Graphic reasoning

by Antonin Wild, CSc, P.Eng.

Logic trees help the engineer to see each branch of the problem without losing view of the whole.

The true problems are often not what they seem to be. Mistaking a symptom for the real problem or inefficient communication among various specialists is common and leads to financial and other losses. Use of Logic Trees may help.

A Logic Tree is a tool for defining problems. It is especially useful in a multidisciplinary environment, which involves not only professionals and technologists from different fields of engineering, but also various scientists and lawyers. Communication within a multidisciplinary team is difficult. As an example, both an engineer and a psychologist understand reliability as the ability to perform as expected. However, to an engineer the expectation implies a success without failures, while to a psychologist it just means predictability of the outcome. Also, a failure may be understood either as not meeting the specifications, or as not meeting the expectations of the customer, maybe under conditions the designer did not envisage. Similarly, complications can arise in the differences in reasoning by engineers and by lawyers (see "Engage at earlier stage of development to cross law-engineering divide: telecom law professor," *Engineering Dimensions*, July/August 2001, p. 10).

The logic tool makes reasoning clear, simplifies communication, and defines problems. It can help overcome the lack of education in formal logic. Logic trees are also effective tools for bridging the gap between applied science and computer science, in such areas as integral knowledge for a software engineer, for example.

Assisting understanding

Logic Trees (Success Trees and Fault Trees) present in a graphic form the various factors contributing to a "top event," which may be a goal to be achieved, or some undesirable event or situation that should be avoided. The graphic form assists in a correct understanding of the relationship of the factors, and it increases the chance that all relevant factors have been identified. Logic Trees thus soothe semantic arguments when an agreement actually exists, and decrease the risk of reaching an agreement that may have opposing interpretations.

The way in which logic trees can also help to overcome a lack of education in formal logic can be demonstrated with the example of a driver who invites a friend to an outing in the driver's jalopy. When the friend expresses some doubts about reaching the destination, the driver dismisses the concern, because the gas tank's just been filled up.

The driver's confidence is based on the following argument: If my car is out of gas; it will stop running. My car is not out of gas, therefore, it will not stop running.

If the driver were aware of the rules of formal logic, this argument would be identified as a "mixed hypothetical syllogism" that "denies the antecedent" and would realize that the reasoning is wrong.

However, the driver can easily avoid the erroneous conclusion by visually representing the situation in the form of a Fault Tree.

A Fault Tree is a Logic Tree, where the "top event" identifies an event or situation to be avoided.

In the Fault Tree on Figure 1 the undesirable event is "the car will stop." The possible causes include the lack of gas, which the driver of the jalopy has considered. By visualizing the problem in the form of the Fault Tree, however, it's likely the driver will think of the other possible cause: a mechanical failure of the jalopy. The driver will also probably recognize the possibility of other reasons that might achieve the goal.

The top event is linked to its contributing factor by a "gate." Gates may be represented by graphic symbols, or by letters. The letter R identifies the gate as an "or" gate, which means that any of the contributing factors (either alone or in a combination) can cause the top event to occur. Several other types of gates exist, e.g. the letter "A" marks an "and" gate, where all the contributing factors have to occur simultaneously for the top event to occur.

Each of the contributing factors can be further developed by defining its contributing factors, as in the case of the top event. In this example, they were not developed and are considered "primary events" that may be divided into several groups, each identified by a "terminating symbol" that again, may be represented by a graphic symbol, or by a letter. The letter "O" marks a "basic event" that doesn't need to be developed further. The letter "D" marks an event that might be developed as the need arises. The letter "E" marks an event that is analyzed elsewhere.

Defining the "top event" requires an answer to the question: "What are we after?" Asking this question is often an important step in the proper definition of the problem at hand, as it helps to focus the attention on the real problem, rather than merely on some of its symptoms.

