The Connective Approach in Expert System Technology

Tay Boon Hou, PhD
Artificial Intelligence Laboratory
IN Technology Pte Ltd
10, Haig Lane, Singapore 438814
Tel: (65)63455419 / Fax: (65)63458251
Email: intech01@singnet.com.sg

ABSTRACT

The connective approach is developed by Sir Peter Strawson, Emeritus Professor of Philosophy at Oxford University, as an effective way to understand the fundamental structure of human thinking in the field of analytical philosophy. This paper provides the justifications and insights for extending the work of Strawson to expert system technology. A distinct advantage of adopting a connective approach stems from the fact that it does not suffer from the circular problem encountered by a typical expert system in the reductionist approach. The connective approach uses three dimensions, namely, ontology, epistemology and logic. The completed diagnostic model is an elaborate network of connected items and concepts that enables equipment troubleshooting to be carried out at higher level of complexity.

Key words: Expert System; Connective Approach; Ontology, Epistemology, and Logic.

OVERVIEW

A typical expert system makes use of a set to rules to arrive at its conclusion. This set of rules represents the experiences of a human expert in problem solving. However, these rules are just illusions. They cannot become objects of perception as they really are. This lack of "things as they really are" or "that physically exist in the real world" is the main cause for the brittle nature of expert systems. In this paper, a connective approach is adopted to enhance an automotive diagnostic expert system (DES). As described in Tay (2003), the connective approach uses three dimensions, namely, *ontology*, *epistemology* and *logic*. The ontological dimension expands the scope of the knowledge base that is normally dominated by the epistemological dimension. The treatment of logic as a distinct dimension enables the DES software engine to compute the logical states derived from the other two dimensions. The conclusion drawn from these logical states enables the DES software engine to narrow down its scope of search within the knowledge base saving valuable time.

A distinct advantage of using the connective approach stems from the fact it does not suffer from the circular problem encountered by a typical expert system. A typical expert system uses the reductionist approach. A reductionist approach is the process of dismantling a complex problem into simpler elements. The process only terminates when pieces that cannot be further dismantled are reached. However, the reductionist approach has not even begun if one of the alleged pieces turns out to be the very thing or the very concept that is to be dismantled. The connective approach does not suffer from this circular problem. In this approach, the model is an elaborate network of connected items and concepts. The function of each item or each concept could be properly understood only by grasping its connections with the others. In this model, there is no reason to be worried if the process of tracing connections from one point to another of the network returns to the starting point. Suppose the process of fault finding has returned to a starting component such as the engine. Although the actual failure within the engine cannot be elucidated, one can still confirm that the fault resides in the engine as loops keep routing back to the engine. Therefore, a vehicle can be recovered by replacing the entire engine instead.

This paper focuses on the feasibility of implementing the connective approach in an automotive diagnostic expert system. The concepts and theories relevant to this study are presented in the following sections:

- The meaning of ontology, epistemology and logic.
- The difference between expert system and simulation. The former focuses on the epistemological dimension whereas the latter concentrates on the ontological dimension.
- Problems encountered in expert system and simulation.
- The feasibility of incorporating the three dimensions of ontology, epistemology and logic in a DES.
- Enabling equipment troubleshooting at higher level of complexity.

THE MEANING OF ONTOLOGY, EPISTEMOLOGY AND LOGIC

At the heart of the connective approach are the three distinct dimensions of ontology, epistemology and logic. The meaning of each dimension is explained in this section.

Ontology is defined by Reber (1995) and Zuber-Skerritt (2001) as an aspect of metaphysical inquiry concerned with the question of existence apart from specific objects and events. It is one’s assumptions about the nature of being and reality. As pointed out by Nita (1999), a fundamental ontological question is whether “reality” is something waiting “out there” to be found or revealed by investigative effort, or whether human consciousness “creates” its own reality. Based on Nita’s explanation, ontology takes on two meanings. The first meaning takes reference to the real world, where experience is characterized in terms of what is “out there”. The second meaning includes belief in the existence of the things in question such that these things are separated and related in time and space. It does not include belief in the existence of the properties or attributes in question. Strawson (1992) calls the second meaning an objective reality. Objective reality is explained by Strawson (1992, p55) in the following paragraph:

> *Our picture of objective reality is a picture of a world in which things are separated and related in time and space; in which different particular objects coexist and have histories; in which different events happen*

> *successively and simultaneously; in which different processes compete themselves over time.*

The notion of objective reality brings out the fact that what we perceive or think about an object is not simply a photographic image of the object. Firstly, as pointed out by Heylighen (1999), this image is not isomorphic to the phenomenon it is supposed to represent. For example, when we think about the number 168, we do not see 168 dots in our mind. There is no structural similarity between the concept of 168 and the collection of 168 screws in the warehouse. Secondly, consider the interesting question brought up by Heylighen (1999, p15): "if perception is nothing but the projection of images onto a screen, and memory not different from a set of photographic prints of those projections, then who is looking at the screen, and shuffling through the photographs?" To answer this question, one needs to understand the notions of "world" and "I" described by Sokolowski (2000). The meaning for "world" is stated below by Sokolowski (2000, p44):

> *The world is not an astronomical concept; it is a concept related to our immediate experience. The world is the ultimate setting for ourselves and for all the things we experience. The world is the concrete and actual whole for experience.*

Thus, the world is like a context, a setting, a background, or a horizon for all the things that are given to us. The world is given to us as one that is encompassing all the items. The notion of "I" is explained by Sokolowski (2000, p44) in the following paragraph:

> *Paradoxically, the "I" is a thing in the world, but it is a thing like no other: it is a thing in the world that also cognitively has the world, the thing to whom the world as a whole, with all the things in it, manifests itself. The "I" is the dative of manifestation. It is the entity to whom the world and all the things in it can be given, the one who can receive the world in knowledge.*

Therefore, the answer to Heylighen's question is to introduce "I" as the little person who is sitting somewhere in the human brain. "I" is looking at different incoming and stored mental images in order to decide what they really mean, what should be done within them and which aspects should attention be paid to.

Sokolowski (2000) has brought up another important point. The "world" as a whole and the "I" as the centre are two singularities between which all things can be placed. The "world" and "I" are correlated with one another to provide an ultimate dual, elliptical context for everything. Therefore, the mind is a public thing. It acts and manifests itself out in the open. It is not just inside its own confines. The mind and the world are correlated with one another. Things do appear to us. And we, on our part, do display, both to ourselves and to others, the way things are. Thus, the notion of "world" and "I" reinforces the fact that ontology is concerned with things that are "out there".

Epistemology is defined by Reber (1995) and Zuber-Skerritt (2001) as the branch of philosophy that is concerned with the origins, nature, methods and limits of human knowledge. It is our assumptions about the nature of knowledge and knowing. As pointed out by Nita (1999), epistemology is either something objective, to be accumulated independently of the perceptions of any particular observer or something subjective, a product created by the observer. In other words, epistemology is the use of concepts in judgement or belief. It refers to the personal and subjective phenomenon. The experience is characterized in terms of what is "in the head" of humans. To appreciate its distinction from ontology, consider whether or not the "pink" monkey seen by an alcoholic counts as an experience.

Logic is described by Reber (1995, p423) as:

> *The normative branch of philosophy that deals with the criteria of validity in thought, the cannons of correct prediction and the principle of reasoning and demonstration. Logic concerns only the reasoning process, not the end result. Incorrect conclusions can be reached through logical means if the original assumptions are faulty.*

In Strawson's view, logic is the study of the general forms of the proposition and of their relations of logical dependence and independence. It has no concern with the internal structure of uncompounded propositions that enter into its compounds. It has nothing to say about the content of logically simple propositions. It has nothing to do with an ontological order. According to Bench-Capon (1990), the main concern of logic is with the soundness and unsoundness of arguments. Its goal is to represent an argument in such a way that it will be uncontroversial as to whether that argument is acceptable or not. An argument comprises a set of premises and a conclusion. The premises are known to be true or are accepted as true for the purposes of argument. The conclusion is then said to be true based on the strength of these premises. Although the statements used in an argument have content and form, the soundness of these arguments follows from their form and is entirely independent of their content. In other words, any argument of this form, no matter what it is about, will also be sound. The term "soundness" is defined by Richards (1989, p15) as "A proof theory is sound if it will not permit the deduction of a false conclusion from true premises. Often called consistency."

