Development of an algorithm for automated cause-consequence diagram construction.

This item was submitted to Loughborough University's Institutional Repository by the/an author.


Additional Information:

- This is a conference paper.

Metadata Record: [https://dspace.lboro.ac.uk/2134/3643](https://dspace.lboro.ac.uk/2134/3643)

Publisher: © Loughborough University

Please cite the published version.
This item was submitted to Loughborough’s Institutional Repository by the author and is made available under the following Creative Commons Licence conditions.

![Creative Commons License](https://creativecommons.org/licenses/by-nc-nd/2.5/)

**Attribution-NonCommercial-NoDerivs 2.5**

You are free:

- to copy, distribute, display, and perform the work

**Under the following conditions:**

### Attribution
You must attribute the work in the manner specified by the author or licensor.

### Noncommercial
You may not use this work for commercial purposes.

### No Derivative Works
You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license).

[Disclaimer](https://creativecommons.org/licenses/by-nc-nd/2.5/)

For the full text of this licence, please go to:

[http://creativecommons.org/licenses/by-nc-nd/2.5/](http://creativecommons.org/licenses/by-nc-nd/2.5/)
Development of an Algorithm for Automated Cause-Consequence Diagram Construction

Akvilina Valaityte, Sarah Dunnett and John Andrews
Aeronautical and Automotive Engineering, Loughborough University,
Loughborough, UK

Abstract

Cause-consequence analysis is one of the best tools available for a comprehensive reliability study. The cause-consequence diagram (CCD) method, like fault tree analysis, represents the failure logic of the system, but in addition the CCD also identifies the complete set of consequences following a given initiating event. While there are well-developed commercialized software packages for fault tree evaluation and construction, no satisfactory methodology has been published for automated cause-consequence chart synthesis.

Hence this paper outlines the development of an algorithm for automated cause-consequence diagram construction. The algorithm builds on methods developed previously for fault tree construction, such as topology diagrams, describing how components are linked together in a system, and component decision tables which model component behaviour. Using this information rules have been developed which enable the construction of the CCD. Once constructed the diagram can be quantified to give exact system reliability. To demonstrate the construction the algorithm is applied to a simple example.

Introduction

In his study, Nielsen [1] developed the cause-consequence diagram as a graphical tool for the analysis and description of relevant accidents in complex process plants. The method is based on a combination of standard reliability techniques. The cause diagram is the conventional fault tree used to describe all causes of an undesired event. The consequence diagram is an event-sequential diagram (decision-tree diagram) describing the alternative failure sequences that an abnormal event leads to if one or more of the accident preventing/limiting provisions fail [1]. By using a combination of the reliability methods, the logical connections between independent accident causes and accident consequences can be established [2]. The main symbol in a CCD is a decision box which contains a component/subsystem condition. It is an identical representation of the YES/NO branches in an event tree. Following the YES/NO branches of the decision box the diagram is developed until it terminates in consequence boxes. As an example of the method consider a simple lamp circuit consisting of battery, switch and lamp. When the switch is closed power is supplied to the lamp which then lights. The CCD for the system is shown in figure 1. The first decision box condition “Switch contacts close” is considered. If the condition is not satisfied then the consequence is that there is no light (NL). If the condition is satisfied then the condition “Current reaches lamp and it lights” is considered. Once again if this is not satisfied then the consequence is NL, however if it is satisfied the consequence is light (L). As all paths have terminated in consequence the diagram is complete. The causes of the switch contacts not closing and current not reaching the lamp are developed using fault trees as shown by the arrows to the first and second
decision boxes respectively shown in figure 1, where SW, BATT and LAMP are the basic events switch, battery and lamp failure, respectively.

Figure 1. Simple cause-consequence diagram

The CCD method has advantageous features in both its representation of a systems failure logic and its subsequent quantification. However traditional cause-consequence analysis is based on manual construction of the diagram. This requires a detailed knowledge of the system, plus experience and practice in understanding the failure modes of the components within the system and their effects. The construction is time-consuming and expensive, moreover, it can be a source of human errors. A faster and error free analysis can be performed if the CCD is automatically generated by computer from the system description [2]. Past work on automating reliability techniques has concentrated on fault tree analysis. As a result, a variety of methods for computerized fault tree synthesis have been developed and published. The most successful are based on the diagraph method [3] and the decision table method [4]. The features of these two methods were combined [5] in a new method for automating fault tree construction appropriate to modelling two-state and continuously variable safety systems.