Figure 1

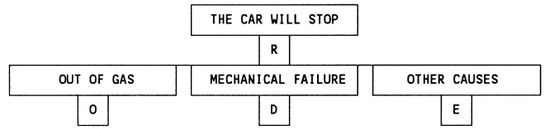
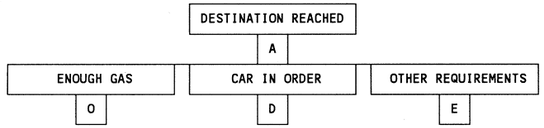


Figure 2



SYMBOL		TYPE	CONDITION
	R	OR	at least one input event must occur
	A	AND	all inputs must occur together
	W	WHILE	input must occur while the condition exists
	m	m-out-of-n	at least m events must occur (m is a digit)
	N	NOT	input must not occur
	S	Summation	sum of weights of inputs occurring reached 100%
	X	X-OR	one input only
	P	Priority	input events must occur in a defined sequence
	G	General	undefined

Gate symbols

SYMBOL		DESCRIPTION
	O	Basic event
	D	Undeveloped event (may be further developed later)
	E	Event developed elsewhere (usually on another tree)
	K	User defined (usually as an external event)
	H	"House", certain event, event which has happened
	Z	Impossible event

Terminating symbols

Presentation of the factors in a graphic form assists in a correct understanding of the factors and their relationship. If necessary, the meaning of any of the factors (e.g. "mechanical failure") may be clarified by identification of its causes, perhaps by development of a whole new branch of the tree. The insistence on a systematic identification of the immediate causes, followed by the exploration of their causes in the next step, helps to recognize factors that might otherwise be overlooked.

The graphic presentation also assists in comparing some of the deductive reasoning (identification of the causes for a given event or condition) with experience, which is primarily inductive (identification of the consequences of an event or condition).

The Fault Tree is the preferred tool of risk analysts, who try to look for problems. Managers and design engineers are, in general, positive thinkers who prefer to define factors leading to a success. Figure 2 shows the "Success Tree" for "the destination reached." The "Fault Tree" and the "Success Tree" are complementary. Their comparison often leads to identification of factors that could be omitted by considering only the one side of the problem (success or failure). Sometimes comparing the complementary (dual) trees even changes the definition of the top event, with a possible subsequent change in the scope of the analysis.

Figure 3

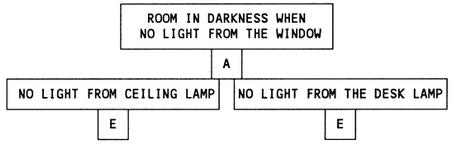


Figure 4

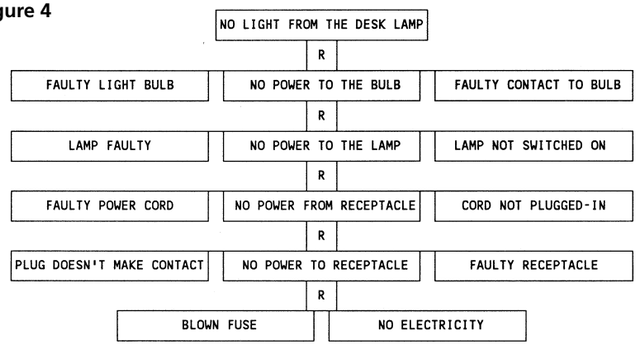


Figure 5

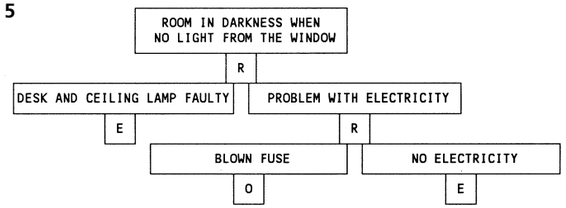
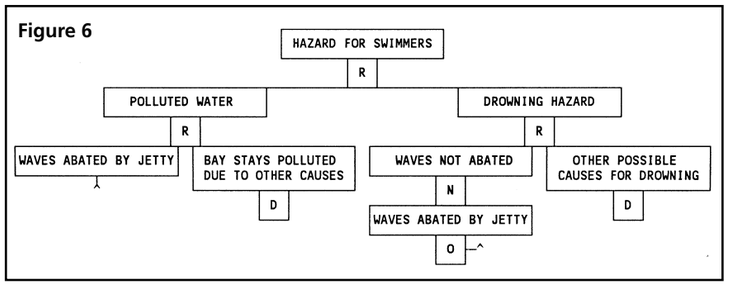


Figure 6



Partitioning of problems

A proper definition of problems usually needs identification of a large number of factors, and may require input from various specialists, who may use different terminology and may have other differences. A Logic Tree helps by partitioning the problem in parts that can be handled separately, without losing the view of the whole.

