The Contributions of Edsel Murphy to the Understanding of How the Behavior of Inanimate Objects Affects Computing, Computing Devices and Computer Science.

A Report to Dr. Robert Cupper
University of Pittsburgh
Computer Science Department

by John P. Breen

illustrated by Bob Heman

Abstract

Edsel Murphy's contributions toward understanding the behavior of inanimate objects as they relate to Computer _ Science is discussed. Murphy's General Law, Special Laws and their corollaries are presented, with supportive examples.

I. Introduction

Although *most* Computer Scientists are aware of the varied manifestations of Edsel Murphy's General and Special •Laws governing the^ behavior of inanimate objects, few of us have studied these laws in detail. This oversight is undoubtedly another manifestation of the General Law..

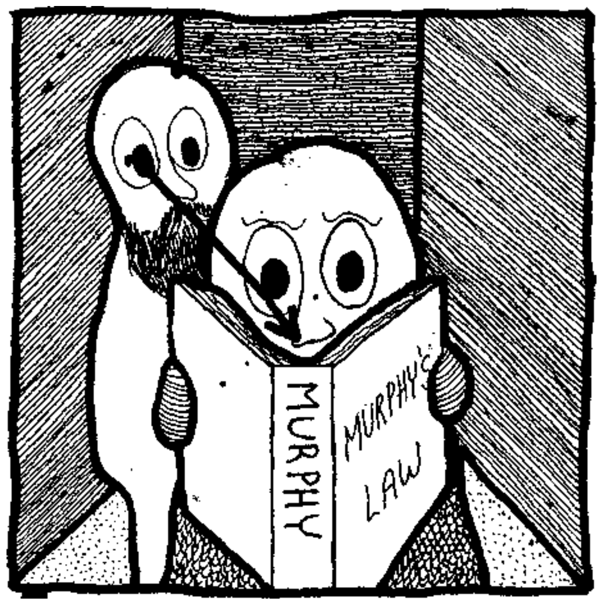
In this paper, the Author intends to show the all pervasive nature of Mr. Murphy's work by formal statement of the Laws and Corollaries find by presenting examples of special applications in the discipline of Computer Science.

II. The Nature of the General Law

Any concept or interdependency can be represented as a form, i.e. a function as a graph, or a more complex relationship as a topological surface. Consequently processes, per se, can be thought of as transitions from one steady state form to another. When the transition is continuous, predictable and smooth it can be*described by classical mathematical methods. It is seldom the case in nature, however, for these form evolutions to progress smoothly. Typically, natural dynamic evolution is abrupt and usually involves perplexing divergences.

This discontinuous and divergent phenomena resisted

"If anything can go wrong, it will."



formal mathematical representation for years, until the recent work of Rene Thorn.[1] Thorn proved that in a space having no more than four dimensions (such as the four-dimensional time-space relationship found in nature) there exists just seven types of transformation. Because of the discontinuous and disruptive nature of these transformations, he termed them the seven elementary "catastrophes", and the science they delineate, "catastrophe theory".

Since Murphy preceeded Thorn by several centuries the inadequacies of contemporary mathematics precluded the description of his theory in formal mathematic symbolism, but by calling on catastrophe theory we may not describe the General Law:

$$2 : 1+1 \qquad (1.)$$

where : is the catastrophe theory symbol for "hardly ever is equal to". In the vulgar, the equation may be stated, "If anything can go wrong, it will."

It is probable that the reader has been informally exposed to the central concepts of the General Law and 'that equation 1. has therefore struck a respondent chord. In an effort to solidify the reader's understanding of the concepts at issue, especially as they apply to computing, computing devices, and Computer Science, we shall present the General Law, the Special Laws and their corollaries in a tabular form.

III. The General Law, the Special Laws and their Corollaries

A. The General Law of Science

In any field of scientific endeavor, anything that can go wrong will go wrong.

Corollary 1: Everything goes wrong at the same time.

Corollary 2: If there is a possibility of one of several things going wrong, the one that will go wrong is the one that will do the most damage.

Corollary 3: Left to themselves, things will go from bad to worse.

