

# 医薬安全性研究会

会報 No.22

Jan. 1987

## 目次

無影響量の推定をめぐって

日本ロシュ(株) 高橋行雄/東洋醸造(株) 松本一彦…………… 1

発癌性試験の評価に関する統計学的検討 [連載第1回] John J. Gart,  
Kenneth C. Chu, and Robert E. Tarone 著

住友化学工業(株) 山岡秀明 訳…………… 12

“コンピュータシステムの信頼性”に関する文献紹介(2)

「信頼性低下の法則」 Tom Gilb 著/(株)生技研 佐野正樹 訳… 24

第28回定例会出席者名簿…………… 35 事務局だより…………… 36



問題は「対照群との間に有意の差があるときには毒性変化とみる方がよい」という点に我々は、ひっかかるのです。これは、有意の差というのが統計的なのか生物学的な有意の差なのか、どちらなのか大きな問題と思っています。もう一つ問と答があります。

問23 亜急性・慢性毒性試験（生殖試験も含む）で、無影響量を求めるとあるが、薬剤本来の薬理作用と切り離せない副作用の場合は、無影響量は薬効量以下となるので、無影響量を出す必要はないと考えてよいか。

これは、我々がしばしば経験することです。この答として

【答】 亜急性・慢性毒性試験等の目的の一つは、その薬剤の影響が、客観的にとらえられない程度の量を把握し、よって、その薬剤の安全性評価に利用しようとするものである。従って、毒性変化か、薬理変化かというように変化の区分に固執せず、対照群と有意の差、もしくは生物学的意義のある差を生じない量としての無影響量を求める必要がある。

が示されています。このガイドラインが出された後に、説明会がありその時に、ある抗生物質メーカーの方が質問を出しました。この時の記録が印刷物になっていますので、紹介します。

Q ビーチャム：戸部先生に伺います。無影響量のことで、たとえば抗生物質の場合ですと、無影響量で盲腸の拡張とか、それがずっと低投与で出る場合があります。臨床例よりもずっと下回る場合があるわけですね。その場合の評価をどのようにしたらいいかということですが。

A 戸部先生：盲腸重量の増加がくるということを記録して、報告書の中にも記載されれば、それでいいんじゃないかと思います。

というのは、その事実は非常に有名ですが、ただ、それがどのくらいの量からおこるだろうかということ、資料を見る側から言えば、一つのインディケーターになるわけですし、従来のものに比べてどうかという対比になります。

Q その場合、無影響量になるまで投与量は下げなければならないということでしょうか。

A 戸部先生：必ずしもそれがなくなるまで低くする必要はないのではないのでしょうか。

Q 無影響量の考察のところで、そういう薬物本来の薬理作用に基づいて考えた無影響量と、あと、肝臓への影響とか、明らかに毒性の影響から考察した無影響量は別々に考察するのは、いけないでしょうか。

A 戸部先生：それは非常にいいと思います。本来ならば、それぞれ指標別に、影響する量はどのくらいかを別個に記載する方がいいと思います。

これは、盲腸重量についてですが、話しが少し変わってきているようで、対処の仕方が難しくなってきました。

この無影響量の定義からすると、肝臓の場合には無影響量だが、他の臓器では、違うといったような回答の仕方も可能になっています。

## 2. 製薬協でのアンケート調査

5年程前に、製薬協の基礎研究部会で、あるアンケート調査をしたことがあります。この中に、最大無作用量についての質問に対する回答がありましたので紹介します。

問 それぞれの試験で最大無作用量だけを求めているか、あるいは他に最大安全量とか最大耐量とか最大無毒量とかの言葉を使い分けているか。使い分けている場合には、どれを用いているか。最大無作用量を決定する判断基準として3つの考え方を示す。

- i) すべての検査・観察項目に対照群とまったく有意差を生じない最大量
- ii) 薬剤投与によって生じたと判断される変化がない量を最大量とする
- iii) 薬剤投与によって生じた有害と判断される変化が出ない最大量

24社から回答がよせられました。答として、試験ごとに最大無作用量だけを使用しているのは11社で、最大無作用量と最大安全量の2つを使い分けているのが11社で最大無作用量と最大安全量および最大耐量を使い分けているのが1社ありました。

この回答からも、最大無作用量という言葉の解釈にはかなり幅があります。ちょっと差が出てはだめ、とする施設が半分と、そうではない施設との差が出てきたわけです。

この質問の後に設けたエピソード欄に書かれていたものに基づいてのフリーディスカッションの結果を紹介します。

主効果を判定する項目が毒性試験の検査項目に含まれている場合は、有害と思われる最大量を最大無作用量とする。このような例は少なからず認められる。

これは、有害であるが否かを重要視するような行き方です。

制がん剤とホルモン剤については、最大無作用量についてはとらわれないで書く

このような人がいましたが、それで大丈夫だったか聞いてみたところ、やはり厚生省よりコメントを求められたことがあったそうです。

薬物の種類によっては薬理作用が、動物実験ではごく低用量で、人体投与量よりかなり低い用量より発現することがある。そのような薬物で最大無作用量を求める意味またはその必要があるのかどうか非常に疑問がある。

このような素朴な意見も出てきました。

ホルモンなどでも、きわめて少量でも薬理的な変化が現れるため、それらの過剰な変化を除いて毒性学的に意味のない最大量を最大無作用量としているが、主作用が極めて強く現れたときに判断が難しい。

確かに、カルシウム低下剤などを投与しますと、投与した後何時間かは必ず下がってい

るのですが、24時間たつと元に戻っている。どこでとるかによって、最大無作用量が変わってくる。この疑問はこのようなものです。

ステロイド剤の最大無作用量の指標として、血中コルステロンを定量したことがある。

厚生省の通達の中に最大無作用量を明記しなさいとの記述があるが、その定義が不明確である。

これは今、議論の対象となっているものです。

最大無作用量が求められなくて実験をやりなおしたという施設もかなりありました。1つの亜急性実験をやりなおしますと委託試験では2,000~3,000万円かかります。必要な実験であれば金額は問題となりません。しかし、果たしてかなり低用量で、薬理量あるいはヒトでの常用量よりかなり低いところについてまで求めなければ毒性診断ができないのか否か疑問をもっているわけです。

厚生省より、これを指摘されないようかなり広く投与量の設定をしたり、何もなしのところ、といったような投与量を設定しはじめたりしているわけです。

これとは別に、なんとか統計的に外挿をして、最大無作用量は、ここだと推定することができるのかどうか、次に高橋さんに話しをして頂くことになっています。

果たして、そういう方法があるのか、もしあれば言葉としては、ちゃんと無影響量を出せることになります。

高橋さん、よろしくお願いします。

## 1. 無影響量の統計的意味

(高橋)

毒性試験データの解析には、よく2標本の有意差検定が用いられている。さて無影響量についての検定は行えるのであろうか？無影響量は推定の問題であり検定ではありません。動物試験の世界で、量の推定を非常に多く行っている分野があります。特に薬理試験では検定よりも量の推定が好まれています。

ある薬理試験のED<sub>50</sub>の推定値について、厚生省より説明をもとめられたことがあります。ここに示すのは、その時の回答のために検討した結果の一部です。薬理試験では、用量を何とか推定しようとしています。通常はED<sub>50</sub>なのですが、ED<sub>50</sub>が出ない場合には、ある抑制率、例えば20%ぐらいではどうかと逆推定をしてED<sub>20</sub>を求めます。薬理では、さらに信頼限界を出そうとします。毒性で検定が好きなように、薬理では95%信頼限界が好きなのです。

これにより、いろいろ問題が起きてきます。図1に示すように、ED<sub>20</sub>の95%信頼限界を求めようとするとも10<sup>xx</sup>というように極めて大きな値となってしまいます。ED<sub>20</sub>値は、このことから意味があるのだろうか、というのが厚生省の疑問と思われました。

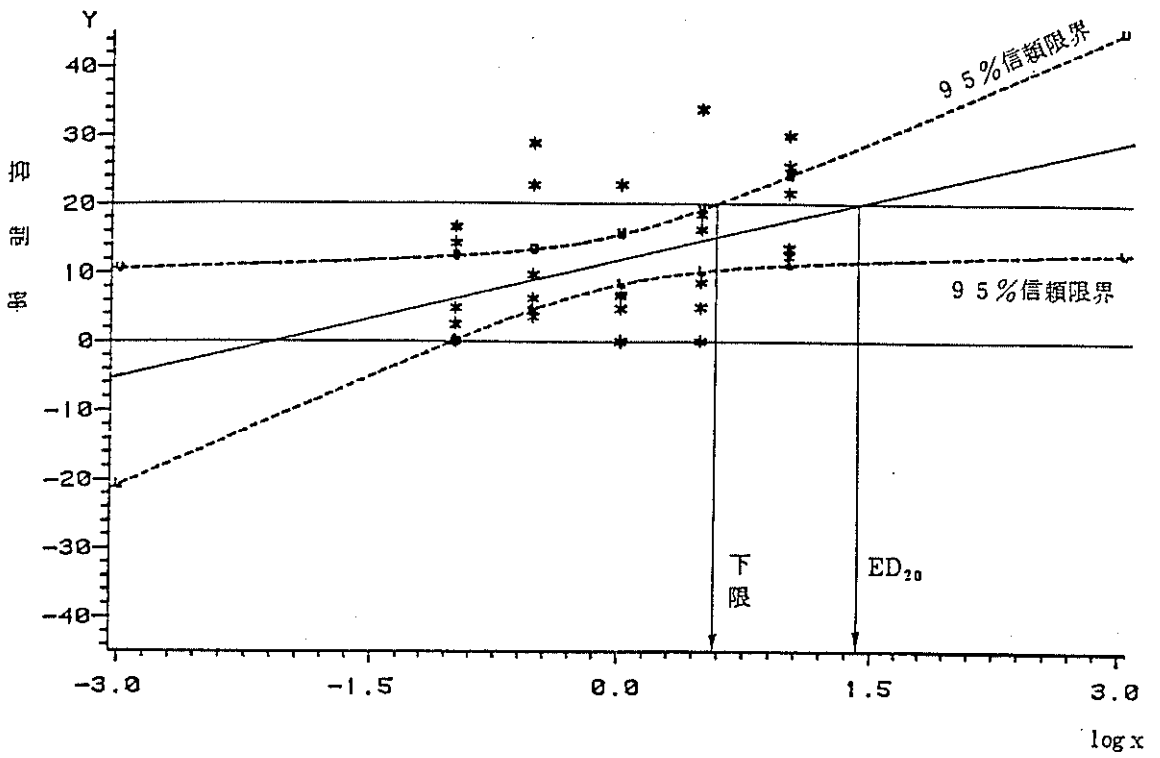


図1 95%信頼限界の表示

SAS

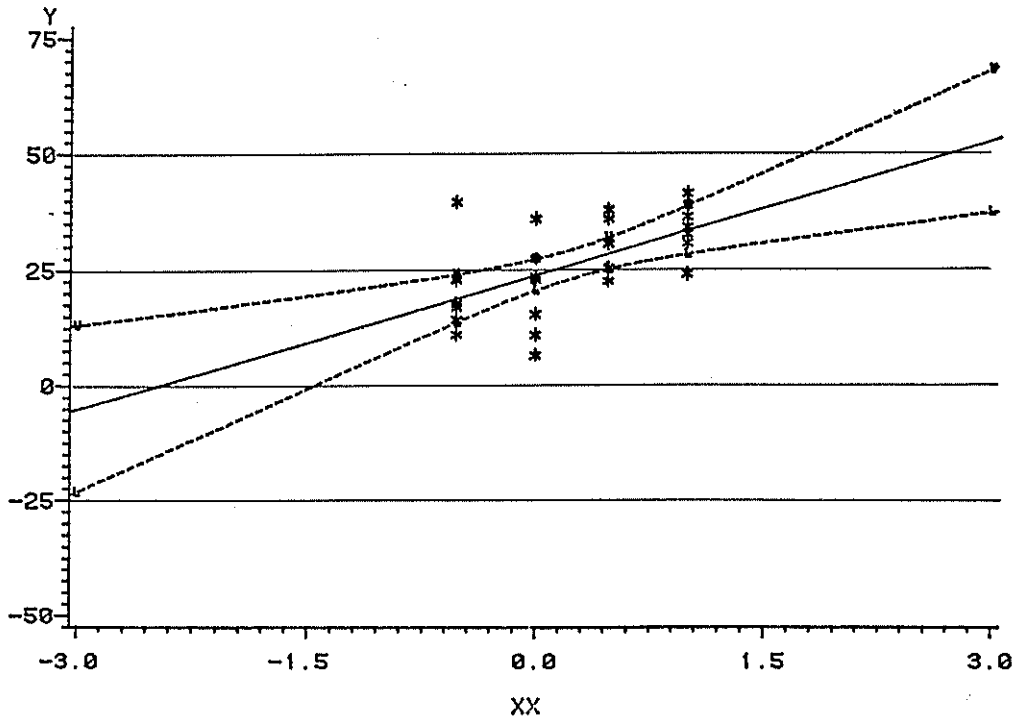


図2

## 2. 無影響量の推定

毒性試験では、必ずコントロール群が置かれることが薬理試験との差です。図1の上でいろいろなコントロール群の出方を示してみます。

コントロール群が、低用量群と同じ分布状態であったとします。もし、有意差検定をしたとしたら、高用量群のみに有意差が出るはずですが、コントロール群の平均値に対する用量をとりあえず無影響量としてみよう。ところが、図から明らかなように95%信頼限界の下限が $-\infty$ となってしまう、無影響量が求まったとしても、あまり意味がないとの結論となります。コントロール群が、もう少し上の方になったらどうであろうか？なんと95%信頼限界を求めることができるかもしれません。