The importance of consistency in logic is highlighted in the following statement by Sokolowski (2000, p104):

> *Logical consistency is a necessary condition for truth of statements. If the statements contradict themselves by virtue of their logical form, then a priori they cannot be verified by our experience of the things themselves.*

Therefore, as pointed by Ross (1995) and Blackburn (1996), the aim of logic is to make explicit the rules by which inferences may be drawn. It is not to study the actual reasoning processes that people use. In fact, these actual reasoning processes may or may not conform to those rules used by logic.

THE DIFFERENCE BETWEEN EXPERT SYSTEM AND SIMULATION

The terms “expert systems” and “knowledge-based systems” can be used interchangeably as highlighted by Mcleod (1995) and Turban and Aronson (2001).

Widman and Loparo (1989) and Zikmund, Middlemist and Middlemist (1995) describe an expert system or a knowledge-based system as a program that is based on the expertise, rules of thumb, and knowledge of specialists in the field. It reproduces the behaviour of a human expert within a narrow domain of knowledge. However, as pointed out by Buchanan and Smith (1989) and Dubrin (1990), designers of expert systems captured only human knowledge in problem solving. They do not commit to building psychological models of how an expert thinks. The captured knowledge is used to support decision-making. It is not used to automate the decision-making process. An expert system can be thought of as a model that is more of the expert’s model of the domain than that of the expert. The epistemological dimension of an expert system is reflected from its use of human expertise stored in its knowledge base.

Simulation, on the other hand, is the process of predicting the future state of a real system by studying an idealized computer model of the real system. The term “simulation” is described by Scheid (1988) and Darby (2000) as the representation of selected characteristics of the behaviour of one system by another system. Simulation experiments are usually performed to obtain predictive information that would be costly or impractical to obtain with real devices - for example, a simulation can be used to determine the optimum capacity and layout of a military vehicle. The ontological dimension of simulations is inferred from the fact that these applications attempt to match their predicted behaviours closely with those observed in the real systems.

Apart from the difference between simulation and knowledge-based programming, the worldviews of the two fields also differ considerably. The different worldviews of these two fields are stated below by Widman and Loparo (1989, p2):

> *The worldview of knowledge-based programming favours abstraction, generality, and elegance whereas the worldview of simulation favours practical utility, precision, and reliability.*

PROBLEMS ENCOUNTERED IN THE EXPERT SYSTEM AND SIMULATION

In expert system technology, two important techniques, namely, forward chaining and backward chaining, are used to make decisions from the set of rules. These rules are stored in the knowledge base. The rules are represented as a form of interconnected and nested IF-THEN statements as shown in the Table 1

In the forward chaining approach described by Laudon and Laudon (2000) and Levine, Drang and Edelson (1991), the inference engine begins with the information entered by the user. It then searches the knowledge base to arrive at a conclusion. The strategy is to fire, or carry out, the action of the rule when a condition is true. As pointed out by Mcleod (1995), when the condition is true, the rule is fired and the next rule is examined. When the condition is false, the rule is not fired and the next rule is examined. Let me illustrate the forward chaining approach using the set of rules captured in Table 1.

Rule 1: IF coolant = Low, THEN Engine Coolant = Low.

Rule 2: IF coolant = Low, THEN Transmission Coolant = Low.

Rule 3: If Engine Coolant = Low, THEN Vehicle Temp = High

Rule 4: If Transmission Coolant = Low, THEN Vehicle Temp = High

Rule 5: If Vehicle Temp = High, THEN Vehicle Power = Low.

Rule 6: If Vehicle Power = Low, THEN Vehicle = stall

Rule 7: If Engine Filter = Choked, THEN Engine Coolant = Low

Table 1 The knowledge base of an expert system.

In Table 1, when the user enters a condition (Coolant = Low), the inference engine will fire all the rules that meet this condition. It triggers Rule 1 and Rule 2 and obtains two conclusions from these rules, namely, “Engine Coolant = Low” and “Transmission Coolant = Low”. The user is then prompted to select one of these conclusions. Suppose the conclusion “Engine Coolant = Low” is selected by the user. This conclusion is then converted into the new condition “Engine Coolant = Low”. The inference engine will scan through the entire Table 1 again. At this stage, it triggers Rule 3 and retrieves the conclusion “Vehicle Temp = High”. The user is then prompted again to confirm the conclusion “Vehicle Temp = High”. Upon confirmation, the inference engine will convert it into a new condition. Rule 5 is triggered after it has performed its search in Table 1. From the confirmation of Rule 5, the inference engine will repeat its search in the table to trigger Rule 6. After Rule 6, the inference engine cannot find any rule whose condition matches the conclusion of Rule 6. The inference engine will stop after that. In this case, the expert system predicts that the vehicle would stall should the coolant level be low.

Although the same knowledge base is adopted, the backward chaining approach operates in the opposite manner of the forward chaining approach. As pointed out by Laudon and Laudon (2000) and Levine, Drang and Edelson (1991a), in the backward chaining approach, the strategy for searching the knowledge base starts with a hypothesis. It then proceeds by asking the user questions about selected facts until the hypothesis is either confirmed or disproved. In other words, the event has already happened, and the goal is to find out why. The backward chaining approach works by searching for conclusions, instantiating causes, and seeing if these causes link to earlier conclusions.

In Table 1, when the user enters a conclusion “Vehicle = Stall”, the inference engine will fire all the rules that meet this conclusion. The inference engine scans through the entire table. Rule 6 is invoked as its consequence matches the conclusion entered by the user. The reason for the vehicle stalling is the condition “Vehicle Power =

Low" used in Rule 6. The chain of reasoning then continues with the question "Why Vehicle Power = Low ". This question is addressed by Rule 5 which identifies the consequence of "Why Vehicle Power = Low". From Rule 5, the inference engine gets the new reason of "Vehicle Temp = High" and uses it to invoke Rule 3 and Rule 4. At this stage, there are two possible reasons, namely, "Engine Coolant = Low" or "Transmission Coolant = Low". The user is then prompted by inference engine to select one of these reasons. Suppose the user select the reason "Transmission Coolant = Low". The inference engine will continue its chain of reasoning to invoke Rule 2. As there are no more rules to be invoked after Rule 1, the inference engine concludes that "Coolant = Low" is the main cause of the vehicle stalling.

The lack of the ontological dimension constitutes the brittle nature of both approaches. In the forward chaining approach, the responsibility of the ontological aspect lies with the crew. As shown in the above example, the context was based on the fact that coolant dissipated through the cooling system. However, when there was a leak in the coolant tank, the set of rules in Table 1 could no longer be used to predict the leak. In the case of backward chaining, the lack of ontological dimension was reflected in its derivation of the main cause for the stalled vehicle. If we observe the chain of reasoning carefully, the search should be completed at the point when the user had confirmed that the transmission coolant level was low. Instead, the inference engine went further to conclude that the main cause was due to the low coolant level. This was due to the inability of the inference engine to understand the structure of the cooling system leading the user to an incomplete conclusion.

Problems were also encountered in simulations. The models used in simulators can be represented by a set of quantitative mathematical equations, a set of qualitative mathematical equations or a combination of both. According to Northrop (1944) and Devlin (2000), in order to derive a mathematical theory, the mathematicians must first define the objects with which they are going to work. These objects can be numbers, points or lines. They then lay down certain laws called axioms to govern the behaviour of the objects they have defined. After this point, there is no longer any need to know about the phenomenon that led to those axioms in the first place. On this foundation they built, through a series of logical arguments, a whole structure of mathematical propositions, with each proposition resting on the conclusion established preceding it. They are not interested in the truth of the axioms but ask only that these axioms be consistent.

If all truths can be deduced from its axioms, it is called complete. However, the claim to completeness of the mathematical models were dashed in 1931, when an Austrian mathematician, Kurt Gödel proved a result that was to change our view of mathematics forever.

As mentioned in the works of Paulos (1991), Dossey (1992), Dehaene (1997), Lavine (1998), Nolt, Rohatyn and Varzi (1998), Barrow (1999), Dewdney (1999), Kaplan (2000) and Barrow (2001), Gödel's Theorem says that if we write down any consistent axiom system for some reasonably large part of mathematics, then that axiom system must be incomplete. There will always be some questions that cannot be answered on the basis of the axioms.

The proof of Gödel's Theorem is difficult because it refers to itself. If Gödel's Theorem is true, then it cannot ultimately be proved to be true within the system of logic within which it is true. This is because the axioms or fundamental assumptions on which Gödel's Theorem rests cannot be proved within its own system, otherwise, it would be like the judge of a court investigating complaints against himself or herself.

