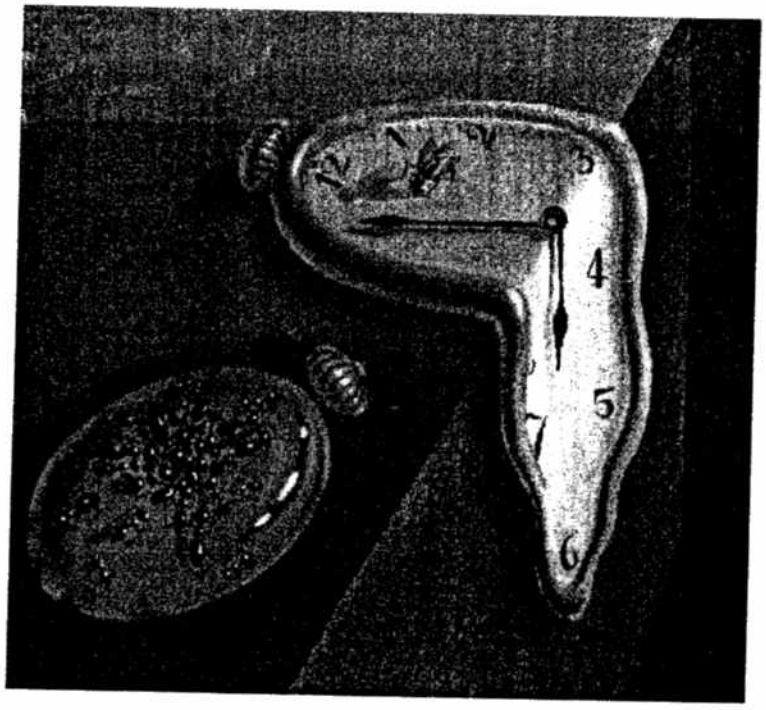
Institut für Automation
Abt. für Automatisierungssysteme

Technische
Universität
Wien

Projektbericht Nr. 183/1–9
Dezember 1989

On Diverse Programming for Vital Systems

G.H. Schildt



Ausschnitt aus: Salvador Dali, "Die Beständigkeit der Erinnerung"

ON DIVERSE PROGRAMMING FOR VITAL SYSTEMS

G. H. Schildt

Institut für Technische Informatik, Wien, Austria

ABSTRACT: After an introduction to safety terms and basic structures of vital process control systems a fundamental definition of diversity is given. Nowadays there are different interpretations of diverse system design: Diversity at run time of a system as 'on line diversity' and diversity according to software development process and the try to reproduce statements of system specification out of the object code (so called 'respecification') as 'off line diversity'. Problems of efficiency, man-power for diverse development as well as the proof of sufficient diverse design will be discussed. Futhermore it is pointed out that there does not exist a complete theory on diverse design principle, but some chance may exist to use diversity principle in checking system specifications against each other.

Keywords. Diversity; safety; fail-safe systems.

1. Introduction

In recent years the field of safety engineering has won importance in transportation systems and nuclear/chemical power plants. In these fields devices and control systems relevant to safety are used. Complex system design impacts the use of computer control, although one has to recognize that programs of high complexity will never be error-free. Thus one approach may be to use software redundancy. Before discussing 'diversity' as a design principle fundamental safety terms and system structures shall be presented.

1.1 Safety terms

Basically, some safety terms for automation systems shall be introduced /1/:

- System relevant to safety (vital system): control systems causing no hazard to people or material in case of constructional element failure, design fault or environmental influence
- Safety: property of an item to cause no hazard under given conditions during a given time; i.e. avoidance of undue fail conditions. Undue fail conditions are caused by
  - technical system failures,
  - design faults or
  - malfunction of an electronic device interfered by electromagnetic noise.
- Hazard (acc. to vital control systems): condition of a system that cannot be controlled by given means and may lead to injuries to people
- Safe condition of a system: property of a system condition to cause no hazard; i.e. in many cases the vital control system has a safe side. In some systems it is possible to achieve the safe system condition by switching the system off. Thus, e.g. in a guideway transit system, a 'safe halt' means a condition, in which the kinetic energy is zero, ($E_{kin}=0$).
- Fail-safe: Technical failures or design faults within an item may lead to fail conditions of a system ('fail') which, however, have to be safe ('safe'); i.e. a control system designed according to the fail-safe principle causes no hazard by a first admissible failure. The safe turn-off condition has to be an absorbing collective condition for all fail conditions of the system, which may cause any hazard.

1.2 Vital system concepts

System design of vital control systems can be done according to the following system structures. Basically, these fundamental system structures are valid for hardware as well as for software systems /2/.

