

**Developing an equipment selection decision  
tool as the basis for knowledge management at  
a material handling equipment supplier**

by

**LS VAN RENSBURG**

**26231728**

**Submitted in partial fulfilment of the requirements for  
the degree of**

**BACHELORS OF INDUSTRIAL ENGINEERING**

in the

**FACULTY OF ENGINEERING, BUILT  
ENVIRONMENT AND INFORMATION  
TECHNOLOGY**

**UNIVERSITY OF  
PRETORIA**

**SEPTEMBER 2011**

## Executive Summary

Loadlift Equipment is a retailer of material handling equipment and servicing thereof. Their main focus is to satisfy customer requirements with the correct products. The company has a need for a tool or standard procedure to optimise new sales team induction and training time and reduce the chance of experienced employees leaving the company.

The aim of the project is to create a knowledge-based expert system for the selection of material handling equipment. The system will use inputs from the customer and process the inputs to deliver the best material handling equipment to meet requirements.

The success of the project relies on the elicitation of the correct field knowledge and experience from the founders, and to use artificial intelligence methods for the calculation of the preferred equipment with its related appropriateness to the requirements.

Fuzzy logic is explained in depth in the literature review. And the concept is applied to the project.

## Table of Contents

Executive Summary .....	2
List of Figures .....	4
List of Tables .....	4
Chapter 1: Introduction .....	5
1.1. Background .....	5
1.2. Project requirements.....	5
1.3. Project aim.....	6
1.4. Project scope .....	7
1.5. Deliverables.....	7
1.6. Project plan.....	8
Chapter 2: Literature review .....	10
2.1. Material handling equipment selection .....	10
2.2. Software development process.....	10
2.3. Requirements and specifications .....	11
2.4. Software design .....	12
2.5. Software construction.....	13
2.6. Rule-based expert systems .....	13
2.7. Inference engine .....	14
2.8. Fuzzy expert systems .....	14
2.8.1. Fuzzy logic.....	14
2.8.2. Fuzzy sets.....	14
2.8.3. Fuzzy set membership functions.....	15
2.8.4. Fuzzy rules .....	16
2.8.5. Fuzzification .....	17
2.8.6. Intersection fuzzy sets (AND operator) .....	17
2.8.7. Modelling of fuzzy expert systems .....	17
2.9. Past methods used in the design of knowledge-based expert systems for the selection of material handling equipment .....	18
Chapter 3: Development of conceptual design .....	20
3.1. Functional specification (Requirements analysis).....	20

3.1.1	Description of general requirements .....	20
3.1.2.	External interface requirements .....	21
3.1.3.	Functional requirements.....	21
3.1.4.	Other Non-functional Requirements.....	24
3.1.5.	List of Products .....	24
3.2.	Handling attributes and membership function design.....	25
3.2.1.	List of Crisp Attributes .....	26
3.2.2.	List of fuzzy attributes .....	27
3.3.	Knowledge-base .....	30
3.4.	Program operating environment.....	31
3.5.	Testing and validation .....	34
Chapter 4: Conclusion and findings.....		34
References.....		35

## List of Figures

Figure 2.1.....	11
Figure 2.2.....	12
Figure 2.3.....	13
Figure 2.4.....	16
Figure 2.5.....	19
Figure 3.1.....	25

## List of Tables

Table 3.1.....	28
----------------	----

# Chapter 1: Introduction

## 1.1. Background

Loadlift Equipment was established with the guiding principles of quality, integrity, service and commitment to clients. They strive to bring the best products and services to the material handling and forklift market.

Loadlift Equipment provides the following products and services:

- Used forklifts
- Material handling equipment
- Parts
- Repairs and maintenance

Their first commitment has always been client satisfaction. Their mission is to offer their clients with the most exceptional material handling equipment products, services and client service in the industry. They strive at all times providing the client with the highest quality equipment and best service for all their forklift and warehousing needs. Loadlift Equipment's main focus is on supplying clients such as warehousing and container yard industries. The company is still small but aims to grow their market share against their bigger competitors.

Loadlift Equipment's sales department is dedicated to establish true partnerships with their clients while offering exemplary services and superior products at a truly competitive price. One of the most important goals of the department is to offer the correct product for the client's application requirements and advice that clearly puts the needs of the client above the short term gains of the company. To accomplish this, the technical sales representatives require broad technical product knowledge on how the products can satisfy the individual needs of the clients. For a sales representative to know all the ins and outs of each industry and their different material handling needs, he typically must have years of experience and knowledge of the specific products and their different applications in the various industries the company serves.

## 1.2. Project requirements

Loadlift Equipment has identified a major problem in the materials handling industry. Larger material handling companies seem to have a tendency to attract their technical sales representatives from each other by offering higher basic salaries, and/or improved staff benefits, making it difficult for the smaller competitors to keep their experienced technical sales reps. As a result the smaller companies who can't afford to employ the older experienced technical sales reps, are forced to employ and train the younger generation. As soon as these new recruits have their three to five years experience, they get head hunted by the larger companies again. Some big companies

rather poach than train because they want to reduce the unproductive time of new trainees and thus improve their profit margin.

Veteran technical sales reps are getting older and starting to retire, thus creating a need for younger new technical sales reps. Because these younger sales skills are not generally available, most companies are struggling to maintain a quality sales department. Inexperienced sales reps are inclined to provide incorrect advice and sell the wrong equipment to the client, ultimately creating a loss in client after sales.

Loadlift Equipment employs sales representatives with some marketing background. When employment commences they are trained on how the different material handling equipment can be used, their different applications and the different terms used in the industry. Normally it takes a new technical sales rep up to three months to start producing sales for the company. In those three months the company loses potential clients, which could have been prevented if the sales representative had the experience and knowledge on what the products could offer when he started at the company.

Loadlift Equipment currently train the sales reps by teaching them the different machines the company offers, their application and the calculations needed to determine what length of forks on the machine is needed and the weight capacity of the equipment. Calculations for mast height, turning radius and floor ramp angles in the work environment are also taught.

Loadlift Equipment is in search for a means to eliminate job hopping of experienced sales representatives and reducing the time it takes new sales reps to make intelligent sales. Loadlift Equipment believes that a system that can use the knowledge and key calculations from its founders to improve client satisfaction with regards to the selection of the best equipment for each application. The problem can thus be stated as the need for a tool or standard procedure to optimise new sales team induction and training time and reduce the chance of experienced employees leaving the company.

### **1.3. Project aim**

The aim of the project is to create an analysis and decision making (diagnostic) tool based on best practise, available technology and some practical experience. The tool should be able to analyse client requirements and recommend a product that optimises the objective function of the client (lifecycle cost, minimise downtime, flexibility, etc.). The tool will minimise the risk of the young sales rep making calculation errors, and reduces the need for a well qualified technical sales representative at the company. This also reduces the chance that a sales representative can give the client incorrect advice on product selection. The system can be considered as a knowledge management system, continuously updated with new techniques, experience and trends, supporting a learning organisation.

The following objectives describe the aim of the project:

- Determine the calculations and information needed for the three different industry applications, i.e. warehousing and shipyards, being Loadlift Equipment's target markets.
- Develop a program that uses the information to analyse requirements for a specified solution and, based on key design parameters, determine the best solution.
- Design a user interface for the sales reps to understand and to use the program to its full potential.

The system will fast track the training of new young sales representatives and enables them to provide advice with confidence, resulting in satisfied customers and increased sales. Loadlift Equipment will be able to protect and continuously update its expertise and knowledge in a tool that can't be poached like a veteran technical sales representative. It is believed that the tool can provide the company with a competitive edge in the market.

#### **1.4. Project scope**

The scope of the project includes the following:

- The design and testing of the product selection program
- Developing a user manual and training material
- Plan to regularly review and update the questions, input format, calculations, objective functions and outputs (maintaining the knowledge base)
- Presentation to and training of users and managers
- Project documentation

The following are exclusions from the project scope:

- Training of sales reps in the art of selling

#### **1.5. Deliverables**

The key deliverables of the project are stated as follows:

- Proposal
- Requirements analysis
- Design of program and user interface
- Testing
- Training and presentation to the company
- Final project report
- Presentation for University of Pretoria

## 1.6. Project plan

The following steps will be followed when planning and implementing the project:

### Phase 1: Initialization of project and proposal presentation

- Define the problem statement
- Perform a preliminary literature study on methods and topics related to project
- Scope project and finalise project proposal
- Present the proposal and perform changes
- Hand in proposal

### Phase 2: Data analysis

- Observe and investigate sales operations
- Determine their constraints and user needs
- Establish design criteria for the program
- Document findings in a system specification for the user
- Receive feedback from users and stakeholders on the system specification
- Analyse the requirements of the complete system specification
- Elicit information from stakeholders and users to clarify ambiguities found after analysis
- Document the detailed requirements in a functional specification

### Phase 3: Design and construction of program

- Develop a system level concept of the program structure (defining interfaces, subsystems and their functions, and the data flows between them)
- Present concept design for approval and modifications to managers
- Subdivide the design process (Top level designing) and define each sub-function and its specific module
- Map out and define data flow and data storage
- Define all algorithms and mathematical processing
- Define data and variables for the interfacing of modules
- Document detailed design
- Have a critical design review for formal approval
- Code the program in partitions (modules)
- Test the modules after coding (unit testing) and resolve issues
- Integrate the modules into subsystems
- Test integration and resolve issues
- Integrate the system
- Document findings and issues
- Implement beta version for testing and validation

### Phase 4: Validation and verification of the program

Lawrence van Rensburg

Page 8 of 35



- Validate if the program meets the requirements specified initially
- Verify if the program meets the design criteria
- Redesign the final program after data is analysed