For example: A person is afraid of a "room in darkness when no light from the window," and so ensures that light is always available by having a desk lamp as a back up for the ceiling lamp. The simple Fault Tree on Figure 3 was used to demonstrate that the availability of light is guaranteed. The top gate is an "and" gate (marked A), indicating that both contributing factors (events) have to occur simultaneously for the top event to occur. The chance of that occurring may seem slim.

However, because the simple Fault Tree doesn't present the whole story, a separate tree shown on Figure 4 was developed, perhaps by a specialist. The Fault Tree for the top event "no light from the desk lamp" identifies systematically the various factors (events or conditions) that might cause the desk lamp not to provide the light when needed.

A similar Fault Tree for the top event "no light from the ceiling lamp" was also developed. It may have been produced by another specialist, working independently, but following the same rules, to ensure that if the same event (e.g. "no electricity") occurs in both trees, both occurrences have the same identification.

Combining all three Fault Trees reveals that the back-up function of the second lamp has been degraded by a dependence on electricity and a fuse, which are common to the ceiling lamp and the receptacle for the desk lamp. The finding may be presented in the form of the simplified restructured tree on Figure 5, which shows that a proper alternative to the ceiling lamp would not depend on electricity. (This could have been obvious, but "jumping to conclusion" leads to missing some factors.)

Dealing with contradictions

A proper definition of a problem is particularly difficult if the problem involves inherent contradictions. In those cases, a well-meaning action to resolve one problem may in turn create another problem. An effective action requires a thorough identification of the contradictions.

However, the presence of contradictions is often not obvious, because a part of the problem may have been mistakenly presented as the whole problem. Or perhaps the factors involved interact in a complex way so that their impact is not easily traced. It helps to use a Logic Tree (a Fault Tree or a Success Tree) to depict the problem, including all the possible factors.

A simplified example of a Logic Tree for a real problem could be the following: A sandy beach in the bay is a great place to swim, except when the wind blows from a certain direction. In this case, high waves create a risk of drowning. A jetty is therefore built to shelter the bay. It controls the waves, but creates a health hazard, because the standing water accumulates pathogens and other contaminants. Should the jetty be removed?

The simplified Fault Tree on Figure 6 can depict the quandary of the city councillors. As in the example of Figure 1, the letter R represents an "or" gate. The letter N represents a "not" gate. The event "Waves abated by jetty" is a replicated event, occurring in the tree as an input to several (in this case two) events, although only one occurrence is marked with a terminating symbol (or a gate symbol, if the event is developed further). The arrow leading to the other occurrence indicates that the event is defined elsewhere in the tree.

A clear identification of replicated events is important, since they may degrade a redundancy built into a system to improve its reliability (as in the example with two lamps), or may create a connection with a "not" gate, which is a "no-win" situation due to inherent contradiction.

If the Logic Tree reveals a "no-win" situation, it may be necessary to review whether the problem was properly described. In the example of the bay, the pollution may not be a hazard for swimmers. However, in many cases the solution requires weighting the different factors and basing the decision on probabilities.

Strengths and weaknesses

Logic Trees can significantly cut costs and potential losses by helping to define clearly:

- what we are after, and what are the major factors to be considered;
- how a problem can be partitioned into manageable parts without losing the view of the whole; and
- what the contributing factors are and their interactions.

Logic Trees are particularly useful for problems that require input from various specialists who may use different terminology and who may have contradictory views on what constitutes proper practices.

Logic Trees for complex systems may be quite large, but suitable computer programs for handling them are on the market.

Logic Trees are also powerful tools for evaluating probabilities—but only within the limits of their applicability. Logic Trees cannot directly calculate probability of an event when the occurrence of the event depends not only on what happened, but also on the sequence in which it happened. The simplicity of many "number-crunching" computer programs on the market is deceiving, and numerical evaluation of probabilities should not be attempted without proper training. ◆

Antonin Wild, CSc, P.Eng., is an expert on techniques for analyzing reliability and technological risk. He is based in Ottawa and can be reached at www.treemaster.ca.