Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Electronic Notes in Theoretical Computer Science 134 (2005) 19–31
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Failure Mode Modular De-Composition Using Spider Diagrams

R.P. Clark[1]

Research and Embedded Development Department
Energy Technology Control Ltd.
25 North Street, Lewes, East Sussex, England

Abstract

This paper concentrates on a possible application for spider diagrams; using them to simplify the task of identifying the effect of individual component failures, leading to system failures in safety critical software or hardware designs. http://www.energytechnologycontrol.com/

Keywords: Failsafe, EN298, gas-safety, burner, control, fault, double-fault, single-fault, fault-tolerance.

1 Introduction

This paper identifies a possible application for spider diagrams, and in particular that :

- Spider Diagrams can be used to model component operational and failure modes
- A diagram can be derived from a spider diagram, where each spider is represented by a set in the new diagram.
- These derived spider diagrams can be combined with other derived diagrams, thus nesting spider diagrams.

[1] Taken and modified from MSc Dissertation Submitted 2003 NOV Brighton University
Email: robin@energytechnologycontrol.com

doi:10.1016/j.entcs.2005.02.018

- This process should dramatically reduce the number of checks to perform in safety critical analysis.
- Spider diagrams can include a default spider, being that which is is not specifed explicitly (a kind of catch all undefined cases spider).

Note this paper does not concern itself with the reliability of the system (i.e. mean time between failure etc) but with possible resultant states of the system due to component failures. The safety of potentially dangerous or explosive industrial plant is far more important than its reliability. The overriding philosophy here is that a system should be able to detect that it has become faulty, and revert to a safe operational mode. Human intervention can then assess and repair faults.

It has been long known that components of systems and operational modes can be modelled with Z [1] : thus Spider diagrams can be used to model components and their operational and/or failure modes.

2 Background - Safety Critical Modelling

2.1 Components

Currently safety critical control systems are built from components with known failure modes. For instance, resistors can be purchased that only fail by going open circuit.

Electrolytic Capacitors, however, can fail by both shorting and going open circuit.

2.2 A Typical Safety Measure

To give an idea of measures already in place in safety critical systems such as industrial gas burner controllers, consider the independent watchdog function, implemented in hardware.

If a burner controller's micro processor (or the software running on it were to fail) the system could obviously become unsafe. So in the event of microprocessor failure, a watchdog system will (with a tight time window) wait for a pulse to indicate all is well every say, 50 ms. The watchdog is normally a second microprocessor. Should no pulse arrive, or a pulse arrive outside the time window, the watchdog processor will shutdown the system using relays to main power independent to relays controlled by the main processor.

Note that the watchdog processor must also have its own clock. This is because if they both ran from the same clock signal, a single failure, the clock/oscillator, could stop the system from shutting down. The main proces-

sor must periodically check the watchdog by providing a false late signal and then cancelling it on seeing that the safety relay has been ordered open.

Note the implied philosophy here. Should one of the two microprocessors fail, the system will revert to a safe state.

2.3 Safety Standards

A feature of safety critical systems specifications is to demand, at the very least that single failures of hardware or software cannot create an unsafe condition in operational plant. Further to this a second fault introduced, must not cause an unsafe state, due to the combation of both faults.

This sounds like an entirely reasonable requirement.

However, to ensure complete coverage, each of the effects of the failure modes must be applied to all the other components. Thus each component must be checked against the failure modes of all other components in the system. Mathematically with components as 'c' and failure modes as 'Fm'.

$$checks = \{\ (Fm, c)\ \mid \hat{c} \neq\ c\} \tag{1}$$

Where demands are made for resilience against two simultaneous failures this effectively squares the number of checks to make.

$$doublechecks = \{\ (Fm_1, Fm_2, c)\ \mid\ c_1\ \neq\ c_2\ \wedge\ Fm_1 \neq Fm_2\ \} \tag{2}$$

If we consider a system which has a total of N failure modes (see equation 1) this would mean checking a maximum of

$$NumberOfChecks = \frac{N(N-1)}{2} \tag{3}$$

for individual component failures and their effects on other components when they fail. For a very small system with say 1000 failure modes this would demand a potential of 500,000 checks for any automated checking process.