In this paper previous methods applied to fault tree construction are extended to automate the construction of the CCD.

Algorithm for the automated construction and analysis of CCD

In the following sections an algorithm will be described that accepts a description of the system under construction and generates a CCD which can be quantified in a straightforward manner.
The automatic generation is based on the following steps:
   i) Initially information about the system required by the algorithm is given;
   ii) The CCD is then automatically constructed by applying the developed algorithm;
   iii) The CCD is reduced to its most efficient form by excluding irrelevant decision boxes;
   iv) Fault trees describing the failure of subsystems are constructed by employing the decision table method [4, 5];
   v) The cause-consequence diagram is then used to analyse the system considered.

\textit{i) Algorithm requirements}

In order for the construction algorithm to be implemented detailed information about the system being considered must be given. This information is broken down into: component models, system topology diagram, and the failure rate data for the components.

The component models used in this work are in the form of decision tables [4]. For each component its state is combined with all possible inputs from the components with which it is linked to give outputs which are passed on to other components within the system. The construction of such tables and their use in other automated procedures is described in detail by Salem et al. [4].

The system topology diagram details how the components inputs and outputs are linked together within the system.

The failure rate data includes a description of the failure modes of component as well as their rate of failure and is used in the construction of fault trees and their quantification, and hence the final CCD quantification. For some systems it is also necessary to specify initial states for the components.

The algorithm also requires that an initiating event is identified. At the current stage of the algorithm development this event is given by the user. It depends upon the system under construction and is generally a component, with associated function, which initiates the system. This is the first event considered in the algorithm.

In order to ensure that the CCD construction process completes and the algorithm does not continually consider components that are linked together, stopping criteria must also be identified. These indicate at which point the algorithm has reached a consequence and hence that path is completed.

\textit{ii) Construction algorithm}

A set of rules are applied to construct the consequence diagram. These rules have been developed here but in order to present the work in a concise manner they have not been listed. In summary, initially the order in which the components are considered is determined by use of the topology diagram. If the system contains circuits then these are used to determine the order. A circuit is a path containing a power supply which starts and ends at the same component and with all components passing current. Starting form the initiating event the functionality of each component
or subsystem is investigated and, by use of the decision tables and topology diagram, the consequences of these sequences determined. The procedure continues until the stopping criteria is reached.

**iii) Diagram reduction**

If any decision boxes are deemed irrelevant (e.g. the branches attached to the NO and YES branches are identical) then these boxes should be removed and the next decision box or consequence box in the path put in its place. Each decision box should be inspected and if no further redundancies exist the cause-consequence diagram is deemed minimal [6].

**iv) Development of the fault trees**

If a decision box is governed not by a component but by a sub-system then the probability of failure will be obtained via a fault tree [7]. These fault trees are produced automatically using fault tree construction methods developed previously [4, 5].

**v) Analysis**

Having constructed the CCD it can be used to analyse the system under consideration.

A qualitative analysis will produce the list of causes for each outcome condition. These are established by considering each decision box on a path to the particular outcome and listing the components failure or success as indicated by the exit path from the decision box.

A quantitative analysis will produce the probability of each system outcome. As the algorithm ensures that the probabilities of the decision boxes of the CCD are independent these are obtained by simply multiplying the probabilities of the component events in the branch leading to that consequence. Each consequence for a system is then determined by summing the probability of each set of events which lead to this particular outcome, as stated by Nielsen [1].

In order to demonstrate the construction and analysis process described above it is applied to a simple system.

**Application to a simple system**

The cause-consequence diagram construction algorithm is now described in detail in relation to the simple example of a lamp system illustrated in figure 2. The system consists of a switch, relay, two power supplies and a lamp. The switch can be closed or opened manually. When the switch is closed current reaches the relay which should close the contacts, causing current to reach the lamp which would then light. When the switch is open the contacts will open and the lamp will no longer light.

Initially the lamp system is considered to be in a dormant state. Therefore the switch and the contacts are open.
Automated CCD construction

Following the automated algorithm described the following steps are taken:

i) Algorithm requirements: A topology diagram for the system is constructed, see figure 3.

Where SW is the switch, R is the relay, PS1 and PS2 are power supplies 1 and 2, respectively, LP is the lamp and CN is the contacts (a component linking the relay to the second circuit of the system).

