CHAPTER 4

KNOWLEDGE BASED SYSTEM, GAUGING ABSENCES OF PRE-REQUISTES AND ANALYTIC HIERARCHY PROCESS

4.1 Introduction to Artificial Intelligence

Intelligence is the ability to learn, understand, solve problems and to make decisions (Negnevitsky, 2002). Before reaching this level, the human thought process starts with *data*, *information* and *knowledge*. According to Awad (1996), *"knowledge is understanding gained through experience or study which is the accumulation of facts, procedural rules, or heuristics"*. It requires familiarisation in dealing with something in order for a person to perform a task. Andersson (2008), added that the definition of *knowledge* has been debated long before it was used in engineering, back in the era of the ancient Greek Plato who defined it as *"justified true belief"*.

Knowledge can be arranged in the hierarchical form of a pyramid model as shown in Figure 4.1. *Data* are unprocessed facts: for example 100 meters. This *data* will become *information* after certain assembling of facts process such as it took two years to build the 100 meter tall building. This *information* when interpreted by a person will become *knowledge* and adding experience to it, the person will have *expertise/wisdom*. As in the above example, it is not profitable to have this kind of building if the company wants to rent the building for a one year period. In addition, *tacit knowledge* or *implicit knowledge* is the kind of *knowledge* that is difficult to express (Lintern, 2006) and is normally known to the individual such as the painting skills.

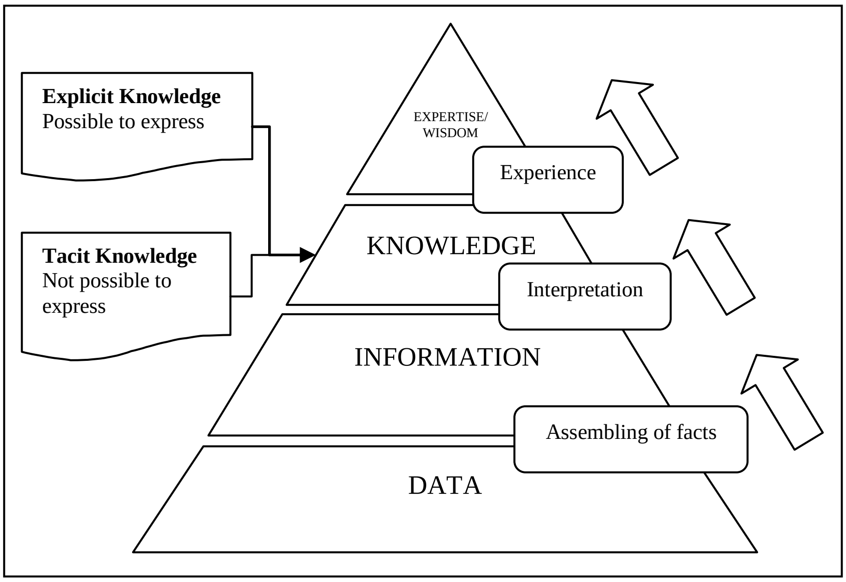

Figure 4.1: Pyramid model, the hierarchy of data, information, knowledge, and expertise/wisdom (Adapted from Andersson (2008))

On the other hand, *explicit knowledge* is the *knowledge* that you are able to express such as in manuals and procedures. Therefore, it is important to transform *tacit knowledge* into *explicit knowledge* to accommodate the transfer of knowledge, especially in building an expert system. Experts in the field of Artificial Intelligence (AI) try to use human intelligent processes to be represented by a machine that can capture the expertise of human beings and think like human beings. Udin (2004) stated that AI is a field of knowledge in computer application development that imitates the human behaviour in completing tasks. According to Pham and Pham (1999), AI has emerged as a computer science discipline since the mid 1950s and has produced a number of powerful tools, many of which are of practical use in engineering to solve difficult problems normally associated with human intelligence.

In building an AI system such as Knowledge Based System (KBS), normally knowledge representation scheme uses logic to represent facts and relationships (Awad, 1996). Logic is the study of reasoning which can be categorised as propositional logic and predicate logic. Propositional logic uses connectives such as

AND, OR, and NOT to evaluate the truth and falsity of the statements. Therefore, a proposition is a statement that can be true (*T*) or false (*F*) about certain statements such as *John is intelligent*. The combination of propositions can form more complex facts such as (*John is intelligent*) and *(John is hardworking).* Based on these established truths and logical extension, new knowledge is formed.

On the other hand, predicate logic is more appropriate for representing knowledge in KBS due to its finer detail reasoning with mathematical properties. . Predicate logic is dealing with some or all of objects in the universe and relations among them (Russel and Norvig, 2003). The object or the argument of the predicate determines the truth value of the predicate, arguments can make the predicate either true or false. Predicate logic uses two standard quantifiers, namely *universal* (∀) and *existential* (∃) to express the properties of objects in a more natural way (Russel and Norvig, 2003). Universal quantifier is used for making statement about *all* objects, whereas the existential quantifier is used for making statement about *some* objects. An example of a sentence in the context of knowledge representation using universal quantifier (Awad, 1996):

All cars have engine.

This sentence would be represented by a mathematical formula as:

"For any object X, if X is a car, then X has an engine."

$$\forall X[Car(X) \rightarrow Has\ engine(X)]$$

The following example of existential quantifier is adapted from Russel and Norvig (2003):

The crown is on John's head.

This sentence would then be represented by a mathematical formula as:

"There exists an x such that Crown, and the x is on John's head"

$$\exists x\ [Crown\ (x) \wedge OnHead(x,John)]$$

Among the advantages of predicate logic are the set of logical formulas which can be changed or derived from the old ones as required, clear rules for easy interpretation and the use of modular format for easy update. Therefore, this KBS rules and methods can support the TQM and QFD in the organisation. This KBS will address quality issue relevant to TQM and QFD, although it is not explicitly link to the TQM methodology existing in the organisation.

Teti and Kumara (1997) categorised the use of AI into its functions, techniques and manufacturing sector as shown in Table 4.1. It is clearly shown that the applications of AI are widely used for various usages in manufacturing environment.

Table 4.1: AI functions and techniques in manufacturing (Teti and Kumara, 1997)

Artificial Intelligence in Manufacturing		
AI functions	***AI techniques***	***Manufacturing Sector***
Learning	Genetic algorithms	Design
Knowledge	Neural networks	Planning
Reasoning	Fuzzy logic	Production
Goal-seeking	Neuro-Fuzzy	Scheduling Systems
Pattern recognition	Simulated Annealing	Assembly
Decision Making	Expert Systems	Monitoring
Advice	Knowledge Based Systems	Control
Communication	Hybrid systems	Inspection
Control	Multi Agents	Maintenance
Self-improvement		
Self-maintenance		
Self organisation		

Seven of these most commonly used techniques will be discussed in this chapter. They are: Genetic Algorithms (GA), Artificial Neural Networks (ANN), Fuzzy Logic (FL), Simulated Annealing (SA), Case Based Reasoning (CBR), Frame Based System (FBS) and Expert System (ES)/ Knowledge Based Systems (KBS).