Although Gödel's Theorem is not strictly a logical paradox and its existence does not prove Gödel's Theorem, it does help us to understand it. Devlin (2000, p62) has provided an easy-to-understand liar paradox as shown below:

> *The liar paradox refers to a person who stands up and says. "I am lying." If this assertion is true, then the person is indeed lying, which means that what he says is false. On the other hand, if the assertion is false, then the person must not be lying, so what he says is true. Either way the assertion is contradictory.*

A logical paradox is a specific kind of contradiction. A logical paradox is a statement that is false if it is true; and true if it is false. Gödel's Theorem is not a logical paradox because the truth of the Theorem does not imply that the Theorem is false, but only that the Theorem cannot be proved.

The incompleteness of arithmetics using the Gödel's Theorem was summarised by Casti (1996, p163) in the statement: "No single set of rules will able to fence in all possible true statements about numbers".

In the axiomatization game as suggested by Northrop and Devlin, the best we can do is to assume the consistency of the axioms and hope that our axioms are rich enough to enable us to solve the problems of highest concern to us. We have to accept the fact that we will be unable to solve all the problems using our axioms. There will always be true propositions that we cannot prove from these axioms.

Therefore, the inherent incompleteness of the axiom systems used in simulation software means that there are some real system behaviours indescribable by the adopted mathematical expressions used by these applications.

FEASIBILITY OF INCORPORATING THE THREE DIMENSIONS OF ONTOLOGY, EPISTEMOLOGY AND LOGIC IN DES

The difficulties encountered in expert systems and that of simulations can be explained by the reductionist approach used in these applications.

The aim of the reductionist approach is to get a clear grasp of complex meanings by reducing them, without remainder, to simple meanings. For example, radar analysis stops with photons, and electrical circuit analysis stops with electrons.

However, the reductionist approach suffers from the problem of circularity when one of the alleged pieces turns out to be the very thing or the very concept to be dismantled. As illustrated in the backward chaining example, the inference engine cannot proceed if the user enters "Coolant = Low" as the conclusion at the start of the session. And in simulations, the predicted behaviours are confined to the set of behaviours asserted by the axioms.

The connective approach is more realistic than the reductionist approach. The connective approach is one of tracing connections in a system rather than reducing the complex system into simple or simpler elements. Therefore, it will not suffer from the circular problem if the tracing of connections returns to the starting point.

As pointed out by Strawson (1992), the concepts to be included in the connective approach must be highly general, irreducible, and non-contingent.

The term “general” is described by Reber (1995) as a judgement or decision that is applicable to an entire class or category of objects, events or phenomena. For example, the word “body” can be used as a general term to refer to the engine, the radiator and the ignition switch.

The term “irreducibility” does not mean or imply “simple”. A concept may be complex, in the sense that its elucidation requires the establishment of its connections with other concepts. At the same time, it is also irreducible, in the sense that it cannot be defined away, without circularity, in terms of those other concepts to which it is necessarily related.

The term “non-contingency” is defined by Reber (1995) as something strengthened by an event. It is considered to be the learned response of a bond between that response and stimulus. It occurs independently of any behaviour. It has a role to play in the development and maintenance of that behaviour. Beyond that, the very concept of experience itself would be lost.

Ontology, epistemology and logic are identified by Strawson (1992) as the three dimensions of a unified enquiry. Therefore, a connective approach is established when these three dimensions can be incorporated in a DES.

In a DES, the ontological dimension is represented by a functional block diagram. The epistemological dimension is represented by a set of fault trees. The logic dimension is by the DES software engine. A functional block diagram is a diagram that shows the function, interrelationships, and interdependencies of components used in a vehicle operation. A fault tree, on the other hand, provides the facts about the failure effects and causes of a component used in the functional block diagram.

The use of the three dimensions in DES can be justified using the notions of generality, irreducibility and non-contingency.

Firstly, the three dimensions are general in nature. The functional block diagram is based on the general theory of being. It applies the notion of ontology by ensuring all its components take reference to the real world. All the components used in the functional block diagram must be found in the physical vehicle being modeled. A fault tree adopts the general theory of knowledge. It represents the collective body of information possessed by the modelers and designers. The DES software engine uses the general theory of proposition. It is concerned with what is true or false.

Secondly, all the three dimensions are irreducible. This is based on the fact that against judgements or beliefs used in a fault tree is the natural world or the functional block diagram to which the judgements or beliefs relate. In order to determine whether the judgements or beliefs are true or false, the DES software engine is required to process the states gathered from the relations of judgements or beliefs used in a fault tree with that of the natural world or the functional block diagram.

Thirdly, the three dimensions are non-contingent. Each element contains a distinct feature that is non-contingent. The functional block diagram is concerned with things that are "out there". A fault tree is concerned with things that are "in the heads" of humans. And the DES software engine is only concerned with the reasoning process. It interprets neither the content of the things in a functional block diagram nor the concepts used in the set of fault trees.

The application of the three dimensions can be illustrated using the electrical circuit shown in figure 1.

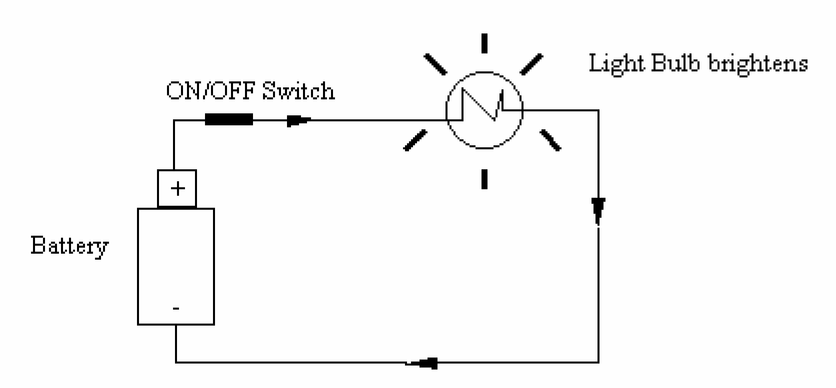


Figure 1 The headlight circuit of a car.

Using the functional approach, the above electrical circuit is represented in figure 2.

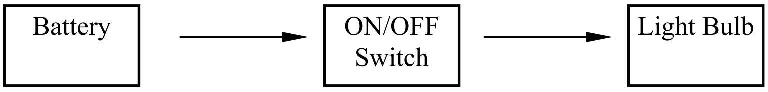


Figure 2 The functional block diagram for the headlight circuit of a car.

The behaviour of the functional block diagram is described as: On activation of the ON/OFF switch, the battery provides electrical current to light up the bulb.

In figure 2, the rectangular boxes and arrows are the symbols adopted for the functional block diagram representation. The three rectangular boxes indicate the objects. Each arrow is used for two purposes. Firstly, it is used to indicate the temporal relationship between two components. This temporal relationship is asymmetrical in nature as pointed by Iwasaki and Simon (1986). For example, the battery must come before the ON/OFF switch. Secondly, it serves as the conduit to propagate the logical state generated by any component used in the functional block diagram.