European legislation[3] directs that a system must be able to react to two component failures and not go into a dangerous state.

This raises an interesting problem from the point of view of formal modelling. Here we have a binary cross product of all components (see equation 2). This increases the number of checks greatly. Given that the binary cross product is $(N^2 - N)/2$ and has to be checked against the remaining $(N - 2)$ components.

$$NumberOfchecks = \frac{(N^2 - N)(N-2)}{2} \tag{4}$$

Thus for a 1000 failure mode system, roughly a half billion possible checks would be required for the double simultaneous failure scenario. This astonomical number of potential combinations, has made formal analysis of this

type of system, up until now, impractical. Fault simulators [7] are commonly used for the gas certification process. Clearly this massive task needs breaking down.

Modularising the problem so that unnecessary checks can be factored out is desirable!

3 Failure Mode Modular De-Composition

Imagine a complicated system built from components combined to form modules. The modules are combined to form sub-systems and the sub-systems combined to form the final system. If each module/sub-system is modelled and has a finite number of failure modes, the massive number of checks to be considered is reduced dramatically.

For instance the failure of a capacitor a the power supply producing a high ripple noise level does not need to be considered for every other component. Rather it has become another well defined failure mode of the power supply, which must be considered in relation to how that affects other modules. Most modules (say a heating element) would not consider this failure mode as a fault.

This means that each spider diagram will derive a new spider diagram, which contains sets for all the spiders defined in the original diagram. This derived spider diagram can then be used to interact with other modules in the system that depend upon it.

In figure 1 three modules ψ χ *and* ϕ are developed from components. Each of these produce a new spider diagram which simply holds the failure modes of the modules. The modules ψ *and* χ are combined to produce a new spider diagram τ. This produces a new spider diagram representing the fault modes of τ and thus becomes a module/sub-system. The entire system is then defined by combining τ *and* ϕ and the results of that spider diagram create a new spider diagram consisting of the failure modes of the entire system.

The use of predefined euler diagrams, used as templates to produce more complicated euler diagrams is discussed in [4]. The work described here extends this concept to add an additional stage. That of converting a spider diagram into a new diagram where each spider is represented as a set. this new diagram is then taken for inclusion into higher level diagrams.

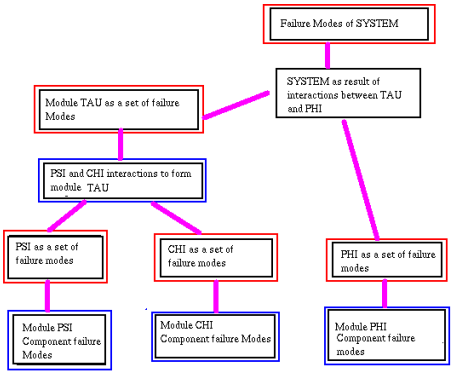

Fig. 1. Formal Modular Dependency De-Composition

4 Using Spider Diagrams to model operational modes of components

Consider a system where basic components have given sets of failure modes.

Each component has a set of contours corresponding to operational/failure modes. A module built from these components includes all these contours, where necessary.

The module is then analysed looking into the effect of all operational/failure modes of components on the module. Spiders are used to describes operational/failure modes of the completed sub-system.