#### Phase 5: Program review and finalisation

- Develop user manual
- Present the final program to users with a training session
- Receive feedback and recommendations for changes
- Perform required changes to the program
- Present the program to board members

#### Phase 6: Presentation

- Document the changes made to the program
- Document the presentation outcome
- Write the final document
- Design the slideshow
- Make poster
- Write presentation
- Prepare presentation
- Present the final presentation to the examiners of the project
- Receive feedback from examiners
- Conclusion of project

## Chapter 2: Literature review

### 2.1. Material handling equipment selection

Mirhosseyni (2009) defines a material handling system as, “A Material Handling (MH) system is responsible for transporting materials between workstations with minimum obstruction and joins all workstations and workshops in manufacturing and storage systems by acting as a basic integrator”. From this definition we become aware of the importance a MH system has in the industry. A MH system defines the link between operations and processes; it gives value to the way we move our material. The associated value becomes more efficient by reducing lead times, improving the travel frequency between operations, increasing productivity and utilising manpower. Thus the selection of MH equipment is crucial in the design and use of a MH system. (Mirhosseyni, 2009)

Park (1996) states that “a key task in material handling system design is the selection and configuration of equipment, requiring extensive technical knowledge and systematic analysis”. The process for selecting MH equipment shows its complexity by the different facility, material and equipment constraints present. A person has to be an expert in the field and have the necessary credentials to provide a customer with the most suitable equipment for his application. (Park, 1996)

### 2.2. Software development process

A software development process defines the plan in designing the software to specification and implementing the software for the required use. There are different kinds of models which can be followed in developing the software. The classical phases in a software development process are the most common phases used in life-cycle models. The phases are defined as follows:

- The *requirements phase* is the process of eliciting requirements for what the software should comply with and the analysis there off.
- The *design phase* uses the specifications from the requirements to design the software architecture and produces a detailed concept design.
- The *implementation phase* is when the concept turns into the coding part of the construction of the software. Testing occurs between the coding phases and after the final program.
- The *deployment phase* is the installation of the software and when training of the end user occurs.
- The *maintenance phase* is where errors in the software are corrected and updated versions of the software are developed.

(DOAF, 2003, p. 2-3)

The shark tooth development model defined by DOAF (2003, p. 10-5) will be used for the purposes of this project. It incorporates prototyping in the development phase and as a result obtains feedback from the user and the managing body of the company. Therefore the developer can design the program as close as possible to the requirements specified by the stakeholders after each feedback session (prototyping).

Figure 2.1 illustrates the model defined by DOAF (2003, p. 10-5)

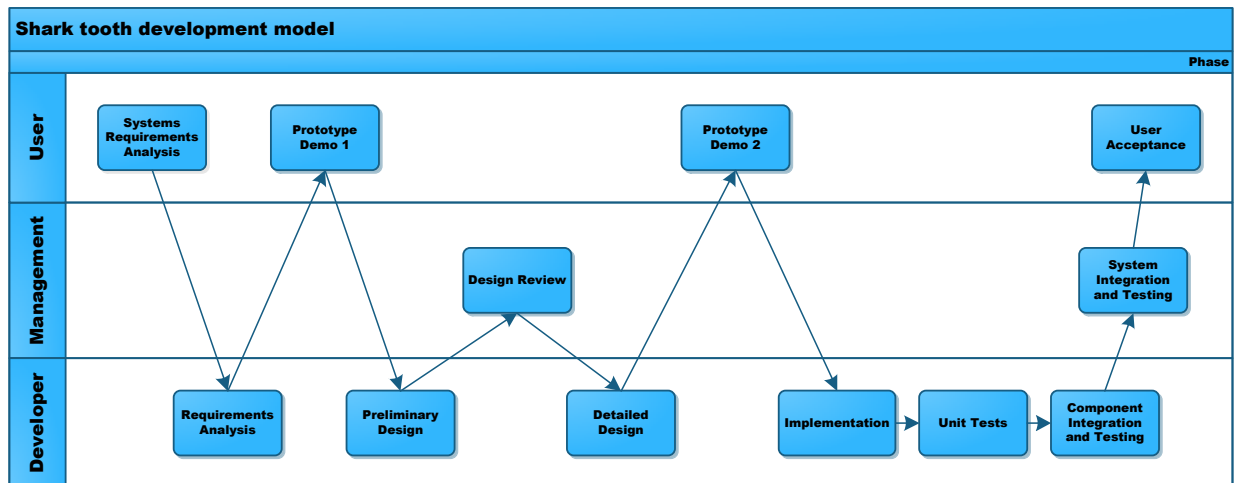


Figure 2.1 Shark tooth development model

### 2.3. Requirements and specifications

Before a program can be designed there has to be requirements to measure the validity of the outputs and to verify the process of inputs required from the user. It forms the basis of the design process and helps the developers to understand what the user wants from the final product. Communication is an important factor in the elicitation and analysis of the requirements. System specifications are used to provide the user with a common understanding on what perception the developer has on the requirements specified. It's written in a manner for the user to understand. System specifications are analysed and amended after the user identifies and clarify ambiguities (prototyping). After the user is satisfied with the requirements, the developer will compile a Functional specification in a higher language for the developer to understand. This document is then used to verify if the program meets the requirements specified. (DOAF, 2003, p. 10-6)

Figure 2.2 from the DOAF (2003, p.10-6) illustrates the followed process.

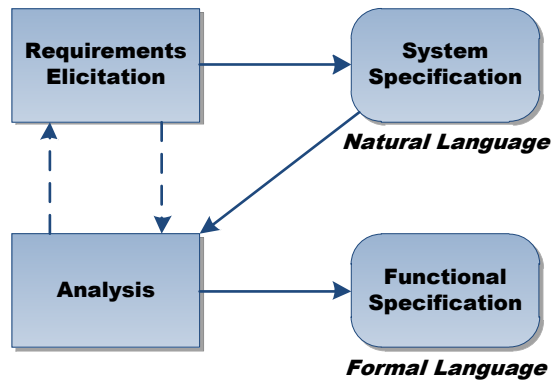


Figure 2.2 Requirements elicitation and analysis

The developer must address certain topics that should be discussed in the specifications. The IEEE Std 830-1998 lists the following topics to be addressed:

- *Functionality.* What shall the software do?
- *External interfaces.* How does the software interact with the user and other software?
- *Performance.* What performance attributes is expected?
- *Attributes.* What security and safety features should be used, and what is the maintainability of the software?
- *Design constraints imposed on an implementation.*

## 2.4. Software design

The design phase is where the planning of the program structure and objects are initiated. Two key levels are used in the design phase, namely systems level designing and program design. Systems level designing describes the hierarchy of the program structure. It describes the interface between the system and user, the sub-systems and their functions. It utilises the specification by designing the system as required to specification. Between the two levels prototyping takes place to clarify if the correct inputs and outputs where defined and if the required functions are performed by the correct sub-systems. (DOAF, 2003, p.10-7)

Program design adds more detail to the subsystems by dividing the processes and functions of the system into modules and defining them. Data flow, data storage, algorithms and modules are all defined in the design. After the concept design a critical review is prepared to identify inconsistencies, errors and wrongfully defined functions in the final program design. If no anomalies where found the design is approved and software construction can commence. (DOAF, 2003, p.10-7)

## 2.5. Software construction

A good developer always starts to code after a well formulated design process has been tediously developed and planned. The best practise for coding is to program the individual modules and functions first and then to integrate them into subsystems and then the entire system. Programming the modules individually gives the developer a chance to concentrate on the module until it's completed and to move on to the next one. If the developer codes everything as a hole it will result in functions being half done and won't provide him with an efficient manner in giving the user what he requires. (DOAF, 2003, p.10-8)

## 2.6. Rule-based expert systems

Abraham (2005) states that “conventional rule-based expert systems use human expert knowledge to solve real world problems that normally require human intelligence”. Therefore rule based expert systems use human expert knowledge and use different rules to determine the correct course of action to be taken in a certain area to solve expert problems through reasoning. Knowledge is captured from the human expert and converted into rules and constraints. The captured data and rules are stored in a knowledge base system to preserve and use the human expert knowledge. This minimizes the need for experienced human experts and the system also reduces the time needed for solving a problem. “Rules link given conditions to actions or outcomes” (Abraham 2005). An inference engine is used to find information in the knowledge base, which is related to the requirements stated, and to convert the information and rules into suggestions like a human expert would do. The system has a user interface and an explanation facility to show the user how the system concluded the results. (Abraham 2005)

Fig. 2.3 from Abraham (2005) presents the architecture of a simple expert system.

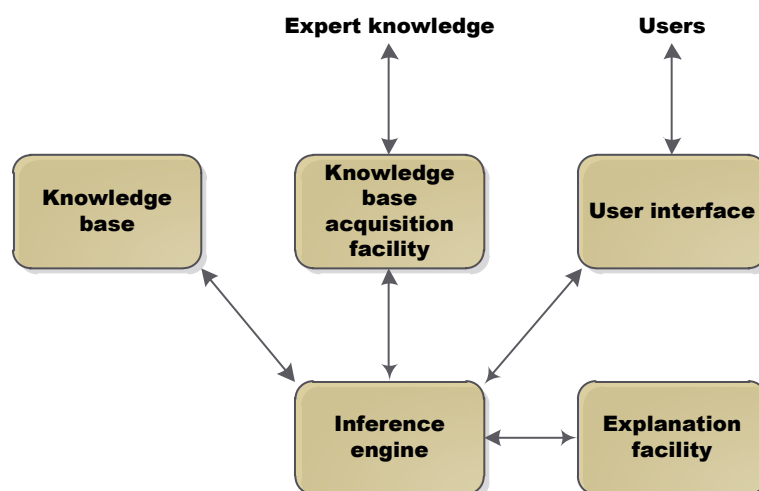


Figure 2.3 Architecture of a simple expert system

## 2.7. Inference engine

The inference engine uses if-then statements to compile rules for the knowledge base. These rules contain *premise* (if this is A) and *consequent* (then it will be B) statements. There are different types of inference engines. The *forward chain* inference engine will be explained. The inference engine will have a set of if-then rules, by using forward chaining the engine determines if the first rule is fired (true), if its fired it will use the data from the previous rule to determine if the next rule is met. The inference engine will go through the same process for the whole set. If all the rules in the set aren't fired the inference engine will go through the set again until all the rules are fired with the data from the input of the interface and the knowledge base. (Abraham 2005) Therefore as stated by Abraham (2005) "Forward chaining systems are primarily data-driven".

