*. Available at: http://www.genworks.com/downloads/gdl-pamphlet.pdf. (Accessed May 29 2011)

[84] Cooper, D., et al. “Keys to success with Knowledge-based Techniques”, *In Society of Automotive Engineers Incorporated.* SAE Paper Number 2008-01-2262, 2008.

[85]Maine, E., Probert, D., Ashby, M. “Investing in new materials: a tool for technology managers” *in Technovation 25* (2005) 15–23

[86] Granta Material Intelligence. (2011) *The Ashby methods – the power behind the CES selector.* Available at: http://www.grantadesign.com/products/ces/ashby.htm. (Accessed 29th May 2011)