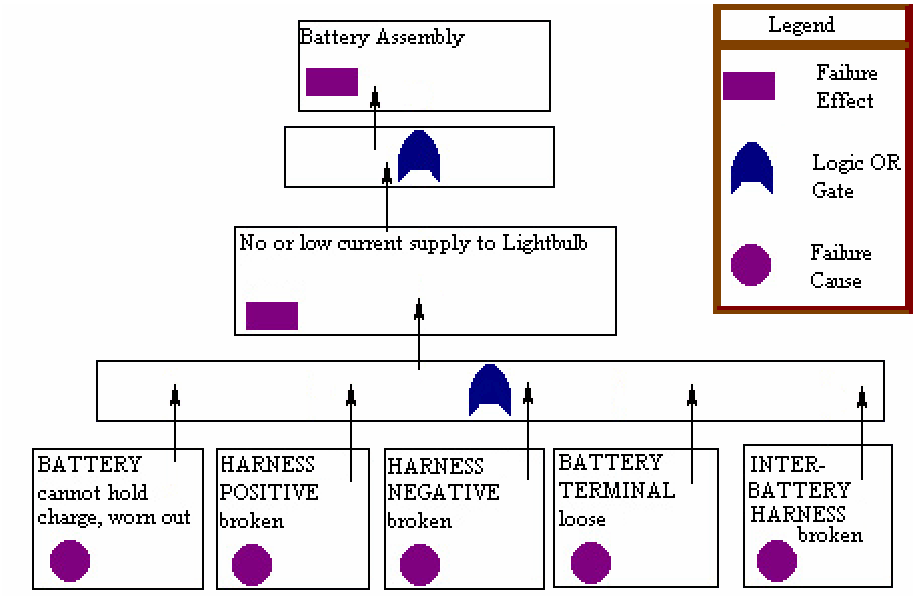
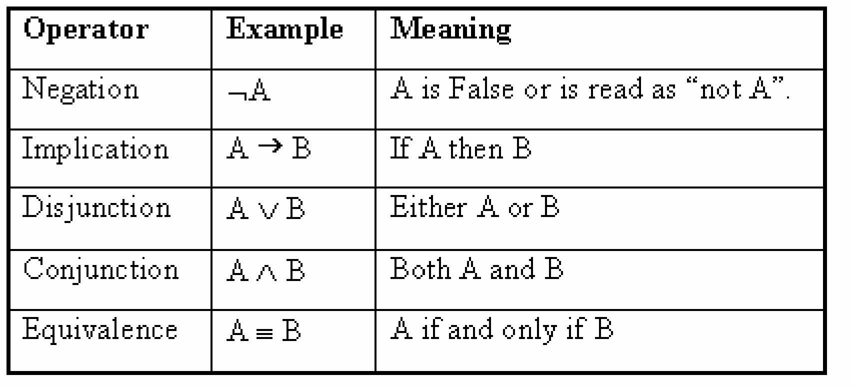


Figure 3 **The fault tree for the battery assembly of a car.**

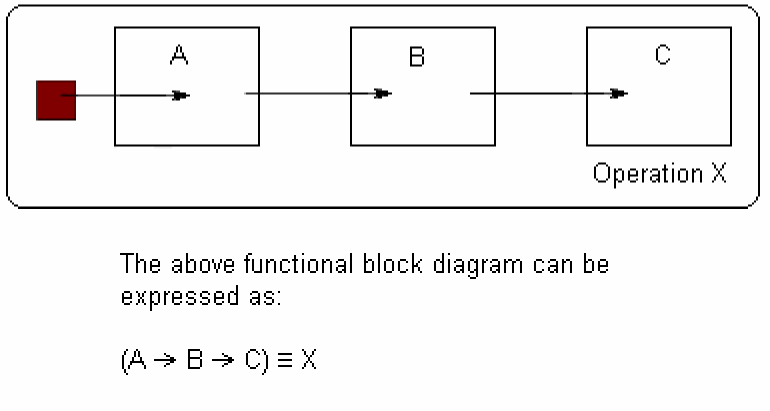
Figure 3 shows the fault tree of the battery. The sub components of the battery, also known as the failure causes, are represented by rectangles with a circular symbol. The failure effect is represented by the rectangle with a rectangular symbol. The failure effect and the failure causes are connected using the rectangles with a logic OR gate symbol. The arrows are the conduits used to propagate the logical state generated by the failure causes or the failure effect in the fault tree. The connection between a function block diagram and a set of fault trees can be viewed as a mind map. A mind-map is described by Buzan and Buzan (2000) as an expression of radiant thinking. It allows the function block diagram to be viewed as the central image of attention. The set of fault trees then radiates from this central image.

The next task is to connect logic with the ontological dimension of the functional block diagram and the epistemological dimension represented by the set of fault trees. As shown in figure 4 and figure 5, a functional block diagram and a fault tree can be represented using the information rules of propositional calculus as described in the works of Klir and Folger (1988), Bench-Capon (1990), and Manktelow (1999). These information rules are listed in Table 2. A proposition is expressed by a sentence that says something is true or false. Parkin (2000, p171) states that: “a proposition represents the smallest unit of meaning to which we can answer true or false”. For example, “Light bulb is turned on.” is a proposition but just “Light bulb” alone is not. A compound proposition is made up of two or more propositions. As pointed by Blackburn (1996), a well-formed formula (wff) is a compound proposition that satisfies the information rules of propositional calculus. A well-formed formula uses two types of symbols in its notation. First, it uses upper-case letters to represent the propositional variables. Secondly, it uses operators to represent the ways in which propositions are connected within arguments.



Operator	Example	Meaning
Negation	¬A	A is False or is read as “not A”.
Implication	A → B	If A then B
Disjunction	A ∨ B	Either A or B
Conjunction	A ∧ B	Both A and B
Equivalence	A ≡ B	A if and only if B

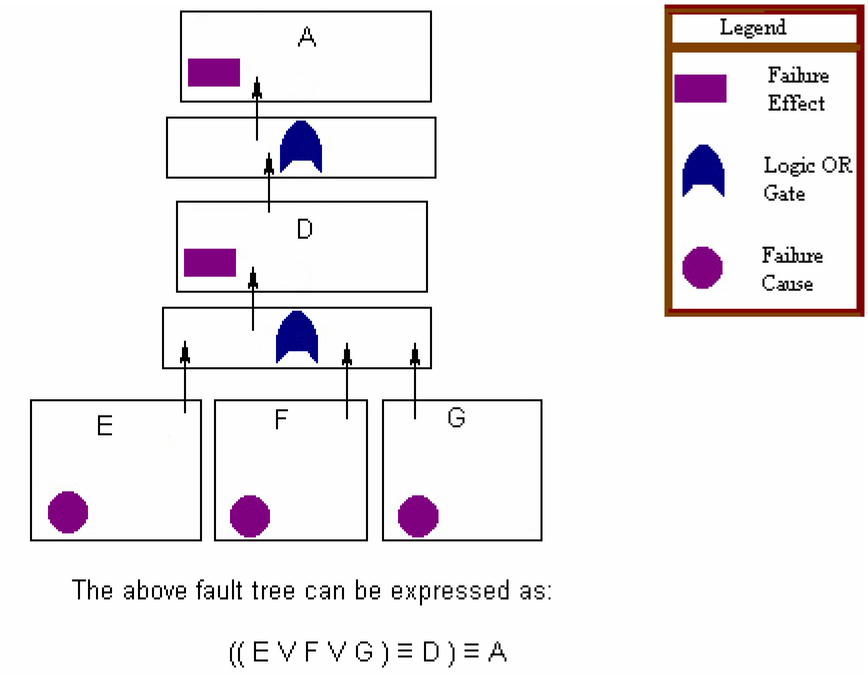
Table 2 The information rules of propositional calculus.



The above functional block diagram can be expressed as:

$(A \rightarrow B \rightarrow C) \equiv X$

Figure 4 A functional block diagram expressed as a well-formed formula (wff).



The above fault tree can be expressed as:

$((E \vee F \vee G) \equiv D) \equiv A$

Figure 5 A fault tree expressed as a well-formed formula (wff).

By representing the functional block diagrams and fault trees as well-formed formulae, they can be converted into a single form that can be processed by the DES software engine. For example, the headlight circuit can be represented by combining the expressions in figure 4 and figure 5 to yield the expression:

$$((((E \vee F \vee G) \equiv D) \equiv A) \rightarrow B \rightarrow C) \equiv X$$

The inter-relationship of the three aspects of DES is illustrated using figure 6a, figure 6b and figure 6c. In figure 6a, the DES software engine concentrates on the ontological dimension by walking through the entire functional block diagram to decide which is the best component to start with. In this case, it has decided to start with component 3.