## 2.8. Fuzzy expert systems

### 2.8.1. Fuzzy logic

Eberhart and Shi (2007) provide the following definition: "Fuzzy logic comprises of fuzzy sets, which are a way of representing non-statistical uncertainty, and approximate reasoning, which includes the operations used to make inferences in fuzzy logic." From the definition we can derive that the reasoning behind fuzzy logic resembles the common sense reasoning of a human being by placing a degree of uncertainty on the premise and consequent rules. (Eberhart & Shi 2007)

Traditionally "crisp rules" where used in the programming logic for knowledge based systems. Crisp rules statements are only satisfied with a 1 or a 0. For example:

If x is A then y will equal B, if not then C.

The statement can therefore only have two outcomes. With fuzzy logic there is a degree of outcomes. For example:

We can't say that if it's cloudy then it's going to rain. We can say though, if it's *completely* cloudy then it can rain a *little bit*.

From the example, it can be said that cloudiness and rain are sets which have a degree of truth to them. The rule statement is defined in linguistic terms like a human would define a real world function. As a result fuzzification took place in the example when we changed the statement into a linguistic term. (Eberhart & Shi 2007)

### 2.8.2. Fuzzy sets

In traditional set theory, an element is either part of a set or not. In fuzzy set theory, an element is part of a set membership occurring by a degree. Let's take the set of *fastest sports cars* for instance. From research it can be said that a Bugatti Veyron does the quarter mile less than 10.1 seconds. Therefore it belongs to the set *fastest sports cars*. When we look at the VW Golf GTI we can relate that it doesn't belong to the set *fastest sports cars*, because it does the quarter mile in 14.5 seconds which is one of

the slowest times for a sports car at the moment. Using traditional set theory, sports cars can only be stated as fast or not fast. What happens when we look at the BMW M3 quarter mile time of 12.25? Does it belong to the set or not? That's where fuzzy logic is more appropriate in this case because it "assigns a degree of set membership to everyone." (Eberhart & Shi 2007) For that reason the degree of *fastest sports cars* membership for a Bugatti Veyron would be 1.0 because it's one of the fastest sport cars in the world. For The VW Golf GTI it's a bit more difficult because we can't assign 0 to the car because it's not the slowest sports car, so it will be given a degree of 0.1. Now we have a method of placing membership to the BMW M3. The car is still a very fast sports car even if it's not the fastest in the world and therefore the degree of membership would be 0.5. The fuzzy membership function can be written in the form:

$$\text{Fastest sports cars} = 1.0/\text{Veyron}, 0.5/\text{M3}, 0.1/\text{GTI}$$

For a more standardized way it can be written in the form:

$$\mu_{\text{Fastest sports cars}}(C) = m$$

which states that the membership value of  $C(\text{car})$  in the fuzzy set *Fastest sports cars* is  $m$ , where  $0 \leq m \leq 1$

It can be derived that statements about *fastest sport cars* can be applied to all sport cars. Suppose we provide the statement "that the best competitor in a quarter mile drag race, are the fastest sport cars." With traditional logic we would have inferred that the Bugatti Veyron would be the best and a VW Golf GTI would be the worst. The BMW M3 can't be part of the inference because it doesn't belong to a set. With fuzzy logic, inference is possible for all the cars and more. Therefore we would infer that to the degree that the statement is true, the Bugatti Veyron will be the most preferred competitor, and the BMW M3 will be the better preferred competitor and the VW Golf GTI the least preferred competitor. "Thus we are able to reason by degree, applying logical operations to fuzzy sets." (Eberhart & Shi 2007)

### 2.8.3. Fuzzy set membership functions

From the previous discussion it's noticed that a function can be drawn from the set on a numeric value. This function is called a fuzzy membership function and can be represented in two ways. Either by a graph or by specifying the membership function points. Figure 2.4 shows the graph of the function of the *fastest sports cars* set.

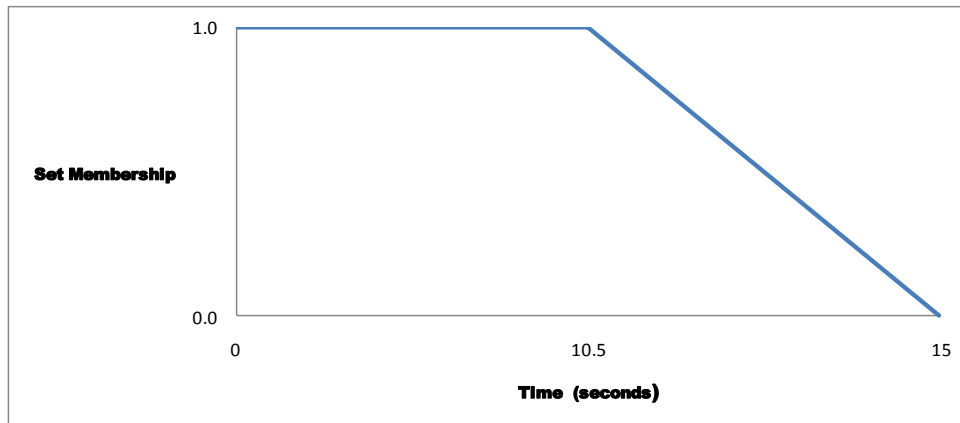


Figure 2.4 Fastest Sports Cars fuzzy membership function

The 1.0 value in the set membership axis is the normalization of the function. It's the highest point found in the membership set. After this point everybody has full membership to the set. From the 15 to 10.5 seconds the degree of membership rises from 0 to 1. Function can be drawn in many ways; this is a representation of a trapezoidal function and will be used in the scope of this project.

Membership function points are represented as follows:

$$\text{Fastest sports cars} = \left\{ \frac{0.0}{15} + \frac{0.5}{12.25} + \frac{1.0}{10.5} \right\}$$

Eberhart and Shi (2007) states that “we should note that in this type of specification the plus signs don't represent addition but rather the aggregation or collection of representative domain points, and that the horizontal lines do not represent division but delimiters, with the membership value above the line associated with the domain point below the line”.

#### 2.8.4. Fuzzy rules

Fuzzy rules, like the traditional rules, also consist of premise and consequent statements as described in section 7 of this chapter. There are different forms of fuzzy rules used in practise. The Mamdani type fuzzy rule will be used in this project and will be described. The consequent of the Mamdani rule is a fuzzy output variable and the domain there off is defined in a fuzzy set. This is the form, described by Eberhart and Shi (2007), of the rule:

*If X is A<sub>1</sub> and ... X<sub>n</sub> is A<sub>n</sub> then Y is B<sub>j</sub>*

*where X, ..., X<sub>n</sub> are the fuzzy input variables, A<sub>n</sub> represents one of the fuzzy sets defined over the domain of the fuzzy variables X<sub>n</sub>. Y is a fuzzy output variable, B<sub>j</sub> is one of the fuzzy sets defined over the domain of Y.*

### 2.8.5. Fuzzification

Bevrani (2010) defines fuzzification as “the definition of fuzzy sets, and determination of the degree of membership of crisp inputs in appropriate fuzzy sets”. From the definition we can derive that the membership functions defined on the crisp inputs are applied to their numeric values so that the degree of membership for the fuzzy rule premise can be determined. The fuzzy input variables will then relate to the consequent part of each rule. (Abraham 2005)

### 2.8.6. Intersection fuzzy sets (AND operator)

When there is more than one fuzzy set in an IF-THEN rule, the AND operator is used in the rule. The AND operator in binary logic is described as the intersection between sets. Meaning that “the intersection of A and B contain those elements that are in A AND B”. (Eberhart & Shi 2007) When the sets are defined by degree, another method has to be followed. Intersection in fuzzy rules use the *minimum* degree of membership of the sets defined in the rule. For instance let's say we have the following fuzzy sets with discrete values  $(a,b,c)$ :

$$X = \left\{ \frac{0.1}{a} + \frac{0.5}{b} + \frac{1.0}{c} \right\}$$

and

$$Y = \left\{ \frac{1.0}{a} + \frac{0.8}{b} + \frac{0.2}{c} \right\}$$

Now

$$X \cap Y = \left\{ \frac{0.1}{a} + \frac{0.5}{b} + \frac{0.2}{c} \right\}$$

The result is unchanged irrespective of the order of the fuzzy sets X and Y. The standardized formula for the intersection of fuzzy sets can be written in the following manner:

$$\mu_{X \cap Y}(x) = \min(\mu_X[x], \mu_Y[x])$$

where  $\mu_X(x)$  and  $\mu_Y(x)$  represents the degree to which  $x$  is a member of fuzzy sets X and Y, respectively, and sets have the same domains.