4.1.1 Genetic Algorithms (GA)

Genetic Algorithms (GAs) have been developed based on the natural evolution of living organisms. The natural evolution is the process to reach an optimum form of new generations of animals and plants (Hopgood, 2001). GA methodology is a sequence of chromosomes reproduction processes from generation to another. If the characteristics of the newly formed generation is favourable than the previous generation, the offspring is more likely to inherit the characteristics and pass them to the next generation. On the other hand, the offspring with unfavourable characteristics will become extinct without reproducing. According to Noor (2007), each cell of a living thing consists of set of chromosomes that determine these characteristics. Chromosomes are made up of genes, the blocks of DNA, where each gene encodes a particular protein which defines a particular trait such as the colour of eyes. The possible settings for the genes are called alleles and each position of the genes along the chromosome is called locus.

Therefore, GA is a stochastic or random search technique based on biological evolution by representing chromosomes as strings of ones and zeros (Negnevitsky, 2002). In essence GA are an optimisation search techniques; which simulate an evolutionary process to produce fitter populations containing better solution chromosomes until a final criterion is accomplished. The artificial chromosomes in GA work by copying themselves during iterative procedure in a specific domain problem until a solution is achieved (Udin, 2004). Each iteration is called a generation and the entire set of generations is called a run (Maqsood et al., 2011). There are three primary operations in GA, namely *Reproduction*, *Cross-over* and *Mutation*, as shown in Figure 4.2.

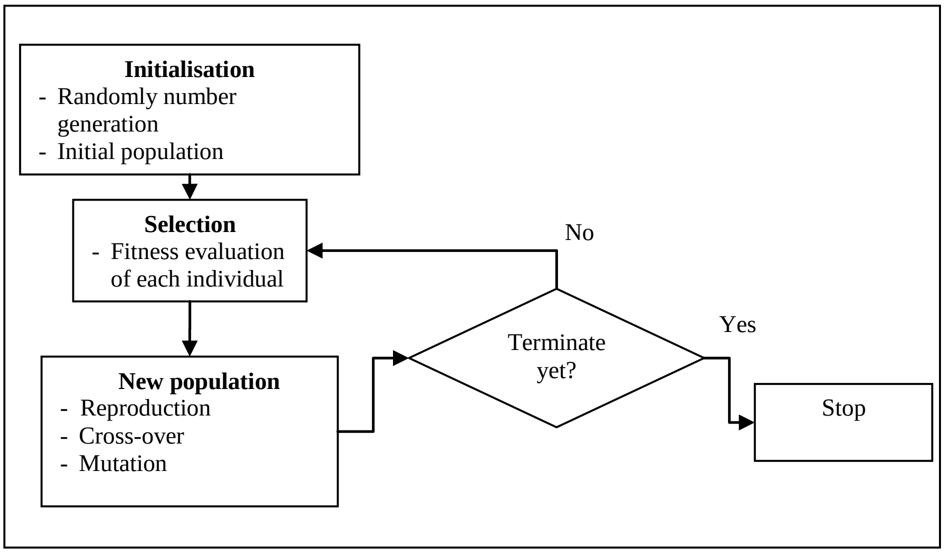

Figure 4.2: Genetic algorithms processing (adapted from Sun et al. (2007))

In the *Reproduction* operation, the GA technique normally starts by creating an initial population of chromosomes using a random number generator of ones and zeros (Sun et al., 2007). It then evaluates each chromosome to find the highest fitness values to be used for subsequent operations which is *Cross-over*. In the *Cross-over* operation, the combination of genes from parents occurs by choosing a random position of the binary string which is exchanged with another string to form a whole new chromosome (Udin, 2004).

According to Noor (2007), the simplest way is to choose randomly some Cross-over point by copying all loci (each position of chromosome is called locus) before the cross-over point from the first parent and then copy all loci after the cross-over point from the other parent. It can be illustrated as follows: (| is the cross-over point):

Chromosome 1: 11011 | 00100

Chromosome 2: **10011 | 11000**

Offspring 1 : 11011 | **11000**

(all loci before cross-over point of Chromosome 1 | all loci after cross-over point of Chromosome 2)

Offspring 2 : **10011** | 00100
(all loci before cross-over point of Chromosome 2 | all loci after cross-over point of Chromosome 1)

Then, this newly created offspring is mutated whereby the elements of DNA are slightly changed. The changes are due to errors during genes copying process from parents. *Mutation* is an operation of changing the binary value of the offspring from zero to one or vice versa and this process will create new possibilities for gene combinations (Hopgood, 2001). As suggested by Noor (2007), *mutation* can be illustrated as follows:

Offspring 1 : 1 1 0 $\boxed{\mathbf{1}}$ 1 | 1 1 0 0 0

Mutated Offspring 1 : 1 1 0 $\boxed{\mathbf{0}}$ 1 | 1 1 0 0 0

Offspring 2 : 1 0 0 1 1 | $\boxed{\mathbf{0}}$ 0 1 0 0

Mutated Offspring 2 : 1 0 0 1 1 | $\boxed{\mathbf{1}}$ 0 1 0 0

Then, a new population is obtained and the cycle is repeated and re-evaluated until the stopping criterion is achieved (Gaafar et al., 2008). Finally, after a number of successive reproductions, the best fit chromosomes will gradually dominate the populations and the less fit will die (Negnevitsky, 2002).

4.1.2 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) is an AI application that is based on computational model of the human brain (Chen and Yan, 2008), meaning that this branch of AI reasoning's model is adopting the biology and neurology knowledge. Therefore, ANN is a mathematical model that tries to follow the human brain's working. ANN consists of interconnected processors, also known as artificial neurons, that imitate the biological neurons in the human brain (Negnevitsky, 2002). These artificial neurons are connected by weighted links in a network architecture that passes signals from one neuron to another neuron in parallel operation. The weight factor expresses the strength and importance of each neuron input. A neural network learns through repeated adjustments of these weights. Hence, ANN is trained so that a particular input finally reaches the specific target output (Noor, 2007). Through this network connections, each neuron receive a number of input signals in the form of raw data or output from other neurons, process the data, and then produce a single output signal. The delivered output could be either the final result or an input to other neurons.

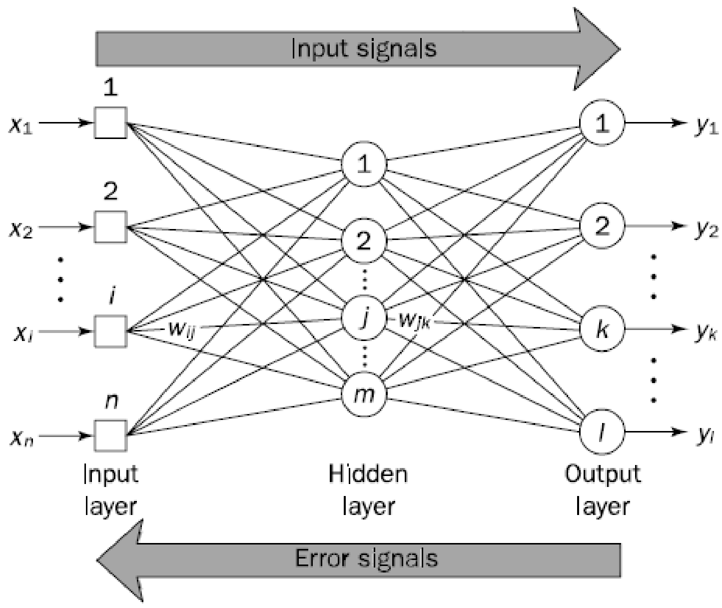

Figure 4.3: Architecture of a typical artificial neural network (Noor, 2007).

