2. Scope of Traps, Trap-handlers

2.1 Simple example of various scopes

Later we discuss traps.

For discussion scope of IP setting, there we concentrate on the simpler case of mode flags.

Langhans controle shows: "false exception is like an invisible "goof."

Furthermore, exceptions cannot be easily fit in with ordinary

dynamically bound variable, or in some other way.

We don't necessarily agree on how settings of flags should persist. Like a lexically.

Because the setting of modes and flags seems to be at least in some cases orthogonal to


1. Prerequisites

Jump right in.

We assume that the audience is familiar with the IEEE 754 standard for binary floating-point

abstract

August 10, 1998

University of California at Berkeley

Computer Science Division, EECS Dept.

Richard Pateman

Course

Compilers/Languages: Contributions to the FP98
Making sense of scope

Redflag suggestion: rounding, and then [gloar] a little fiddle up or down as you seem to have requested (static, default, post-round) if I might just execute the sine routine with default (DEQ A [plain] compiler might do this since the rounding mode by default is part of the dp-integer, use of an optional compromise that says to generate FP code that looks in no event). set to whatever was in force at compile time (probably the not necessarily round-to-

Don't: compile-time). if I might just execute the sine routine with the rounding mode

Or

necessary smaller

to round, and the result would likely be some number quite close to 0.84147, but not 0.84147." I might just execute the sine routine with the rounding mode set to round-

(Static, sticky) if I might just execute the sine routine with the rounding mode normally

Or

made modes should return with modes altered from the caller's settings -

Your system might have

What does your system provide?

Your intention is to obtain an answer like 0.84147 or some other nearly but distinctly

compute sin 0.1 radians to towards infinity?

set rounding mode to towards infinity?

half of a wrong result? your expression what you had in mind (there is a problem) how can

If the language design and implementation does not correspond to what you had in mind

a language specify these settings? (or worse, folks)
... set round-trip mode old-round-trip-mode
    this is compiled to new-round-trip-mode
    (x) (x)
(set round-trip-mode nearest)
(save current mode)
    (def (x x)
call)
define (x (def (x (x x)
(set (round-trip-mode nearest))
    (def (x (x x)
define (x x)

round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
round-trip-down, (then x round-dp) to round up.
that most methods don't guard against, but strictly speaking is not part of the language.

2.4 Implementation issues

----------

...
2.5 Discussion: Why bother?

More specifically, why would one want a language to support different roundings?

Why bother?
3.1 Optimizations

1.0. Do we need more examples?

\[ (y + z) - y \neq x \] is not the same as \[ x \neq x \] as parentheses must be respected. Example: \( (1.0 + 1.0) - 2.0 \)

There are a few of possible changes to code that can be examined for doing violence to

**Simplification of Optimization**

[Link: https://www.researchgate.net/publi...xample.html]

Simplify near \( x = 0 \). This is right accurate to any \( x \) that enclose the points. This looks like a
table of 100 (1000) that enclose the points. This looks like the
Consider plotting the intervals \( [1.0, 2.1] \times [0.001] \) instead. This means drawing the
This is a poor (numerical) description of the function.
will plot something with a collection of random numbers between -1 and 1.
Near \( x = 0 \), a naive plotting program based on sampling at fixed intervals (say every 0.001)

\[ \sin(1/2) \]

Let's return to our computation of \( \sin(1.0) \) or more generally, consider plotting

### An Example

Is justifiable (see also: how much is accuracy worth?)

\[ \text{is a perfect example of how much is accuracy worth?} \]

This is right accurate (may be that your calculations may involve
paradoxical results, hence \( \int \text{exact} \) and \( \int \text{exact} + \int \text{approximate} \) for two of (two)
permitted representations).\n
So far, so slow: the problem of finding a collection of random numbers between -1 and 1.

Get by directed rounding of a single operation on a fixed format: this may not seem so bad.

The difference of two floating point doubles the width of the interval that you'd

A real trade to some assurance that the result is not cloudy bones.

Cayley's theorem: two numbers can produce results with guaranteed (but not high error bounds).