図1は、かなりひどい例でしたが、図2のようにdose responseがはっきりしている場合には、コントロール群の分布にかかわらず、無影響量と95%信頼限界も求めることができます。もし、コントロール群に対し、低用量群が明らかに有意差がでる位低かったとしましょう。この場合には、従来の検定の方式では、無影響量は実験の範囲外であるのでわからないとなってしまう問題が起きます。図2の場合であれば、実験の用量の範囲外に、コントロールの平均値があったとしても、もし、95%信頼限界が求めることができるとしたら、無影響量を外挿したとしても安心できるのではないのでしょうか。

## 3. 95%信頼限界の例

ED<sub>50</sub>とその信頼限界の出方について、実際の薬理試験の論文からひろってみると、いくつかの表現に表1のように分類できました。

表1 95%信頼限界の表示例

区分	薬剤	ED <sub>50</sub>	基準
A	X	1.21 (0.40~2.61)	対数用量反応に有意な一次回帰がみられ、95%信頼限界が算出できた場合
B	X	27.3 (3.83~ $\infty$ )	対数用量反応に有意な一次回帰がみられ、95%信頼限界も算出できたが、片側の数値が大きすぎて表示できない場合
	Y	85.9 (49.5~)	
	Z	1.22 (0.57~-)	
C	Y	36.9 ( ~ )	probit法によりED <sub>50</sub> 値を求めたが、95%信頼限界はFiellerの式におけるg値が1を越えたため算出が不可能であった場合
D	Z	ca 4.7	対数用量反応に有意な一次回帰はみられなかったが、求める抑制率が試験した用量範囲内であると推定された場合
E	X	>30	対数用量反応に有意な一次回帰がみられず、最高あるいは最低用量でも求める抑制率が得られなかった場合
	Y	<80	

区分Aは、一番素直な例です。区分Bは、ED<sub>50</sub>は求まったが、95%信頼限界の上限がわからない例です。区分Cは、なんとかED<sub>50</sub>は求まったが、信頼限界は計算できなかった例です。区分DとEについては、用量反応があるかどうかすら分からない例です。

この結果をもう少し整理して、回帰分析についてのフローチャートを図2のように作ってみました。

このフローの中で、低用量群にのみ有意差が出て、高用量には、有意差が出ない逆用量反応の例を示しましたが、表2に示すように実際の毒性試験の報告書の中に見出すことができます。

表2 Blood chemical analysis—13 weeks oral toxicity study of X X X in rats (Female)

Dose(mg/kg/day)	Control	15	35	85	200
No. of animals	10	10	10	10	10
GOT (KU)	100.25±31.90	90.85±15.15	87.65±17.12	94.70±31.33	76.65±17.08
GPT (KU)	22.95±5.81	33.25±10.02*	28.10±12.30	23.25±6.85	19.75±3.43
Al-P (K-AU)	13.36±2.64	10.58±2.31*	12.93±5.44	9.96±2.67*	7.06±1.89**
LAP (G-RU)	104.1±10.5	110.0±7.7	102.2±11.6	110.3±12.8	110.1±11.1
Glucose (mg/dl)	151.6±24.8	154.7±31.5	156.4±13.8	145.9±13.8	139.9±10.6
Cholesterol(mg/dl)	71.55±9.89	69.97±11.02	75.44±9.87	66.59±5.02	74.66±7.86
Urea-N (mg/dl)	13.3±1.1	13.4±1.5	12.1±1.7	11.7±1.6*	12.9±0.9
T-Protein (g/dl)	6.16±0.32	6.11±0.26	6.21±0.37	6.11±0.18	5.87±0.29*
A/G ratio (-)	1.178±0.113	1.352±0.089**	1.277±0.137	1.234±0.082	1.195±0.098
Na (mEq/l)	142.24±2.08	144.81±2.29*	143.86±2.68	139.84±1.96*	142.60±1.24
K (mEq/l)	4.38±0.54	4.36±0.32	4.05±0.22	4.21±0.29	4.64±0.33
P (mg/dl)	6.92±0.39	7.78±0.79**	7.06±1.02	6.59±0.95	6.34±0.53*
Ca (mg/dl)	9.34±0.36	9.78±0.44*	9.73±0.37*	9.51±0.26	9.36±0.26
T-Bilirubin(mg/dl)	0.1403±0.0380	0.1496±0.0295	0.1217±0.0238	0.1222±0.0243	0.1175±0.0248

Mean±S.D. Significance of treatment-control difference: \*  $p \leq 0.05$  \*\*  $p \leq 0.01$

GPTは、低用量群のみで有意差が出ているだけで、高用量群になるにつれてコントロール群に近づくといった逆用量反応の例です。Al-Pは、無作用量が実験範囲外の例です。A/G比も逆用量反応となっており、どうしても不可能です。

この毒性試験では、コントロール群と15mg/kg投与群間で有意差検定は、304項目行われています。このうち、危険率5%以下で有意となっている項目は、20項目もあります。この報告書では、無影響量を

以上の成績より、本薬のラットに対する最大影響量は35mg/kgと推定された。

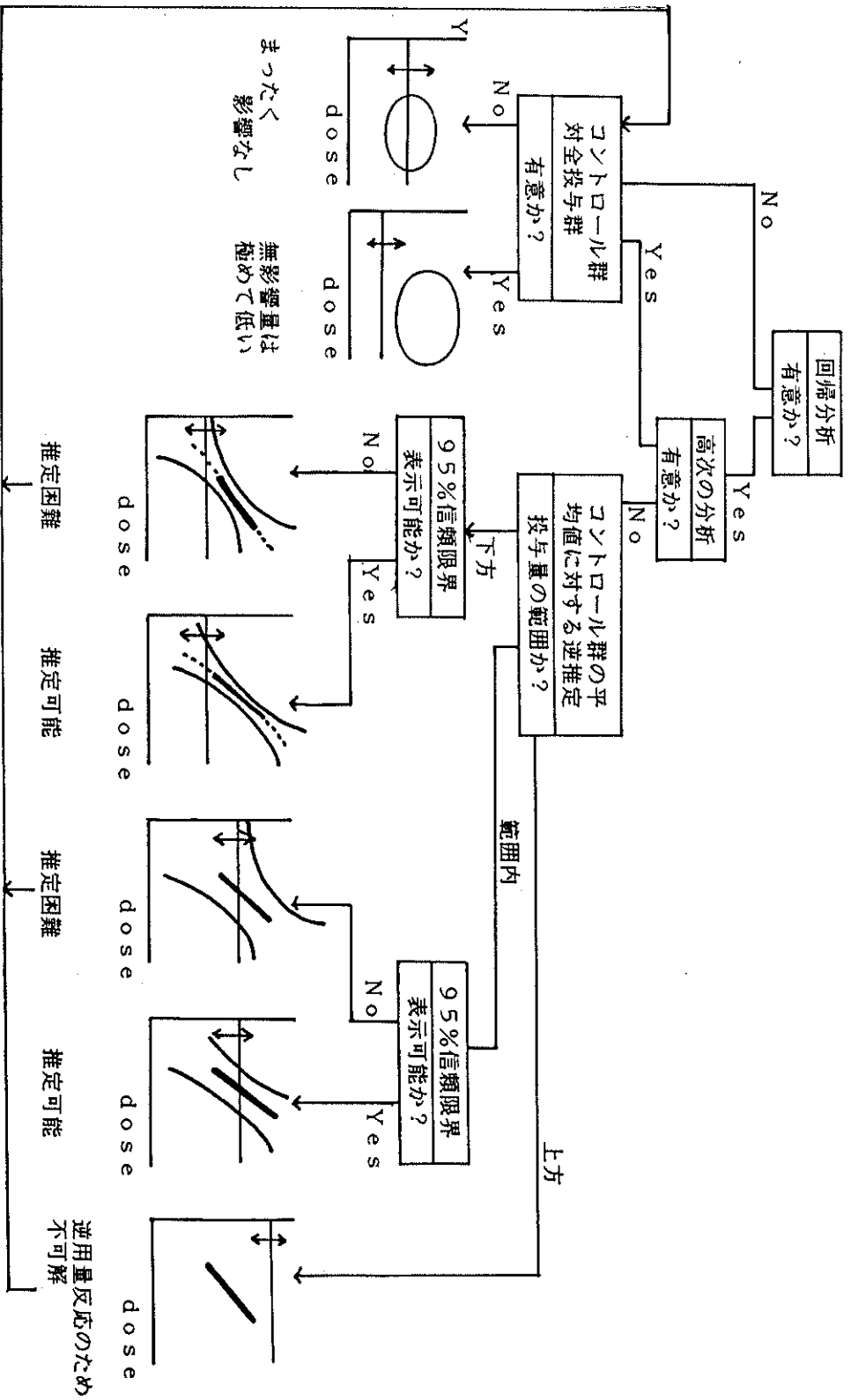
としています。何を基準に決めたのでしょうか。生物学的な判断が入っていると思うのですが、その基準については示されていません。

Naはコントロール群の上と下で差が出ており理解に苦しみます。KおよびPも同様でわかりません。Caも逆用量反応です。

現実の実験結果からは、図1に示した回帰分析のフローでは非現実的と思われる例も実際には存在しているわけです。コントロール群との間の有意差検定だけを行っている、無作用量を推定しようとする立場からは、結果の表現上いろいろと損をすることになります。もう少しマクロ的に回帰分析という見分からすると、無影響量も推定できますし、もしできない場合には、毒性学的に良く考えなければいけないとの結果をだしてくれます。

以上、無影響量について、最近考えていることを若干お示しいたしました。





## 質疑応答

**増山** 無影響量の定義が何か良く分からないのですが、具体的な例を思い出したので付け加えたいと思います。

戦後、例の黄変米ですが、これを配給していかどうかいくつかの班に分かれて実験をしました。最初出された結果は「黄変米の量が、ある程度以下ならば混入していてもよろしい」とこのことでありました。その根拠は、ある動物実験で肝臓の切片からある用量以下では変化がみられなかったので、その程度ならば配給しても良いとの結論が出されたのです。

その時、私は奇妙な気がしたのです。肝臓の切片を調べるといっても全部を調べているわけではないのです。従って、肝臓の一部しかおかされていなければ、見逃されている率がかなり高い。それで、東大で私の講義を聞いていた学生をつつついて、1匹の動物から1000枚のプレパラートを作ってもらい病理検査をしました。その結果、黄変米を与えたものは100%おかされてしまいました。そこで、慶応の病理でもやってもらったら同じように100%なのです。