The feedforward ANN is structured in three layers, known as input, intermediate (hidden layer) and output as shown in Figure 4.3. The first level is known as the input layer, which receives input signal from outside the network and redistribute these raw data to all neurons in the hidden layer. Normally the input layer rarely processes the input patterns (Hayajneh et al., 2009). In the hidden layer, neurons detect the features through the weight of the input signals and finally the output layer will use these signals to determine the output pattern for the entire network. The learning process in a multilayer network proceeds by making small adjustments in the weights to reduce the difference between the actual and desired outputs.

The learning algorithm starts the training by passing the input pattern to the network input layer. Then, the network propagates the input pattern from layer to layer in the hidden layer until the output pattern is formed by the output layer. If the output result is different from the desired output, an error is calculated based on the differences of the outputs. Then, the error signals will be propagated backwards through the network from the output layer to the input layer. The weights are modified as the error is propagated until the desired output is achieved. In other words, the neuron uses the following transfer or activation function (Negnevitsky, 2002):

$$X = \sum_{i=1}^{n} x_i w_i \qquad \text{Equation (1)}$$

$$Y = \begin{cases} +1 & \text{if } X \geqslant \theta \\ -1 & \text{if X } < \theta \end{cases} \qquad \text{Equation (2)}$$

Where, X is the net weighted input to the neuron, x_i is the value of input i, w_i is the weight of input i, n is the number of neuron inputs, and Y is the output of the neuron. This type of activation function is called a sign function. Thus the actual output of the neuron with a sign activation function can be represented as

$$Y = sign\left[\sum_{i=1}^{n} x_i w_i - \theta\right] \qquad \text{Equation (3)}$$

The learning method in ANN may be supervised or unsupervised. In supervised learning, the ANN needs a guide which is a training set of examples of input and output. The actual output is compared to the desired output based on this training set of acceptable response. During the network passes through the examples, feedback errors will force the weights to be adjusted towards the desired output. The process will continue until it reaches a stage whereby the weights pass without error. At this stage, the network has completed the supervised learning in associating a set of input patterns with the desired output.

In contrast, unsupervised learning or self-supervised learning (Maqsood et al., 2011), no guide is involved in determining the adjustment of the input's weights. In this type of learning, the ANN works independently without the proper indication of the desired output. Through several passes, the ANN faces new experiences and learns how to adjust itself to reach the final goal. Although widely used, ANN's back-propagation learning is not without problems. The main problem is the calculations are extensive before it reaches the reasonable result.

4.1.3 Fuzzy Logic (FL)

Fuzzy Logic (FL) is a concept that simulates the process of human reasoning in dealing with uncertainty (Udin, 2004). Fuzzy, or multi-valued logic is also known as fuzzy sets which is fundamental to mathematics. In real reasoning, processes are most likely not so clear cut and often contain inexact, incomplete or even immeasurable data (Negnevitsky, 2002). Hence, FL attempts to adopt a human sense of the words, decision making and common sense in its model architecture. The FL theory is that an element belongs to a fuzzy set based on a certain degree of membership values in the interval [0,1] (Mascle and Zhao, 2008). The degree or value is not either true or false, but may be partly true or partly false. In essence, an element can belong to more than one set.

The sources of uncertainty are needed to be handled differently and can be anticipated by using statistical approach such as Bayesian updating. According to Hopgood (2001), Bayesian updating suggests that it is possible to infer the probability to every hypothesis and that probabilities can be updated based on evidence. Bayes' rule or Bayes' theorem is an equation which is used by all modern AI systems for probabilistic inference (Russel and Norvig, 2003). The Bayes' rule is written as (Hopgood, 2001):

$$\mathrm{P(H|E)} = \frac{\mathrm{P(H) \; x \; P(E|H)}}{\mathrm{P(E)}}$$

Where:

P(H|E)	=	Posterior probability
P(H)	=	Prior probability of hypothesis
P(E|H)	=	Likelihood of evidence, given H
P(E)	=	Prior probability of evidence

Therefore, Bayes' theorem performs abduction which is determining causes by using deductive information such as the likelihood of effects and evidence (Hopgood, 2001). Users provide information about the evidence observed and the expert system computes P(H|E) for hypothesis H in light of the user-supplied evidence E. Probability P(H|E) is called the posterior probability of hypothesis H upon observing evidence E.

FL is based on a compilation of IF-THEN rules, where the IF part is known as the antecedent (prior) and the THEN part is known as the consequent (resultant). FL commonly has many antecedents that are combined using fuzzy operators such as AND and OR. Once combined, it generates a single truth value that drives the rule's outcome. FL replaces the precise value of a variable by a linguistic description such as short, medium or long. FL provides flexibility compared to a more rigid conventional thinking and decision making process. Unlike the conventional digital logic, FL can accommodate a range of values called a membership set instead of only two extreme sets such as hot and cold (Nawawi, 2009).

Hence, FL gives users an option to make guess or intuition when dealing with a number of possibilities. Thus, FL attempts to define such ambiguous terms such as tall, short, heavy and others (Udin, 2004). For example, a particular height is, tall or short, depending on one's relative definition of tall or short, and it is different from one and another. FL tries to quantify the range with a probability associated with that range so that it indicates whether tall is in the specified range.

Crisp sets or non-fuzzy sets are used to categorize things such as 400°C is in the range of high temperature, then it is neither medium nor low temperature (Hopgood, 2001). The boundary between each set is already clearly defined although the tiny difference between the boundaries such as medium and high temperature is set at 300°C as illustrated in Figure 4.4.

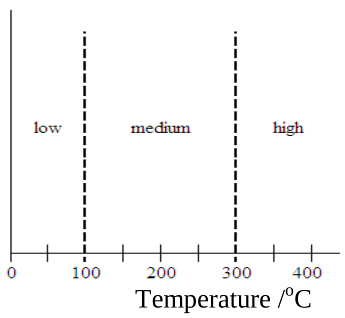

Figure 4.4: Conventional crisp sets (Hopgood, 2001)

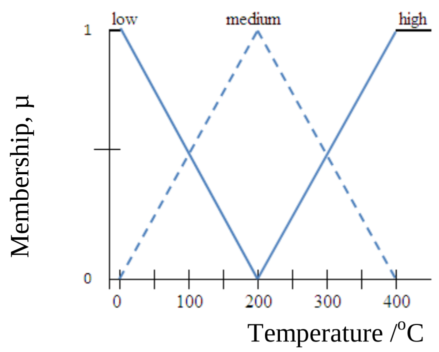

Figure 4.5: Fuzzy sets (Hopgood, 2001)

Hence, a reading of 301°C is considered high, whereas 299°C is considered medium. On the other hand, fuzzy sets are used to smooth out the boundaries so that it clears the dispute of the membership categories as illustrated in Figure 4.5. Therefore, the temperature 350°C may have share of membership to both fuzzy sets medium and high. The sum of the membership functions can be arranged to equal to 1, but it is not always necessary (Hopgood, 2001).

4.1.4 Simulated Annealing (SA)

Simulated Annealing (SA) is a method under the AI branch which adopts the random behaviour of molecules during an annealing process which finally make the metal become stronger. SA uses this annealing process as a random search technique that may be used to solve optimization problems. The actual process of annealing involves slow cooling from a high temperature (Bureerat and Limtragool, 2008). In the cooling process of heated metal, the atoms line in the ordered manner and form a crystal, which is the state of minimum energy in the system or the global minimum. To adopt this analogy, SA uses a temperature as the control parameter that is decreased by iterations until it becomes close to zero (Teti and Kumara, 1997).