There are five decision tables relevant to the components in the example. These are shown below. As an example of how to construct a decision table for a component consider the switch, see table 1. Two failure modes have been considered for this
component, failed open (FO) and failed closed (FC) as described in table 6. From the
topology diagram, figure 3, it can be seen that SW has two inputs, IN1, – the manual
closing or opening and IN2 the current from PS1. IN1 has two possible states, closed
(CL) and open (OP), and IN2 has current (C) or co current (NC). Table 1 considers all
possible combinations of inputs from IN1 and IN2 and all possible states of SW and
the effects these will have on the output to the relay. The sign “–” in the inputs and
state columns indicates the “don’t matter” condition, which means that the specified
input states will result in the specified output state regardless of the value of the
variable. For example, there are two causes of current in the output, these are:

1. Input 1 is closed (IN1 = CL), current in input 2 (IN2 = C) and switch working
   (W).

2. Current in input 2 (IN2 = C) and the switch failed closed (FC).
The other tables are constructed in a similar manner.

<table>
<thead>
<tr>
<th>IN1</th>
<th>IN2</th>
<th>State</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CL</td>
<td>C</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>OP</td>
<td>–</td>
<td>W</td>
</tr>
<tr>
<td>3</td>
<td>–</td>
<td>NC</td>
<td>–</td>
</tr>
<tr>
<td>4</td>
<td>–</td>
<td>–</td>
<td>FO</td>
</tr>
<tr>
<td>5</td>
<td>–</td>
<td>C</td>
<td>FC</td>
</tr>
</tbody>
</table>

Table 1. Decision table for Switch

<table>
<thead>
<tr>
<th>IN</th>
<th>State</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>–</td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>NC</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 2. Decision table for Power Supply

<table>
<thead>
<tr>
<th>IN</th>
<th>State</th>
<th>OUT1</th>
<th>OUT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
<td>W</td>
<td>CL</td>
</tr>
<tr>
<td>2</td>
<td>NC</td>
<td>W</td>
<td>OP</td>
</tr>
<tr>
<td>3</td>
<td>–</td>
<td>FO</td>
<td>OP</td>
</tr>
<tr>
<td>4</td>
<td>–</td>
<td>FC</td>
<td>CL</td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>6</td>
<td>NC</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 3. Decision table for Relay

<table>
<thead>
<tr>
<th>IN1</th>
<th>IN2</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
<td>CL</td>
</tr>
<tr>
<td>2</td>
<td>–</td>
<td>OP</td>
</tr>
<tr>
<td>3</td>
<td>NC</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 4. Decision table for Contacts
Table 5. Decision table for Lamp

Where $L$ is light and $NL$ is no light, and $F$ is failed.

The input data for the components in the system is shown in table 6. In this example failure rates, necessary for quantification, have not been included as the example has been taken in order to demonstrate the construction process.

<table>
<thead>
<tr>
<th>Component</th>
<th>Failure mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW (Switch)</td>
<td>SW_FO</td>
<td>Switch failed open, operator cannot close it</td>
</tr>
<tr>
<td>PS# (Power supply)</td>
<td>PS#_F</td>
<td>Failed, no power to a circuit. # takes values 1 or 2</td>
</tr>
<tr>
<td>R (Relay)</td>
<td>R_FC</td>
<td>Relay contacts fails closed</td>
</tr>
<tr>
<td>LP (Lamp)</td>
<td>LP_F</td>
<td>Failed, no light</td>
</tr>
</tbody>
</table>

Table 6. Failure data for the components

The initial conditions, are that the switch and the relay contacts are open, i.e.

SW: OUT = NC;
R: OUT1 = OP.

In order to start the algorithm the initiating component and its function must be identified. The system considered is initiated when the switch closes. Hence the initiating component is SW with function $CL$. In the notation adopted here this is written as:

SW: IN1 => CL,
which states that input 1 of switch closes.

The stopping criteria in this case are the two possible consequences, lamp lights or lamp doesn’t light, i.e. the diagram path terminates in one of these consequences:

LP: (OUT1 => L $\cup$ OUT1 => NL).

ii) Construction algorithm: The algorithm now has all the information it requires to generate the CCD. Initially it decides the order in which the components are to be considered. Starting from the initiating event components are traced through the topology diagram and any circuits identified. If the output of a component within a circuit is also connected to another component which is contained within a different
circuit, then, these two circuits must be considered in turn when constructing the cause-consequence diagram.