結局「大丈夫だ」というのは、従来の病理のサンプリング方法では見つからなかったということなのです。この結果、配給は中止となったのです。

この例から、無作用量というのがどういうことなのか疑問なのです。

松本さんの示した問22についての回答は、理解しがたい。この回答でいいとすれば、なるべく検査しない方がいいのです。サンプリングの率を減らしてしまえばいいのです。見逃しの問題が全然考慮されていないところが不可解なのです。

**高橋** 一問22の答を読み上げる。

**増山** その答の中の「対照群と比較して有意の差、もしくは生物学的意義のある…」が問題なのです。サンプリングの割合を減らしてしまえば、有意差が出てこないのがあたりまえになっている。

**松本** 逆に、1群の動物数を増やせば増やすほど危険が伴ってくるのです。

そうすると最低限でいこうということになる。「無影響量を求めよ」ということは、毒性学にとって非常にマイナスの方向となっています。毒性試験の感度を上げれば上げるだけ有意差が出やすくなります。我々は、分散を小さくしよう小さくしようと努力してきたのですから。厚生省の先生方からも無影響量を求めるのは、まずいのではないかともいわれています。それでは、何をとらえて、影響ありというのか、厚生省から逆に考えてほしいといわれるわけです。

**増山** 前に経験したことなのですが、与えたものの影響をいつ見ているのかも問題です。極端な例かも知れませんが、ある病気の患者の体温を夜も自動的に連続記録するようにしたら真夜中に高熱を出していることが分かりました。通常の検温だけでは分からなかったことだったのです。何をいつ計るのがきいてくるのです。やはり、こういう場合には見逃しの方が問題だと思います。

**柳本** 無影響量ということをつき進めていくと、結局のところデータから確実に言え

るのは、常にゼロしかないのではないのでしょうか。有限の標本からの限られた情報で推定しようとするのですから。

**増山** おそらく、薬理の方々は、薬物の量が薄くなれば処理できると考えがあるのでないのでしょうか。放射線に対しては、弱くてもDNAが切れたり、重合を起こしたりしますけれども、その場所が少なければ修復酵素が直してくれるのですね。

放射線がある程度以下ならば、回復過程が存在しうるわけです。したがって薬の場合でもある程度以下ならば、自分自身の持っている酵素のシステムで処理できるという考えが出ているのではないのでしょうか。

それ以下を無視してというのではなくて、良心的に考えている方は、そういう可能性があるからその値を決めろと言っているのではないのでしょうか。しかし、我々統計屋さんから見ると、 $\beta$ の問題でおかしいと思うのですが、無影響量を考えた方はたぶん放射線だとか、あるいは薬物の処理に関してはある程度以下ならば処理できるのだから、したがって、その量を捜せというのがこれを考えた人のアイディアだったろうと思います。

**長谷** 臨床を担当している者にとって、無作用量と聞くとすぐに人に投与するときはどうするか考えます。現在は、60分の1か120分の1かをまず投与してみようというのが一般的かと思います。非可逆的なところは困るのですが、ちょっと胃が痛くなるような量だったら120分の1でも人に投与してもいいのではないかと思うわけです。私としては、人に投与しなければならぬ立場ですので、何でも良いからとにかく“量”を決めてほしいのです。

**増山** それは確かにそうなのです。放射線科の先生方は、当面の治療の目的にはいろいろな副作用があってもやむを得ないと考えるのです。臨床家として他に手がなければやらざるを得ないのです。

**高橋** 無影響量が毒性実験をした用量の範囲外であったときに一番困るのは、さきほど松本さんが話したように実験をやりなおししなければならない時です。そうすると、メーカーにとっては、死活問題なのです。最低投与量で有意差が出ているか、出ていないかといったところで議論しなければならないとしたらバカバカしいと思うのです。

再実験が必要なのは、どういう場合なのか、あるいは再実験の必要のないときはどういう場合なのか、もし私がそれを判断しなければならないとしたら、図1に示したように、「無影響量が極めて低い」といった結果が多く出ている場合には、再実験が必要ないと思います。それ以外の結果ならば、良く考察すれば必要ないと思います。

**柳本** 高橋さんの提案は、極めて矛盾に富んだ示唆だと思うのです。それは、コントロール群のバラツキを無視していること、平均を用いていることがおかしいと思います。もっとおかしいのは、用量反応直線に対し、コントロール群の平均値が、低用量よりも上の方で交わっている例です。用量が少なくなればコントロール群に近づくことを考えると、あきらかにおかしいのです。

**吉村** 現実にこの様な例がなければ問題がないのですが、高橋さんが示されたのは、ほんもののデータですか？

**高橋** そうです。現にいくつかの検査データは、とても理解に苦しむものも含まれています。

**柳本** コントロール群を含めたモデルを作るべきだと思うのですが。例えばVSDのような形で1年間に1.5人ぐらい癌になってもしかたがない、というぐらいに腹をくくった量が考えられるのではないのでしょうか？結局、無影響量なんて嘘なんです。だからこそVSDという量が考えられたと理解しています。

**増山** 私も賛成です。統計的な手法でどこまで保証できるかという問題と、どうしても犠牲になる人がさけられないならば、それがどういう人であるかということをはっきりとすることだと思ふのです。アレルギーと分かっている人には、アレルギーを起こすような物はやるなということによって解決すれば犠牲は出なくなるのです。

いろいろ細かな分布の隅の方を議論してもナンセンスだと思うのです。統計的に議論できることは、平均的な人間の近傍だけでの話であって、それからずれている人間はいったいどういう人間だということをむしろやるべきなのです。それをやらないで、統計的に処理しようというのはおかしいと思ふのです。

**司会** 議論が多く出て興味がつきないのですが、時間が参りましたので、このへんで打ち切らせていただきます。この問題については、さらに企画を練って続けたいと思ひます。

発癌性試験の評価に関する統計学的検討<sup>1</sup>

Statistical Issues  
in Interpretation  
of Chronic Bioassay Tests  
for Carcinogenicity

John J. Gart,<sup>2</sup> Kenneth C. Chu,<sup>3</sup> and Robert E. Tarone<sup>2,4</sup>

山岡秀明（住友化学） 訳

要約 -- 発癌性試験の評価には、データの適切な記録が必要である。各動物の個体別記録("case history")は、各動物各臓器の病理学的データを寿命の長さに関連づける。この情報は、生存情報を考慮した腫瘍発生率の評価に用いることができる。ヒストリカルコントロールの役割について論じ、腫瘍発現パターンの評価に対する、いろいろな分野からのアプローチにおける有意差検定の役割について提言した。連続(定量)値の評価に有効な多重比較の方法は、これらの研究で使われる計数データの解析には適していない。これらの検討の考え方と手法は、クロロホルムについての動物実験のひとつに適用されている。

-- J Natl Cancer Inst 62: 957-974, 1979.

物質の発癌性試験とは、化合物の発癌性の可能性を調べる長期間の動物実験のことである。データの統計学的解析は、NCIの発癌性試験手順(1)のいくつかを理解することから始まる。

発癌性の可能性を調べる典型的な慢性試験においては、被験物質は、マウスでは、雌雄に18-20カ月、ラットでは各性別ごとに20-24カ月間連続的に

---

略号: MTD = 最大耐量; NCI = 米国国立癌研究所; TBA = 担癌動物

投与される。短期の亜慢性試験をもとに、使用動物種の性別ごとにMTDを決める(1)。一般に、被験物質は、2つの投与群に投与される。1つは、MTDである高用量であり、もう一方は、MTDの1/2である低用量である。同時に走らせる対照群は、同じ期間中観察される。投与に媒体を使う場合は、溶媒対照が設定される。理想的には、実験動物は、群間で無作為化され、普通は、試験開始時平均体重が、群間でほぼ等しくなるようにランダム化される。試験中死亡した動物は、完全に自己融解していない限り病理解剖が行われる。死亡は、自然死か事故死かに識別される。しかし、普通それ以上の詳しい死因は必要ない。瀕死動物を故意に屠殺した場合は、自然死とみなす。すべての生存動物は、予定された処置終了時に屠殺及び剖検される。普通、定期的に動物を殺していくことはしない(例：試験終了以前に動物を予定を立てて殺すことはない)。

いくつかの組織・臓器部位(表1)から組織が採取され、病理学者によって検査される。彼らは、顕微鏡で調べ、病理組織学的な診断(表2)をくだす。統計学的解析がよりどころとする基本的な実験情報("end points")は、死亡までの期間、及びいろいろな部位について腫瘍があるかないか、ということである。発癌性試験の評価は、マルチ学間的な作業である。実験動物の、投与物質に対する反応の評価において、病理学者は、採取した組織の異常変化の範囲、及び腫瘍の類型や広がり具合に注意しなければならない。そのような複雑なパターンを病理学者が評価することにより、採取した組織において、特別な組織学的類型を持つ腫瘍などといったしばしばよりすぐれた実験情報("end points")が得られることがある。そのようなたくさんの実験情報("end points")に対し、潜在的発癌性の実験的証拠の強さが、この論文で取り上げる統計学的手法によって数量化できるのである。よりこみいった反応パターンは、簡単には数量化できないであろう。しかし、それらは、発癌性試験の評価において、病理学者の専門的意見とともに、考慮すべきである。

## データの記録

実験の単位は、個々の動物である。各動物は、コード番号で識別され、種、性

別、実験群（対照群、低用量群、又は高用量群）によって、分類される。動物が、実験に使われて死ぬまでの期間もまた週単位で個々に記録される。死亡は、自然又は事故による途中死亡か、又は、最終屠殺のどちらかである。それぞれの動物について、どれだけの病理学的検査がされたのか報告される。これにより、表1に挙げられている中の、どの臓器が病理組織学的に調べられたか知ることができる。時々、自己融解のため動物の剖検がなされないことがあり、又たまたま、肉眼的剖検のみが行われることもある。部分的に自己融解した動物については、ある特定の臓器（普通、肝臓）についてのみ病理組織学的検査をすることもあるし、そうでない場合として、上皮小体のような小器官についてのみ病理組織学的検査が欠落している場合もある。剖検された各動物について、調べた各臓器に対し、腫瘍病変とその他の病理学的所見が記録される。この情報はすべて動物番号に結びつけられ、各動物の個体別記録が作られる。

時には、動物個体は、統計学的な目的のための実験単位としてはふさわしくないこともある。実験動物に子宮内投与した場合、同産群が適当な単位であろう。

（（2）及び色々な統計的手法が提案されている引用文献を見よ）。ケージ内でデータが似かよう可能性も考慮するべきである。大規模なマウスにおける皮膚塗布試験において、Gart (3) はケージの影響についてほとんど何も見い出せなかった。

また、生存動物は、試験を通じて定期的に体重を測定される。個々の体重はあまり有用ではないが、試験の早い時期については特に、群平均体重によって群間における体重差に関する何らかの知見が得られる。

## 生存分析

生存に対する処置の影響を評価するためには、生命表法が用いられる。事故死又は偶発的な動物の欠落（どちらもまれだが）は、観察中断("censored observations")として週単位で記録される。自然死は、"uncensored observations"として記録される。生存曲線は、Kaplan-Meierのプロダクトリミット推定量によって求められる。近似的な標準誤差は、死亡が発生する各時点ごとに計

算することができる。詳細と計算例は、(4)に述べられている。

生存曲線間で有意差検定ができる。Cox (5) [(6)も見よ]によって示された検定は、2つ以上の生存曲線の同等性を調べるのに用いられる。生存率の用量相関性を調べるTaroneの検定、つまりトレンドテストも、死亡率が、直線的に対照群から低用量群さらに高用量群にわたり増加するかどうかを明らかにするために用いられる。又それに関連した、直線性からの隔たりを調べる検定も行われる。この検定は、例えば低用量群の死亡率が、高用量や対照群のものよりも高くはないかどうか、及びその他の線型傾向からの逸脱を検出するものである。そのトレンドテストは、例えば、処置によって生存率が増加するといったような、死亡率と用量との逆の相関も検出することができる。