Therefore, in solving an optimization problem, a control parameter T, analogous to a temperature, and a constant C, analogous to the Boltzmann's constant is used.

By comparison, both SA and GA start with initial random population and proceed until optimization is achieved. SA simulates the metal cooling and freezing process whereas GA is based on the genetic processes. According to Bureerat and Limtragool (2008), the searching procedure of SA starts with an initial solution or known as parent which will be mutated in the process and leading to a set of children. Then, only the best offspring will become the candidate to challenge its own parent. For minimization purpose, the parent will be replaced by the offspring if it has a lower objective value than that of the parent. However, the offspring can still challenge its parent although it has a higher objective function value than its parent provided that Boltzmann probability accepted it. The best solutions and the parent are the same initially but they can be different during the optimisation process. The iteration stops when the system is frozen or has reached the crystalised state.

SA is an iterative improvement strategy given an objective (cost) function $C(x)$ and initial state vector x_0 trying to improve the current state by perturbing x_0. If the new state x_i produces lower cost, then it replaces the current condition and perturbation process continues from the current condition (x_i) (Oliver and Theruvakatti, 1994). However, if the perturbed condition has higher cost than the original, it will be rejected and the perturbation re-starts from the original condition. This perturbation process continues until no further cost improvement. The algorithm to simulate the SA process can be illustrated as follows (Oliver and Theruvakatti, 1994):

1. Randomly perturb x_i, to some new state x_{i+1}, and calculate the change in cost $\Delta C = C(x_{i+1}) - C(x_i)$
2. If $\Delta C < 0$, accept the new condition
3. If $\Delta C > 0$, accept the new condition with probability $P = \exp(-\Delta C/T)$

4.1.5 Case Based Reasoning (CBR)

The essence of Case Based Reasoning (CBR) is the ability to recall previous experience as what normally the characteristic of human intelligence (Yang and Wang, 2009). CBR recalls and uses previous encountered experience whenever the similar problems happen. The CBR method works by solving the new problems through adapting previously successful solutions to the similar problems. According to Meziane (2000), the intelligent components of CBR contains a history of past problems and the rectification solutions to those particular problems. Thus, the users can solve the future problems by using the analogy of the past cases.

In CBR, a computer learns from refinement of the stored knowledge or experience. Machine learning in CBR adopts reasoning by analogy as a type of algorithm to determine cases comparison to find the best match (Tsai et al., 2005). The processes involved in CBR are in a cyclical form comprising the four *Re*s: *Retrive, Reuse, Revise, and Retain* as shown in Figure 4.6. According to Kim et al. (2002), once a problem occurs, it will be matched against cases in the case base.

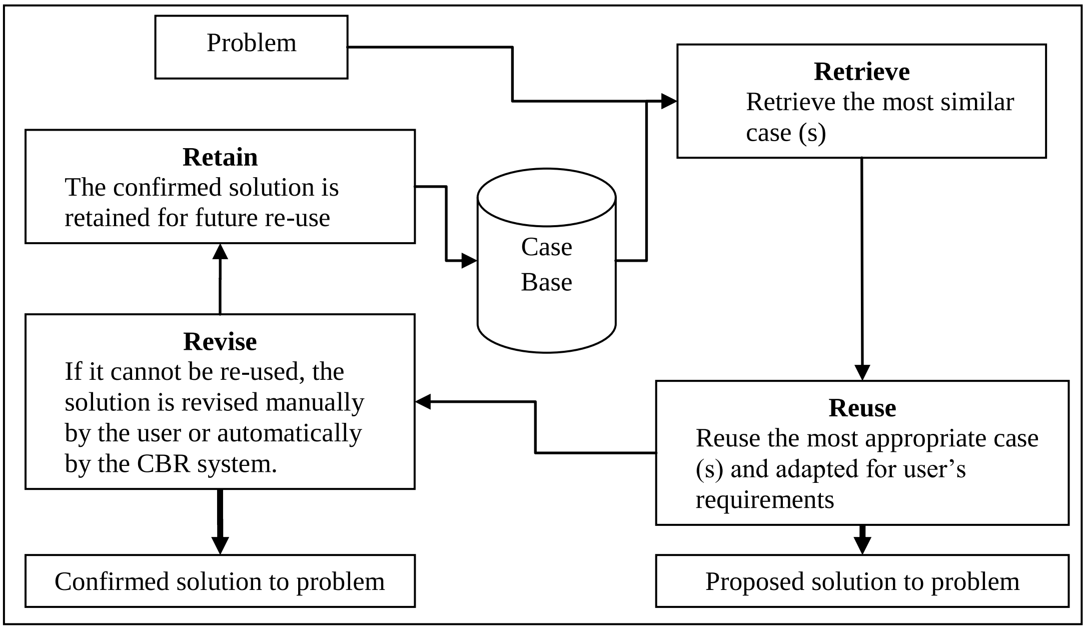

Figure 4.6: CBR cycle (adapted from Kim et al. (2002))

The Case Base is the place where all the previous cases are stored. One or more similar cases are then retrieved to solve the problem. A solution suggested by the matching cases is then reused and tested for a solution. If the match case is not successful, then it will be revised until a confirmed solution is achieved. Then, it becomes a new case which will be retained in the Case Base for future use. The majority of CBR software uses the frame/object representation to deal with the cases.

4.1.6 Frame Based System (FBS)

Frame Based System (FBS) is another knowledge representation and organising technique (Hopgood, 2001). A frame is a data structure that captures and represents knowledge in a frame based expert system (Negnevitsky, 2002). A frame organises knowledge in slots to describe various attributes and characteristics of the object.

FLYJET BOARDING PASS

Carrier:	FLYJET AIRWAYS
Name:	MR. B BROWN
Flight:	FJ 1234
Date:	5MAR2009
Seat:	5A
From:	BRADFORD
To:	LONDON
Boarding:	1120
Gate:	3E

Figure 4.7: Example of a typical frame

An example of a frame is shown Figure 4.7, which shows that a frame combines all necessary knowledge about a particular object or concept such as this boarding pass in one place. Each frame has its own name and a set of slots associated with it. For instance, carrier, name, and flight are slots in the frame

Boarding Pass. These slots have their own value associated to them, such as slot *Flight* has a value of FJ 1234. Whenever the slots are filled with values, an instance of the frame is created. Instance frame is used when referring to a particular object, and the class-frame when referring to a group of similar objects (Shehab and Abdalla, 2001).

A class frame describes a group of objects with common attributes such as computer, insects and food whereas the instance frame describes a specific object such as ABC 345 (computer class-frame) as illustrated in Figure 4.8. In FBS, the system supports the class inheritance which means all the characteristics of a class-frame are assume by the instance frame. In FBS, matching technique is used to match the values of a given entity with the slot values of frames. With a sufficient match, an instance is assumed to be completed.