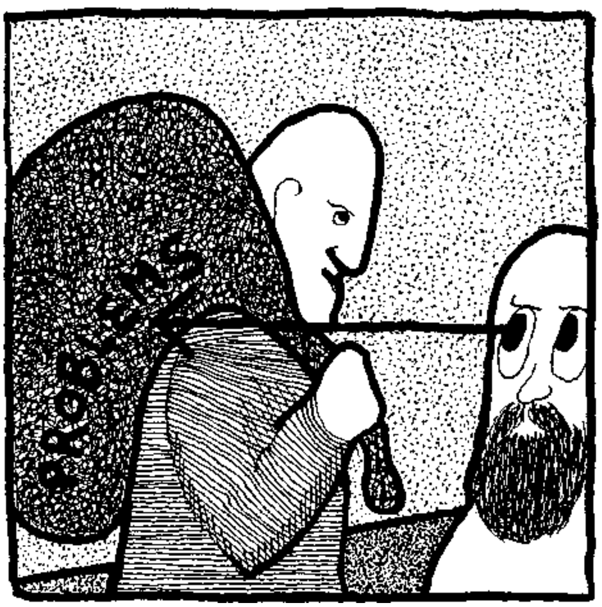
Corollary 4: Experiments must be reliable; they must always fail at the same place.

Corollary 5: Nature always sides with the hidden flaw.

Corollary 6: If everything seems to be going well, you have obviously overlooked something.

B. The Special Laws of Computing, Computing Devices, and Computer Science.

1. It is generally impractical to worry beforehand about problems; if you have none, someone will supply them for you.



Corollary 1: The need to change the basic algorithm will be communicated to the programmers after, and only after the coding is complete.[8]

Corollary 2: In simple cases, when choosing between two algorithms, one obviously right and one obviously wrong, it is often wiser to choose the wrong one so as to expedite subsequent revisions.

Corollary 3: The more innocuous a modification appears to be, the further its influence will extend, and the more the algorithm will have to be changed.[8]

Corollary 3a: The necessity of making major changes in an algorithm increases as the system approaches completion.

Corollary 3b: Firmness of completion date is inversely proportional to the tightness of the schedule.

Corollary 4: Suggestions made by the systems optimization committee will increase runtime and decrease capability.

Corollary 5: Any paper submitted to the ACM for publication, will be preceded by two weeks by a similar paper from a member of your department.

2. In any code or collection of data, the elements that are obviously correct beyond all need of checking, will contain the error.

2a. In any coded program, the modules which are correct beyond any shadow of a doubt, are causing the execution errors.

Corollary 1: No one whom you ask for help will see the error.

Corollary 2; The most nagging intruder who stops with unsought advice, will spot the problem immediately.

Corollary 3: In any miscalculation, the source will never be found if more than one person is involved.[7]

Corollary 4: Any error that can creep in, will; and it will be in the direction that will do the most damage to the calculation.

Corollary 5: All constants are variable.[3]

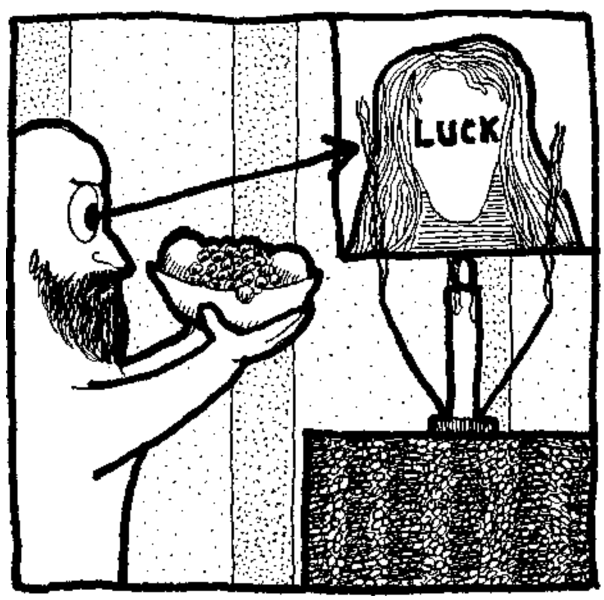
Corollary 6: In a complex algorithm, at least one factor from a numerator will move into a denominator, and a decimal point will simultaneously be misplaced.

3. All record of the third law has been lost. During the preparation of this paper the source material for the third law was misplaced; another manifestation of Murphy's Law. In keeping with the Law, these misplaced documents will be found on the day this paper is presented.

3a. Rules for Optimizing Software Design with Consideration given to Murphy's Laws.

The following precepts are offered to the systems designer who wishes to optimize his designs according to the Laws and their implications.

1. Do not simplify the design of a program if a way can be found to make it complex and wonderful.[6]
2. Always keep your disk area filled with obsolete files; it indicates that you have been busy.[3]
3. Before studying a problem, be sure you first under stand it thoroughly.[6 7]
4. Do not take luck for granted; believe in it and revere it.