生存曲線は、生存動物の平均体重のみから得られる平均体重曲線の適切な評価になくしてはならない。途中死亡が生じた場合、それをはさむ平均体重曲線上の2時点の体重測定のうち、後の方は、前の方で生存していた動物の一部しか測らないことになる。ひとつの体重曲線上の2時点の比較は、その2時点間で生存曲線が水平でなければ(この期間中には、事故死や自然死がないことを当然仮定している。)適当ではない。この注意は、群間の平均体重曲線の比較に対しより強くあてはまる。普通その比較は、どの群もまだ全然またはほとんど死んでいない試験の初期の週に限るべきである。その期間では、処置群における平均体重の増加抑制は、処置群がMTDかそれに近いところで試験されているということのひとつの証拠となる。以上のことから、個別に体重が記録され、経時的に連結されているなら、体重曲線はより簡単に分析及び評価できるということがわかる。

## 腫瘍発生頻度の分析

群、及び臓器部位ごとに、その部位を調べられた動物数、または剖検された動物数に対する、1次腫瘍("primary tumor")を持った動物数の比率が算出される。表1は、各部位について、母数として組織学的に検査した標本数( $t_e$ )を用いるのか、また剖検した動物数( $n_e$ )を用いるのかが示してある。この比率を、頻度又は未修正比率と呼ぶ。この比率が「未修正」であるというのは、それを計



算する際、生存の情報を無視しているからである。生存情報を加味した統計解析は、時間依存分析、生命表分析、ないし修正された分析として言及されている。

## 未修正分析

特定の部位における腫瘍発生頻度を、各投与群ごとに对照群と比較するために Fisher-Irwin の直接検定が用いられる。この順列("permutation")検定により、そのような比較に対し正確な片側確率("one-tailed")の P 値が導かれる。腫瘍の発生頻度が、自然発生率が低い 2 項変数と見なされる場合 (8) は、ほとんどの部位について、その片側確率("one-tailed")検定を用いるのがより適している。もし 2 つ以上の検定がなされるなら (各投与群について 1 回)、解析には検定の多重性を考慮に入れなければならない。したがって、群数  $r$  に対して (一般に  $r$  は 2) Bonferroni の修正 (9) では、計算した P 値に  $r$  をかけて、総危険率がこの臓器内腫瘍の比較において保たれるようにしている。総危険率の  $\alpha$  に対し、 $P$  が  $\alpha/r$  (一般に  $\alpha/2$ ) 以下の時のみに有意差ありとされる。

对照群と個別に 1 対 1 群間比較をすることの他に、すべての群を含んだ比較もある。このトレンドテストは、对照群から低用量群さらに高用量群にわたって、腫瘍発生率が増加しているか、減少しているかを検出する。この検定は、Fisher-Irwin 検定の直接的な一般化 (10) として、もしくは概算的な検定 (11、12) として用いられる。その概算的な検定は、線型傾向からの偏差に対するおおまかな検定と組み合わせられていて、そこでは両側確率の P 値が扱われる。これによって、例えば低用量群は、高用量群や对照群に比べて腫瘍発生率が高いという可能性がわかることもある。

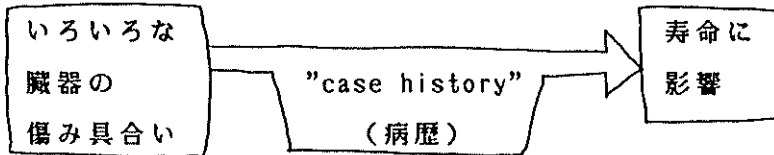
注意しないといけないのは、これらの統計検定、及び以降に述べる検定はことごとく、腫瘍産生に対して処置が正に相関しているのか負に相関しているのか示す上で対称性があるということである。どちら向きの相関なのか注意することが必要である。

修正されていない腫瘍発生率に対する統計検定を評価するには、各群の生存パタ

ーンを見てみないといけない。生存性と腫瘍発生との関連性にもよるが、発癌性に関係するかなり明確な知見が、未修正分析に基づいている事も有り得る。内包する概念を明確にするために、生存分析のケースを3つの可能性に単純化する。処置と死亡との正の相関(+)、処置と死亡と相関なし(0)、処置と死亡との負の相関(-)、同様に、腫瘍発生が処置と正の相関を持つ(+)、相関を持たない(0)、負の相関を持つ(-)場合がある。もちろんこれらの可能性の間に、その他の細かいケースが生じるかもしれない。また更に、高用量と低用量間で不可解な事象が起きるかもしれない。死亡と腫瘍発生に関して、これら3つの場合に限定すると、表3に示すような9つの可能なケースが得られる。完全と言うわけではないが、これら9つのケースは、少なくとも原則的には、現実的なほとんど全ての場合を包含している。評価を改善することにおける基本的な概念は、動物が生き長らえば長らえる程腫瘍が出やすい、と言うことである。これらの9つのケースにおいては、十分な数の動物が腫瘍を持つまで十分長生きすることを仮定している。つまりたとえば、試験初期に全群の大部分の動物が死んでしまうような感染症(従って死亡との関連性は0)のような状況は考えない。そんな場合は、再試験である。表3の評価は、ほとんどの場合、十分に大きな投与量が使われたことをも仮定している。

解説及び質疑応答 (P.- Page L.- Left line R.- Right line)

- 1) P.957/L. 3 「A "case history" for each animal links the pathology data for each organ of each animal to the length of it's life.」:



- 2) P.957/L. 6 「survival information」: 生死だけでなく生存期間も含む。
- 3) P.975/L.24 「MTD」: 最大耐量 (MTD) は発癌性試験に先立つ90日間の亜慢性毒性試験で決定する。その試験は後続の発癌性試験と同一の、動物種・系統、投与経路・投与方法で行う。MTDは、用いられる動物の各系、各性別ごとにそれぞれ決められねばならない。MTDは発癌性試験において、動物を死亡させず、中毒症状を示さない。また腫瘍以外死因となるような病理学的な所見を示さないそのような量であり、対照群に較べ、体重減少が10%以内にとどまる最大量とする。(文献(1)より)
- 4) P.957/L.35 「be autolyzed」: 死後変化で、Lysosome (細胞の自爆機能有り) が破裂し、細胞が液化すること。
- 5) P.975/R.10 「however, they should be considered in the evaluation of the bioassay with the use of the expertise of the pathologist.」: = 「(より込み入った反応パターンは) 生物学的視点に立った数量的な処理で考えるべきだが、その時には病理学者の専門的意見もちゃんと考慮に入れなといけない。」
- 6) P.958/R.8 「the survival curve」: P.966 図1 参照
- 7) P.959/R.22 「the mean weight curves」: P.970 図5 参照
- 8) P.959/R.21 「The survival curves are essential to the proper interpretation of the mean weight curves, ...」: た例えば1匹だけ体重が少ない動物がいた時、その群平均体重がどんどん下がって行き、それが死んだとたんに群平均体重が著しく上がることもある。これは、死んだということの影響であり、平均体重曲線を評価する上で気をつけるべきである。  
(吉村先生)

9) P.959/R.39 「This discussion makes it apparent that the weight curves would be more easily analyzed and interpreted if the individual weights were recorded and linked over time」: “profile”として解析せよと言うこと。

profileとは、例えば、controlは一定であり、dose levelを上げると落ちてくるような場合、profileとして差があると言える。また、どの辺から差が出てくるか等の評価については、個体について経時的に観測をし、profileが与えられたものとしてきっちり解析をすべきだ。平均値ではなくて、profileを見て行けということ。検定するとしても、profilから得られたデータで平均を検定するということを考慮しなければいけない。

具体的には、検定の多重性を考慮して、profile解析の一環として、平均体重の検定等を行うべきであるということが最近の風潮。

repeated measurment analysis として多重性を考慮した解析をせよということ。実際には、臨床等で推奨されているのは、分散分析をmodificationするやり方である。相関構造がある場合、一旦平均より高い人はずっと高い値をとるというように、時点間の観測値に相関がある。それを考慮した分散分析をやれということ。

さらに最近では、小数例の事象において、正規性がわからず分散も群間でバラバラな場合、ノンパラメトリックな、あるいは、「並べかえ検定」と言う、コンピュータを使った検定をやってはどうかという提案も出はじめている。(大橋先生)

10) P959/R.51 「This ratio is “unadjusted” because in calculating it we have ignored the information on survivorship.」: 表3参照

11) P.960/L.9 「Thus for  $r$  doses ( $r$  is usually 2), the Bonferroni correction (9) requires that the computed  $P$  be multiplied by  $r$  so that an overall error rate will be maintained for this organ-tumor comparison.」: Fisher の確率計算法で片側検定を行う。いくつかのdose level に対し検定をすると、ひとつの検定では有意水準は $\alpha$ であっても、複数個の検定をやれば有意と出やすくなるから、その辺

を調整しなくてはいけない。したがって、検定を行った数ぶんで、 $\alpha$ を割ったもので有意水準を設定しておかないといけない。(吉村先生)

12) P.960/L.29「Note that these statistical tests and later tests are entirely symmetric ...」:これらの検定は、2群で違うかどうかを言うものであり、増えているか減っているかは問題にしていない。同じように有意差ありと言ってしまうと言うことで対称性があるということ。

13) P.960/L.37「Depending on the interplay of survivorship and tumor production, it is not unusual that a fairly definite finding regarding tumorigenicity may be based on the unadjusted analyses .」:未修正分析だけでもしばしば明確に発癌性ありということがわかることもある。

14) 表3:どのタイプに該当するかは、処置と腫瘍との相関及び処置と死亡率との相関を見ることでわかる。死亡と処置に相関がない(ゼロ)場合、修正しなくても正しい評価が得られる。

影響があるかないかということでは、腫瘍発生について過小評価であってもなおかつ差があるのなら、再試験等はしなくて良いが、未修正分析と修正分析で結論が逆転しそうな場合は、補正(修正分析)を行う必要がある。

15) 両側検定か片側検定か?

<質問> Fisher検定は片側で、トレンドは両側検定を行うことになっている。これは多分表3において+、0、-の相関を見るための検定と思うがそれぞれ検出力がちがうのではないか?

<答> これ1回の検定で断定をしようということよりも、傾向をこれで見たいこうという面がかなりあるから、Fisherもトレンドも片側検定でやっていくべきだ。(吉村先生)

4~5 dose levelの場合、片側でやるべきだという主張が、NIHSの人達から出ている。変異原性試験でもしかり。(大橋先生)

有意水準というものは、ある意味ではひとつの目安なわけで、そうした場合に、例えば5%でやるということは、両側であれば10%で行うことに結果として相当する。今のところは片側で決論を出しておく方が、このような

毒性の発見には良いのではないかと言うことだと思う。(吉村先生)

16) profile検定について。

〈質問〉「曲線が何らかのパラメータで固定できるような場合には、群間の比較ができる。」というような話だったと思うが、例えば、直線と見なせる場合、パラメータとしては、傾きと切辺の2つになるが、群間の比較は、パラメータの平均を比較するということか？もっと複雑なものか？

〈答〉仮説をどう設定するかだが、群内でパラメータの値が同じであって、群間ではその値が違うという対立仮説を設定するならば、それでひとつの仮説設定ができるから、統計の一般論で検定のやり方を作ることが出来る。問題は、そのパラメータが群内では同じであって群間では違う可能性があるという仮説を設定していいかどうかということにもなる。場合によっては、同じ群の中でも、パラメータに多少の変動は許すが、その多少の変動に比べると群間の変動はもっと大きいという仮説を設定することも考えられる。これはいわゆる変量模型的な仮説の設定になる。その辺は実情しだいである。(吉村先生)