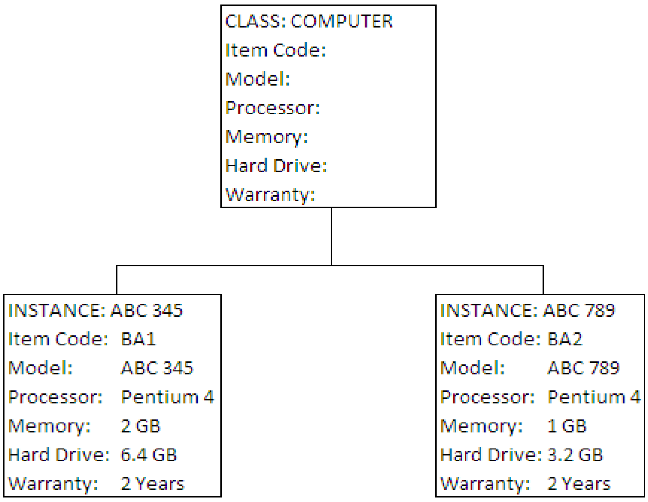

Figure 4.8: Example of class and instances

This is very useful in dealing with hierarchical knowledge because the FBS organises the knowledge for easy inferences improvement (Lau and Mak, 2001). Frames communicate with each other by using methods and demons. A method is a procedure used for frame attribute or slot and normally is in the form of WHEN

CHANGED and WHEN NEEDED. A Demon, on the other hand is a synonym term usually used for the IF-THEN statements.

4.1.7 Expert Systems (ES)/ Knowledge Based System (KBS)

The Expert Systems (ES) or Knowledge Based System (KBS) is a branch of AI that deals with knowledge representation and reasoning. In the old days, they all called Expert Systems. However, when the ES were developed, their performance was limited and was not capable of matching the human expert capabilities. However, the ES did contain considerable knowledge, even though not to the level of human expert performance, hence, these systems were called KBS (Khan et al., 2011). However, these two terms (ES and KBS) are used synonymously. Since the KBS technique will be used as the methodology tool for this research, therefore the understanding of knowledge concept will be necessary in building a hybrid KBS.

4.1.7.1 Knowledge Based System (KBS)

Knowledge Based System (KBS) represents an evolutionary step in Computer Aided Engineering (CAE). KB engineering is an engineering method that merges the Object Oriented Programming (OOP), Artificial Intelligence (AI) techniques and CAD technologies. This system allows the user to model engineering design process and then automate all or parts of the process (Chapman and Pinfold, 2001). The system contains practical knowledge obtained from a human expert, is a problem-solving tool with explicit comprehensive information which is able to explain the reasoning on demand.

According to Sapuan et al. (2006), KBS is one of computer-based tool for concurrent engineering used by automotive manufacturers to reduce cost, increase quality and time compression. Pham and Pham (1999), suggested that KBS is usually made up of two main elements, a knowledge base and an inference mechanism. According to them, "the knowledge base contains domain knowledge which may be expressed as any combination of 'IF–THEN' rules, factual statements, frames, objects, procedures and cases". Hence, according to Chi (2009), a knowledge base is a special type of database, based on expert-centric design approaches for representing domain expertise which comprises collections of facts, rules, and procedures. Therefore, human knowledge must be modelled using these facts, rules, and procedures so that a computer can process the information (Kroo and Manning, 2000).

The final goal of a KBS is to have all the best experts' experience into a single knowledge base (Chapman and Pinfold, 2001). Therefore, to achieve this goal, commitments from all related parties are required during the development stage so that the system represents the actual knowledge requirements in solving problems. Careful consideration must be put forward during the implementation process of KBS, because it is expensive and relative investments are not reversible (Sunnapwar and Kodali, 2006).

According to Udin (2006) and Nawawi (2009), KBS is widely used in business organisations to support the decision making process. Furthermore, KBS are usually defined as computer programs that help solve problems which would normally be done by a human expert. This expert system commonly consists of knowledge core, knowledge exploration and ability to show results. The acquired knowledge of the

KBS is the input from various sources such as human expert, research papers, and books (Benavides and Prado, 2002).

Moreover, the application of KBS provides the opportunity to interact with users, assist in the decision making process and also can be used as a learning device for all members in the organisation (Khan and Wibisono, 2008). KBS also can help in the manufacturing process (Abburi and Dixit, 2006); product design planning, which integrates product specifications and parts characteristics, and link them to a design schedule and cost table (Hung et al., 2008). Manufacturing evaluation system also uses KBS to evaluate the overall manufacturing performance based on Analytical Hierarchy Process (AHP) in identifying opportunities for improvements, comparing internal performance and comparing external competitors (Yang et al., 2009).

The general architecture of the KBS is shown in Figure 4.9, which consists of: user interface and the knowledge acquisition interfaces, inference engine, knowledge base, blackboard, knowledge engineer, expert domain and end user.

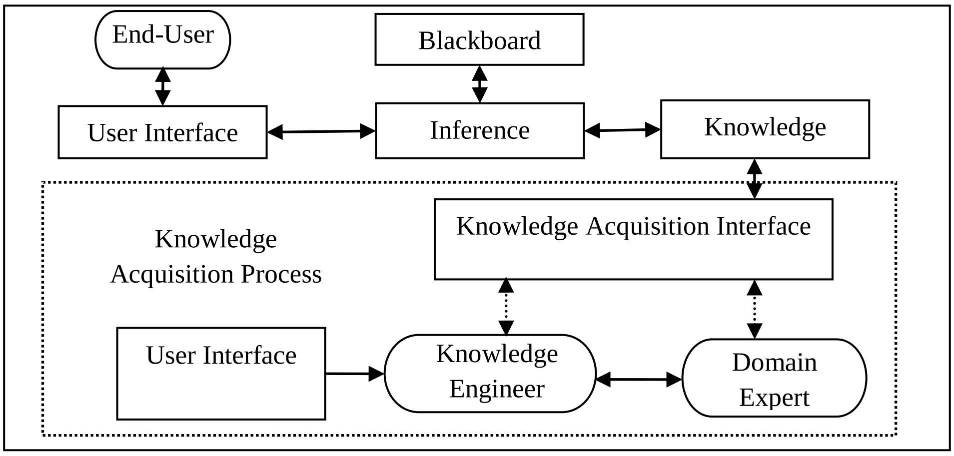

Figure 4.9: Architecture of knowledge based systems (Nawawi, 2009)

The user interface is the communication medium between a user and the KBS. The user wishes to solve the problem and the system will act as the problem solver.

It is important that the communication between the user and the system is as meaningful as possible so that it becomes an effective KBS. For friendly communication between the user and the system, a commonly presented natural language processor should be installed and accompanied with menus and graphics.

The second interface in the KBS is a knowledge acquisition interface. The Knowledge engineer uses this component to supply the expertise and knowledge in the form of rules and facts to the knowledge base. Domain experts can also use this interface to supply the knowledge and expertise directly into the system during the knowledge acquisition process (Udin, 2004).

4.1.7.1.2 Inference Engine

The inference engine is the brain of a KBS and is also known as the control program or rule interpreter (in the rule-based KBS) (Wibisono, 2003). The inference engine is a key component that performs the inferencing and decides how and when the facts and rules in the knowledge base are to be used in solving the problems. To reach the conclusion, the inference engine utilises reasoning techniques in making inferences by accessing the knowledge base.

General programming languages such as *C* and *Pascal* requires the programmer to build the KB system's user interface right from the beginning with the appropriate inference engine, whereas the "shell" programs offer a better advantage by having inference engine and a user interface built in the system (Nawawi, 2009). Shells give an easy starting point for KBS development since they are types of KBS with empty knowledge. Hence, the developer can concentrate on acquiring knowledge in the KB without having to build everything from the beginning of the system development. In KBS, there are two approaches that are used in controlling the inferencing process, known as Forward Chaining and Backward Chaining.