(Eberhart & Shi 2007)

### 2.8.7. Modelling of fuzzy expert systems

The following steps can be followed when designing an expert system using fuzzy logic, described by Abraham (2005):

1. Select the relevant input and output variables and define the number of linguistic terms relevant to the variables. Also define the fuzzy membership functions, fuzzy operators, and reasoning mechanism.
2. Choose the fuzzy inference engine to be used. In the scope of this project it would be the Mamdani inference type with intersection operators.
3. Compile the fuzzy IF- THEN rules for the knowledge base.

## **2.9. Past methods used in the design of knowledge-based expert systems for the selection of material handling equipment**

The first expert system for the selection of material handling equipment was designed in 1988 by Fisher and Farber, which was named MATHES (MATERial Handling Equipment Selection). MATHES is a rule based expert system which uses heuristic selection rules to select appropriate types of equipment for intra factory moves. MATHES takes the input and feedback from the user, by interface, and relates it to the degree of appropriateness of the material handling equipment. The programming for MATHES was very tedious because the rule base consisted of 172 rules and had to be coded for a system that only had 16 choices of equipment. (Fisher & Farber, 1988)

In 1996 Park saw that with the previous expert systems they never evaluated the performance of the selected model. He created ICMESE (Intelligent Consultant System for Material Handling Equipment Selection and Evaluation) which consisted of a knowledge base with decision trees for the system to relate attributes to the equipment, a multi criteria decision making (MCDM) process and simulators for the evaluation of the equipment selected. The MCDM consisted of two stages, namely a screening process and an analytic hierarchy process. The screening process required the user to specify the minimum criterion for certain attributes that were related to the equipment. As a result the equipment who doesn't meet the criteria is eliminated. The equipment left after the screening process goes through an analytic hierarchy process that uses goal driven techniques (backward chaining) to select the best candidate. Park (1996) presents the following illustration of the AHP:

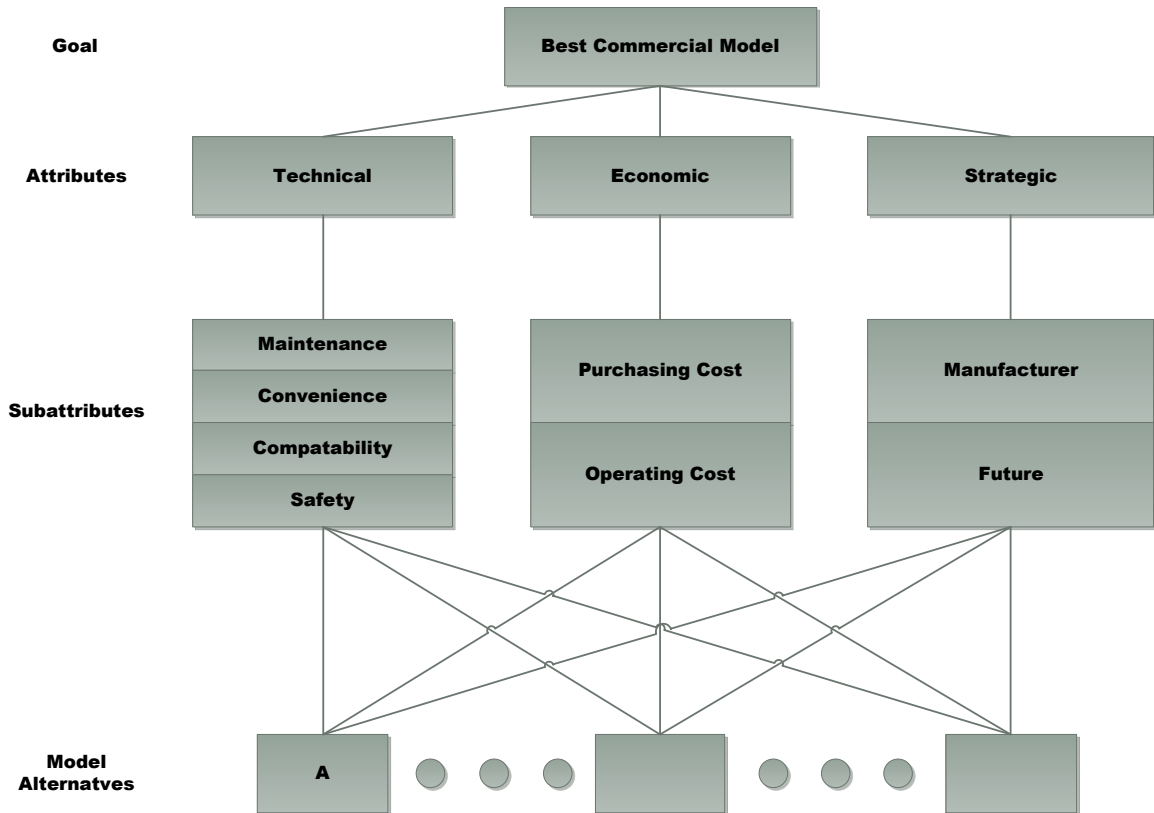


Figure 2.5 Analytical hierarchy procedure model

Chan also used the AHP to choose the best equipment for MHESA (Material Handling Equipment Selection Advisor). Only difference between the two is that he introduced artificial intelligence in his inference engine for the knowledge base, and the inference engine performed forward chaining. (Chan, 2001)

The project is based on most of the principles from the next expert system to be described. Mirhosseyni used a hybrid of fuzzy reasoning and crisp reasoning for the knowledge base and genetic algorithms for the selection of the fittest material handling equipment to meet the requirements. Genetic algorithms are based on the concept of a gene pool of chromosomes (feasible equipment), that are evaluated by the fittest (objective function) genes, after making the pool mutate. The chromosomes mutate until the fittest piece of equipment of all is selected. (Mirhosseyni, 2009)

## Chapter 3: Development of conceptual design

The Shark tooth method was employed and the following steps for the design were followed:

- A functional specification was compiled with the requirements for the project.
- The membership functions for the attributes were prepared
- The rule base was developed using crisp and fuzzy reasoning
- Design principles were followed for the conceptual design
- Concept design was presented to stakeholders and approved
- The program was coded, using VBA(Visual Basic) in excel, and tests were run incrementally after the completion of each module, function and subsystem, using the debugging tool in VBA
- The program was incorporated in the company and maintenance is still taking place

### 3.1. Functional specification (Requirements analysis)

#### 3.1.1 Description of general requirements

##### *Product Perspective*

The program will be a new system for the company and will meet the requirements of the company. Systems like this one have been developed previously by other institutions and the final product will contain some of the main coding and design principles from them.

##### *Product Features*

Some of the features which the program will contain are the following:

- Reports on the best solution
- User-interface with drop down lists for effective inputs
- Sharing restrictions

##### *User Classes and Characteristics*

The program will be used mainly by new young sales reps for training and to reduce the chances of them making mistakes when suggesting a product to a client. The more experienced sales reps will be entitled to use the product for information and to improve their technological knowledge of the products.

##### *Operating Environment*

The program will be developed and operated in Excel. VBA is used as the coding language for the program. Access will be restricted by using the online permission function provided by Microsoft. Users have to be registered to the online function and permission given to them by the administrator. The reports generated from the

program must be able to be stored on the company database. The MS Access is the platform used for the company database and can coexist with Excel.

### *Design and Implementation Constraints*

The main constraint for the program is the security of the sharing of the program. Another major constraint is time.

### *User Documentation*

A user manual will be compiled for the program and tutorials will be added at the end of each chapter to make the user more familiar with the product.

## **3.1.2. External interface requirements**

### *User Interfaces*

The form function and VBA, in Excel, will be used to create a user interface for the inputs needed for processing. Some of the questions will have a drop down list for selection of the correct criteria. Visual basic in Excel will be used to make the forms and GUI of the program.

### *Software Interfaces*

Macros function will be used for the import and export of data from the reports of the program to the company database.

## **3.1.3. Functional requirements**

### *Main menu page*

The main menu will be the first page the user sees when he opens the program up.

### **Functional requirements**

The function must:

- Capture customer details
  1. What is the customer's name?
  2. What kind of industry does the customer cater for?
- Capture the user details
  1. What is the user's name?
  2. What is the date?

### **Inputs**

- Customer name
- Type of industry
- User name
- Date

## Outputs

- Customer needs checklist for specifications, for the type of industry stated.

### *Warehousing and manufacturing page*

This page will come up after it was specified in the industry entry of the main page. It will contain the needs checklist for warehousing and manufacturing operations.

## Functional requirements

The function must:

- Capture the load shape and handling method
  1. What kind of load is used?
  2. What is the shape of the load?
- Capture the load type
  1. What type of load will be handled?
  2. What is the size of the unit load?
  3. What is the weight of the unit load?
  4. What is the size of the combined load to be handled?
  5. What is the weight of the combined load to be handled?
  6. Is a clamp attachment needed?
- Capture the work place and environment
  1. What kind of work place is used by the customer?
  2. What is the entrance height of the building?
  3. What is the ceiling height of the building?
  4. What is the aisle width of the building?
  5. What is the stacking aisle width?
  6. What type of environment is to be worked?
  7. What is the height of the top stacking beam?
- Capture the move attributes
  1. What is the distance to be travelled in the facility per shift?
- Capture the area constraints
  1. What type of storage area is used?
  2. How deep is the rack?
- Capture the type of operation
  1. What type of operational handling method is used?
  2. What type of loading and unloading operation is used?
  3. Does the facility have a FIFO or a FILO operation?
  4. What is the order picking level of the facility?
- Capture attachment specifications
  1. What type of attachment is needed?
  2. Does the attachment need a rubber contact?
  3. Does the attachment need a pressure regulator valve and gauge

## Inputs

- Shape of the load
- Load handled
- Kind of load
- Unit load size
- Unit load weight
- Combined load to be handled at one time size
- Combined load to be handled at one time maximum weight
- Kind of work place
- Building entrance height
- Building ceiling height
- Aisle width
- Stacking aisle width
- Type of environment
- Similar model currently in use
- Maximum loading height
- Distance travelled per shift
- Storage area
- Rack deep
- Operational handling method
- Loading /unloading type
- Order picking level
- Storage /retrieval order
- Attachment make/type/model
- Contact rubber
- Pressure regulator valve and gauge

### *Container yard page*

The page is used for the container yard industry

### **Functional requirements**

The function must:

- Capture the container type
  1. What types of container/containers are used by the customer?
  2. What is the size of the containers to be handled?
  3. Are the containers to be handled empty or loaded?
- Capture the stacking method
  1. What stacking method is used by the yard?
  2. What is the maximum stacking height?
- Capture the yard area criteria
  1. What is the aisle width?
  2. What is the container row length?