表 1 - 局所解剖学コードと比率計算用分母

組織、器官	局所解剖学コード	分母*	組織、器官	局所解剖学コード	分母*
外皮系			生殖器系		
皮膚	01,02	ne	精巢上体	791	ne
皮下	03	ne	精管	792	ne
呼吸器系			精索	793	ne
肺と気管支	26,27,28	te	尿道球腺	7517	ne
気管	25	te	(クーパー腺)		
鼻腔	21,22,23,2X12,2X32	ne	外陰	80,8041を除く	ne
喉頭	24,2Y22	ne	陰核龟头	8041	ne
造血器系			腺	81,8X22	ne
骨髓	06	te	子宮	82-86,8660を除く	te
脾臓	07	te	卵巣	87,8660	te
リンパ節	08	te	神経		
血液	0X	なし	神経	X05-X07,X8,X9	te
胸腺	98	te	脳	X100-X187	te
扁桃	61	ne	(X117,X130,X139, X148,X159を除く)		
循環系			X2-X6,X70-X73		
心臓	32-39	te	脊髓	X-74-X79,X130,	ne
血管	40-49	ne	X139,X117,X148, X159,X188,X189		
リンパ管	09	ne	脈絡叢	X19	なし
消化器系			特殊感覚器官		
口腔	51,52,53,54,60,6X12	ne	眼	XX00-XX74	ne
唾液腺	55	te	眼付属器	XX8	ne
肝臓	56	te	涙腺	XX90-XX97	ne
胆嚢・胆管	57,58	ne	ハーダー腺	XX98	ne
膵臓	59	te	耳	XY0,XY1,XY2	ne
食道	62	te	(XY24,XY25を 除く)		
胃	63	te	XY3-XY9		
小腸	64,65,6420を除く	te	耳道腺	XY24	ne
大腸	66,67	te	ジンバル腺	XY25	ne
直腸・肛門	68,69	ne	筋骨格		
泌尿器系			骨	11	ne
腎臓	71	te	関節	12	ne
腎盂	72,7X22	te	骨格筋	13,14	ne
輸尿管	73	ne	軟骨	1500-1510	ne
膀胱	74,7X42	te	椎間	1520-1526	ne
尿道	75,7517を除く	ne	関節半月	1530-1537	ne
内分泌系			滑液包	16	ne
下垂体	91	te	腱	17	ne
松果体	92	ne	靱帯	18	ne
副腎	93	te	体腔		
旁神経節	94,95	ne	胸膜	2Y70-2Y72,29	ne
甲状腺	96	te	Y210-Y222,Y300		
上皮小体	97	te	縦隔	Y230-Y234	ne
ランゲル ハンス島	99	te	心膜	31,3X	ne
生殖器			腹膜	Y4-Y7	ne
乳腺	04	ne	精巢精膜	7840	ne
精巢	7800-7831	te	体腔	Y171	ne
前立腺	7710-7740,7758	te	腸間膜	6420	ne
精嚢	7750-7757	ne			
凝固腺?	7759	ne			
陰莖	7600-7634	ne			
包皮腺	7635	ne			

ne=剖検した動物数を分母とする。

te=組織学的検査をした組織数を分母とする。

表2 - 全身癌、循環系及び造血器系

循環系		造血器系	
形態学コード	癌部位	形態学コード	癌部位
2400-2495		7122	リンパの枯渇
240	先天性心臓血管奇形	7208	リンパ腺症
241	心位置異常	7600-7798	
242	転位	7600-7698	異形成
243	心房中隔欠損	7700-7798	造血組織異常
244	心室中隔欠損	8580-8586	胸腺腫
245	心弁膜奇形	9590-9753	
246-248	先天性血管奇形	9590-9633	リンパ腫
249	その他の心臓血管奇形	9643	細網肉腫
3417	リンパ管拡張症	9653-9683	ホジキン病
3440-3466	動脈瘤、静脈瘤	9693	巨大濾胞性リンパ腫
3481	弁膜閉鎖不全	9703	菌状息肉腫
3700-3722	血栓症および塞栓症	9713-9723	その他の網内系腫瘍
4922-4927		9731-9733	形質細胞腫瘍
4922	巨細胞性動脈炎	9741-9743	肥胖細胞腫
4923	多発性血管炎	9753	パーキット腫瘍
4924	結節性動脈周囲炎	9793	骨髄腫
4927	血管周囲炎	9803-9943	白血病
5200-5261	動脈硬化症		
8710-8716	血管球腫		
9120-9161	血管腫瘍		
9170-9179	リンパ管腫瘍		
解剖学コード	癌部位	解剖学コード	癌部位
0901-0990	リンパ管およびリンパ液	0X00-0X50	血液
2106	鼻海綿叢	0500-0887	
3000-3010	心臓血管系	0500-0550	造血器系およびリンパ系
3200-4968	心臓、血管	0600-0670	骨髄
5624	肝類洞	0700-0710	脾臓
7933	蔓状静脈叢	0800-0887	リンパ節
9520	尾骨小体	6100-6114	扁桃およびアデノイド
9530	頸静脈小体	6118-6135	
		9800-9820	胸腺

表3 - 腫瘍発生率の未修正分析に対する生存分析による評価

結果タイプ	処置に関連した腫瘍	処置に関連した死亡率	腫瘍発生率の未修正分析に対する評価*
A	+	+	未修正分析は処置による腫瘍発生を過小評価するかも知れない。
B	+	0	未修正分析は処置による腫瘍発生を正しく反映する。
C	+	-	処置群に見られる腫瘍は処置群の寿命が長い場合かも知れない。時修正分析が望ましい。
D	-	+	腫瘍が少ないのは処置群の寿命が短い場合かも知れない。時修正分析や、より低用量での再試験が望ましい。
E	-	0	未修正分析は処置の潜在的腫瘍予防力を正しく反映する。
F	-	-	未修正分析は処置の腫瘍予防力を過小評価するかも知れない。
G	0	+	処置群で死亡率が高いため、未修正分析は処置による腫瘍発生を見過ごすかもしれない。時修正分析や、より低用量での再試験が望ましい。
H	0	0	未修正分析は処置が腫瘍と無関係なことを正しく反映する。
I	0	-	処置群の寿命が長い場合、腫瘍予防力が隠されるかも知れない。

\* これらの評価の多くは、MTDが使われたこと、および十分な動物数が使われ、適当な期間を通じ十分な生存率であったことを仮定している。



## “コンピュータシステムの信頼性”に関する文献紹介(2)

LAW S O F  
UNRELIABILITY  
by Tom Gilb

訳：佐野 正樹

医薬安全性研究会会報(No.16, April, 1985)に、GLP規制化のコンピュータシステムと題して、FDAの見解を拙訳にてご紹介しました。また、これに関連した内容について、第24回定例会で各社のQAUの方々のご協力を得て活発な討議がなされました。

結局のところ、安全性試験にコンピュータを使うのは、楽にしかも迅速に実験データを処理したいというだけでなく、処理の信頼性をも同時に高めたいと望むからに違いありません。しかし、コンピュータの使用は、その一方で、データの信頼性に関して多くの新しい問題を生み出します。とくに、各社のQA業務に就かれている方々はこれらの新しい問題に真っ向から直面させられることとなります。

こういう状況を反映してか、最近では、“コンピュータバリデーション”はどうしたらいいか、という質問がわたしのところにくるようになりました。しかし、この件にお答えするには、わたし自身の知識が足りないというだけでなく、いくつかの問題があります。その一つは、“コンピュータバリデーション”が、コンピュータ自身(ハード)の妥当性検査を意味したり、時にコンピュータのプログラムの妥当性検査を意味しており、両方の目的が混同されがちなことです。

そこで、Gilb(ギルブ)の“Laws of Unreliability (信頼性低下の法則)”を拙訳にてご紹介します。コンピュータと信頼性について、いまや古典的といえる(?)観点から総括的な意見が述べられています。著者はシステム管理者向けに著したとおもわれますが、内容自体はそう難しいものではありません。したがって、システムを設計する方のみならず、QA作業をなさっている方にも、“コンピュータの信頼性”を検討する上で、かならず役立つと確信します。

なお、読んでいて、意味が通りにくい文章がありましたら、誤訳しているのかも知れません。また、原文のあちこちに比喩的説明が加えられていたため、すこしだけ余分な訳をしたところがあります。そこで、原著も末尾に縮小して掲載しましたので、これも適宜参照していただければ幸いです。

---

株式会社 生物科学技術研究所 研究部  
〒433 浜松市葵町95-10 (TEL 0534-36-9021)

In data processing, the only  
thing we can rely on is unreliability

L A W S   O F  
U N R E L I A B I L I T Y  
by Tom Gilb

データ処理において、ただひとつ  
われわれが請けあえることは信頼性が低いということである

信頼性低下の法則  
トム ギルブ 著

著者紹介

ギルブ氏はノルウェーでコンサルタント業を営んでいるが、1956年以来ヨーロッパに住んでいるアメリカ市民でもある。

彼はIBMサービス部ソフトウェアサポート課の主任で、しかもIBMノルウェーの主任インストラクターである。また、同時に、ノルウェーコンピュータ協会会長をも務めている。

われわれは職業がら、“コンピュータ科学”あるいは実用的な“ソフトウェア工学”が存在することを実証する必要にせまられる。これらのものが実在することに疑念を表明する人たちは、法則、原理、方法の評価や機器の比較が“科学”と“工学”の特徴であることに注目している。たしかに、これらの存在を明らかにするにあたっては、とてつもなく大きな困難があるわけではないが、ある種の悪循環が働きはじめる。つまり、われわれはひどく忙しく生産だけをしているので、より生産性を高めるために勉強するような余裕も能力も持てないでいる。

わたしは研究者や学者ではないし、自営のお金持ちでもないのだから、決定的な解決策を提供することはできない。しかし、次のいくつかのすばらしい例である——Parkinson（彼の法則は、仕事の量は利用可能な能力の一杯まで増える、というものである）、Peter（人はふさわしくない地位に昇るものだ、という原理で知られている）、そしてTownsend（組織化の秘訣を著した）——を思い起こすと、コンピュータ科学上の問題がいわゆる科学的自然の簡明さほど単純ではないことに、わたしははっとさせられる。この難問は、それがビジネス社会に存在しているので、人々を考えさせること、もっとはっきり言うなら人々を以前と違った風に考えさせようとする点にある。

それゆえ、先ほどの紳士たちの仕事に続いて、いくつかの“法則”をつくりあげてみた。データ処理は初めての人、すぐにはそのポイントがわからない人のために、ひとつひとつの法則に説明を加えておいた。たとえわれわれがそうすべきだと未だに主張しているとしても、これらの人々はさまざまなことを失敗の経験から学ぶべきではない。

## 1. コンピュータは信頼できないが、人はなおいっそう信頼できない。

コンピュータはけっして誤ちをおかさない、それ自身のいかなる誤りも確認できる能力を持つ、という驚くべき作り話が、しばらくの間たいへん強力に世間にもてはやされたので、次のような見出しを新聞が掲げるとはっとさせられる。

### コンピュータ、 106才の老婦人へ小学校の入学通知

今なお“ニュース性の高い”この例外的事件は、コンピュータが普通そこまでおろかな誤りをしでかさなことを、われわれに印象づけるのに役立っている。もし、人間がこの手の間違いをしたのであれば、その場合まったく報道されなかったろう。

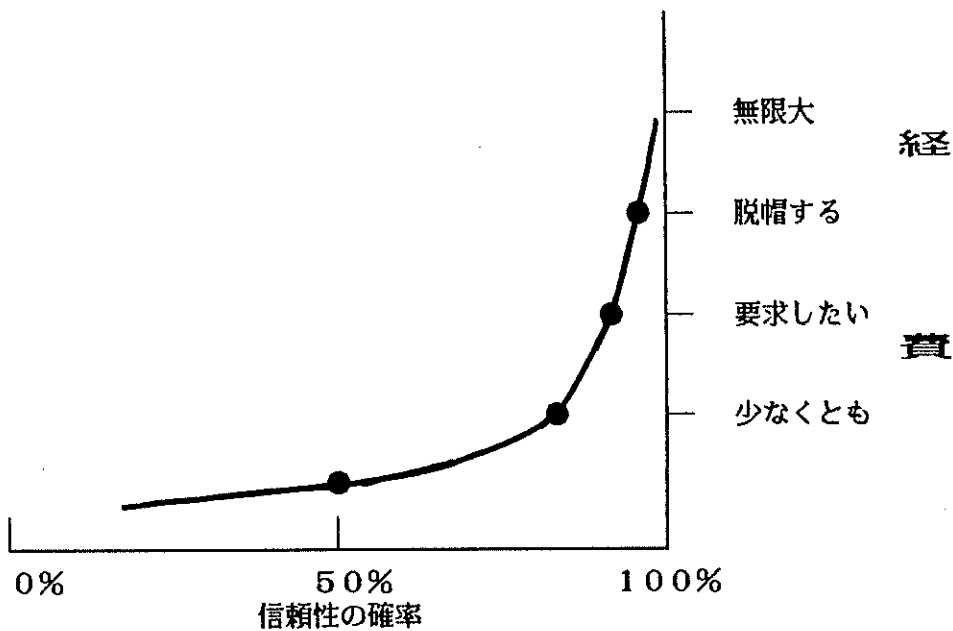
この例の問題の本当の原因、さらに（自分のすむ町がどれだけ進んでいるかにもよるが）わたしたちがすでに経験もしくはすぐにも経験することになるコンピュータがらみのもめごとの殆どは、まったく人間の制御という領域に属している。非難を避けようとして、コンピュータの衣のすそに隠れるのは卑怯である。