2. Software diversity

Up to now there exist many diverse hardware realizations for example in the field of nuclear power plants. Because programs of high complexity will never be error-free, there may arise a certain chance by applying diverse system design on the software development process. The term of diversity has to be defined first:

"Diversity is to perform
a required function
by different means"

Program diversity as software redundancy bases on the assumption, that for a certain set of input data different programs performing the same required function will not react the same way. Yet one has to prove, that there is no correlation between these different programs, otherwise the whole concept will fail.

In some publications different interpretations of diverse design principle can be found. Therefore one has to distinguish between ...

1. an one-channelled software to control the process directly, but supervised by a monitoring software system (acc. to Fig. 2) computing, if all data correspond to a safe process control and are produced in time acc. to process control requirements /4/. If the software system design of the control system as well as the monitoring system are based on the same operational system specification, there is given no sufficient decoupling.
2. an one-channelled software for process control combined with a try to reconstruct any statements of system specification out of the object-code (so called re-specification or decomposition). This interpretation should not be named as a diverse system structure, but as a try to validate an one-channelled software system with the means of re-specification. But to have the possibility of re-specification it is necessary, that the software controlling the process has been specified formally before.
3. a n-channelled software system for process control at run time (so called 'on line diversity').

In the following only the diversity at run time will be discussed together with all problems and challanges.

2.1 Fundamentals of diversity

The intensions of applying software diversity are ...

1. to increase safety within a vital process control based on the assumption that at run time not all programs will react the same way.
2. to increase of availibility by using a majority voting strategy.
3. to detect software design failures by parallel testing and comparison of results (fig. 4).

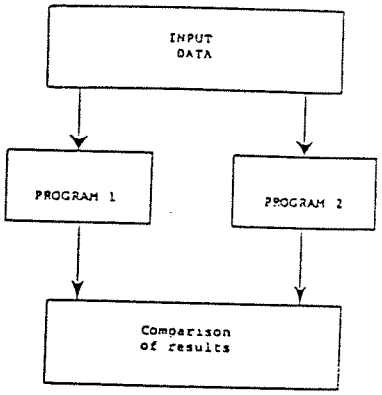


Fig. 4: Detection of software design failures

2.1.1 Physical diversity

How to realize physical diversity may be illustrated by the following examples:

- if a temperature within a boiler plant has to be measured, there exist two different approaches, using a convection thermometer and a radiation pyrometer.
- realization of a communication link using electric signals on a coax cable and optical signals on fibre optics.

The advantage of these different realizations is, that an independent testing laboratory will at once certificate sufficient diversity, because different physical processes have been applied. Thus, physical diversity is given, if in the different control channels run different physical processes.

2.1.2 Logical diversity

The following example shall illustrate logical diversity:
The calculation of the equation $x^2 + px + q = 0$ with constant coefficients leads to the well-known solution $x_{1,2} = -p/2 \pm \sqrt{(p/2)^2 - q}$
(for example to predict the stopping distance of a train at constant deceleration).

Another algorithm to solve the same problem as above may be the estimation method first described by I. Newton:
Dependent on a first estimated value n iterations have to be done, until the condition for two adjacent estimated values $|x_n - x_{n-1}| < \varepsilon$ is fulfilled (fig. 5).

1.2.1 Single channelled fail–safe control system

This system structure is described as follows:

- a single–channelled item will be fail–safe, if a failure or design fault relevant to safety leads the control system to a safe condition.

Condition graph and temporal description illustrate the system reaction in case of failure or if a design fault becomes effective (fig. 1).

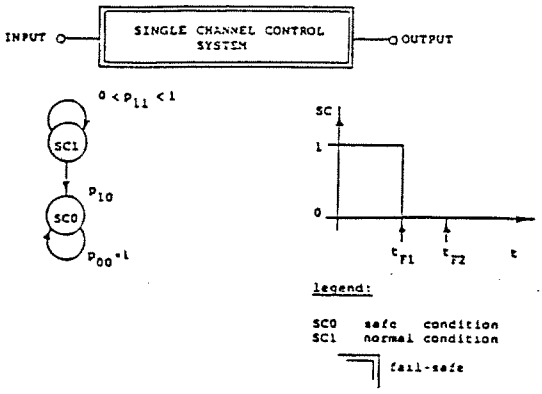


Fig. 1: one–channelled fail–safe control system

If a first failure occurs at t_{F1} system changes its condition from SC1 to SC0. In case of a second failure at t_{F2} system remains in the safe condition SC0. An essential feature of an one–channelled fail–safe system is that transition probability $p_{11}=1$, so that safe system condition cannot be left by itself. A system restart can only be done by an operator after system diagnostics and repair. Because only simple devices may be designed acc. to one–channelled system structure as shown in fig. 1, otherwise complex computer control systems have to be installed, one has to use other modified system structures.