## Inputs

- Container handling type
- Container type
- Container size
- Container stacking method
- Stacking height
- Empty or Loaded containers
- Aisle width
- Container row length

## *Output report*

The following outputs should be in the report:

- the product name
- the specifications of the equipment
- the attachments for the equipment
- the mast to be used for the equipment
- and the initial and operating cost

### 3.1.4. Other Non-functional Requirements

#### *Security Requirements*

Unauthorised copying of the program and company information must be prevented and protected. The user will need to have a password to identify the user. Only company employees should be able to access the data from the program. When an employee's employment is terminated the company must be able to restrict the employee from ever using the program again.

#### *Software Quality Attributes*

The information and calculations coded in the program must be precise and up to the standard. The program must be maintainable to change product information and add future products.

### 3.1.5. List of Products

#### *Warehousing and manufacturing*

- Diesel/gas forklift trucks
- Three wheel electric counterbalance trucks
- Four wheel electric counterbalance trucks
- Electric pedestrian pallet truck with platform
- Electric stand-on pallet truck
- Hand pallet truck
- Scissor pallet truck
- Electric pedestrian stacker with platform
- Reach truck
- Pedestrian reach truck with platform
- Horizontal order picker

- Low level order picker

### Container yards

- Reach stacker 45ton
- Stuffing forklift 1-3.5ton
- Empty container handler 7-9ton
- Masted container handler 40ton

## 3.2. Handling attributes and membership function design

In this design phase the crisp sets and the corresponding attributes were defined first, and secondly the fuzzy sets and their corresponding attributes. In the design of the fuzzy sets, the membership functions had to be defined for each numerical attribute. First the universe of discourse had to be defined for each attribute. This depended on the specifications and constraints of the products defined in the functional specification. Trapezoidal functions were used for the shape of the membership functions. As an example the membership function for the attribute *unit load weight* will be presented. The universe of discourse was determined by using the largest load capacity offered by a product, which was 3.5 ton. Fig. 3.1 shows all the points on the graph where the load capacity of the products are present on the universe of discourse. From these points we can determine how many fuzzy sets we can have along the universe of discourse. The upper limit of the set defines the range in which the equipment can have a degree of membership of. The lower limit was set on the breaking points to avoid more number of sets.

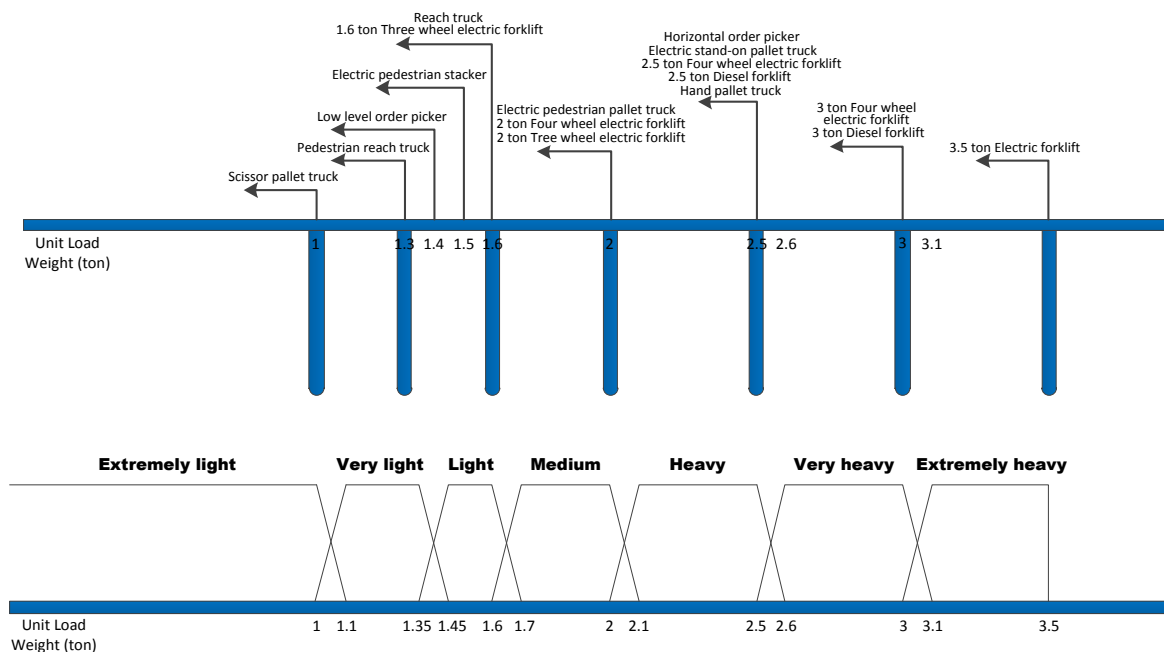


Figure 3.1 Membership function of *unit load weight* attribute

The same procedure was followed for determining the membership functions for the rest of the fuzzy attributes. The list of crisp and fuzzy attributes were identified and their inputs defined.

### 3.2.1. List of Crisp Attributes

#### *Main menu*

- **Industry:** Warehousing and manufacturing, Container yard

#### *Warehousing and Manufacturing*

##### **Load attributes**

- **Shape of the load:** box/case, flat goods, circular cylinder goods, bags, concrete block, loose products
- **Kind of load:** individual unit, pallet unit load
- **Load handled:** uniform, variable

##### **Work place and environment attributes**

- **Kind of work place:** warehouse, factory, rough road
- **Type of environment:** salty, dusty, wet, hot, sub-zero, hazardous, normal

##### **Operational attributes**

- **Operational handling method:** loading/unloading, storage/retrieval, order picking
- **Storage/retrieval order:** FIFO, FILO
- **Loading/unloading type:** loading bay, ground level
- **Order picking level:** level 1, level 2

##### **Area attributes**

- **Storage area:** floor, rack
- **Rack deep:** single, double

##### **Attachment specifications**

- **Attachment type:** paper roll clamp, side shift, adjustable forks, rotaters
- **Contact rubber:** with, without
- **Pressure regulator valve and gauge:** with, without

#### *Container yard*

##### **Container attributes**

- **Container type:** closed, open-top, insulated, platform, reefer, open wall, tank
- **Container size:** 20 foot, 40 foot

##### **Handling attributes**

- **Container handling type:** top spreading, side spreading, forks, stuffing
- **Empty or loaded containers:** empty, loaded, both

### Stacking attributes

- **Container stacking method:** with stacker, without
- **Stacking height:** 1 high, 2 high, 3 high, 4 high, 5 high, 6 high
- **Container row length:** 1 row, 2 row, 3 row

### 3.2.2. List of fuzzy attributes

#### *Warehousing and manufacturing*

#### Load attributes

- **Unit load size:** extremely small, very small, small, medium, large, very large, extremely large
- **Unit load weight:** extremely light, very light, light, medium, heavy, very heavy, extremely heavy
- **Combined load to be handled at one time size:** extremely small, very small, small, medium, large, very large, extremely large
- **Combined load to be handled at one time maximum weight and load centre:** extremely light, very light, light, medium, heavy, very heavy, extremely heavy

#### Workplace and environment attributes

- **Building entrance height:** slight, medium, high
- **Building ceiling height:** slight, medium, high, very high, extremely high
- **Aisle width:** slightly wide, medium width, wide, very wide, extremely wide
- **Stacking aisle width:** slightly wide, medium width, wide, extremely wide
- **Maximum loading height:** slight, medium, high, very high

#### Move attributes

- **Distance travelled per shift:** extremely short, very short, short, medium, long, very long, extremely long

#### *Container yard*

#### Area attributes

- **Aisle width for 20 foot container:** slightly wide, wide, very wide, extremely wide
- **Aisle width for 40 foot container:** slightly wide, wide, very wide