In the example it can be seen that there are two circuits: \{PS1, SW, R, PS1\} and \{PS2, CN, LP, PS2\}, see figure 4. As the circuits contain different components they are treated individually.

![Figure 4. Circuits of lamp system](image)

Starting from the initiating component, switch, the CCD is constructed. Since the initiating event is switch closed, the Switch decision table, table 1, with entry CL in column IN1 is considered. Row 1 satisfies the criteria and has output C. Hence, as component SW is traced within circuit 1 the first decision box is created with the question related to the output OUT, see figure 5: “Is C in Circuit1?” i.e. does the switch close and power supply 1 work? The comments in brackets in the decision box in the figure are added for clarity.

From the rules developed the YES branches of the decision boxes are traced first until the consequence box is reached. Hence the YES branch of decision box 1 is traced (Figure 5). The topology diagram is checked to determine if any components within circuit 1 have outputs that are external and therefore connect to components within other circuits. Only one external output is found – OUT1 from component R, and hence this is considered next.

Following the YES branch of box 1 there is current C in OUT of SW and hence, as OUT of SW is connected to IN of R, see figure 3, the input, IN, of relay R is current C. Hence all rows in the Relay decision table 3 with C in IN are considered. “Don’t matter” states are also included since in this case the state of the input could be C. There are 4 such rows (1st, 3rd, 4th and 5th). However in the 4th row the relay is in a failed closed state, which contradicts the initial assumption that the relay is open, thus only 3 rows describe the given situation. The external output of R, OUT1, column 3 in table 3, must now be considered. This determines the next decision box, box 2 (Figure 5). Considering the rows in turn row 1 is considered first and in this case OUT1 of relay R is closed CL, hence the decision box contains the question “Is R: OUT1 = CL?”
The algorithm proceeds by tracing the component connected to the external output of relay. From the topology diagram, figure 3, it can be seen that output $OUT1$ of R connects to input $IN2$ of contacts CN. Therefore component CN is now considered. Following the YES branch of decision box 2 $IN2$ of CN is $CL$ and hence the rows of the decision table for contacts, table 4, with $IN2 = CL$ are considered. There are 2 such rows, 1 and 3 and considering row 1 first results in $C$ in $OUT$. As this component is contained within the second circuit a new decision box 3 is introduced (Figure 5), which asks the question “Is $C$ in Circuit2?”

Following the YES branch results in current in circuit 2. Components with external output within circuit 2 are then identified. There is one such component, lamp. Two rows in the lamp decision table (table 5) satisfy the condition that there is current in circuit 2, these are rows 1 and 2. Considering row 1 decision box 4 is created which questions the value of the external output of lamp, $OUT1$, which in this row is $L$, i.e.:

“Is $LP: OUT1 => L$?”

Following the YES branch of decision box 4 results in light being output from the lamp. Therefore the stopping criteria for this branch is reached and a consequence box has to be created. Hence the consequence box “$LP: OUT1 = L$” is added, see figure 5, where the complete branch is shown.

When all YES outputs of the branch are traced, the algorithm proceeds by returning to the NO outputs of the last decision box, which in this case is box 4. Following the NO branch of decision box 4, all possible values of $OUT1$ in table 5 except for $L$ must be considered. There is only one other value, $NL$. From table 5, row 2 is the only row satisfying the conditions, as from decision box 3 there is $C$ in circuit 2. Hence the stopping criteria for the second branch is reached $LP: OUT1 = NL$ and a consequence box is created (Figure 5).

From the algorithm the NO branch of decision box 3 is now considered which results in no current in circuit 2. Considering the component in circuit 2 with external output, LP, results in the external output, $OUT1$, being $NL$. As from the decision table for lamp, table 5, rows 2 and 3 satisfy $NC$ in $IN$ and both result in $NL$ in $OUT1$. Hence the consequence box “$LP: OUT1 = NL$” is added.

A similar situation arises following the NO branch of decision box 2. If the relay doesn’t close the contacts, from table 4, it is apparent that no current circulates in circuit 2 and therefore the lamp wouldn’t light.