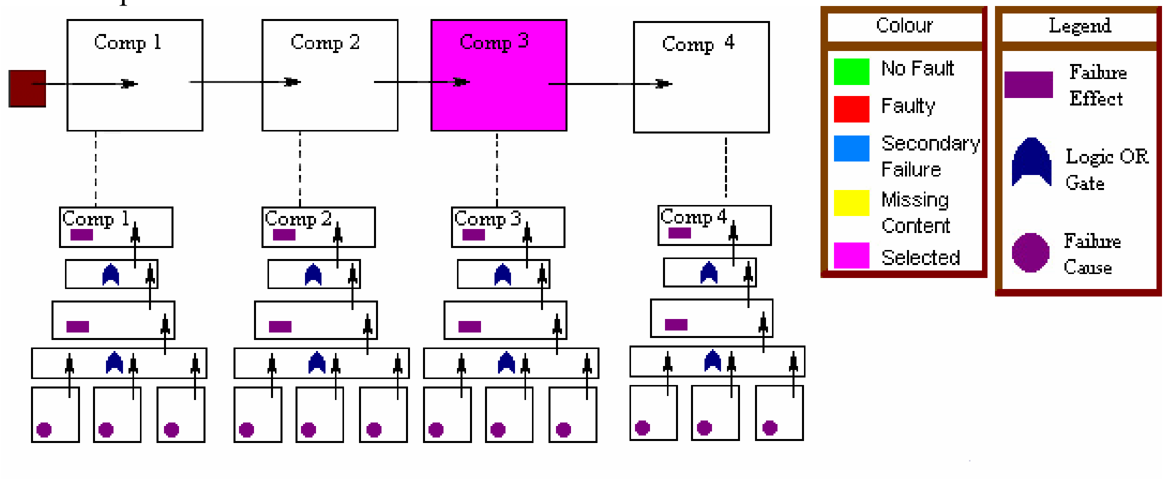


Figure 6a **The DES software engine focuses on the ontological dimension.**

Next, the DES software engine switches its attention to the epistemological dimension by looking at the fault tree of component 3. From the uppermost node (i.e. layer 1) of the tree, it evaluates the logic expression using layer 2 (the logic OR gate operator) and layer 3 (the failure effect). As there is no logical value (i.e. true, false or unknown) associated with the failure effect, it selects the failure effect as shown in figure 6b. It then prompts a set of inspection procedures for the user to check against the failure effect.

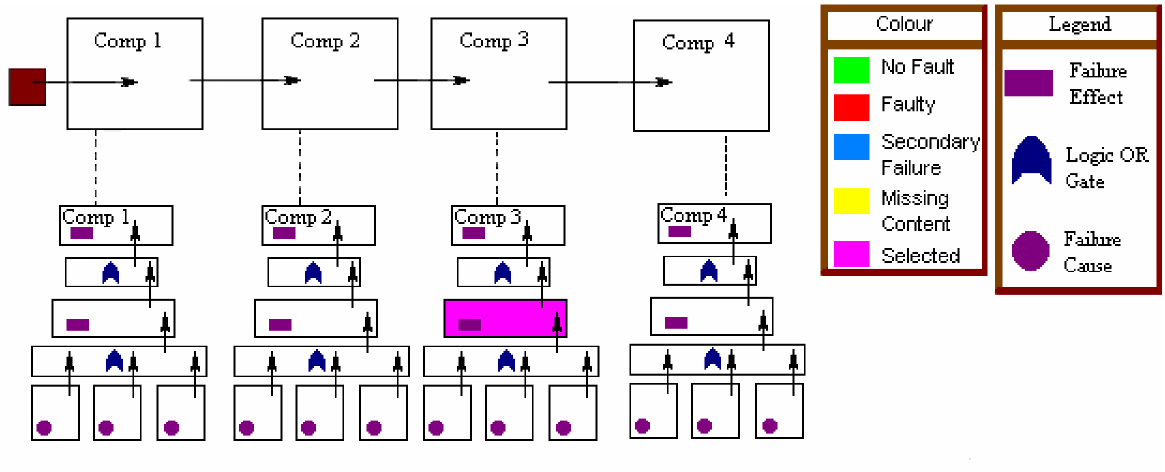


Figure 6b **The DES software engine focuses on the epistemological dimension.**

The user's answer in response to the inspection procedures is converted into a logical value. This logical value is assigned for that particular failure effect. Once the logical value is established (i.e. a true value in this illustration), the DES engine switches to its logic dimension. It propagates the logical value within the fault tree and then upwards into the functional block diagram as shown in figure 6c.

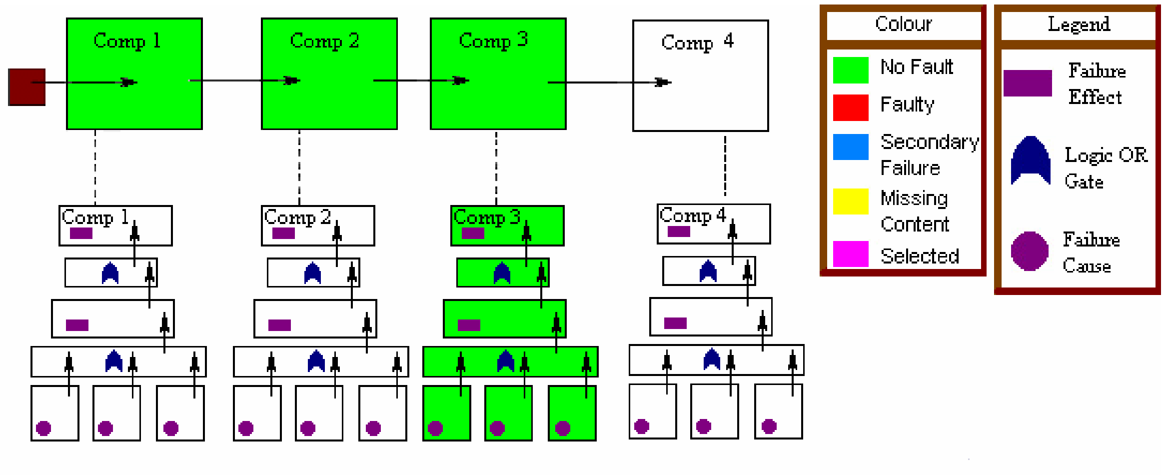


Figure 6c **The DES software engine focuses on the logic dimension.**

Therefore, it is feasible to implement the three dimensions of ontology, epistemology and logic in DES. As shown in the above illustrations, the three dimensions are integrated and interrelated. They cannot be reduced to one another and no one is dominating over the rest.

However, the use of logic has led to the next problem: How to model a system completely if the axiomatic system adopted is always incomplete based on Gödel's Theorem?

This problem can be easily addressed by incorporating the notion of systems thinking in the DES model represented by the functional block diagram and the set of fault trees. As described in the works of Heylighen (1995), Day (1995), Checkland (1999), Davies (2001) and Maani and Cavana (2002), the notion of systems thinking is where the whole is more than the sum of its parts.

Based on Gödel's Theorem, it is not possible to solve all problems using the axioms. There will always be true propositions that cannot be proved from these axioms. However, the completeness of the system is restored if the final conclusion of the well-formed formula that is used to represent the system can be generalised further to include unprovable but true propositions. This approach is feasible in the context of this study. For example, the final conclusion may be used to represent the entire vehicle. When a fault cannot be elucidated, the whole vehicle is replaced with a new one.

Therefore, a top-down modeling approach is preferred over a bottom-up approach. The top-down modeling approach helps the modelers to capture the wholeness of the vehicle first before they proceed to work on the details of sub-components.

Figure 7 illustrates the top-down approach used in modeling a selected vehicle operation such as starting, driving, stopping or parking a vehicle. The modelers begin their modeling at point X. They look at the vehicle as a whole and learn how to perform the selected vehicle operation. Details to be captured by the modelers at this stage are descriptions on performing the selected vehicle operation. Before they proceed to refine their descriptions to include the set of components used in the selected vehicle operation, they have to restart from point X by performing a quick review on the overall aspect of the vehicle. This is to ensure that the modelers remember the wholeness of the selected vehicle operation before they proceed to work on the details of their models. Likewise, to expand the components into smaller sub-components, the modelers have to return to point X again and to refine their models in the direction of the spiral arrow. Ultimately, the modelers will stop at resolution level B as shown in figure 7 due to the given project schedule. It is important to note that the wholeness of the vehicle operation is retained throughout the modeling process as represented by the vertical line C in figure 7. Otherwise, the model will not be complete should the wholeness of the vehicle operation identified at the level indicated by the horizontal line A be lost.

Therefore, the incorporation of the notion of systems thinking in DES has not only solved the issue of incompleteness in Gödel’s Theorem but it also justifies the use of the logic dimension.