Consider the following theoretical components with sets of operational modes modes thus:

- Type α having operational modes $\{ \alpha_{ok},\ \alpha_A \}$
- Type β having operational modes $\{ \beta_{ok},\ \beta_A,\ \beta_B,\ \beta_C \}$
- Type γ having operational modes $\{ \gamma_{ok},\ \gamma_A,\ \gamma_B,\ \}$

Imagine a system ψ that is built from one of each of the above components, that has a set of operational modes, say A, B, C and D.

$$\psi_{states} = \{ \psi_{ok},\ \psi_A,\ \psi_B,\ \psi_C,\ \psi_D \}$$

These failure modes are due to combinations component failure modes. If all components are "ok" then the sub-system will have a state "ok".

$$\psi_{ok} = \{ \alpha_{ok},\ \wedge \beta_{ok},\ \wedge \gamma_{ok} \}$$

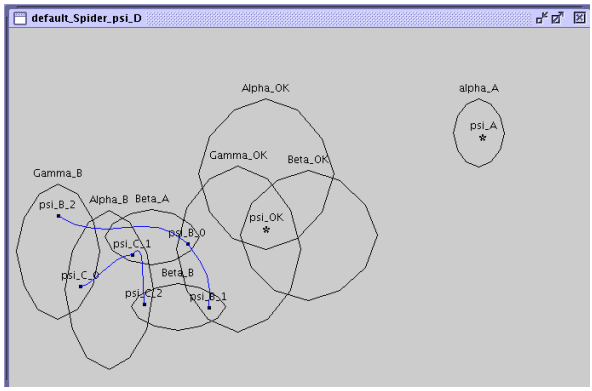

Fig. 2. Component to Module Failure Mode Diagram

Now lets makeup some more rules for these failure modes for this example. Lets say

$$\psi_A \triangleq \alpha_A$$

$$\psi_B \triangleq \alpha_{ok} \wedge (\beta_A \vee \beta_B) \wedge \gamma_B$$

$$\psi_C \triangleq \alpha_B \wedge (\beta_A \vee \beta_B \vee \gamma_B)$$

The definition of ψ_D implies a default state, it could be described as being the default not OK state.

$$\psi_D \triangleq \neg \psi_{ok} \wedge \neg (\psi_A \vee \psi_B \vee \psi_C)$$

This can be more clearly represented by a spider diagram with the addition of a 'default spider', ψ_D. See figure 2. Without the default spider all failure modes would have to be represented on the diagram and a very large spider drawn to represent it.

The facility of having a default spider will help in producing practical models of modules. Imagine a power supply for instance. For most combinations of failure modes the power supply will simply not work, but for some selected combinations of failures it will behave in incorrect but well defined modes.

In tabular form, or as a Karnaugh Map [6] each of the 2^N states would have to be represented; in this case 2^9 entries. The spider diagram with an added 'default spider' is much easier to understand.

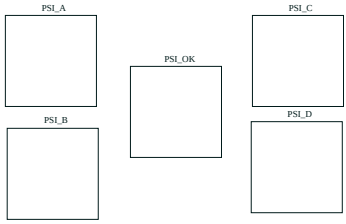

Fig. 3. Component to Module Failure Mode Diagram

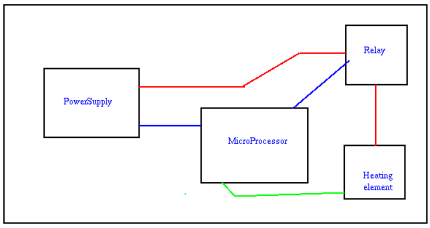

Fig. 4. Simple Heater Control System

5 Hierarchical Ordering of Spider Diagrams

In the example above the spider diagram in figure 3 was derived from figure 2. This diagram could now be combined with other diagrams to form a higher level diagram. A hierarchy of diagrams could thus be produced ending with a representation of an entire system. .

5.1 Simplified Example of a Hierarchical Spider Diagram

In order to run through a example to show modularisation of a safety system, consider a very simple heater controller. It has a power supply that provides AC for the heating element and DC for a micro processor which uses an on board ADC to measure the temperature and a relay to turn on/off current to the heater. See figure 4.

Taking each of these modules in turn (in a simplified and incomplete way).