4.1.7.1.3 Forward Chaining

According to Negnevitsky (2002), forward chaining is the data driven approach where the reasoning starts from the known data and proceeds forward with that data until it reaches a conclusion or goal. Using this approach, the KBS will analyse and examine the input with the IF condition in the IF-THEN rule (Udin, 2004). At each step, only the top most rule is executed in the forward direction until the goal is satisfied. This approach is for gathering information and then inferring from it whatever can be inferred. Figure 4.10 illustrates forward and backward chaining techniques.

Initial Facts: A, B, C, D, E

Rules

Rule 1. **IF** A **AND** B **THEN** F

Rule 2. **IF** C **AND** D **THEN** G

Rule 3. **IF** F **AND** G **THEN** H

Rule 4. **IF** E **AND** H **THEN** I

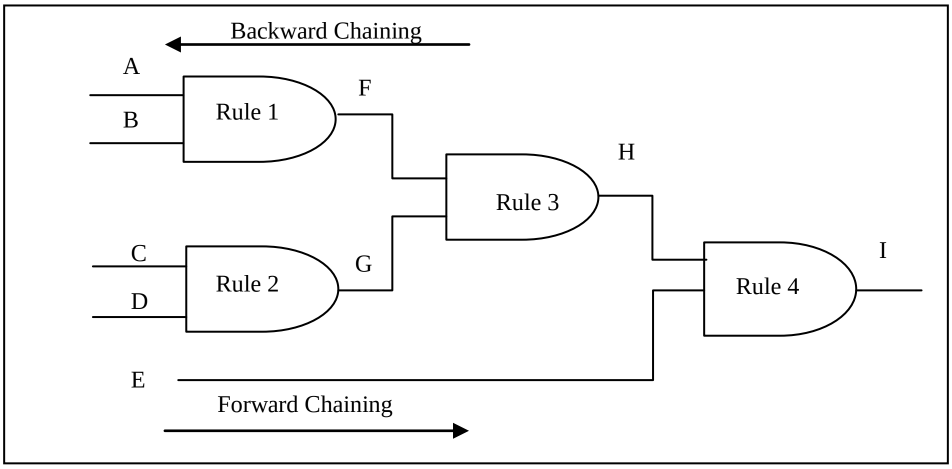

Figure 4.10: Forward chaining and backward chaining approaches

Facts A and B are known, then Rule 1 allows us to know F, so that F is now a known fact. We also know facts C and D, Rule 2 allows us to know G. Since we know now F and G, Rule 3 allows us to know H. Since we now know H and we already know E from the beginning, Rule 4 allows us to know I. Since I is the solution, the chain will stop here.

4.1.7.1.4 Backward Chaining

The second approach in rule-based KBS is the backward chaining or the goal driven reasoning. The KBS starts with a goal and the inference engine attempts to seek the evidence to prove it. It works by searching rules that might have the desired solution and must have the goal in their THEN parts. If such a rule is found and its IF part matches the data in the database, then the rule is triggered and the goal is concluded.

According to Negnevitsky (2002), the choice of forward and backward chaining is dependant on how the domain expert solves a certain problem. Use forward chaining if the experts need to gather some information and later tries to infer from it. On the other hand, use backward chaining if the experts begin with hypothetical solution and then tries to find facts to prove it. Again, referring to Figure 4.10, initially known facts and rules are as for forward chaining in order to illustrate backward chaining.

Backward chaining begins with I as the solution, then, to know I, we need to know E and H by using Rule 4. E is known from the beginning, then, to know H, we need to know F and G by using Rule 3. Next, to know F, we need to know A and B by using Rule 1, and to know G, we need to know C and D by using Rule 2. Therefore, we verify that we know I by using Rules and Facts backward.

4.1.7.1.5 Knowledge Base

The knowledge base contains rules, facts and knowledge acquired from human experts. It contains the knowledge necessary for understanding, formulating and solving specific problem of particular domain (Saaty, 2008). The heart of a knowledge base is the rules that are categorised as definitional and heuristic. Definitional rule defines direct relationships among objects whereas heuristic rule requires experience to solve a certain problem. The knowledge base is not static; as new knowledge becomes available the knowledge base needs to be updated (Khan et al., 2011). The knowledge in the knowledge base is incorporated with the system through a process known as knowledge representation (Maqsood et al., 2011). In representing the knowledge, there are various approaches such as logic representation, semantic networks, production rules, and frames (Nawawi, 2009). Below is the brief explanation about facts and rules.

Facts: Fact is a general statement which refers to either permanent or temporary knowledge applied by an expert (e.g. "stamping is a type of car body part forming process").

Rules: Rules or production rules are the most common method of representing knowledge (Negnevitsky, 2002). Production rules generally apply IF, AND, THEN and OR statements. The IF, AND, THEN and OR are 'condition', 'action', and 'alternative' parts of a statement, respectively (Wibisono, 2003). Normally, each rule comprises of one or more actions and alternatives. A rule implies that if an action taken is true then it will result in some conclusions. Furthermore, the statements between IF and THEN are known as procedural part, whereas the individual statements are known as premises. The statement which follows THEN is known as consequence or conclusion. In a KB System, knowledge is represented in the form of production rules, i.e. IF and THEN, which is the most popular and easiest method to apply (Hopgood, 2001). An example of a KB rule-base is given below:

IF *the manufacturer does have Human Resource Development (HRD) programmes*
AND *the manufacturer's Human Resource Development (HRD) programmes includes helping employees develop their personal skills*
AND *the manufacturer's Human Resource Development (HRD) programmes includes helping employees develop their organisational skills*
AND *the manufacturer's Human Resource Development (HRD) programmes includes helping employees develop their knowledge and abilities*
AND *the manufacturer practices Key Performance Indicator (KPI) for setting the company's target*
AND *the manufacturer does have Performance Measurement System*
THEN *the manufacturer's achievements in HR development programme is good and capable to improve human resources capability*
OR *the manufacturer needs to review its HR development programme to improve human resources capability.*

4.1.7.1.6 Blackboard

The blackboard or working memory (during an operation) is an area for the description of a current problem specified by the user-input data (Udin, 2004). The blackboard system uses the analogy of a physical blackboard for the experts to communicate their ideas in response to the available information on the blackboard (Hopgood, 2001). It attempts to get the experts or knowledge sources having specialised area of knowledge to agree on a solution which cannot be solved by a single expert. Using the blackboard, the experts have equal chance to propose a solution until the process reaches the final solution.

The blackboard records intermediate hypotheses and three types of decisions: a plan, an agenda and a solution. According to Nawawi (2009), as a user enters description on a current problem into the blackboard, the system matches this description with the knowledge contained in the knowledge base to infer new facts.

Then, the new facts are continuously added to the working memory. Therefore, the blackboard is a repository that stores all partial solutions from the independent cooperating experts (Awad, 1996). These experts try to contribute a higher level partial solution to be applied to the current blackboard status.

The blackboard is not a database, however, it is similar to the concept of RAM in computer systems (Udin, 2004). The databases are normally integrated with the KBS whereas the blackboard is a part of the KBS. The contents of the blackboard are changed according to the problem situation and very useful for complex problems that require group of experts.