Moving up to the NO branch of decision box 1 no current is detected in circuit 1. The algorithm detects that either the switch doesn’t close or the power supply is failed, rows 3 and 4 in table 1, and rows 2 and 3 in table 2, respectively. This results in $NC$ in $IN$ of relay and hence from table 3 the contacts remain open leading to $NL$ from $OUT1$ of lamp.

Having returned to decision box 1 with all branches terminating in a consequence box the CCD is complete.

iii) Diagram reduction: The cause-consequence diagram presented in figure 5 cannot be reduced as no redundant decision boxes are identified.
iv) Development of the fault trees: Fault trees are now developed for each decision box starting with the first one. The fault trees: \(F_{t1}, F_{t2}, F_{t3}\) and \(F_{t4}\), are constructed according to the decision table method and are shown in figure 6.

v) Analysis: The final cause-consequence diagram for a simple lamp system, figure 2, is presented in figure 5 with the corresponding fault trees in figure 6 which are shown in reduced form.

![Figure 5. Cause-consequence diagram for a lamp system](image-url)
Having obtained the diagram it can now be analysed and quantified in a straightforward manner.

**Qualitative analysis**

For the example considered the failure event is *no light* from the lamp, i.e. the consequence “LP: OUT1 = NL”. In the diagram there are 4 consequence boxes with this outcome. To obtain the minimal cut sets the failure events leading to these boxes must be traced. For example for the consequence box on the NO branch of decision box 1 the component failures leading to this are given from Ft1, i.e. PS1_F or SW_FO. Considering the other consequence boxes in the same manner leads to the complete list of minimal cut sets:

\{PS1_F\}, \{SW_FO\}, \{R_FO\}, \{PS2_F\}, \{LP_F\}.

**Quantitative Analysis**

Considering the paths of the CCD leading to NL the probability of the system failing and producing *no light* can be obtained. The four paths leading to the consequence “LP: OUT1 = NL” are:

1) no current in circuit 1;
2) current in circuit 1, but the relay doesn’t close the contacts;
3) current in circuit 1 and the relay closes the contacts, but power supply 2 doesn’t work;
4) current in circuit 1, the relay closes the contacts, there is current in circuit 2, but there is no light from the lamp.

The probability of NL is obtained by adding the probability of each path. Then

\[
\begin{align*}
P(\text{No Light}) &= P(\text{Path 1}) + P(\text{Path 2}) + P(\text{Path 3}) + P(\text{Path 4}) = \\
&= q_{F1} + (1-q_{F1})q_{F2} + (1-q_{F1})(1-q_{F2})q_{F3} + (1-q_{F1})(1-q_{F2})(1-q_{F3})q_{F4} = \\
&= q_{SW\_FO} + q_{PS1} - q_{SW\_FO}q_{PS1} + (1-q_{SW\_FO} - q_{PS1} + q_{SW\_FO}q_{PS1})q_{R\_FO} + \\
&+ (1-q_{SW\_FO} - q_{PS1} + q_{SW\_FO}q_{PS1})(1-q_{R\_FO})q_{PS2} + \\
&+ (1-q_{SW\_FO} - q_{PS1} + q_{SW\_FO}q_{PS1})(1-q_{R\_FO})(1-q_{PS2})q_{LP}
\end{align*}
\]
where
\[
q_{SW\_FO} \quad \text{is the probability of switch (SW) failure to close (fails open)},
\]
\[
q_{PS1} \quad \text{is the probability of power supply 1 (PS1) failure},
\]
\[
q_{R\_FO} \quad \text{is the probability of relay (R) failure to close the contacts (fails open)},
\]
\[
q_{PS2} \quad \text{is the probability of power supply 2 (PS2) failure and}
\]
\[
q_{LP} \quad \text{is the probability of lamp (LP) failure.}
\]

This is the same results as would be obtained by taking the four minimal cut sets listed earlier and using the inclusion-exclusion expansion [8].

**Conclusions**

An algorithm to automatically construct cause-consequence diagrams is developed in this paper. From an initial description of a system the algorithm develops the CCD in a logical manner. Such an automatic construction saves an analyst from a laborious and time consuming task and ensures that all possible system behaviour is modelled. The diagram generated can be quantified simply to give exact probabilities of all system outcomes considered. The algorithm employs topology diagrams and decision tables, methods developed previously for fault tree construction.

In order to demonstrate the algorithm a very simple example has been considered here, however the algorithm has been tested with several substantial industrial examples.

**References**