1.2.2 Control system with monitor channel

For complex applications only commercial computers can be used although these computers have not been designed according to safety aspects. One approach to build a vital control system with commercial computers may be to install an additional monitor channel checking all input data, intermediate results and output telegrams of the not–intrinsically safe control system. On the one hand the monitor channel has to decide, if all data correspond to a safe process control, on the other hand monitor channel has to compute, if output telegrams are produced in time acc. to process control requirements. Fig. 2 presents the system structure together with a condition graph and temporal system description.

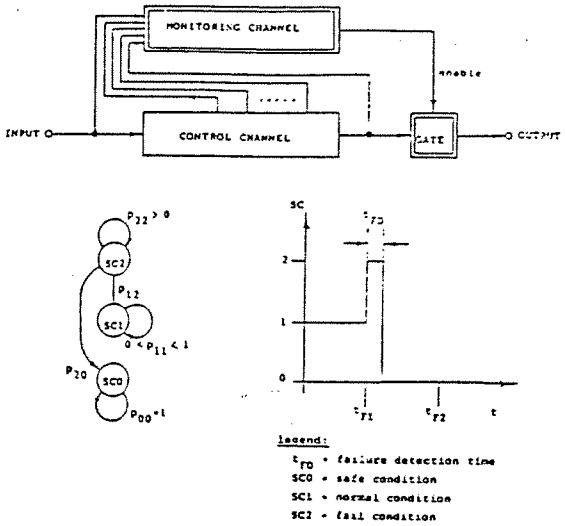


Fig. 2: Control system with monitoring channel

This monitor channel turns out to be a high sophisticated component because not only more facilities have to be implemented into the monitor channel in comparison to the control channel, but also the monitoring functions have to be done safely. Additionally one has to recognize that in case of a first failure monitor channel needs a certain time to detect the fail condition, i.e. there will exist a failure detection time t_{FD}.

1.2.3 N–channelled control system with (m–of–n) voter

Fig. 3 illustrates a control system consisting of n control channels together with an (m–of–n)–voter. This fault–tolerant system structure bases on a majority decision strategy. The implementation of voters can be made by specific complex hardware or as a distributed voting process by software on different computer channels. For m=n=2 the system structure becomes a double–channelled control system with a fail–safe comparator, but degraded availibility /3/.

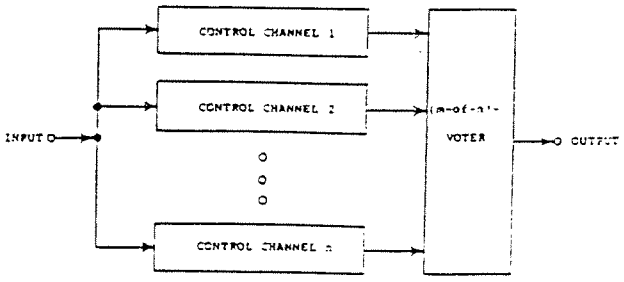


Fig. 3: System structure containing n control channels and a (m–of–n)–voter

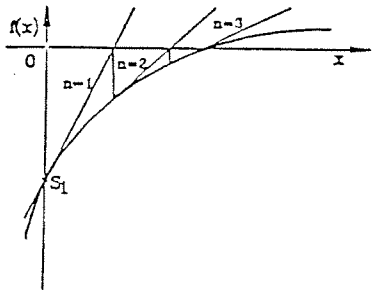


Fig. 5: Estimation method (I. Newton)

Thus logical diversity is given, if there exist different algorithms to perform a required function.

2.2 Typical problems of diverse software

Basically, diverse software design causes the following typical problems:

- non-planable waiting-time for results to be compared
- results R(x) may differ caused by rounding effects or different succession of statements
- increase of failure detection time t_{FD}
- a high sophisticated fail-safe voter will be necessary (not a simple comparator)
- one may not find a sufficient diverse algorithm in any case
- more than double man power effort in comparison to development costs of a one-channelled software system

How complex the function of the fail-safe voter is, will present fig. 6 for a double-channelled software system (for example to compute the brake parabola for a guideway transit system).

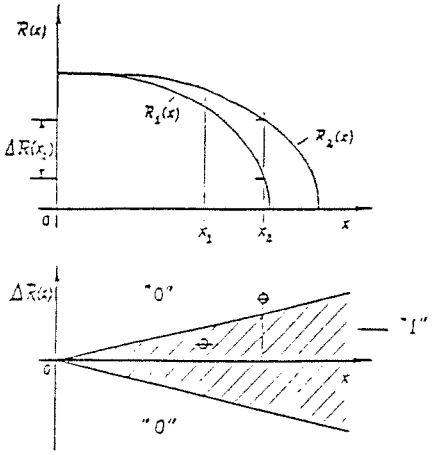


Fig. 6: Results R_1, R_2 versus x combined with tolerance zone evaluation

The graphs $R_1(x)$ and $R_2(x)$ are almost similar. Dependent on the value of x there will appear the difference of results $\Delta R(x) = R_2(x) - R_1(x)$. The voter has now to decide, whether both results will be evaluated as to agree within a given tolerance zone or not. The tolerance zone function can be defined optionally, in fig. 6 a symmetrical function is chosen ($\Delta R(x) = px$ with percentage p). Additionally, this high sophisticated voter has to operate safely. If this device cannot be realized by a relatively simple one-channelled hardware circuit, possibly systems structure as shown in fig. 2 and 3 have to be used for hardware or software implementation. An additional problem of diverse programs running parallely on two computer as shown in fig. 7 is, that one may find possibly some implementation faults, but because of an one-channelled specification no failures within the specification itself can be detected.