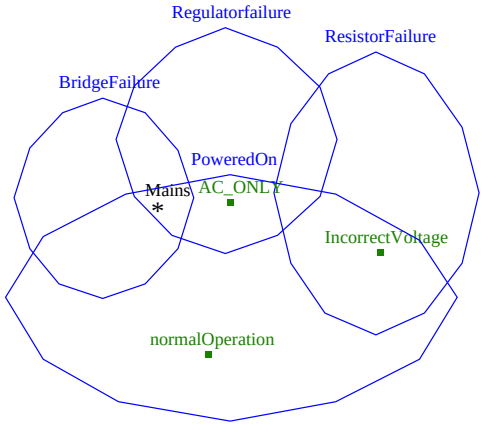

Fig. 5. Power Supply Components

5.1.1 The Power Supply

This simplified power supply can fail in 4 ways.

- Normal Operation
- Supply No Current (a default condition if no other is true)
- Supply AC but no DC to the microprocessor
- Supply DC but no AC
- Supply the wrong voltage (mains) to the entire circuit

These are derived from the spider diagram in figure 5. Note that "No Current" is the default spider for this diagram.

Each of these failure modes can be used as a set to determine the behaviour of the systems which rely on it. A conversion process could be used to create a diagram to model other circuit elements dependent upon it. The diagram in figure 6 shows this after 'conversion'.

The next stage is to take a system which depends upon the power supply. The Microprocessor is the obvious choice. This will function correctly when given DC, and will be damaged by mains voltage. It will not care if the AC voltage (for the heater element via the relay) is present.

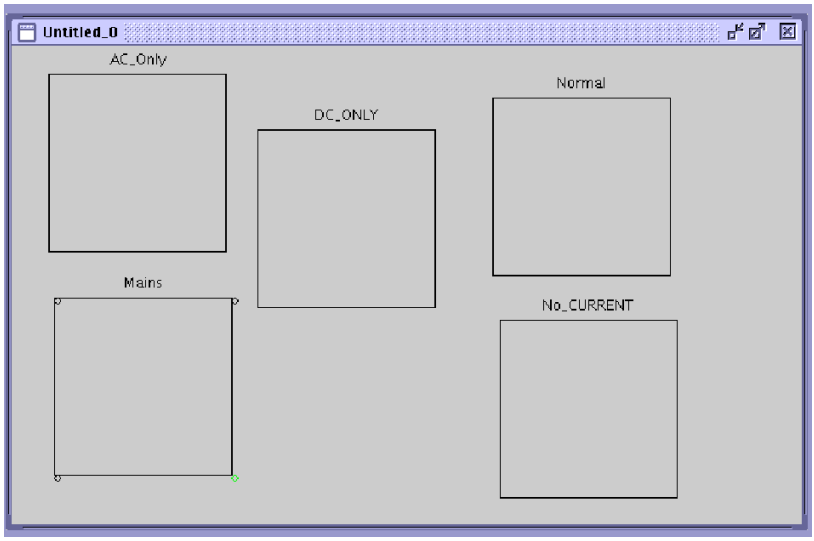

Fig. 6. Power Supply Spiders Converted into Sets for Further Modelling

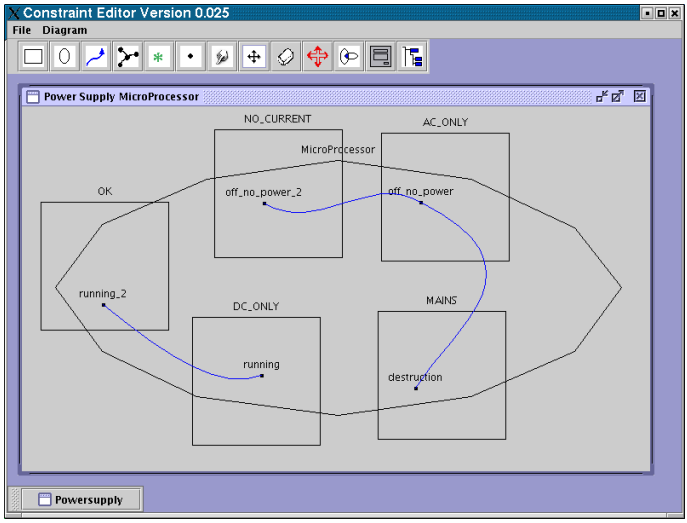

Fig. 7. Power Supply Failure Modes Combined with Microprocessor Operation

The Relay and Heating element depend on both the power supply and the microprocessor, and is dealt with later.