Theorem: transcendental functions Newonian iteration-based root-finding etc. are possible.

A large body of literature has emerged on numerical computing with real numbers, where
3.1.1 Loops

Other ways that "constant expressions" could change:

\[
\text{store of loop result} = q
\]

++\(t\);
\[
\text{for (t=0; t<n; t++)}
\]
\[
\text{a and c are loop invariant}
\]

++\(t\);
\[
\text{rest of loop result}
\]
\[
\text{only perform one advance} //
\]
\[
\text{for (un=0; un<n; un++)}
\]
\[
\text{a and c are loop invariant}
\]

\[
\text{nested loop into}
\]

\[
\{<\text{loop result}>\}
\]
\[
\text{a/c = q}
\]
\[
\text{for (t=0; t<n; t++)}
\]
\[
\text{a and c are loop invariant}
\]

Use common components in subsequent iterations. For example:

Loop peeling is when you execute the first iteration of a loop "unpeeled" and then cannot be moved out a loop. There's a trick though.

Any expression that could cause an exception but would not necessarily be evaluated, and then move outside the loop, it forces an exception without raising the exception. A moderately defined the 0.0/0.0 is a constant, and hence could be evaluated out of the loop. It is

end

\[
\text{e(x) = compute(x)}
\]
\[
\text{force exception if trap}
\]
\[
\text{if (condition(x) then 0.0/0.0 returns a NaN)}
\]
\[
\text{for x in range}
\]

By comparison:

Moving operations out of a loop may be incorrect, even if they merely replace computations.
It is hard to know what use can be made from arithmetic that behaves as given below.

If you have the time, try linking each of the following rules to its appropriate time. Otherwise, you can use the expression below, which is a good approximation of the time.

\[ t = \frac{1}{10} \cdot \frac{1}{10} \]

Where does it mean to compare two numbers of different precision?

Nevertheless, the result may be meaningful.

The existence of signed zeros means that two memory patterns that are different may be equal or non-equal.

Thus, there are several predicates: Eq, Ne, Lt, Le, Gt, Ge.

If not all, consider books that are in other respects arithmetic.

The kinds of branching and loop exits that are compiled must respect the NRM. Many, if not all, error messages have to be considered.

for any two ordinary numbers. False for NaNs, either a or b. False for NaNs, either a or b.

Similar problems may occur when apparently common subexpressions are eliminated. but

end

**Exercise**: To evaluate the expression "evaluate expression + expression" for 1 from 0 to 4.
Expression evaluation with mixed precisions (etc)

some computation and eliminates some spurious exceptions.

This saves

of a sphere — rather than the infinite rectangular model of the Cartesian plane. This saves

of the complex numbers with one complex infinity (think of complex numbers as the surface

advantages of dishonouring (q.f.) from | are a consequence of preferring a projective model

whose real part is zero. No standard language has pushed this into implementation, but the

WIK has proposed a symbol "i" for the imaginary unit; this is distinct from a complex number

3.2 Treatment of complex numbers

This works quite well for integer operations.

end

for i from 1 to 10 do

z:=z

for j from 1 to 10 do

z:=z+x

end

end

Do computers still change

3.1.4 Strength reduction

But it is part of the Mathematica design, based on a bad idea called significance arithmetic.
More exactly from "Java How" (Kahan/Parley) resolution.

If methods are overloaded, there is a potential for additional scenes to achieve method.

kind of activity already require this

interoperable burden for the compiler, and in fact some languages (Ada) already require this

allows propagation upward and downward in an expression tree. This is not an

preventing what appears to be the most reasonable precision intended by a com-

IS THERE A WAY TO DO THIS RIGHT?

right (1) is usually available only in interpreted languages with a very loose or non-existent

about 3 decimal places.

input, uncertainty even in its decimal digit, but the answer is accurate. You can do this

function may in fact give a higher accuracy answer. To get the precision

right (1) is the advantage of allowing for the possibility that a low precision input to a

which is hardly ever (never) done

The old O rules were (III). This is better than (I), but perhaps wasteful compared to (II)

needed.

value has a type. Compute the appropriate type of each operation at runtime, as

(a) Runtime type-value system. Valuables have no types, but every number

(c) dotProduct(c correctiveProduct(a),

(c) correctiveProduct(b),

(b) correctiveProduct(c)

(v) Require everything that is not obvious to be explicit, e.g. sometimes the

wide...tions (perhaps narrowings) in c.