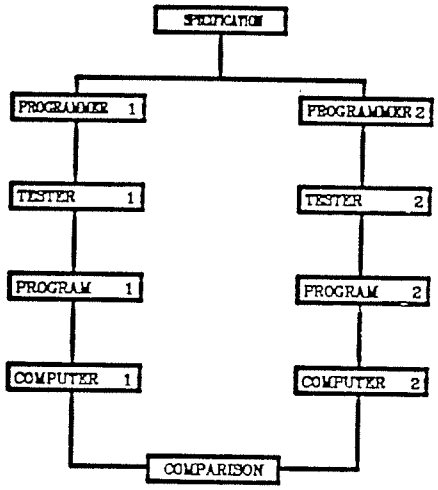


Fig. 7: Diverse software implementation

Because the importance of specification faults is much greater than implementation faults, one recognizes that efficiency of diverse design principle diminishes.

Up to now there have been made some experience on diverse software systems for example in the field of operation centers for guideway transit systems. There are two possibilities to run n programs: on the one hand n programs may run on n computers parallely connected with high hardware costs, on the other hand n programs may run on one computer sequentially connected with the disadvantage of high CPU-load (often CPU-load greater than 90 % is found).

Because of the problems discussed above the number of diverse software installations has remainded low.

3. On the fail probability of diverse programs

The fail probability p of a program shall be the probability that for a random data set a wrong result will be calculated. This probability has to be distinguished from the probability of the occurence of implementation faults within a program. If there are two diverse programs a and b with the fail probabilities p_a and p_b, one will find often the assumption that the fail probability of the diverse software system p_{DIV} will be $p_{DIV}=p_a \cdot p_b$. But this formula is only valid, if there is given statistical independence for both failure processes. Recently however, some doubts have been cast on the independence of design faults /5/. Because this statistical independence in most cases is not given (e.g. caused by a common system specification or some communication between both software development teams), the fail probability of diverse software system will be $p_{DIV} \gg p_a \cdot p_b$ /6/.

4. On validation of diverse programs

Futhermore, there is an additional aspect, namely the validation procedure for one-channelled and diverse programs. While an one-channelled program will be validated by a deepened function proof and reviewed by a white box test together with a C1-test coverage as the nowadays state of the art, for the different programs of a diverse software system is no white box test necessary. Otherwise one has to prove sufficient diversity between different programs. This, however, causes nearly the same costs as the validation procedure for an one-channelled program. Futhermore, there does not exist a complete theory how to proof the sufficient diversity of different programs within a diverse software system.

CONCLUSIONS

Because different interpretations of diverse system design are found, it is essential to distinguish between diversity at run time and the try to reconstruct statements of system specification out of the object code. Futhermore because there is no complete theory of diversity, one has the difficulty to proof sufficient diverse system design. There are typical problems as described above: Diverse system design is connected not only with high development costs but also with non-planable waiting time, increase of failure detection time, and a high-sophisticated voter. Because of these disadvantages one should not use diverse system design especially for vital software systems. However, a possible approach to use diversity may be to work out system specifications parallely by different teams and then to check both specifications against each other because of the great importance of faults or incompleteness within the system specification.

REFERENCES

/1/ Schildt,G.H.: Safety control systems interfered with electromagnetic noise, EMC-Symposium, Zürich, 1981.

/2/ Fricke, H., Schildt, G.H.: Conception of Safety and Realization Principles, ATRA-Conference, 1978, Indianapolis, USA

/3/ Schildt, G.H.: Grundlagen für Vergleicher mit Sicherheitsverantwortung, Siemens, Forschungs- und Entwickl.-Ber., Bd.9, Nr.6, S.347-353, 1980.

/4/ Theuretzbacher, N.: 'VOTRICS': Voting Triple Modular Computing System, IEEE-Transactions of Softwareengineering

/5/ Bishop, P.G., Pullen, F.D.: STEM-Software test and evaluation methods. A study of failure dependency in diverse software, Central Electricity Generating Board, Research Report, Febr. 1988.

/6/ Grams, T.: Diversitäre Programmierung: Kein Allheilmittel, Informationstechnik, 28. Jahrg., H.4, S.196-203, 1986.