4.1.7.1.7 End User, Knowledge Engineer and Domain Expert

There are two types of human involvement in KBS. The first type is the end user, who uses the system seeking for a solution to a particular problem. The second type of human involvement is in the knowledge acquisition process. This process requires the participation of a knowledge engineer and the expert. The knowledge engineer, normally the system developer, is the person who seeks and then structures the knowledge from the experts of the particular area through interviews to find out how a particular problem is solved (Khan and Wibisono, 2008). The knowledge engineer is thus responsible for development of a KBS from the initial stage to the final delivery of the system including the maintenance of the system.

The domain expert, on the other hand, is a knowledgeable and skilled person who possesses special knowledge, experience, skills and judgement in solving problems in a specific domain. The expert knows the importance and relationship of the facts and is willing to participate in the system development by providing his knowledge to the knowledge engineer or directly to the knowledge base. Moreover, the expert also provides the skill on how to solve the problem that the KBS will perform. With the help from the expert, the knowledge engineer also extracts the knowledge from written documents and converts it into the KBS. The extracted knowledge should be consistent, accurate and complete in order to make the KBS work effectively (Khan et al., 2011).

4.1.8 Applications of Artificial Intelligent in Manufacturing

Many researchers have discovered the advantages of using AI in the manufacturing environment, especially the applications are widely used in planning and control (Bureerat and Limtragool, 2008), scheduling (Gaafar et al., 2008), design (Mascle and Zhao, 2008), process (Hayajneh et al., 2009), and quality (Kwong et al., 2009). Table 4.2 shows a summary of the research carried out of AI (GA, ANN, FL, SA, CBR, and FBS) applications in various manufacturing areas. It revealed that the applications of AI techniques can be applied for the whole manufacturing processes from planning to the manufacturing of the products.

The applications of KBS are also being utilised in the manufacturing environment. KB Systems were developed for the planning and designing a collaborative supply chain management system (Udin, 2004). Nawawi (2009) developed a KB approach for the planning and designing a collaborative lean manufacturing management. In manufacturing scheduling, Sapuan (2006) utilised the KBS for concurrent engineering technique in the automotive industry. Hung et al. (2008) utilised KB technique for designing a manufacturing cell formation. Abburi and Dixit (2006) developed a KBS model for the prediction of surface roughness in turning process in machining environment. In manufacturing performance measurement systems, a manufacturing evaluation system based on the AHP approach for wafer fabricating industry has been developed by Yang et al. (2009).

In this research, the rule-based KBS is selected to support the planning and designing of LVAM System development. According to According to Udin (2006) and Nawawi (2009), KBS is the most established AI technique and many commercial shells and software are available to support the organisation's decision-making process development. Thus, the process of building the KBS is easier compared to other AI techniques because knowledge and information acquisition from literature and user interactive sessions are easily structured using the KBS rule-base.

Table 4.2: Summary of AI applications in manufacturing

	GA	ANN	FL	SA	CBR	FBS	KBS
Planning and Control	Utilising GA for planning and control (Pham and Pham, 1999)	Utilising NN for planning and control (Pham and Pham, 1999)	Utilising FL for process control (Pham and Pham, 1999)	Utilising SA for planning (Bureerat and Limtragool, 2008)	Utilising CBR for planning and control (Yang and Wang, 2009)	Application of FBS for module planning and control (Negnevitsky, 2002)	Utilising KB in planning and design (Udin et al., 2006)
Scheduling	Utilising GA for scheduling agile environment (Gaafar et al., 2008).	Utilising NN for scheduling (Pham and Pham, 1999)	Utilising FL for manufacturing scheduling (Teti and Kumara, 1997)	Utilising SA for manufacturing scheduling (Teti and Kumara, 1997)	Utilising CBR for scheduling (Watson and Marir, 1994)	Application of FBS for scheduling packet-switched network (Stiliadis and Varma, 1996)	Utilising KB for scheduling (Sapuan et al., 2006)
Design	Utilising GA for product design (Sun et al., 2007)	Utilising NN for product conceptual design (Chen and Yan, 2008)	Utilising FL for product/process design (Mascle and Zhao, 2008)	Utilising SA for commonality design (Shafia et al., 2009)	Utilising CBR for fixture design (Wang and Rong, 2008 , Watson and Marir, 1994)	Application of FBS for design packet-switched network (Stiliadis and Varma, 1996)	Utilising KB for product design planning (Hung et al., 2008)
Process	Utilising GA for process evaluation (Cui et al., 2008)	Utilising NN for manufacturing process (Hayajneh et al., 2009)	Utilising FL for parameter process setting (Lau et al., 2009b)	Utilising SA for multiple process routing (Wu et al., 2009)	Utilising CBR for concurrent engineering process (Qi et al., 2009)	Utilising FBS for process improvement (Lau and Mak, 2001)	Application of KB for manufacturing process (Abburi and Dixit, 2006)
Quality	Utilising GA for quality process setting (Kwong et al., 2009)	Utilising NN for manufacturing quality control (Yu and Xi, 2009)	Utilising FL for intelligent quality management system (Lau et al., 2009a)	Utilising SA for efficient manufacturing system (Wu et al., 2008)	Utilising CBR for a defect prediction system (Tsai et al., 2005)	Utilising FBS for manufacturing cost estimates (Shehab and Abdalla, 2001)	Application of KB for manufacturing evaluation system (Yang et al., 2009)

4.2 Gauging Absences of Pre-requisites (GAP) Analysis Technique

GAP analysis is a method to assess the gap between the manufacturer's necessary pre-requisites for effective (benchmark) implementation compared to its current status quo. According to Nawawi (2009), the GAP technique should be applied in a structured and hierarchical format. There are three main objectives of GAP analysis in developing the LVAM for this research.

The first objective is to identify the main requirement for the effective realisation of LVAM in the automotive industry. By doing the GAP analysis, the main items that are needed for LVAM implementation can be discovered from the proposed hybrid model of the system. The second objective is to compare the current manufacturer's status with the benchmark standards by means of providing a quantitative basis for analysis. The final objective is to identify the strengths and weaknesses of the existing practices in LVAM environment by positioning to the new LVAM proposed model.

4.3 Analytic Hierarchy Process

Analytic Hierarchy Process (AHP) is one of the decision making process tools in selecting the best alternatives for a specific target (Aguilar-Lasserre et al., 2009). It is a powerful and flexible decision making process of guiding people to make the best decisions in situations where both the qualitative and quantitative aspects of a decision are required (Huang, 2009). AHP method processes the complex decisions to a series of pair-wise comparisons until it reaches the judgement by giving a clear rationale for the decisions being concluded. It was developed by Saaty (1980), aimed at combining different measures into a single overall score in prioritising the decision.

The AHP method is widely used in variety of applications such as manufacturing (Wen, 2009), Information Technology (Ngai and Chan, 2005), strategic planning (Huang, 2009), and operation management (Aguilar-Lasserre et al., 2009). AHP provides advantages to decision makers because it is a systematic analysis method designed for multi-criteria decision that decomposes and structures a complex problem into several levels (Aguilar-Lasserre et al., 2009). The AHP then weights the alternatives and makes comparisons amongst the alternatives before suggesting the final decision (Nawawi, 2009). The AHP development process proposes a structured methodology in organising the analytical thought based on three basic principles: structuring hierarchies, setting priorities and logical consistency (Aguilar-Lasserre et al., 2009) as shown in Figure 4.11.