(III) Regardless of the widths of a, b, and c, e := a + b is done by addressing in the machine's

(II) if and b are of different widths, say a wider than b, c := a + b is done by converting a to

possible rules.

How should precision of operations and intermediate results be determined?

4.1 Precisions
Interprets and standardized semantics for error han-
...
There is also the considerable difficulty faced by the programmer in writing secure code, and the consequent risk of mistakes that will result in unhandled exceptions that can be exploited by attackers. The exception handling mechanism can be exploited if exceptions are handled in a manner that makes it easy for the attacker to manipulate the program state.

A higher-level language view is needed to present a more abstract view of the block in which the exception occurred. Exception handling is a key aspect of object-oriented programming, as it allows the programmer to respond to unexpected events in a controlled manner.

5.1 Object level
discretion in a language.

On some platforms, the T* language in Java, where another more comprehensive view of "con

future. For example, efficient and to be avoided. Just as C++ programmers are told not to use object-oriented

Portion is quite the opposite, with many programmers lauding their subroutines as in-

well favored language. Lisp is well-adapted for this. Other functional languages would do as

well about evaluating code in case an exception occurs. This would probably be rather tricky.

Al complete-time there would some hope that the compiler would emit code for ema

with runtime-mode=zero {execution

<replacement-execution

<execution

with tp-arity=tree=1

{T}M: then running the same piece of code

B. pre-determine the possibility of section have to dispatched or enabled could be done at RUN

{replacement-execution

<replacement-execution

with tp-arity=tree=1

...}

if there could be complete-time instructions:

[[]]

reduce the dimension of handshakes

language, "what do these non-pre-substitution specifications look like?". The tools are

(tc: Scheme continuations)

b) Platform considerations: and it is rather not possible to impose upon all implementations

platform considerations, and it is rather not possible to impose upon all implementations

of the exception model not be mallocable and would certainly involve language compiler, and

a higher level of concern. Direct use of some level of concern is usually available, but can be

omitted. This should be implemented cleanly on each platform.

allow for stating (in some convenient manner), the notion that "should the calculation of 0/0 be

example one may wish to change 0 from a NaN to 1. Temporarily, A language designer could

For many purposes, "pre-substitution" is a plausible strategy. There may be, in some cal-

5.3 Is there a useful model

"default method signature" of Java signatures).
5.4 Stages in Common Lisp Error Handling
design of exception handling should not be left for casual design from
summaries: Semantics of exception handling should differ differently from interpreted code
also in many cases the error in compiled code differs from interpreted code
Among the questionable Lisp pieces, it does in/compiler's Lisp pieces, but in a platform-independent way.
that used here on Sun and HP Lisp systems, but not in a platform-independent way.
set of and identifying these faults can be done by a byte-assembly-language or C programs.
since they are not bound
that restart could be restarted (e.g.,
It is possible to use find-restart to see if a handler exists, and even (restarted) if

(((foo (x) (+ x 1)))
 (restar-case (invoke-restart foo 3)))
) named reques (trivial example):
: (gives 2 options.

(restart-case (compute-this (null restart1) (null restart2)))

(restart-case restart-this)

value? [computed in the context of the restart-case] e.g.,
you want to redefine the function with different results, do you want to return an exception
an opportunity to specify programming from the debugging: e.g.,
any other way to restart? restart-case provides alternatives (in an interpreted system)
that can be extended by the programmer, and handlers can be bound to anything here.

warning:

... floating-point-overflow
... division-by-zero
... arithmetic-error
... simple-error
... error
... serious-condition

The call of object system approach to dealing with conditions places a hierarchy in place.

error occurred)

summary includes whatever is potentially useful. E.g. function name / location in which the
which is given to handlers. No standard for built-in floating-point exceptions (should pre-
6. Speed

Why is 100% is achievable, it is probably a bad idea for numerical results.