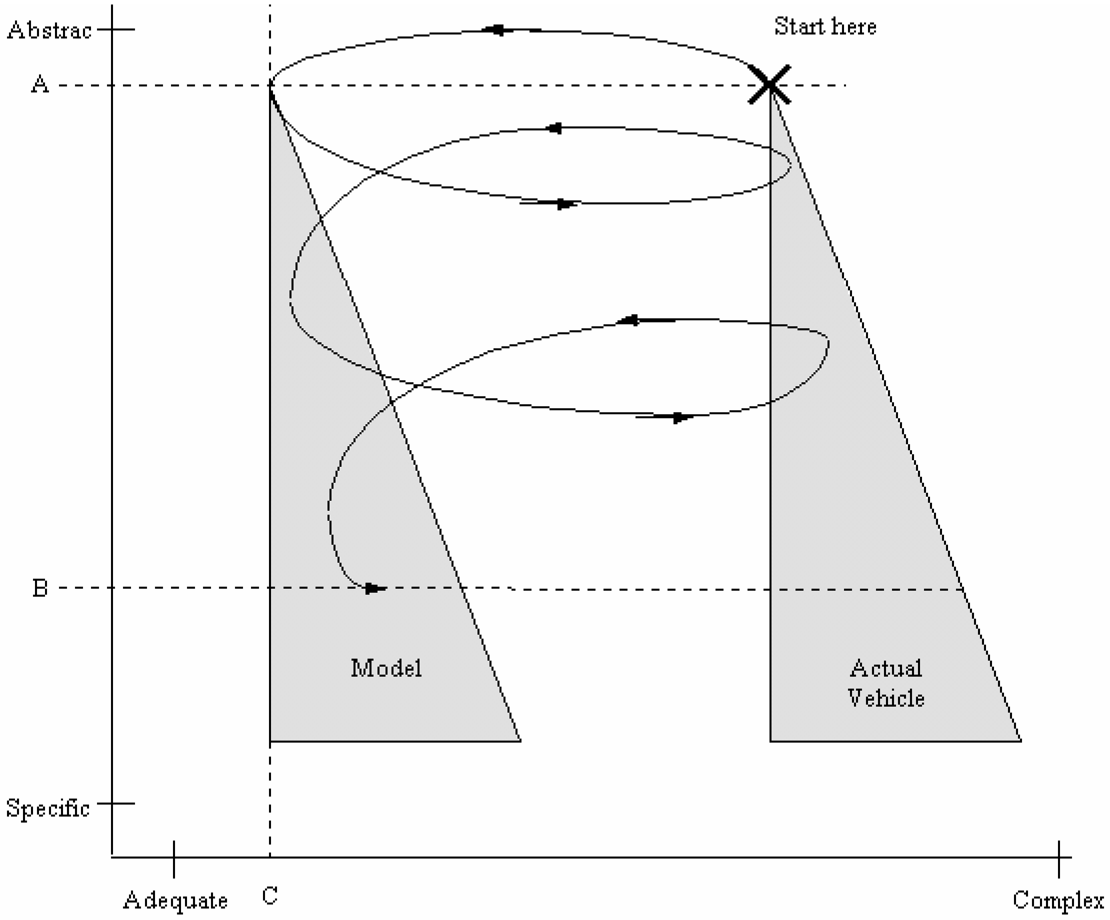


Figure 7 The top-down modeling approach.

ENABLING EQUIPMENT TROUBLESHOOTING AT HIGHER LEVEL OF COMPLEXITY

In this section, let me illustrate how the connective approach can be used for enabling equipment troubleshooting at higher level of complexity. Suppose the operation in figure 8a fails due to a failure cause that has not been captured by the modelers. The missing failure cause (marked in yellow) has caused an invalid logical conclusion to be detected at node 3.1 by the DES software engine.

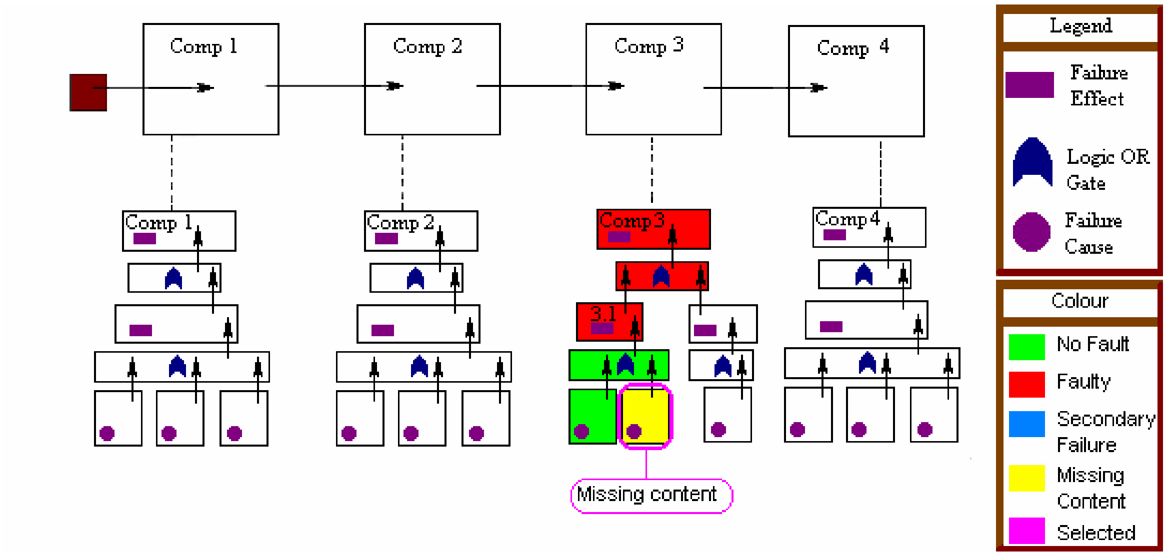


Figure 8a **The missing content has caused an invalid logical conclusion at node 3.1.**

The DES software engine moves upwards into the functional block diagram and applies blue to mark "secondary failure" to components 3 and 4 (see figure 8b).

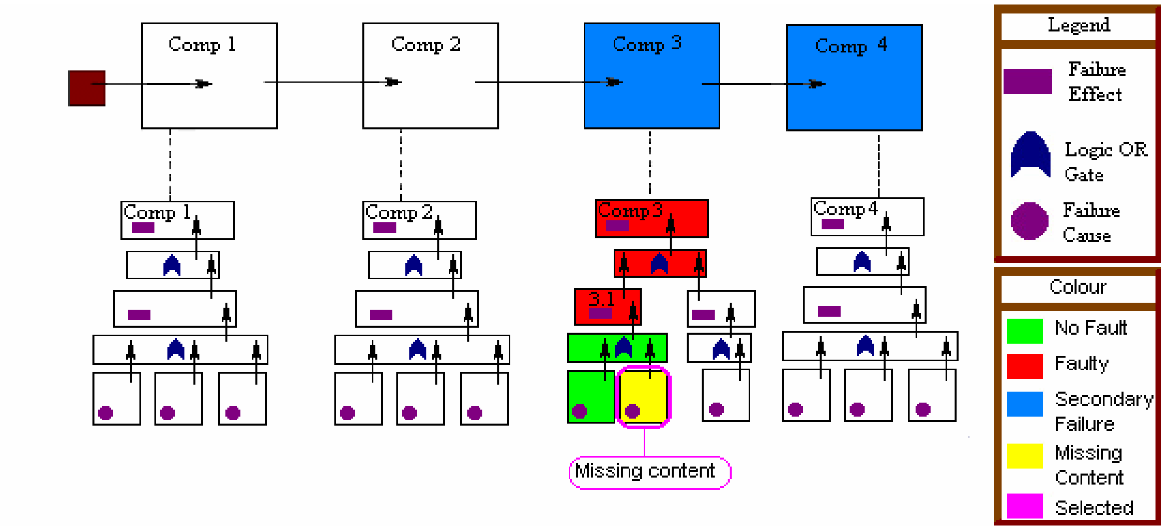


Figure 8b **Components 3 and 4 are marked as "secondary failures".**

The DES software engine enters the fault tree of component 2, confirms it is working and applies green to mark "no fault" to components 1 and 2 (see figure 8c).