The membership functions and the criteria are defined in the Table 3.1

Table 3.1 Membership functions

<b>Unit load size (ton.m)</b>	<b>Extremely small</b>	<b>Very small</b>	<b>Small</b>	<b>Medium</b>	<b>Large</b>	<b>Very large</b>	<b>Extremely large</b>
	$\left\{ \frac{1}{0.6} + \frac{0}{0.61} \right\}$	$\left\{ \frac{0}{0.6} + \frac{1}{0.61} + \frac{1}{0.84} + \frac{0}{0.85} \right\}$	$\left\{ \frac{0}{0.84} + \frac{1}{0.85} + \frac{1}{0.9} + \frac{0}{0.91} \right\}$	$\left\{ \frac{0}{0.9} + \frac{1}{0.91} + \frac{1}{1} + \frac{0}{1.01} \right\}$	$\left\{ \frac{0}{1} + \frac{1}{1.01} + \frac{1}{1.25} + \frac{0}{1.26} \right\}$	$\left\{ \frac{0}{1.25} + \frac{1}{1.26} + \frac{1}{1.5} + \frac{0}{1.51} \right\}$	$\left\{ \frac{0}{1.5} + \frac{1}{1.51} + \frac{1}{1.75} \right\}$
<b>Unit load weight (ton)</b>	<b>Extremely light</b>	<b>Very light</b>	<b>Light</b>	<b>Medium</b>	<b>Heavy</b>	<b>Very heavy</b>	<b>Extremely heavy</b>
	$\left\{ \frac{1}{1} + \frac{0}{1.1} \right\}$	$\left\{ \frac{0}{1} + \frac{1}{1.1} + \frac{1}{1.35} + \frac{0}{1.45} \right\}$	$\left\{ \frac{0}{1.35} + \frac{1}{1.45} + \frac{1}{1.6} + \frac{0}{1.7} \right\}$	$\left\{ \frac{0}{1.6} + \frac{1}{1.7} + \frac{1}{2} + \frac{0}{2.1} \right\}$	$\left\{ \frac{0}{2} + \frac{1}{2.1} + \frac{1}{2.5} + \frac{0}{2.6} \right\}$	$\left\{ \frac{0}{2.5} + \frac{1}{2.6} + \frac{1}{3} + \frac{0}{3.1} \right\}$	$\left\{ \frac{0}{3} + \frac{1}{3.1} + \frac{1}{3.5} \right\}$
<b>Combined load size to be handled at one time (ton.m)</b>	<b>Extremely small</b>	<b>Very small</b>	<b>Small</b>	<b>Medium</b>	<b>Large</b>	<b>Very large</b>	<b>Extremely large</b>
	$\left\{ \frac{1}{0.6} + \frac{0}{0.61} \right\}$	$\left\{ \frac{0}{0.6} + \frac{1}{0.61} + \frac{1}{0.84} + \frac{0}{0.85} \right\}$	$\left\{ \frac{0}{0.84} + \frac{1}{0.85} + \frac{1}{0.9} + \frac{0}{0.91} \right\}$	$\left\{ \frac{0}{0.9} + \frac{1}{0.91} + \frac{1}{1} + \frac{0}{1.01} \right\}$	$\left\{ \frac{0}{1} + \frac{1}{1.01} + \frac{1}{1.25} + \frac{0}{1.26} \right\}$	$\left\{ \frac{0}{1.25} + \frac{1}{1.26} + \frac{1}{1.5} + \frac{0}{1.51} \right\}$	$\left\{ \frac{0}{1.5} + \frac{1}{1.51} + \frac{1}{1.75} \right\}$
<b>Combined weight to be handled at one time (ton)</b>	<b>Extremely light</b>	<b>Very light</b>	<b>Light</b>	<b>Medium</b>	<b>Heavy</b>	<b>Very heavy</b>	<b>Extremely heavy</b>
	$\left\{ \frac{1}{1} + \frac{0}{1.1} \right\}$	$\left\{ \frac{0}{1} + \frac{1}{1.1} + \frac{1}{1.35} + \frac{0}{1.45} \right\}$	$\left\{ \frac{0}{1.35} + \frac{1}{1.45} + \frac{1}{1.6} + \frac{0}{1.7} \right\}$	$\left\{ \frac{0}{1.6} + \frac{1}{1.7} + \frac{1}{2} + \frac{0}{2.1} \right\}$	$\left\{ \frac{0}{2} + \frac{1}{2.1} + \frac{1}{2.5} + \frac{0}{2.6} \right\}$	$\left\{ \frac{0}{2.5} + \frac{1}{2.6} + \frac{1}{3} + \frac{0}{3.1} \right\}$	$\left\{ \frac{0}{3} + \frac{1}{3.1} + \frac{1}{3.5} \right\}$
<b>Building entrance height (m)</b>	<b>Slight</b>		<b>Medium</b>			<b>High</b>	
	$\left\{ \frac{1}{2} + \frac{1}{2.4} + \frac{0}{2.55} \right\}$		$\left\{ \frac{0}{2.4} + \frac{1}{2.55} + \frac{1}{2.95} + \frac{0}{3.25} \right\}$			$\left\{ \frac{0}{2.95} + \frac{1}{3.25} + \frac{1}{\infty} \right\}$	
<b>Building ceiling height (m)</b>	<b>Slight</b>		<b>Medium</b>	<b>High</b>	<b>Very high</b>	<b>Extremely high</b>	
	$\left\{ \frac{1}{2} + \frac{1}{5.05} + \frac{0}{5.15} \right\}$		$\left\{ \frac{0}{5.05} + \frac{1}{5.15} + \frac{1}{5.2} + \frac{0}{5.55} \right\}$	$\left\{ \frac{0}{5.2} + \frac{1}{5.55} + \frac{1}{6.2} + \frac{0}{6.3} \right\}$	$\left\{ \frac{0}{6.2} + \frac{1}{6.3} + \frac{1}{8.25} + \frac{0}{8.35} \right\}$	$\left\{ \frac{0}{8.25} + \frac{1}{8.35} + \frac{1}{\infty} \right\}$	
<b>Aisle width (m)</b>	<b>Slightly wide</b>		<b>Medium width</b>	<b>Wide</b>	<b>Very wide</b>	<b>Extremely wide</b>	
	$\left\{ \frac{1}{2} + \frac{1}{2.3} + \frac{0}{2.7} \right\}$		$\left\{ \frac{0}{2.3} + \frac{1}{2.7} + \frac{1}{2.85} + \frac{0}{3.2} \right\}$	$\left\{ \frac{0}{2.85} + \frac{1}{3.2} + \frac{1}{3.7} + \frac{0}{3.85} \right\}$	$\left\{ \frac{0}{3.7} + \frac{1}{3.85} + \frac{1}{4} + \frac{0}{4.15} \right\}$	$\left\{ \frac{0}{4} + \frac{1}{4.15} + \frac{1}{\infty} \right\}$	
<b>Stacking aisle width (m)</b>	<b>Slightly wide</b>		<b>Medium width</b>	<b>Wide</b>	<b>Very wide</b>	<b>Extremely wide</b>	
	$\left\{ \frac{1}{2} + \frac{1}{2.3} + \frac{0}{2.7} \right\}$		$\left\{ \frac{0}{2.3} + \frac{1}{2.7} + \frac{1}{2.85} + \frac{0}{3.2} \right\}$	$\left\{ \frac{0}{2.85} + \frac{1}{3.2} + \frac{1}{3.7} + \frac{0}{3.85} \right\}$	$\left\{ \frac{0}{3.7} + \frac{1}{3.85} + \frac{1}{4} + \frac{0}{4.15} \right\}$	$\left\{ \frac{0}{4} + \frac{1}{4.15} + \frac{1}{\infty} \right\}$	
<b>Maximum lifting height (m)</b>	<b>Slight</b>		<b>Medium</b>	<b>High</b>	<b>Very high</b>		
	$\left\{ \frac{1}{0.2} + \frac{0}{0.3} \right\}$		$\left\{ \frac{0}{0.2} + \frac{1}{0.3} + \frac{1}{0.8} + \frac{0}{0.9} \right\}$	$\left\{ \frac{0}{0.8} + \frac{1}{0.9} + \frac{1}{5} + \frac{0}{5.1} \right\}$	$\left\{ \frac{0}{5} + \frac{1}{5.1} + \frac{1}{7.52} \right\}$		
<b>Distance travelled per shift (m/shift)</b>	<b>Extremely short</b>	<b>Very short</b>	<b>Short</b>	<b>Medium</b>	<b>Long</b>	<b>Very long</b>	<b>Extremely long</b>
	$\left\{ \frac{1}{0} + \frac{1}{10} + \frac{0}{15} \right\}$	$\left\{ \frac{0}{10} + \frac{1}{15} + \frac{1}{20} + \frac{0}{25} \right\}$	$\left\{ \frac{0}{20} + \frac{1}{25} + \frac{1}{35} + \frac{0}{37.5} \right\}$	$\left\{ \frac{0}{35} + \frac{1}{37.5} + \frac{1}{40} + \frac{0}{45} \right\}$	$\left\{ \frac{0}{40} + \frac{1}{45} + \frac{1}{50} + \frac{0}{52.5} \right\}$	$\left\{ \frac{0}{50} + \frac{1}{52.5} + \frac{1}{55} + \frac{0}{60} \right\}$	$\left\{ \frac{0}{55} + \frac{1}{60} + \frac{1}{\infty} \right\}$
<b>Aisle width for 20 foot container (m)</b>	<b>Slightly wide</b>	<b>Wide</b>		<b>Very wide</b>		<b>Extremely wide</b>	
	$\left\{ \frac{1}{9.32} + \frac{1}{9.7} + \frac{0}{9.8} \right\}$	$\left\{ \frac{0}{9.7} + \frac{1}{9.8} + \frac{1}{13.58} + \frac{0}{13.68} \right\}$		$\left\{ \frac{0}{13.58} + \frac{1}{13.68} + \frac{1}{14.82} + \frac{0}{14.92} \right\}$		$\left\{ \frac{0}{14.82} + \frac{1}{14.92} + \frac{1}{\infty} \right\}$	

Aisle width for 40 foot container (m)	Slightly wide	Wide	Very wide
	$\left\{ \frac{1}{13.59} + \frac{1}{15.4} + \frac{0}{15.5} \right\}$	$\left\{ \frac{0}{15.4} + \frac{1}{15.5} + \frac{1}{16.2} + \frac{0}{16.74} \right\}$	$\left\{ \frac{0}{16.2} + \frac{1}{16.74} + \frac{1}{\infty} \right\}$