Why not 100% identical results?

but perhaps something less on floating-point arithmetic.

Calculations involving floating-point numbers (e.g., division, square root, exponentiation, etc.)

identical results on exact data types (simple, fixed).

identical results with only minor adaptation (most programs continue to work)

A more reasonable approach is to hope for:

Or in the past by full compilation of instruction set exec

Most (e.g., Mandrake, MicroVax, Unix, etc., this is machine architecture).

Rigorous approaches to fully portable code specifically require replacement of all interfaces

and in all implementations — refer to the Java (TM) Language Specification

and in all implementations, Java is non-deterministic. Therefore, the order matter.

Consider:

Remain:

Determine whether a function is written in a standard high-level language does not mean that it will operate 100% identically. In fact, the industry has a consensus bundle of portable

ultimately

6 Portability
6.3 Portable Language Design and Implementation

6.2 Precision

(a) Solution offered by Portable modification of Java to support LPES...
Does this make any more sense than if Microsoft or Java forbid you to use your
beefy DMs to display or a higher resolution printer, or a faster modern

Java, like Microsoft, forbids the majority of us (some 95% of computers on desk-

From John:}

the SNA C SPARC has more floating point registers and larger caches. – J.B}

than the native code on the PPE. The isn’t too surprising when you consider
the SPARC has more floating point registers and larger caches. – J.B

My question is this: does one measurement you may be interested in. I took the

is not associative: a+b+c may not equal a+(b+c)

the same. From two different sequences with the slightly different because of arithmetic +

flops. (Speedups of factors of 10 to 6 are not impressive.) The results (ordinarily, very near-

sizes) same. From two different sequences with the slightly different because of arithmetic +

contrast, the results of matrix multiplication to benefit from different cache organization or

in spite of IDENTICAL arithmetic, different processors or optimizing compilers may seek

(contrast, the results of matrix multiplication to benefit from different cache organization or

particulars to favor certain access ordering to benefit from different cache behavior of

Some computers have MAC (multiply/accumulate or multiply-accumulate)

6.4 Other Specific Portability Problems:


not while interpreting. On only 64-bit values instead of 80

formalism certain days of the week, or only use extended formats after something is printed but

(5) Solution offered by JavaSoft (March 1993) is to allow – not new types – but alternative
double-precision (26 bit) and 32-bit-precision floating point, extended, integer, complex, and imaginary.

before runtime. There are no more suggestions for numeric types for all numbers including

time and precision platform-dependent (computations result is constants are computed
Note: failure to set flags and then setting flags leads to an undefined program.

```
noeflag Error 17 end of transaction unit
```

#pragma STDC FENV_ACCESS on-off-switch
#include <fenv.h>

Additionally:

```
FE_DOWNWARD FE_TONEAREST FE_TOWARDSZERO FE_UPWARD
```

Examines floating names (as macros) for flags in the floating-point environment.

```
C9X provides a type `fenv_t` with the intent to represent the entire floating-point envi-
roment. C9X provides a type `fenv_t` with the intent to represent the entire floating-point envi-
roment. For the bit positions not reserved, mode mapping varies from platform to platform, and there must be names to be used for various flags, types, and actions. Since the bit patterns for registers are names to be used for various flags, types, and actions. Since the bit patterns for registers are
```

## 8 Minimizing: Tomatoes or Tomatoes?

(refer to other papers here)

(too) looks (introduction of post-execution?)

Actually made it into C9X discussion. Clever trap handling and representative debugging

initializing arrays to NaNs (perhaps all 1's instead of 0's or old contents [security risk])

### 7.1 Uninitialized Arrays

To take advantage of IEEE 754 features.

There are numerous opportunities in language design, implementation, and runtime systems

**nooties**

**Language assists for debugging, retroactive diagnosis**

**Summary:** Demanding bit for bit identity of results is too costly for most people. In the absence of a good reason, they should not have to pay for it.
Example. What is the max of (0, NaN)?