5. When writing program documentation, always leave room to add an explanation in case the program doesn't work (the rule of the way out.)
6. Always use the latest developments in the discipline of

Computer Science when designing algorithms.

a. Items such as Finagle's Constant and the more subtle Bougerre Factor (pronounced "Bugger"), are loosely grouped, in mathmatics, under constant variables, or if you prefer, variable constants.

b. Finagle's Constant, a multiplyer of the zero-order term, may be characterized *as* changing the universe to fit the equation.[2]

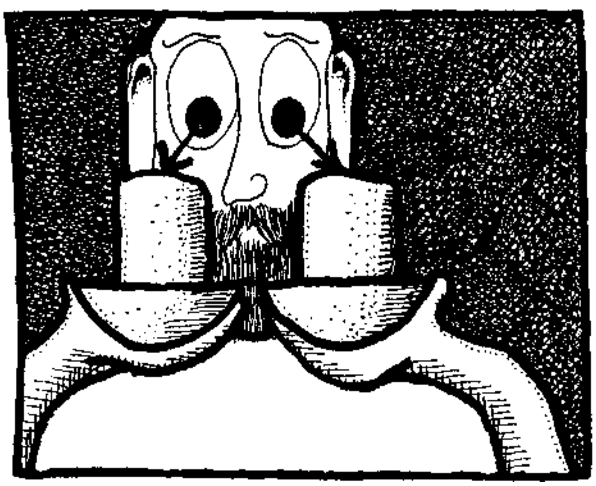
c. The Bougerre Factor is characterized as changing the equation to fit the universe. It is also known as the "smooth ing factor"; mathematically similar to the damping factor, it has the effect of reducing the subject under discussion to zero importance.

d. Any deficiency in rationalizing a casual relationship can be eliminated by the introduction of a variable constant of the invariant type, to wit the Phuje Factor (pronounced "Fudge"). The value for the Phuje Factor is chosen to fill the gap be tween the real and the imagined.[8]

e. A combination of the above three, the Diddle Coef ficient, is characterized as changing things so the universe and the equation appear to fit without requiring a change in either.

4. A critical circuit, required to interface devices to the same manufacturer's mainframe, will not be available from the OEM.

Corollary 1: Interchangeable parts, won't.



Corollary 2: An important Instruction Manual or Operating Manual will have been discarded by the receiving department.

Corollary 3: All Warranty and Guarantee clauses become void upon payment of the invoice.

Corollary 4: In any device characterized by a number of plus-or-minus errors, the total error will be the sum of all the errors accumulating in the same direction.

Corollary 5: The manufacturer's specification will be incorrect by a factor of 0.5 or 2.0, depending upon which coefficient gives the most optimistic value. The salesman's claim for these values will be 0.1 or 10.0 respectively.

Corollary -6 The probability of failure of a component, assembly, subsystem or system is inversely proportional to the ease of repair or replacement.

Corollary 7: A dropped tool will always land where it will do the most harm.

Corollary 7a: The most delicate component will drop. (Also known as Murphy's Law of Selective Gravitation, this is proven when a dropped slice of buttered bread lands with the buttered side down.)

Corollary 8: Device drive motors will rotate in the wrong direction.

Corollary 9: Dimensions will be given in the least usable units. Tape velocity, for example, will be given in furlongs per fortnight.

Corollary 10: A purchased component or instrument will meet its specification until, and only until, it has passed incomming inspection.

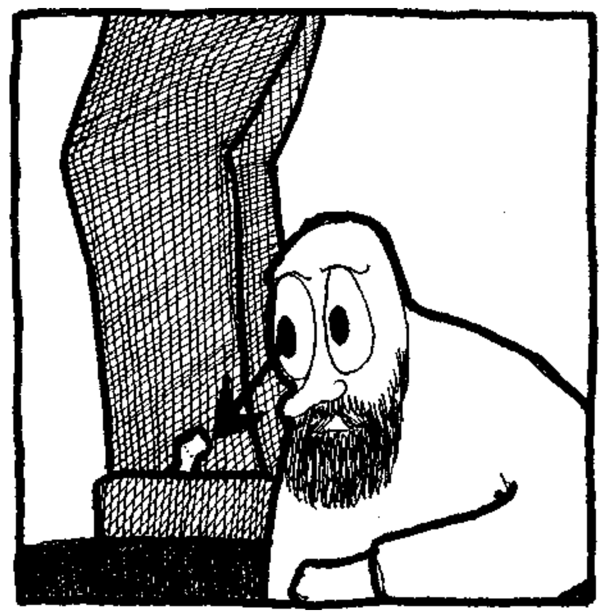
Corollary lOa: A failure will not appear until a unit has passed final inspection.