### 3.3. Knowledge-base

The knowledge base consists of crisp rules and fuzzy rules. The crisp rules defined are implemented in the inference engine by using decision trees to determine the potential equipment for the Fuzzy reasoning part of the inference engine. The fuzzy rules defined use the potential equipment to determine what the appropriateness factor would be for the potential equipment. The following two figures represent an example of the root taken in the decision trees:

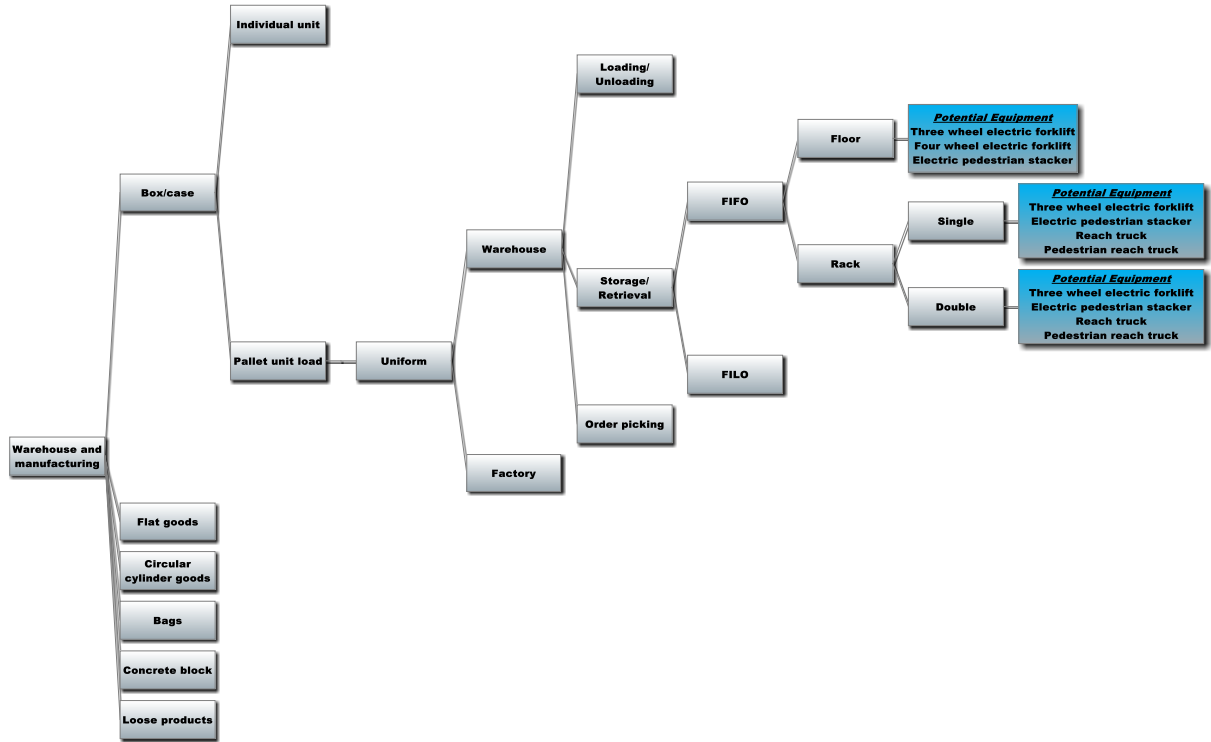


Figure 3.2 Warehousing and manufacturing decision tree

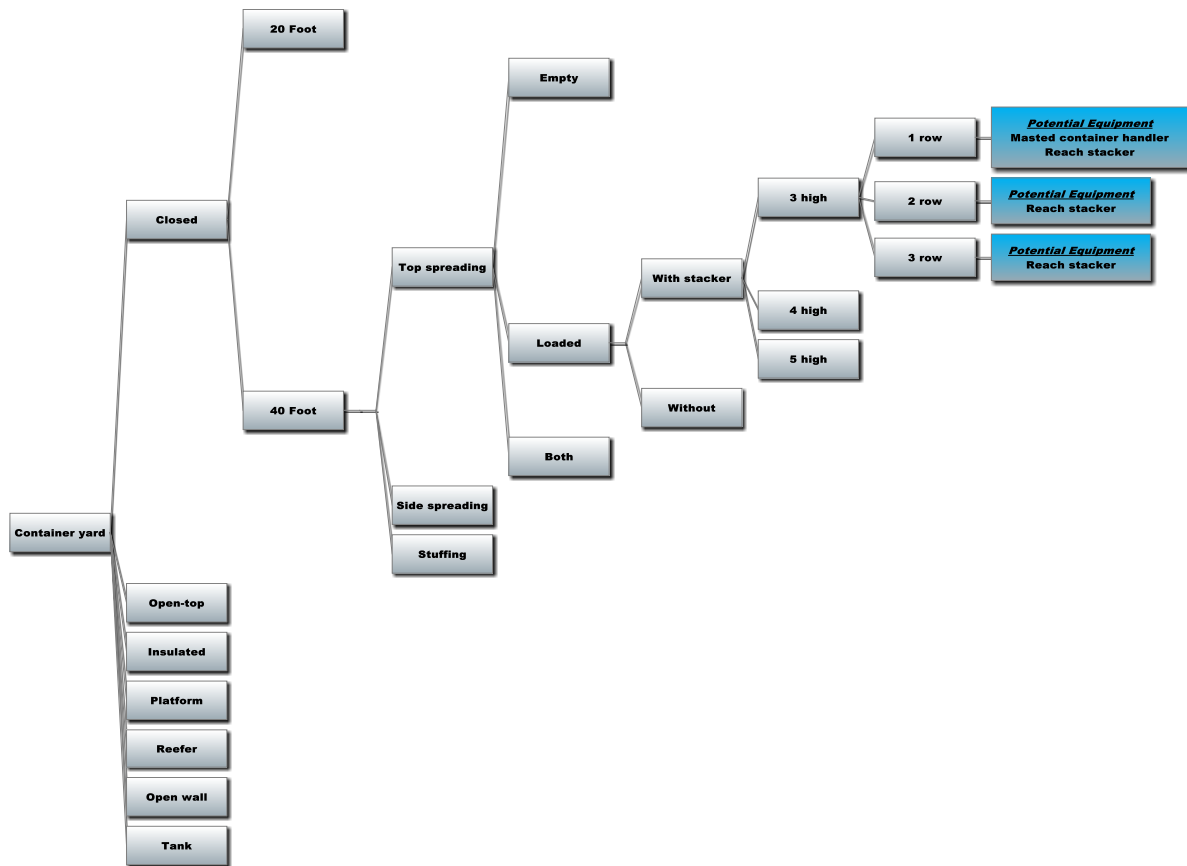
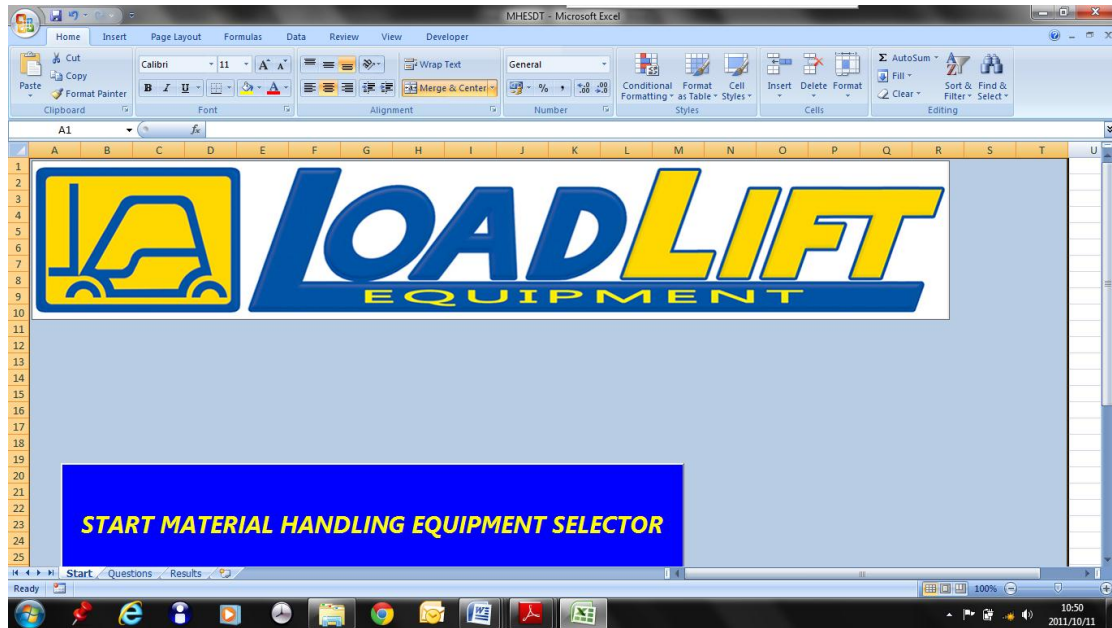