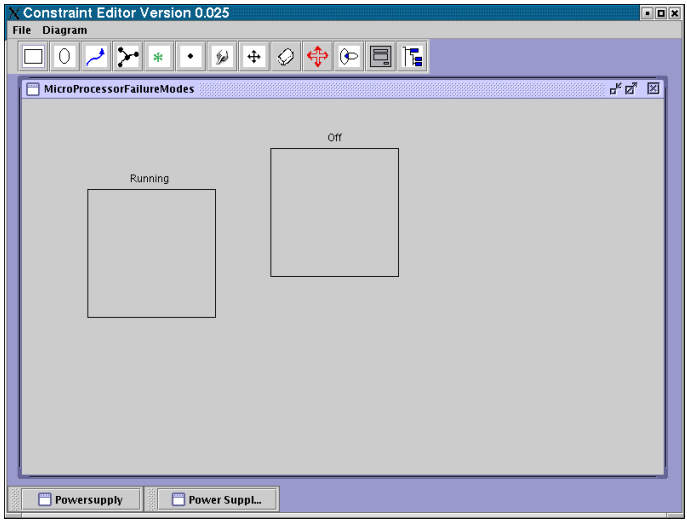

Fig. 8. MicroProcessor Failure Modes

5.1.2 The Micro Processor

This requires stable DC in order to operate. Also internally the ROM, RAM, IO or ADC could fail. However, to simplify this example the internals of the microprocessor will not be dealt with.

Combining the diagrams, gives us two microprocessor modes. OFF, and RUNNING. These spiders (see figure 7) can now be converted into a spider diagram where again each spider is represented as a set. This is displayed in figure 8.

5.1.3 The Relay and Heating Element

The Relay can fail in two modes, it can weld itself ON, or it can fail to respond to a TURN ON signal from the microprocessor IO line. The Heating element can go open circuit and thus stop functioning. The relay depends on both the Microprocessor and the power supply (the A.C. part anyway) to function correctly.

A spider diagram for the relay/heating element combination with a default spider off "OFF" or "NO_HEAT" can thus be constructed. See figure 9.

Thus when the derived spider diagram for the relay and heating element (see figure 10) is combined with the Microprocessor and the Powersupply A.C.

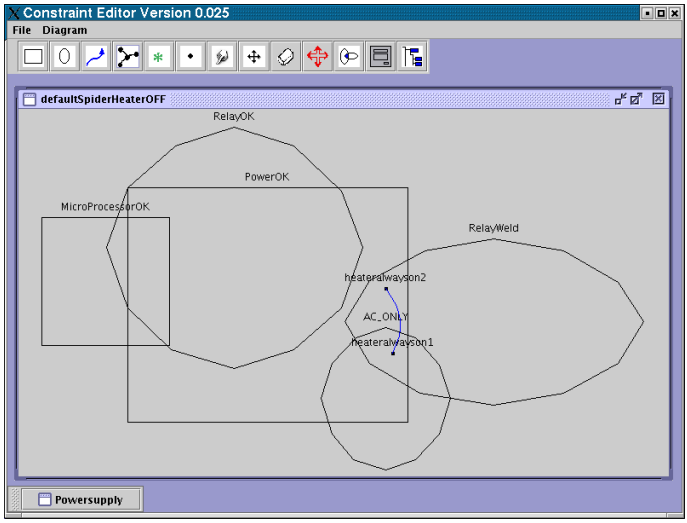

Fig. 9. Spider Diagram combining Power Supply, Micro Processor and relay/heater

Part, we will have a failure modes definition for the entire system.