この法則の系をひとつ述べると、

コンピュータのせいにされる個々の誤りの原因を調べれば、その誤りをコンピュータのせいにする間違いを含めて、人間の間違いを少なくとも二つ見いださざらう。

もしもコンピュータが本当に“信頼できない”ものなら、これを使う人間は不当な決定をしたのだ。もし、誤りをしでかさす傾向を十分に知っていて、コンピュータを使う決定をしていたのなら、その決定もまた人間の判断である。人間は正直にこう言うべきだ。“その手の誤りは、0.03%で起こるものだ。誤りを回避しようとして納税者に数百万ドル費やさせるよりも、起こるまま納税者に発見させておく方が安上がりであるとわれわれは推定する。”

完全な信頼性の代償は無限大の経費である



完全な信頼性の代償は無限大の経費である

## 2. 人間の信頼性に依存するいかなるシステムも信頼性の低いシステムである。

人間の信頼性はシステムの中でもっとも弱い環であるが、人間の常識は強みとなる点のひとつだろう。不適当なシステムを設計する者とは、いわゆる、人間の信頼性の低さに配慮を欠いてきた者であり、この者は自分の不始末を棚に上げて人間的要素をしばしば非難する。非難の対象がすこしばかり違っている。設計家はシステムの信頼性低下に第一義的な責任を負っており、最悪の場合での問題を予見するのが自分達の仕事なのだ。

心ある者のなかから“理想的”な人を使うようには、システムが設計される筈がない。現実の世界は愚か者で満ちており（あるシステム設計家の意見から）、しかも彼らとともに設計家は生きなければならない。幸いにも、十分に柔軟で、人間の弱点に耐え得る道具が設計家の手元にある。そして、すべてが現実的な経費で実現できるので、この道具を使用しているシステムでは、人間が担う貧弱な信頼性の役割を補償することができる。

やや初原的な姿をとる自動データ処理装置 —たとえばパンチカード式電動作表機、複式簿記用機器、および類似の機器— の数十年に渡る歴史は、“ゴミからはゴミしか生み出せない (Garbage In / Garbage Out)” という観点から考えるようにわれわれを条件づけてきた。このG I G Oは、データ入力する人間を非難する根拠である。また、コンピュータ製造業者、データ処理管理者、システム設計家および自分達の失敗の責任をとろうとしない他の者たちは、何年間にもわたって成功裡にこの用語を使用してきた。

その信頼性を人間の能力に依存するようなシステムは設計されるべきではない。そのような必要はないのである。

## 3. システムを攻撃する愚か者と犯罪者の真の違いは、愚か者は予想もしないときにあちこちから攻撃するという点だけである。

業界紙や一般紙はコンピュータに関連したある種の犯罪的な詐欺の記事であふれている。当然、これは扇情的であり、刺激的である。そして、すべての弁護士業の者にとって非常に興味深いことでもある。

わたし自身に関するかぎり、ノルウェーの税率が現在でも不快なほど高過ぎる水準なのに、さらに引き上げられるかもしれないので、役立つ筈の有益な手法をたくさん学んできた（税率が高過ぎるので、ある調査結果では、市民の5人に4人がチャンスさえあれば喜んでごまかすだろうということである。そうでない人たちは、嫌疑をかけられたくないか、チャンスがくるとは未だかつて考えたこともないか、嘘をついているかのどちらかだろう。）。

とにかく、こんなことがらへ注意を向けるので、真の危機である“愚か者”を見失っているのだとおもう。愚か者たちは刺激的なスパイ物語には関与しないが、彼らに絶えず向けられている関心と比べて、ずっとそれ以上に関心を持たれている現在のコンピュータシステムへはすいぶん多くの愚か者が接触している。たとえば、飛行機事故や船舶事故の防止システムにおいて、あるいは“乗っ取り”の脅迫を除けば、ある術後患者分析システムにおいて、犯罪者が金儲けの種を見いだせるなど、ありそうもないことだ。しかし、これらのシステムが果たす危機的な機能にわたしの生活はすっかり依存している。したがって、わたしに関するかぎりは、強盗に入られた銀行や、少しばかりの“秘密”を入手したスパイやコンピュータの“乗っ取り”よりも、これらのシステムの信頼性の方がずっと気になるのだ。

“愚か者”はシステムに接触できる操作員であるかもしれない、また目下のところわれわれの大多数が確かにそうであるように、下級のシステム開発担当者が“愚か者”であるのかもしれない。ともかく、われわれの生活と家計のやりくりにとって、かれらは目前にせまった危険人物なのである。

この“予想もできない”愚か者の問題に対して十分に注意を払えるならば、そのときは同じやり方で“犯罪的な”問題の90%か、おそらくそれ以上をわれわれは解決するだろう、と想像してみても価値がない。本当は専門家にとっても厄介なことではあるが、もっとも重大な問題（愚か者）を認識するところから初めよう。もし、われわれが始めないのなら、だれか他の人がやるだろう。

#### 4. システムは、信頼性の低下が耐え難くならない限り、簡素化よりも複雑化するように成長しがちである。

短期的には、一たぶん必要ですらあるが一システムを“バッチする”ことにより、不適当なシステム設計を修繕することがたいへん手軽に行われる。本来あるべき姿となるに必要な改変を含むように、そのシステムを“再設計”するためには、動機、機会、あるいは能力がほとんどない。

改変は、このようにして、システムに適当な改良を下回るものになる。それは効率を脅かす。信頼性を脅かす。システムの通用力を脅かすかもしれない。複雑さを増させる。全体的な検証を妨げるであろう。

設計が不十分な、ハードウェアシステムならびにソフトウェアや応用型のシステムは（われわれの誰もが、じぶんたちの経験から、いくつかの事例を確かにもっている）、こうして、それ自身の構造の複雑さと論理の複雑さを成長かつ増大させる。ついには何か大きな音とともに崩れ去るにちがいない確率は、改変の数に応じて幾何級数的に増加する。

不十分な才能と時間しか、システムの改変と検査に使われないという悪循環は、その循環が始まったところから成長しつづけるだろう。

システムが競争的な状況下にある場合は、その競争は自分を妨害している災いを感じ取る。そして、背景（購入者が自分のところの問題を自覚するようになっていることと、問題の存在を認識しようとする者だけがそれを解決できると信じるがゆえに自発的行動をとることが組み合わさっている）を有利にあって、競争は売り場に忍び込み、商売上不都合になるものを盗み出しておいてから、以前と比べても

なお早過ぎるくらいの時期に、そこの支配人に対して次期システムの新バージョンを発表するよう強要するだろう。—このようにして新しい循環が始まる。

もし、システムがユーザの手になる応用システムなら、このユーザーは互換性のない新しいコンピュータシステム（互換性はないが、おなじ製造業者からのもの）を選択するだろう。そして、新しいシステムは、現行の応用対象の完全な再設計と再プログラミングを余儀なくさせ、こうして、つぎつぎに循環を繰り返す機会をそのユーザーに与える。あるいは、彼の後任となった管理者がたぶん次の機会に他人の罪を負うので、前任者は傍らのアラバスクの繰り返し模様（ピーターを参照）に心を痛めることになるだろう。

5. 自己確認システムは、それを使用しているシステムの信頼性が低い分だけ、いっそう複雑になりがちである。

通俗的になるが、製造業者がその信頼性の有様を強調すればするほど、その業者は自分の販売競争以上に信頼性の有様を大切にしたいかということが、わたしにはいっそう疑わしくおもえてしまう。ロールスロイスは自社の自動車の信頼性をけって話題にしなかった。そのようなことは誰の目から見ても全く不要なことであった。

この法則の違った方面への適用： 自己確認用数字を使用するシステムのあるものは、わたしが経験してきたところによると（たとえば、ノルウェー国定識別番号は、ふたつの検査桁からなるが、若干の説明書を読んでみても、なお不可解なものである）、非常に複雑であり、その複雑さは長く抽象的な数字を取り扱うことがいかに危険かを警告している。

この法則のまた別の面への適用： 正確さを“保証する”ために複雑に組み合わせられた一連の構成要素からできあがっているデータ処理システム（発案者、コード化担当者、コード確認者、穿孔係、穿孔照合係、試験走行、修正担当者、修正照合担当者、出力確認者、など）では、これらの過程のうちただひとり（発案者）だけを絶対に必要とするにすぎないので、シラミのわいたシステムにせよ誰かが設計したのだという真実をわからせるには、じっさいにずいぶんと手間が掛かる。

自己確認システムは理想的な到達点である。しかし不幸なことには、あまりにも常識的であるが、これらのシステムはすべて、システム中に当初設計されなかったはずの弱点を補償しようとする行為を反映したものである。このように、この法則はわれわれを用心深くさせることを第一義として設計された。

信頼できるシステムとは、おそらく自己確認システムの部品をほとんど必要としていないか、マーフィの法則（“起きそうなことが、起こるものだ”、など）への防御を自身のシステムに注意深く組み込んでいるものなのであろう。

6. いかなるシステムにおいても、誤りを検出かつ訂正する能力は、それらが取り扱えない誤りを知る鍵となる。

また、たとえばコンピュータ供給業者から、彼らにとって夢のようなシステムにおける信頼性能力を堂々と書き並べたものを見せられた場合、この“法則”はわれわれの注意を喚起し、われわれの危機感を生起させる刺激剤として働くよう企図されている。

われわれは安易にも“薬の力をかりて”安全になった気である筈だ。それでも、誤りが起こる確率は無限に大きいので、誤りを検出かつ訂正する手法のうち、われわれが現在使いこなせるものは、起こるべき誤りの広大な浜辺から拾い上げた100個の砂粒でしかない。

簡単なバリティチェックシステム（たとえば1語当たり1ビット）が機械の誤動作のすべてを“検出”すると信じるコンピュータ関係者の数はおどろくほど多い。この事例では、IBM370の1語（32データビット）当たり約11個のバリテ

ビットのうちの10ビットが無駄になる勘定である。そうして買い手は、不満を持って、この無駄をデータ貯蔵の目的に使用するよう主張するに違いない。

自己確認用の数字やカードの検証過程（二回繰り返し穿孔による重複検証；訳注）を“信じ込んで”いるコンピュータ関係者の数もびっくりするくらい多い。われわれは信頼性の問題をかかえてはいるが、このことだけがわれわれの理解の水準だとすると、それもそのはずとおもえる。（これらの手法は両方ともハードウェアの販売に際して重大な役割をするので、このことをつつて風船みたいに膨らませないようにするのが既得権というものだ。）

ふたたびマーフィの引用：“隠れている欠陥のうちこそ、本当の姿がある”そして“すべてが正しくみえるとき、まちがいをなく何かを見逃している”

7. 検出不能な誤りは、定義づければ制限できる検出可能な誤りとは異なり、さまざまな形をとって数限りない。

上の第6法則の解説はここでの目的には高尚すぎる。

検出可能な誤りは、提示されたシステムで実際に検出されるものではあるが、理論的に検出可能とされる誤りそのものではない（もしあなたがある誤りを定義できたら、ひとつの検出手段を考案できるにちがいない。このようにして定まる誤りはすべて、理論的には検出可能である。）。ある種の誤りが本当の意味で“検出可能”となる以前に、あなたは自分のシステムに検出システムを築こうとして、努力とこれとは別の乏しい資源とを費やす決定をしたことがあるにちがいない。