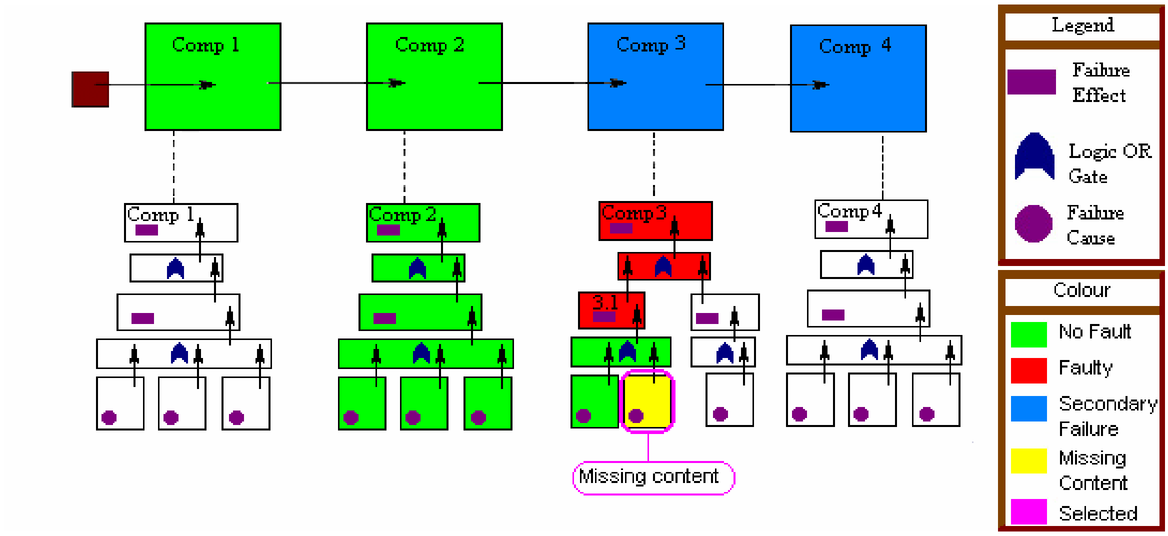


Figure 8c Components 1 and 2 are marked as “no fault”.

At this stage, the DES software engine concludes that the fault lies in component 3. It applies red to mark “faulty” to component 3 (see figure 8d). It also resets the failure effect of node 3.1 except nodes that are already marked in green.

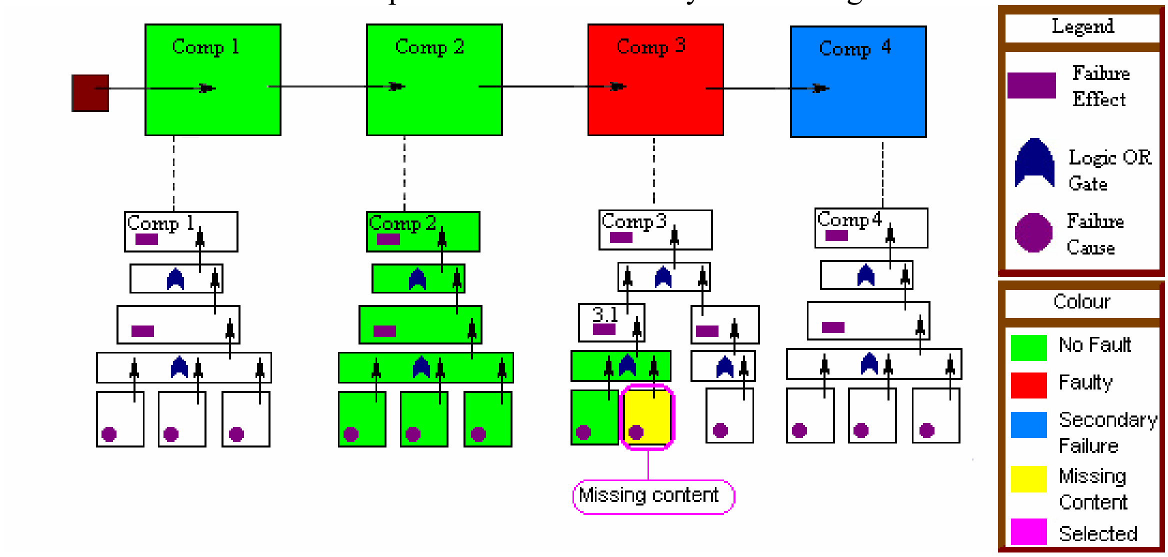


Figure 8d Component 3 is marked as “faulty”.

In figure 8e, an invalid logical conclusion is again obtained at node 3.1. At this stage, the DES registers node 3.1 as having some missing content. It proceeds to prompt the user to replace the entire component 3.

When component 3 is being replaced, the entire fault tree of component 3 is marked green as shown in figure 8f. This is because the fault has been removed. Next, component 4 is being reset to allow the user to perform further troubleshooting should there be another fault encountered in the remaining components of the functional block diagram.

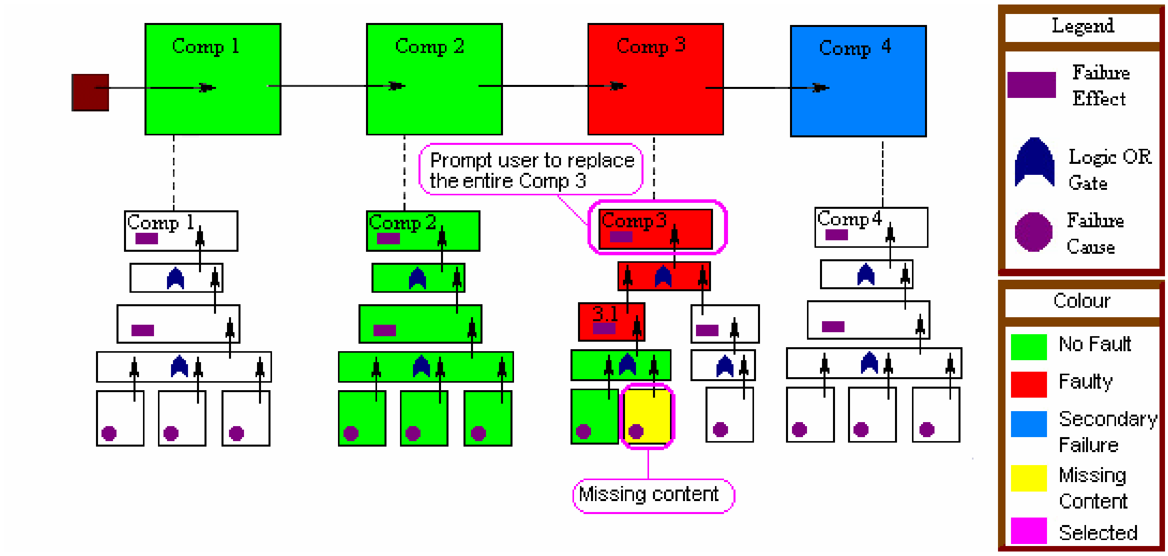


Figure 8e User is prompted to replace the entire component 3.

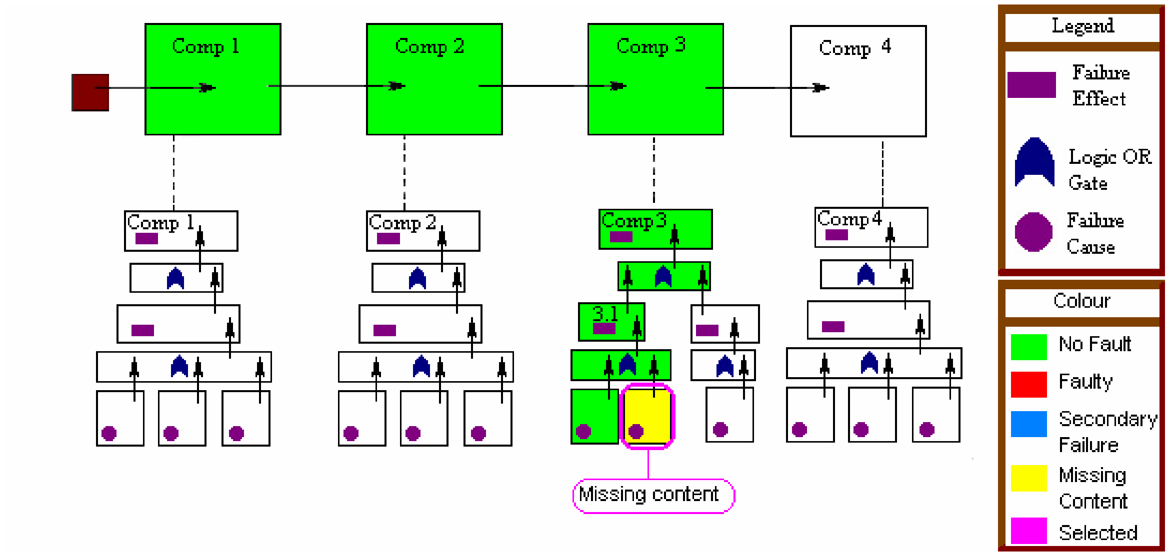


Figure 8f The fault is solved after a new component 3 is installed.