For example, a coherent system as possible, formulate an coherent system as possible. Their consequences be traced through the design so as to ensure consistencies of q, t, s, e a consistent set of conceptions that have been used to create.

/*
 * Marzat extensions don’t always exist; that compromises must be raised by a RAISE function.
 */

assert(same-round(x)); // comp is rounding mode inaccessible

if (same-round(x))
    assert(same-round(x));
else
    same-round(x);

if (same-round(x))
    same-round(x);
else
    same-round(x);

//秦国的STDC FENV-ACCESS NO

**example**

9.2 Borneo

shows that rounding modes are set globally, not per thread.

While there are implementations of the language for a number of platforms, they are not


From for many platforms (courtesy of DEC Systems Research Center. More details can be found
Module-2 is a descendant of Algol-60 in Pascal and Module-1.2, with a free implementation

9.1 Module-3

Where to look for help (shoulders, not feet).

We should be standing on the shoulders, not the feet of those who worked on these
portability (expressiveness, generality, efficiency).

If it is not easy to develop a programming language, constraints (size, speed, environment,

apply

Survey of a handful of languages and some philosophies

Incredibly, the Module-3 language words are different...

"unacceptable standards. Community is this way...

and worse that it is considered acceptable in the
requirement of the implementation. And worse that this language standard leaves important details
in developer's hands. It seems understandable that this language standard leaves important details
instead it backs the programmer into the middle of some under-specified territory without
by the ministers as "not backing the compiler/compiler into a corner by prescinding too much."
valid C9X implementations to do anything of note in so many ways was probably viewed
as a usable guide to current implementation of hardware-based algorithms. The acceptability of a
There are too many loose ends for implementation dependencies to make C9X standard per

8.1 More on C9X

Branch cuts for Complex Elem. Funcs, 1987
for example, x+0. If x is x+0, (IM) paper on
Need clear definitions for Elementary function defintions.

There are uses for signed zero arithmetic vs. distinction,
some solution (or program is unsatisfactory) combinatorial problem might not care, except that he might be wrong and the problem is an undue measure of a one of solution of a problem that in the middle of a very well.

Why should we care? It depends on who is mean by "we". In any case it has been done by some hardware designers, one procedure to the logical conclusion as has been done by some hardware designers, one answer, why are we not happy with it?

9.4 Some Philosophy

(Or Java).

The Apple operating system (coming up something) is under the covers of UNIX.


Generally, PPC is the lower precision. Difference in exception handling, round-off precision, precision of the floating-point library.

PPCN does not necessarily do so. When PPCN may need intermediate operands (imagine using wide intermediate).

PPC has double-double but not double-extended PP.

Some differences:

require some attention to programs.

Apple processor architecture (PPC) is different, requires recompilation and may also

SANE is Standard Apple Numeric Environment, built for the Motorola 680x0. The current

9.3 SANE

Recommended reading for a design team.

Specified for a language (example: How precise are constants?)

The bottom definition documented includes implemented other "library" items that must be

Borrowed?

Influence the Java programmers. This raises political and financial concerns. History of Java.

Roland T. Dreyer. Since the existence of any versions of Java are made to be the differences in

There is no implementation for Borland, a variant of Java developed by J. Darcy and W.
Later use of that result. Accuracy decreases by $2^{-(K-\Delta)}$ the incidence of numerical comparison attributable to the effect of $K$ significantly relies in an intermediate result, the presence law of Precision. An increase of $K$ significantly relies in an intermediate result, from excessive inaccurate computation.

Provisions of software and hardware must protect the (relatively speaking) naive users.

- How inaccuracy might be associated with use?
- How (in)accuracy might be measured by subsequent use?
- Where less accuracy is acceptable (context unknown)?

Library

In particular, the numerical software engineer cannot generally judge for users of some packages etc., then one cannot put a cost on inaccuracy. When one is constructing software for re-use by others (libraries, interactive environments, loss is nobody’s gain, etc.)

We as suppliers of supposed quality hardware and software should care.