多様な誤りが実際の目的では制限されない以上、中くらいの規模の現実のシステムにおいてさえ、目一杯に防御するのはまったく現実的ではない。われわれは生きていくかぎり、そのシステムの発展を止めようとはけっしてしないものだ。

このように、実際に必要なこととして、たぶん諸問題の源でもある誤りの原因を識別しなければならないこと、さらにこれを調査するために必要な方策や経費と関連づけて、この原因を評価しなければならないことを認識するようわれわれは強いられている。検出する能力を向上させるために、これ以上の時間と努力を払う価値があるとは考えられないところまで、検出する手段を加えて行くことで、システムの信頼性を改善するための方策の順位表が生み出されるに違いない。

8. 現実のすべてのプログラムは、そうでないと証明されないかぎり、誤りを含んでいる。そして、その証明は不可能である。

ダイクストラの指摘によれば、単純なプログラム群を構成するのは確かに可能であり、同様なやりかたで、それらのプログラムの論理的正当性を“証明”することも可能である。

しかし、彼も認識していたように、プログラムの構成要素数が増すと、“証明”問題の手間は指数関数的に上昇する。このことはただちに次の結果を導く。実際的な利益を求める目的のために作られた“普通の”コンピュータプログラムでさえ、それ自身を証明してみせることは現有の手法では容易でない。近い将来においてさえ、大きいプログラムを実際的な“証明”付きで構成するための方法論をわれわれが理解することはありそうもない。さらに、要求される方法でプログラムを構成する力を得るようになると弁護士たちにわれわれが確信させることはなおいっそうありそうもない。

その上、奇跡的な洞察力、方法と協力がある日訪れたとしても、せいぜい一つのプログラムの“論理的な”正当性を証明したことが注目されるだけに違いない。

そのようなプログラムの“信頼性”を証明できる以前に、われわれには踏み込んでおくべき、有益な足場がある。コンパイル、データ表現、ハードウェアの論理の各々の癖、そして信頼性が、依然打ち勝ちがたい障害物として立ちはだかつていよう。—これらはあまりに強大なので、論理的正当性の証明を成し遂げるのに必要

な努力は、信頼性の他の局面に使われる方がたぶんより有益であろう。実際のところ、十分に構造化されたプログラムではない他のプログラムに向けて、調査範囲を絞り込むためにだけ、正当性の証明は役立つだろう。

これらの問題のすべてが解決される頃には、あなたとわたしは少なくとも退職しているだろう。したがって、当面の実際的な目的のためには、“バグ”がないと証明できないプログラムやシステムとともに生きることには直面しなければならない。この時の頼みの綱は、“検出”・“防止”・“訂正”・“回復”のためにシステムに組み込まれた部品を使って、効果的で強力な防御柵を築くことである。そして、最終的には若干の誤りを“起こるままにしておく”ことが全く安上がりにつくことを許容することである。完全性の経費は無限に大きい。

9. 信頼性への投資は、それが誤りの推定経費を越えるまで増大する。あるいは、なにか有益な仕事をさせようと誰かが主張するまでは。

あるシステムに信頼性を構築するにあたり、ある適度な投資水準が存在する。もし、われわれがこの最適化を行わないなら、定義によりわれわれは、過剰投資が原因で資源を浪費するか、あるいは（信頼性の低いシステムにおける）非常に高い操作経費が原因で資源を浪費するかのいずれかである。どちらかの資源をわれわれは無駄にする。

システムに一連の信頼性部品を導入する際、効果と経費についての工学的力量、すなわち知識、に応じて、この信頼性の水準をわれわれが適正化できるかどうか。

換言すれば、われわれの工学的力量は、力量を計測することと計測力に依存しており、興味ぶかい正確さの水準に関しては、達成された信頼性の水準と、システムの信頼性部品（もしくは信頼性を高めるその他の部品）の変化に基づくその信頼性の水準の変化とに依存している。少なくとも1%の正確さでは、たぶんこれよりずっと良い場合でも、“適正な”信頼性の水準に近づくようにシステムの進捗を効果的に管理する必要があるとおもわれよう。

当然ながら、実際的にはわれわれはそのような工学的力量、知識、あるいは正確さを計測する技術を保有していないことから、物置の戸を当てるのに“ショットガン”をめくら打ちするような方法を使うよう強いられる。われわれはずいぶん単純なのだが、システムのすべての装置やその他の部品を二重化あるいは三重化することによって（三重化は偉大な原理である。というのは、“多数決の投票”システムが、欠陥部分を自動的に試験するのに使用できるからである。）、もっとも重大あるいは有益な応用分野において、われわれは信頼性の最低水準を確実に達成できる（残された分野も待っているに違いないようにおもう）。

ある日、ある人かグループかが、システムの三重化という既得権によらずに、上で記述された工学的道具を開発するのに必要な研究を企画するであろう。

これらが法則である。学術的野趣に溢れる読者には、これらの法則の数量的定式化をお勧めする。この試みは諸法則をより信頼性の高いものにするだろう。

□



In data processing, the only thing we can rely on is unreliability.

# LAWS OF UNRELIABILITY

Our profession has been challenged to prove the existence of "computer science" or even of a pragmatic "software engineering." Those who express doubts that these things exist note that "science" and "engineering" are characterized by laws, principles, measuring methods, and measuring instruments. Certainly there is no insurmountable difficulty in starting to develop these things, but a vicious circle comes into play: we are too busy simply "producing" to have leisure or capacity to spend on learning how to be more productive.

Being neither a research worker nor an academic, or independently wealthy, I can offer no positive solution. However, recalling the excellent examples of Parkinson (whose Law states that the amount of work expands to fill the available capacity), Peter (known for the Principle which states that a person tends to rise to a position of incompetence), and Townsend (who wrote *Up the Organization*), it struck me that the problem in computer science may not be simply of a scientific nature. It may lie—as it does in the business world—in getting people to think, more specifically to think differently.

Therefore I have patterned some "laws" after the works of those gentlemen. I have included an explanation with each law for the few who are new to dp and do not immediately find the point: they should not be forced to learn everything by bitter experience.

even if we still insist upon doing so.

*1. Computers are unreliable, but humans are even more unreliable.*

The frightening myth of the computer's infallibility, of its ability to check itself in any error, was for awhile so powerfully projected on the public that it is with great relief that the newspapers can print headlines such as:

## COMPUTER CALLS 106 YEAR OLD LADY TO FIRST GRADE

Yet even this "newsworthy" exception only serves to impress upon us that computers don't normally make such stupid errors. If a person made this error, then I doubt it would be reported at all.

The very source of this example's problem, and the source of almost all the forms of computer harassment which we have experienced or soon will (depending on how "advanced" one's country is) are completely within the sphere of human control. It is cowardly to hide behind the skirts of a computer in avoiding blame.

A corollary to this Law may be stated:

*At the source of every error which is blamed on the computer, you will find at least two human errors, including the error of blaming it on the computer.*

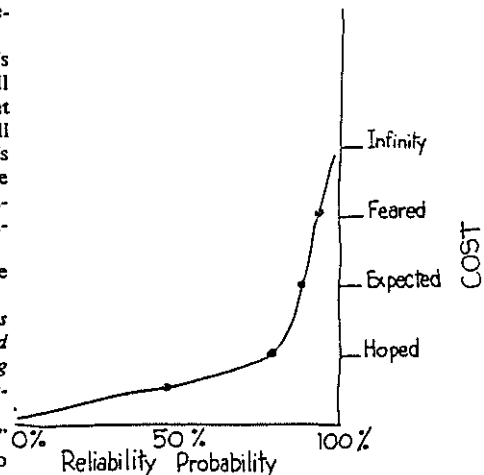
If the computer is "not reliable," then a human made a bad decision to use it. If he made that decision to use it in spite of a known tendency to err,

then that also is a human decision. He should be honest and say: "That error is one of the 0.03% we estimate it is cheaper to allow to occur and be discovered by the taxpayer than to use a few million dollars of those same taxpayers to avoid."

The price of perfect reliability is infinite cost.

*2. Any system which depends on human reliability is an unreliable system.*

Human reliability is the weakest link in a system, although human common sense may be one of the strong points. Those who design inadequate systems, that is, those who have been careless in



The price of perfect reliability is infinite cost.

## LAWS OF UNRELIABILITY

their consideration of human unreliability, often blame the human element for their failure. The assignment of blame is slightly misplaced. The designers have the primary responsibility for any system unreliability; it is their job to foresee worst case problems.

A system cannot be designed with some "ideal" person in mind. The real world is filled with fools (from a system designer's point of view) and the designer must live with them. Fortunately he has, at his fingertips, an instrument which can be so flexible, so tolerant of human weakness—and all at a practical cost—that he can compensate for the weak human reliability part of a system using this instrument.

Decades of tradition with automatic dp devices of more primitive nature—punched card electromechanical tabulators, overdimensioned bookkeeping machines and the like—have conditioned us to think in terms of Garbage In/Garbage Out. GIGO places the blame on human inputters and has been successfully used for years by computer manufacturers, dp managers, system designers and others to avoid taking responsibility for their own faults.

No system should be designed so that it depends on human ability to be reliable. It is not necessary.

*3. The only real difference between the fool and the criminal who attacks a system is that the fool attacks unpredictably and on a broader front.*

The trade press and the popular press is filled with material regarding some form of criminal deception in relation to computers. Naturally this is sensational, exciting—and of great interest to all professional practitioners.

I, for one, have learned a lot of useful techniques which could come in handy if the Norwegian tax rate increases above its present highly uncomfortable level (so high that a poll showed that 4 of 5 citizens would gladly cheat if they had a chance; the others presumably either need to avoid suspicion, don't ever feel they will get a chance, or are lying).

In any case, all this attention seems to me to be missing the real danger: "the fool." Fools don't make for sensational spy stories but there are enough of them in contact with present computer systems to be of considerably more interest than is currently given them. It is, for example, unlikely that criminals can find financial gain in an air traffic or ship traffic protection system, or in a post medical operation patient analysis system except for "hijacking" threats. But since my life literally would depend on the critical functions performed by these systems,

I for one am far more worried about their reliability than about any bank being robbed, or any spy getting a few more "secrets," or any computer "hijacker."

The "fools" may be operational personnel in contact with the system and they may be underqualified system developers, which most of us admittedly are at present. In any case they are a clear and present danger to our lives and to our financial account status.

It is worth noting that if we gave the necessary degree of attention to the problem of the "unpredictable" fool, then we would at the same stroke have solved 90% of the "criminal" problem, perhaps more. Let us begin to recognize the most important problem (the fool) in spite of the fact that it is professionally embarrassing. If we don't, somebody else will.

*4. A system tends to grow in terms of complexity rather than of simplification, until the resulting unreliability becomes intolerable.*

In the short run it is very convenient—perhaps even necessary—to correct inadequate system design by "patching" a system. There is rarely motivation, opportunity or ability to "re-design" the system to include the needed change in an optimal fashion.

The change becomes thus a less-than-optimal addition to the system. It threatens efficiency. It threatens reliability. It may threaten system portability. It adds to the complexity. It may defy systematic verification.

Insufficiently planned hardware systems and software or application systems (surely we all have some examples from our own experience) thus grow and increase their complexity of structure and complexity of logic. The probability that *something* must finally come crashing down with a loud bang increases geometrically with the number of changes.

A vicious circle of insufficient talent and time to make system changes and test them sufficiently will continue to grow from the point at which it began: the initial system planning.

If the system is in a competitive situation, then the competition will recognize the impending disaster and, with the advantage of hindsight (combined with the buyer's growing awareness of his problem and consequent willingness to believe that anybody who will *admit* the problem exists must have solved it), will enter the market, steal an uncomfortable amount of business, and force the market-ruler to release the new versions of his next system even more prematurely than the previous one—thus starting a new cycle.

If the system is a user-application system the user will select an incompatible new computer system (incompatible, but from the same manufacturer) which will force complete redesign and reprogramming of all present applications and thus allow him the opportunity of repeating the cycle. Or perhaps his successor as manager will be the scapegoat in the next round, the previous manager having suffered a lateral arabesque (see Peter).