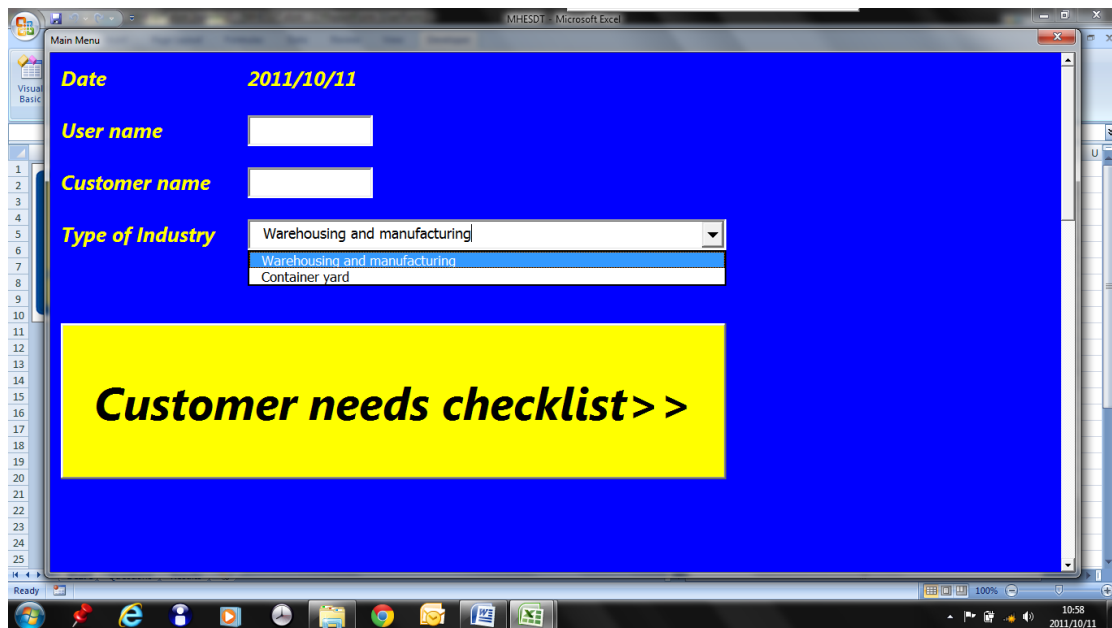
Figure 3.3 Container yard decision tree

### 3.4. Program operating environment

The environment of is specified as a form based environment, where buttons, text boxes and drop down menus are used. Visual Basic editor was used in the design of the forms and the coding architecture of the program. The basic process a user will follow in using the process is defined through the representation of the following screen shots:



After the user has logged in the following sheet will appear with the button **Start material handling equipment selector**. When the button is clicked it will open the **Main menu** form.



The date is automatically represented by the program. If a field is left empty a message box will appear stating that the required fields have to be filled in. When the type of industry is selected and the button clicked the checklist for the selected industry will open and this form close.



**Warehousing and manufacturing**

**Shape of the load**

**Kind of load**

**Load handled**

**Unit load size**

**Unit load weight**

**Combined load to be handled at one time size**

**Combined load to be handled at one time maximum weight**

**Operational handling method**

**Loading/unloading**

**Storage/retrieval order**

**Order picking level**

**Area attributes**

**Storage Area**

**Rack deep**

**Work place and environment**

**Kind of work place**

**Building entrance height**

**Building ceiling height**

**Aisle width**

**Stacking aisle width**

**Maximum loading height**

**Type of environment**

Salty  Wet  Normal

Dusty  Hazardous

Hot  Sub-Zero

**Move attributes**

**Distance travelled per shift**

**Attachments**

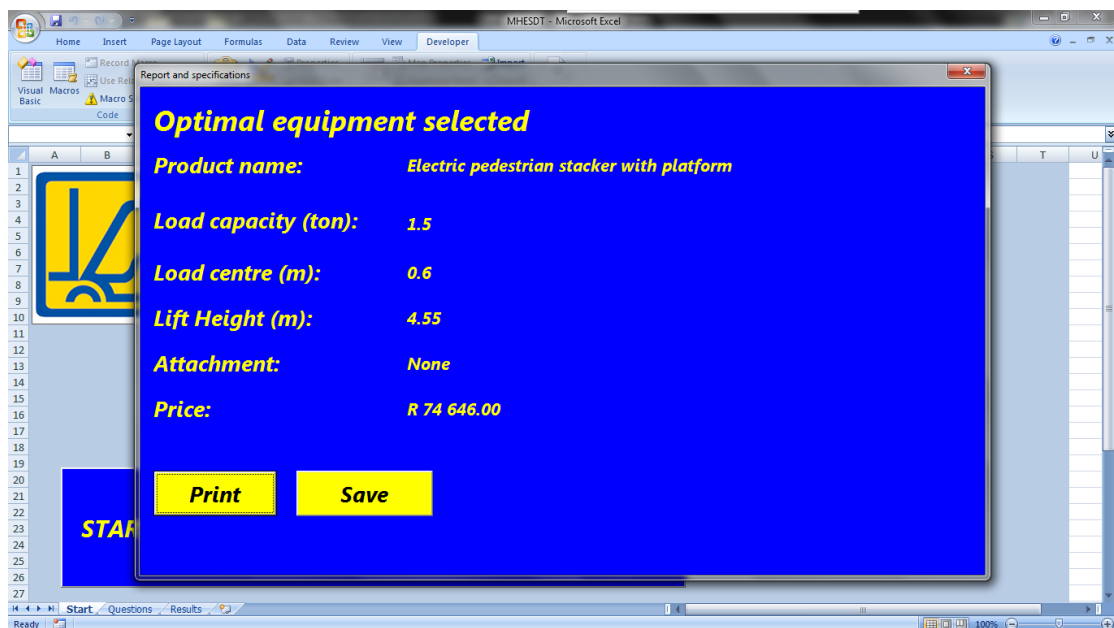
**Attachment needed**  Yes  No

**Attachment type**

**Rubber contact needed**

**Pressure regulator valve & gauge needed**

All entries have to be filled in as for the previous form. When a certain field is selected in the drop down menu, the following drop down menu will only list the criteria for that specific field. When the button is clicked the report form will open and this form close.



**Report and specifications**

**Optimal equipment selected**

**Product name:** *Electric pedestrian staker with platform*

**Load capacity (ton):** 1.5

**Load centre (m):** 0.6

**Lift Height (m):** 4.55

**Attachment:** None

**Price:** R 74 646.00

**Print** **Save**

The report takes information from the checklist and makes the necessary calculations and populates the optimal solution with the specifications for the equipment. The user can print and save the report.

### **3.5. Testing and validation**

The prototyping sessions with stakeholders helped a lot in the compilation of the final functional specification. The stakeholders and users identified requirements not specified and helped with the final formulation of the attributes needed for the knowledge based inference system.

The concept design was validated with the requirements stated in the specification. Testing took place incrementally and prototypes were presented to stakeholders for feedback and validation.

The maintenance phase of the program is still under completion.

## **Chapter 4: Conclusion and findings**

A lot of effort and time has gone in to the study of fuzzy logic. The literature showed concepts that were difficult to understand under some circumstances. Visual Basic was adequate enough for the coding of the decision tool for the company, because it met all the necessary criteria. The design process helped in designing and implementing a sound program. The problem experienced with the design process is the vast amount of time needed to implement all the phases of the process model. The program has some issues that can be solved and improved by diligent maintenance procedures. The student gained the needed experience and knowledge of heading and doing a project on his own with no knowledge or experience on the subject. Time constraints made the student aware of the importance of efficiency and exceptional project planning. The student can conclude that program is a great asset to the company and will provide them with future technical sales reps and improve sales criteria for customers.

## References

Abraham, A 2005, 'Rule-based Expert Systems', in PH Sydenham & R Thorn (ed.), *Handbook of Measuring System Design*, John Wiley & Sons, Ltd.

Bevrani, H 2010, 'Fuzzy Rule Base and Fuzzy Inference Engine', paper presented to the Department of Electrical & Computer Engineering at the University of Kurdistan, Kurdistan, 11 July 2010.

Chan, FTS. and Ip, RWL. and Lau, H. (2001). Integration of expert system with analytic hierarchy process for the design of material handling equipment selection system. *Journal of Materials Processing Technology* [online]. **116 Issues 2-3**, [Accessed 5 June 2011 ], p.137-145. Available from: <<http://www.sciencedirect.com/innopac.up.ac.za/science/article/pii/S092401360101038X>>.

Chaudron, M & Groote, JF & van Hee, K & Hemerik, K & Somers, L & Verhoeff, T 2004, *Software Engineering Reference Framework*, Eindhoven University of Technology, Eindhoven, viewed 4 June, 2011, <<http://www.win.tue.nl/~jfg/articles/CSR-04-39.pdf>>

DEPARTMENT OF THE AIR FORCE 2003, *Guidelines for the Successful Acquisition and Management of Software-Intensive Systems (GSAM)*, Chapter 10, Software Technology Support Centre, OO-ALC/MASE, Utah.

Eberhart, RC & Shi, Y 2007, *Computational Intelligence Concepts to Implementations*, Elsevier Inc.

Fisher, EL. and Farber, JB. & Kay, MG. (1988). MATHES: An expert system for material handling equipment selection. *Engineering Costs and Productive Economics* [online]. 14 Issue 4, [Accessed 5 June 2011 ], p.297-310. Available from: <<http://www.sciencedirect.com/innopac.up.ac.za/science/article/pii/0167188X88900341>>.

Mirhosseyni, SHL. and Webb, P. (2009). A Hybrid Fuzzy Knowledge-Based Expert System and Genetic Algorithm of efficient selection and assignment of Material Handling Equipment. *Expert Systems with Applications* [online]. 36 Issue 9, [Accessed 5 June 2011 ], p.11875-11887. Available from: <<http://www.scopus.com/innopac.up.ac.za/record/display.url?eid=2-s2.0-67349202476>>.

Park, Y. (1996). ICMES: Intelligent Consultant System for Material Handling Equipment Selection and Evaluation. *Journal of Manufacturing Systems* [online]. 15 Issue 5, [Accessed 5 June 2011 ], p.325-333. Available from: <<http://www.sciencedirect.com/innopac.up.ac.za/science/article/pii/0278612596841951>>