The fact that HeaterAlwaysOn is one of the states possible from a single failure should be seen as quite alarming ! In Mathematics this fault could be described as

$$HTR\_ALWAYS\_ON \triangleq RelayWeld \wedge (PS\_AC\_ONLY \vee PS\_NORMAL\_OPERATION)$$

The modular process has proved that this combination is possible. How likely it is to occur is another matter and is beyond the scope of this discussion. The methods described here are designed to tackle the requirements for single and double failure of components leading to unsafe conditions, as described in European Legislation for the safety of Gas burner systems. [3].

5.1.4 Combining all Diagrams - Comparing with cross product of all components

This idea represents bottom-up modular decomposition of interacting systems reducing the number of cross product checks required. This spider diagram (9) shows that one single fault [1] and several obvious double faults can cause this system to fail.

[1] Relay Weld, in practice relays are connected in series and controlled by separate fail safe systems

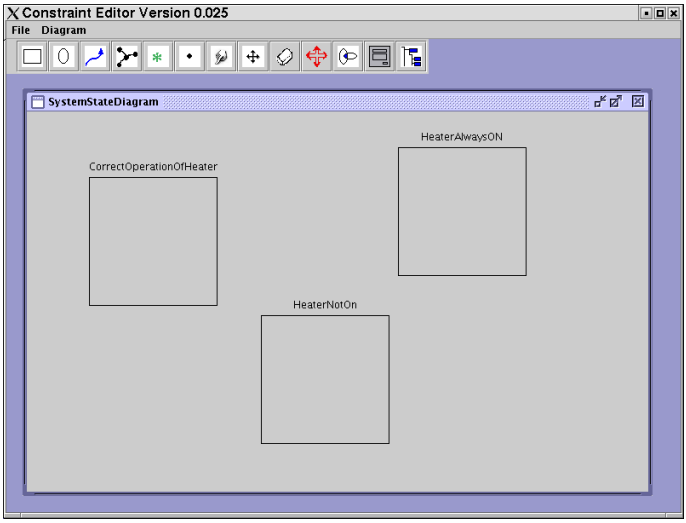

Fig. 10. Derived Diagram representing possible System States

6 Extending Failure Mode Modular De-Composition to Double Simultaneous Component Failure Scenarios

For double failure mode the same bottom up technique can be used but checking for double failures within the components. This will probably normally create some extra failure modes per module (i.e. the binary cross product of failure modes will generally be larger than the single failure modes).

The double failure modules will still be separate modular entities, but each module checked for combinations of two failures, and the resulting hierarchy, using double fail scenarios to progress up to the final system model.

These double failure modules will therefore produce derived spider diagrams, which when combined with other modules, will also undergo double failure mode checking. Obviously this will involve more work and checking than the single failure models, but will not require the astronomical number of checks demanded by a binary cross product of all possible double instances of all component failure modes.

References

[1] D. C. Ince, *An Introduction to Discrete Mathematics, Formal System Specification and Z,* Oxford, ISBN 0-19-853836-7.

[2] J. A. Flower and J. Howse. *Generating Euler Diagrams,* in: *Proceedings of Diagrams 2002, Callaway Gardens, Georgia, April 2002,* Springer Verlag http://www.it.bton.ac.uk/research/vmg/VisualModellingGroup.html

[3] *European Standard For Gas Burner Safety Systems,* European Committee for Standardisation, October 1993.

[4] J. A. Flower, J. Howse and J. Taylor, *Nesting in Euler Diagrams,* http://www.cmis.brighton.ac.uk/Research/vmg/papers/GTVMT02.pdf

[5] J. Howse, G. Stapleton, J. Flower and J, Taylor, *Corresponding regions in Euler diagrams,* http://www.cmis.brighton.ac.uk/Research/vmg/papers/D2K2HSFT.pdf

[6] R. Garnier and J. Taylor, *Discrete Mathematics for New Technology,* pp 436-447, IoP ISBN 0 7503 0135.

[7] F. M. Concalves, M. B. Santos, I. C. Teixeira and J. P. Teixeira, *EDA Tool Development to Support the Design and Certification of Fail-Safe Products,* http://www.inesc-id.pt/pt/indicadores/Ficheiros/1705.pdf