*5. Self-checking systems tend to have a complexity in proportion to the inherent lack of reliability of the system in which they are used.*

Put more popularly: the more a manufacturer stresses the reliability features, the more suspicious I get that he needed them more than his competitors. Rolls Royce never discussed the reliability of their motor vehicles: it was quite unnecessary from everyone's point of view.

In a different line of application of this law: the complexity of some of the self-checking number systems which I have seen (the Norwegian national identity number with its two check digits is incomprehensible even after several written explanations, for example), is a clear warning of the dangers of working with long abstract numbers.

Looking at still another application of this law: a data processing system which consists of a complex series of organizational units to "assure" accuracy (initiator, coder, checker of code, puncher, verifier, test run, corrector, correction verifier, output checker, etc.) is really just an indirect way of demonstrating the fact that somebody designed a lousy system since only one (initiator) of those processes should actually be necessary.

Self-checking systems are a desirable goal. The unfortunate fact, however, is that they all too commonly are reflections of an attempt to compensate for weaknesses which should never have been designed into the system in the first place. Thus this law is primarily designed to make us wary.

A reliable system would probably have little need for self-checking system components but would carefully include them in defense of Murphy's Laws ("If anything can happen, it will," etc.)

*6. The error-detection and correction capabilities of any system will serve as the key to understanding the type of error which they can not handle.*

Again, this "law" is intended to warn us, to serve as a stimulator which will arouse our critical senses when presented by, for example, computer

## LAWS OF UNRELIABILITY

suppliers with an imposing list of reliability capabilities in their dream system.

We could easily be "drugged" into a sense of false security. Yet the possibility of error is so infinite that all the error detection and correction techniques which we presently master are but a hundred sand grains removed from a vast beach of possible errors.

The number of computer people who believe that simple parity checking systems (one bit per word for example) "detected" all machine errors is surprisingly high. In that case, 10 of the approximately 11 parity bits per IBM 370 word (32 data bits) are wasted, and the purchaser should complain and insist on using them for data storage purposes.

The number of computer people who "believe" in the self-checking number or the card verification process is also shockingly high. No wonder we have reliability problems if that is the level of our understanding. (Both these techniques have contributed heavily to hardware sales so there is a vested interest in not pricking that balloon.)

To quote Murphy again: "Nature is on the side of the hidden flaw" and "If everything seems correct, then you must have overlooked something."

**7. Undetectable errors are infinite in variety, in contrast to detectable errors, which by definition are limited.**

The comments on Law 6, above, are highly applicable here too.

Detectable errors are those which are *actually* detectable in a given system, *not* those errors which are *theoretically* detectable. (If you can define an error you should be able to devise a means of detection; thus all errors are *theoretically* detectable). You must have made a decision to spend effort and other scarce resources on building detection systems into your system before a class of errors is really "detectable."

Since the variety of errors is limitless for all practical purposes, even in moderate size real systems, then full protection would never be practical. We would never actually finish developing such a system in our lifetime.

Thus we are forced to recognize that as a practical necessity we must identify the error causes most likely to create problems and weigh them in relation to the alternatives and costs for detecting them. This should produce a priority list of ways of improving system reliability by adding detection elements up to some point where it is not considered worth more time and effort to improve our ability to detect.

**8. All real programs contain errors**

*until proved otherwise—which is impossible.*

As Dijkstra points out, it is certainly possible to construct simple programs in such a way that it is possible to "prove" the *logical* correctness of those programs.

But as he also recognizes, the scale of the "proof" problem rises exponentially with the number of components of the program. This rapidly leads to the result that a "normal" computer program for practical interesting purposes does not easily lend itself to such proofs with present techniques. Nor is it likely in the near future that we will understand the methodology for constructing large programs capable of a practical "proof," and even more unlikely is it that we will convince practitioners to take the energy to *construct* programs in the required manner.

And even if these miracles of insight, method and cooperation should one day come to pass, then it should be noted that we will at best have proven the "logical" correctness of a program.

We still have several interesting stages to go before we are able to prove the "reliability" of that program. The quirks of compilers, data representations, hardware logic, and reliability still would stand as formidable hurdles—so formidable that the effort needed to achieve logical correctness proofs would likely be more profitably spent in other areas of reliability. Correctness proofs would in fact only serve to limit our field of inquiry to these other less well structured problems.

By the time all these problems are solved, you and I will be retired, at least. Thus for immediate practical purposes we must face living with programs and systems which cannot be proved to be without "bugs." The only defense then is to build a sufficiently strong barricade of such things as "detection," "prevention," "correction," and "recovery" components into the system, and finally, to admit that some errors are simply cheaper to "let happen." Perfection costs, infinitely.

**9. Investment in reliability will increase until it exceeds the probable cost of errors, or until somebody insists on getting some useful work done.**

There is an optimum level of investment in building reliability into a system. If we do not achieve this optimum, we are by definition either wasting resources due to overinvestment, or wasting resources due to unnecessarily high operational costs (unreliable systems). Either way we lose.

Whether or not we can achieve this optimum level depends on our engi-

neering ability, that is, our knowledge of the effect and cost of introducing a particular set of reliability components into the system.

Our engineering ability in turn is dependent upon our measuring ability, our ability to measure, with an interesting level of accuracy, the level of reliability achieved and the change in that reliability level due to changes in system reliability components (or any other component for that matter). An accuracy of at least 1%, and probably far better, would seem to be necessary to effectively control system progress near the level of "optimum" reliability.

Naturally, since we have practically no such engineering ability, knowledge, or measuring accuracy, we are forced to use the "shotgun" method of hitting the barn door. We are highly unsophisticated but by duplication or triplication of all equipment and other parts of the system (triplication is a great principle since "voting" systems can be used to determine the wrong thing to do automatically) we can achieve certain minimum levels of reliability in the most vital or profitable applications (the rest must wait, it seems).

Some day some person or group without a vested interest in triplication of systems will undertake the research needed to develop the above mentioned engineering tools. The result of course will be a significant expansion of the possible market for data products.

Those are the laws. Academically inclined readers are encouraged to attempt quantitative formulations of them. This will make the laws even more believable. □



Mr. Gilb is an independent consultant in Norway, an American citizen who has lived in Europe since 1956. He has been chief of software support at an IBM service bureau and chief instructor at IBM Norway. He has also served as the president of the Norwegian Computer Society.

☆☆☆☆☆☆ 第 28 回定例会出席者名簿 ☆☆☆☆☆☆☆

日時：1986年10月11日(土) 11:00~12:00 基礎講座

場所：総評会館

(途中昼食)

13:00~17:00 定例会

御出席頂いた先生方

- |                   |                   |
|-------------------|-------------------|
| * 大橋靖雄 (東京大学医学部)  | * 柳本武美 (統計数理研究所)  |
| * 高橋道人 (国立衛生試験所)  | * 吉村 功 (名古屋大学工学部) |
| * 塩崎裕通 (日本ロシユ研究所) |                   |

- |                       |                        |
|-----------------------|------------------------|
| 1 阿部俊一 (ミドリ十字)        | 42 滝沢 毅 (日本ロシユ)        |
| 2 大塚芳正 (持田製薬)         | 43 井野裕子 ( " )          |
| 3 田中 健 (日本科学技術研修所)    | 44 相馬義徳 ( " )          |
| 4 堀江成光 (参天製薬)         | 45 内田 ( " )            |
| 5 福木 充 ( " )          | 46 北山英太 (日本新薬)         |
| 6 佐野正樹 (生物科学技術研究所)    | 47 河村 寿 (ブリistolマイヤーズ) |
| 7 林 真 (国立衛生試験所)       | 48 今溝 裕 (東洋醸造)         |
| 8 中嶋康彦 (キッコーマン)       | 49 安田 (住友製薬)           |
| 9 渡辺敏彦 (科研製薬)         | 50 半田 淳 (日本化薬)         |
| 10 五島滋喜 (大鵬薬品工業)      | 51 岩間正弘 (藤永製薬)         |
| 11 聶城 豊 (ライオン)        | 52 植村 (ビーチャム薬品)        |
| 12 関 康弘 ( " )         | 53 佐藤喬俊 (サンド薬品)        |
| 13 秦 正弘 (鳥居薬品)        | 54 池田正己 (日本ハイボックス)     |
| 14 下井信夫 (ユックムス)       | 55 瀬川美秀                |
| 15 長谷文雄 (グレラン製薬)      | 56 今井節夫 (動物繁殖研究所)      |
| 16 倉石忠幸 (エスエス製薬)      | 57 小池 敏 (アツプジョン)       |
| 17 白橋賢二 (エッセクス日本)     | 58 小杉典子 (日本シェーリング)     |
| 18 樋口史郎 (わかもと製薬)      | 59 桑山典之 (帝国臓器製薬)       |
| 19 山岡秀明 (住友化学工業)      | 60 大林繁夫 (グレラン製薬)       |
| 20 岡 基 (家畜衛生試験場)      | 61 茂田哲哉 (日本レダリー)       |
| 21 野田 勉 (大阪市立環境科学研究所) | 62 宮入加寿也 (サントリー)       |
| 22 沢田隆博 (信州動物実験センター)  | 63 荻原孝一 (セローノジャパン)     |
| 23 兼沢 敦 (湧永製薬)        | 64 明光英樹 (日本科学技術研修所)    |
| 24 出来俊昭 (シーエスケー)      | 65 高木 悟 (ヘキストジャパン)     |
| 25 大滝 清 ( " )         | 66 松下 博 (雪印乳業)         |
| 26 徳富 淳 (協和発酵工業)      | 67 栗山敏治 (東洋醸造)         |
| 27 山下哲司 (ロート製薬)       | 68 河上喜之 (実中研)          |
| 28 坂本達哉 ( " )         | 69 岡崎正光 (大塚製薬)         |
| 29 大窪 (東菱薬品工業)        | 70 石原敦信 (小野薬品工業)       |
| 30 国兼 ( " )           | 71 早坂郁夫 (三和化学研究所)      |
| 31 田中 ( " )           | 72 酒井芳紀 (小野薬品工業)       |
| 32 佐藤七平 (日本実験医学研究所)   | 73 塚田良雄 (ヘキストジャパン)     |
| 33 中村良治 (アツプジョン)      | 74 沢井保子 (明治乳業)         |
| 34 横井義之 (鐘紡)          | 75 森岡 浩 (ヘキストジャパン)     |
| 35 船山宣夫 (堀内伊太郎商店)     | 76 奥村紘二 (ヒューマンライフ)     |
| 36 渡辺雄二 (富士生物科学研究所)   | 77 " ( " )             |
| 37 福田武司 (日本生物化学センター)  | 78 菅井象一郎 (クミアイ化学工業)    |
| 38 永見俊之 (日本農薬)        | 79 小林 (鳥居薬品)           |
| 39 金子泰久 (アツプジョン)      | 80 鈴木 稔 (帝国臓器製薬)       |
| 40 高橋昌三 (日本チバガイギー)    | 81 笠原義典 (帝人)           |
| 41 桑田雅彦 (大鵬薬品工業)      | 81 (津村順天堂)             |
|                       | 82 (持田製薬)              |

〔事務局だより〕

1987年も、定例会の準備と、『毒性・薬効データの統計解析』の制作進行であわただしく始まりました。1月末の刊行を目示して夢中になって原稿を整理し、初稿に目を通しているのですが、多数の会員の共同制作という性格上、内容をめぐっての議論も多く、もうしばらく時間を貸して下さい。

しかし、私自身、全くの素人ながら原稿 初稿と読んでいくにつれ、動物実験における統計解析の仕組みがようやく呑み込めてきました。と同時に生きた事例に対し、統計というメスが入り、徐々にデータの評価が明らかにされていく記述は、読む者に知的興奮を覚えさせずにはいないと思います。発行者が言うのもおかしいのですが、オススメの一書であります。刊行の際は是非、ご一読下さい。

＊ ＊ 第30回定例会のお知らせ ＊ ＊

日時：1987年4月18日（土）

場所：総評会館

医薬安全性研究会 会報No. 22

昭和62年1月31日発行

編集・発行 醫サイエンティスト社

画101 東京都千代田区神田峻河台3-2

山崎ビル 電03(253)8992

振替 東京8-71335

印刷・製本 ナガノ印刷

1987©