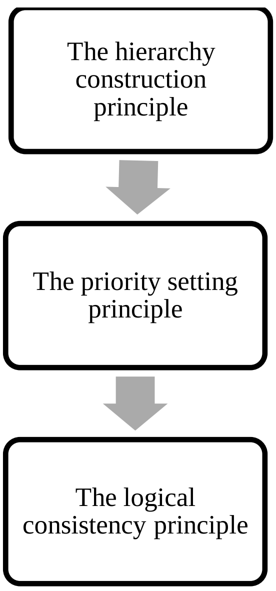

Figure 4.11: Basic steps in implementing AHP

4.3.1 The hierarchy construction principle

The main objective of AHP technique is to decompose the complex systems into levels of hierarchy elements (Aguilar-Lasserre et al., 2009). There are three levels of the hierarchy structure of AHP ranked in decreasing order of importance, as shown Figure 4.12; namely top level, intermediate level and the lowest level (Wen, 2009). The top level is the goal or focus of the system, the intermediate level is the criteria of the system which consists of several attributes. The lowest level of the structure is known as alternatives, which contribute whether positively or negatively towards the main objective through their impact on the criteria in the intermediate level. The intermediate level consists of criteria or attributes that may have several elements that affect the decision (Udin, 2004).

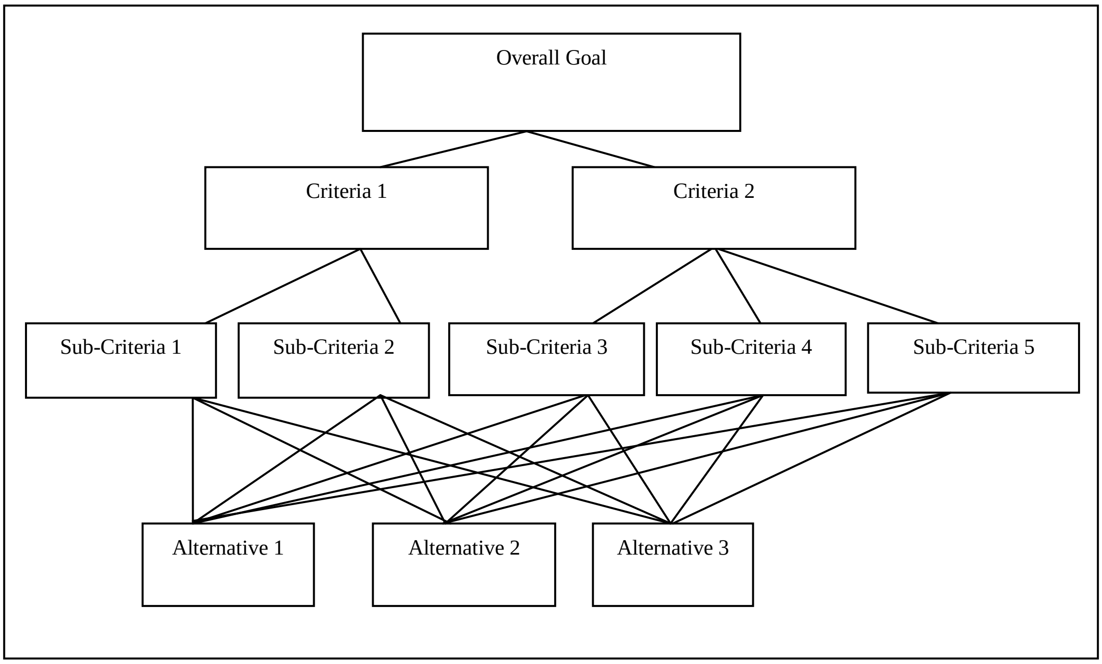

Figure 4.12: Structure of hierarchy (adapted from Ngai and Chan (2005))

4.3.2 The priority setting principle

The structure of hierarchy is used to construct a set of pair-wise comparison matrices whereby each element in an upper level is compared with the elements in the lower level immediately with respect to it (Saaty, 2008). The priorities obtained from the comparison are used to weight each of the elements to determine their importance. This process is continued until the final priorities of the bottom most level is compared. Scale of numbers as in Table 4.3 is needed to make a pair-wise comparison to quantify the judgement.

Table 4.3: Scale for pair-wise comparisons (Saaty, 2008)

Intensity of Importance	Definition	Explanation
1	Equal importance	Two activities contribute equally to the objective
2	Weak or slight	
3	Moderate importance	Experience and judgement slightly favour one activity over another
4	Moderate plus	
5	Strong importance	Experience and judgement strongly favour one activity over another
6	Strong plus	
7	Very strong or demonstrated importance	An activity is favoured very strongly over another; its dominance demonstrated in practice
8	Very, very strong	
9	Extreme importance	The evidence favouring one activity over another is of the highest possible order of affirmation

The elements in the AHP are structured into the form of a matrix for consistency testing. Figure 4.13 below illustrates the sample of a square matrix that provides a consistency testing. The comparison process begins by selecting the property or basis (C) from the top of hierarchy, while elements in the next level of hierarchy are selected for comparison. The consistency testing is done based on the normalised matrix by using Consistency Ratio (CR) in order to make sure the judgement is good (Aguilar-Lasserre et al., 2009).

C	A1	A2	A3
A1	1	A1/A2	A1/A3
A2	A2/A1	1	A2/A3
A3	A3/A1	A3/A2	1

Figure 4.13: Matrix for pair-wise comparison

4.3.3 The Logical consistency principle

The consistency principle in AHP technique determines the consistency of the matrix according to the Consistency Ratio (CR). The consistency of the matrix is important because the decisions made by users are subjective and depending on circumstances (Aguilar-Lasserre et al., 2009). The value of CR should not exceed 0.10 which is the acceptable upper limit for CR, which implies a 10% probability that the elements have not been properly judged and should review the comparisons (Huang, 2009). Then, the mathematical calculation integrates the weights to develop overall evaluation of the decision process determination.

4.4 Summary

This chapter has provided a review of Artificial Intelligence (AI) and Knowledge Based Systems (KBS), their structures, and applications in manufacturing environment. In designing a hybrid KBS, key AI techniques are studied. Among the AI techniques are GA, NN and SA, all of which require complex modelling in their system. The KBS, FL, CBR, and FBS normally use IF-

THEN rules. Each of the AI method discussed has its own advantage for the manufacturing environment, especially the KBS applications which are widely used in five main areas: planning and control, scheduling, design, process, and quality. This application gives some advantages to manufacturers in managing the LVAM, and helps them in reducing cost and improving quality especially when related to decision making, design and implementation processes.

This chapter also reviewed the GAP analysis method to assess the gap between the manufacturer’s existing conditions and the standard practice for benchmarking implementation in a structured and hierarchical format. There were three main objectives of GAP analysis in developing the LVAM for this research: to identify the main requirement for the effective realisation of LVAM in the automotive industry, to compare the current manufacturer’s status with the benchmark standards by means of providing a quantitative basis for analysis, and to identify the strengths and weaknesses of the existing practices in LVAM environment by positioning to the new LVAM proposed model.

Embedded in the proposed LVAM model is AHP technique. AHP is a tool which is used to support multi-attribute problems by prioritising the areas that are needed for improvement, and based on a series of questions that have been analysed by the GAP analysis technique. The process of transferring, calculating and displaying the AHP prioritisation has been discussed in this chapter. The following chapter will now propose a framework for a KBLVAM System which is specific to the planning and design of LVAM model.