Corollary 11: Interactive plotters will deposit more ink on people than on the paper.

Corollary 12: After the last of sixteen mounting screws has been removed from an access cover, it will be discovered that the wrong cover has been removed.

Corollary 13: After the last of sixteen mounting screws has carefully been replaced, it will be discovered that the access cover gasket has been omitted.

Corollary 14: After an instrument has been fully assembled, extra components will be found in the cuff of someone's trousers.



Corollary 15: If an obviously defective part has been replaced in a device which was exhibiting intermittant fault, the fault will reappear after the device is returned to service.

4a. Any attempt to "breadboard" an unavailable circuit will exceed the estimated cost by 3^{21} and will fail.

Corollary 1: Any wire cut to length will be too short.

Corollary la: At least one technician will attempt to solve the problem by cutting off more wire.

Corollary 2: The availability of a component is inversely proportional to the need for that component.

Corollary 3: If a circuit requires n components, there will be n-1 components stocked locally.

Corollary 4: If a particular resistance is needed, that value will not be available; furthermore, it cannot be developed with any available series or parallel combination.

Corollary 4a: In breadboarding, Murphy's Law supersedes Ohm's.

Corollary 5: A device selected randomly from a group having

99% reliability, will be a member of the 1% group.

Corollary 6: A crystal filtered 5 volt D.C. power supply will supply 13.5 volts A.C.

Corollary 7: If a circuit cannot fail, it will. Corollary

8: A fail-safe circuit will destroy others.

Corollary 9: An instantaneous power supply crowbar circuit will operate too late.

Corollary 10: An integrated circuit chip, protected by a fast acting fuse, will protect the fuse by blowing first.

Corollary 11: A crystal oscillator will oscillate at the wrong frequency, if at all.

Conclusion

Since everything in nature, including its sciences and disciplines, is dominated by the phenomenon described by Edsel Murphy it is essential that we learn to live in harmony with it.

The computer threatens to vanquish error. Fortunately nature provides checks and balances and Murphy's Law serves to temper and moderate the discipline of Computer Science. Many of man's advancements came when, in an effort to overcome error, he reached past his limit to accomplish the impossible. "If we begin with certainities," Bacon said, "we shall end in doubts; but if we begin with doubts, and we are patient with them, we shall end in certainties".

Of course we will occasionally be hampered, hindered, frustrated and angered by the intrusion of the Murphy phenomenon. This author submits, however, that a genuine purpose is served by this visitation; it is the nature of our humanness to remind us that we are human. The infusion of error into the grandest of schemes is another service provided to us by nature. Ferris Greenslet wrote, "Give me a good fruitful error, full of seeds, bursting with its own corrections; you can keep the sterile truth for yourself."

The danger to man is not that he will be controlled by his computers as much as it is that he will imitate them. Humor is the ultimate weapon for scientists to deal with frustration. Finding the humor in the inevitable awkward moment causes the awkwardness to evaporate. Sensitivity allows man to find the humor; and sensitivity separates the specialist from the poet. Aristotle reminds us that the specialist expresses only the particular, while the poet expresses the universal.

V. Cited References and Footnotes

1. Thorn, Rene, *Structural Stability and Morphogenesis, D.* H. Fowler, New York, New York, (1975).
2. Wirth, N., Systematic *Programming: an Introduction,* Prentice-Hall, Englewood Cliffs, New Jersey, pp. 14-25.
3. Fisher, D. L., *Data Documentation and Decision Tables,* CAMC 9, No.l, (January 1966), pp. 26-31.
4. *USA Standard FORTRAN,* ANSI Standard X 3.9-1966, American National Standards Institute. New York, New York.
5. Wright, Donald, *A comparison of the FORTRAN lan guage implementations for several computers,* CAMC 9, No. 2, (February 1966), pp. 77-79.
6. Kernigan B. W., Plauger, P. J., *The Elements of FORT RAN Style,* McGraw-Hill, New York, New York, pp. 131-137.
7. Dijkstra, E. W., Notes on *Structured Programming,* T. H. Report WSK-03, Second Edition, Technical University Eindhoven, The Netherlands.
8. Cupper, R. E., *Kluges and Glitches,* Lecture Notes from CS-1, (November 15, 1973). •