SUMMARY

The connective approach is more realistic than the reductionist approach. It incorporates ontology, epistemology and logic as three distinct and equally dominating dimensions in a DES application used for equipment troubleshooting. A distinct advantage of adopting this connective approach in DES stems from the fact it does not suffer from the circular problem encountered by a typical expert system based on the reductionist approach. The connective approach is one of tracing connections in a system rather than reducing the complex to simple or simpler elements. Therefore, as reflected by the work done in this chapter, it is feasible to adopt the connective approach in DES. In addition, adoption of connective approach also helps to reduce the number of diagnostic steps needed in a troubleshooting session as well as enabling equipment troubleshooting at higher level of complexity.

REFERENCES

Barrow, J. D. (1999), *Impossibility: The Limits of Science And The Science Of Limits,* London: Vintage. First published by Oxford University Press in 1998.

Barrow, J. D. (2001), *The Book of Nothing,* London: Vintage. First published by Jonathan Cape in 2000.

Bench-capon, T.J.M. (1990), *Logic,* Knowledge Representation: An Approach to Artificial Intelligence, 27-40, Volume 32, in the APIC series, eds. Shave, M. J. R. and Wand, I. C., London: Academic Press Ltd.

Blackburn, S. (1996), *The Oxford Dictionary of Philosophy,* Oxford: Oxford University Press. First published in 1994.

Buchanan, B.G. Smith R.G. (1989), *Fundamentals of Expert Systems,* The handbook of Artificial Intelligence. Volume IV, 149-192, eds. Barr, A., Cohen, P. R. and Feigenbaum, E. A., New York: Addison-Wesley Publishing Company, Inc

Buzan, T. and Buzan, B. (2000), *The Mind Map Book. Revised edition,* London: BBC Worldwide Limited. First published in 1993.

Casti, J. L. (1996), *Five Golden Rules: Great Theories of 20th-Century Mathematics – and why they matter,* New York: John Wiley & Sons, Inc.

Checkland, P. (1999), *Systems Thinking, Systems Practice. Includes a 30-year retrospective,* Chichester: John Wiley & Sons, Ltd.

Darby, M. L. (2000), *Simulation-based Training – Beneath The Shroud,* Journal of Battlefield Technology, Volume 3, Number 1, 42-53.

Davies, A. (2001) *Action Research and Systems Thinking,* Effective Change Management Using Action Learning and Action Research, 75-86, eds. Sankaran, S., Dick, B., Passfield, R. and Swepson, P., Lismore: Southern Cross University Press.

Day, R. D. (1995), *The Science of Family Science,* in edsResearch and Theory in Family Science, 91-101, eds. Day, R. D., Gilbert, K. R., Settles, B. H. and Burr, W. R., California: Brooks/Cole Publishing Company.

Dehaene, S. (1997), *The Number Sense: How the Mind Creates Mathematics,* New York: Oxford University Press.
Devlin, K. (2000), *Mathematics: The Science of Patterns,* 3rd edition, New York: Scientific American Library. First published in 1994 and second published in 1997.

Dewdney, A. K. (1999), *A Mathematical Mystery Tour: Discovering The Truth And Beauty Of The Cosmos*. New York: John Wiley & Sons Inc.

Dossey, J. A. (1992), *The Nature Of mathematics: Its Role and Its Influence,* in Handbook of Research on Mathematics Teaching and Learning, 39-48, ed. Grouws, D. A., New York: Simon & Schuster Macmillan.

Dubrin, A. J. (1990), *Essentials of Management,* Cincinnati: South-Western Publishing Co. Second edition.

Heylighen, F. (1995), *Downward Causation*, in Heylighen, F., Joslyn, C., and Turchin, V.: Principia Cybernetica Web (Principia Cybernetica, Brussels), URL: http://pespmc1.vub.ac.be/downcaus.html

Heylighen, F. (1999), *Representation and Change. A Metarepresentational Framework for the Foundations of Physical and Cognitive Science,* Web edition, URL: http://pcp.vub.ac.be/books/Rep&Change.pdf. First published in printed version in 1990.

Iwasaki,Y. and Simon, H.A. (1986), *Causality in Device Behaviour,* Artificial Intelligence 29, 3-32, Amsterdam: Elsevier Science Publishers.

Kaplan, R. (2000), *The Nothing That is: A Natural History of Zero,* London: Penguin Books Ltd. First published by Allen Lane The Penguin Press in 1999.

Klir, G. J. and Folger, T. A. (1988), *Fuzzy Sets, Uncertainty, And Information,* New Jersey: Prentice Hall Inc.

Laudon, K. C. and Laudon, J. P. (2000), *Managing Knowledge,* Management Information Systems: Organization And Technology In The Networked Enterprise, sixth edition, 432-465, New Jersey: Prentice Hall Inc.

Lavine, S. (1998), *Understanding the Infinite,* paperback edition, Massachusetts: Harvard University Press. First published in 1994.

Levine, R. I., Drang, D. E., and Edelson, B. (1991), *Forward Chaining,* AI and Expert Systems: A Comprehensive Guide, C Lanuage, second edition, 33-57, Singapore: McGraw-Hill Inc.

Levine, R. I., Drang, D. E., and Edelson, B. (1991a), *Backward Chaining,* AI and Expert Systems: A Comprehensive Guide, C Lanuage, second edition, 58-91, Singapore: McGraw-Hill Inc.

Maani, K. E. and Cavana, R. Y. (2002), *Systems Thinking and Modelling – Understanding Change and Complexity,* New Zealand: Prentice Hall Inc. First published in 2000.

Manktelow, K. (1999), *Reasoning and Thinking.* East Sussex: Psychology Press Ltd.

Mcleod, JR. (1995), *Expert Systems,* Management Information Systems, sixth edition, 459-497, New Jersey: Prentice Hall.

Nita, C. (1999), *Action Research: A Pathway to Action, Knowledge and Learning,* Melbourne: RMIT University Press.

Nolt, J., Rohatyn, D. and Varzi, A. (1998), *Schaum's Outline Of Theory And Problems Of Logic,* second edition. New York: McGraw-Hill. First published in 1988.

Northrop, E. P. (1944), *Riddles in Mathematics,* Reading: Penguin Books Ltd. Reprinted with revisions 1961.

Parkin, A. J. (2000), *Essential Cognitive Psychology,* East Sussex: Psychology Press Ltd.

Paulos, J. A. (1991), *Gödel and his theorem,* Beyond Numeracy: An Uncommon Dictionary of Mathematics, 95-97. London: Penguin Books Ltd.

Reber, A. S. (1995), *Dictionary Of Psychology,* second edition, London: Penguin Books Ltd. First published in 1985.

Richards, T. (1989), *"Clausal Form Logic: An Introduction to the logic of computer reasoning"*, England: Addison-Wesley Publishing Company, Inc

Ross, T. J. (1995), *Classical Logic and Fuzzy Logic,* Fuzzy Logic with Engineering Applications, 183-231. New York: McGraw-Hill Inc.

Scheid, F. (1988), *Schaum's Outline Of Theory And Problems Of Numerical Analysis,* second edition. New York: McGraw-Hill. First published in 1968.

Sokolowski, R. (2000), *Introduction to Phenomenology,* Cambridge: Cambridge University Press.

Strawson, P. F. (1992), *Analysis and Metaphysics: An Introduction to Philosophy,* Oxford: Oxford University Press .

Tay, B.H. (2003), 'Using Action Research to Develop a Social Technical Expert System for an Industrial Environment'. PhD Dissertation, Graduate College of Management, Southern Cross University, Lismore: Australia.

Turban, E. and Aronson, J. E. (2001), *Knowledge-based Decision Support: Artificial Intelligence and Expert Systems,* Decision Support Systems and Intelligent Systems, sixth edition, 396-436, New Jersey: Prentice Hall Inc.

Widman, L. E. and Loparo, K. A. (1989), *Artificial Intelligence, Simulation, and Modeling: A Critical Survey*, in Artificial Intelligence, Simulation, And Modeling, 1-44, eds. Widman, L. E., Loparo, K. A. and Nielsen, N. R., Chichester: John Wiley & Sons.

Zikmund, W. G., Middlemist, R. D. and Middlemist, M. R. (1995) *Business: The American Challenge for Global Competitiveness,* Illinois: Austen Press.

Zuber-Skerritt, O. (2001) *Action Learning and Action Research: Paradigm, Praxis and Programs,* in Effective Change Management Using Action Learning and Action Research, 1-20, eds. Sankaran, S., Dick, B., Passfield, R. and Swepson, P., Lismore: Southern Cross University Press.