Submitted to Datamation

UCRL-19782
Preprint

c. 2

THE USER/MANUFACTURER INTERFACE

David F. Stevens

April 21, 1970

AEC Contract No. W-7405-eng-48

LAWRENCE RADIATION LABORATORY
UNIVERSITY of CALIFORNIA BERKELEY

2/

DISCLAIMER

The User/Manufacturer Interface

The relationship between the manufacturer of a computing system and its users was, until recently, a reasonably well-defined, if somewhat non-productive, affair. Unbundling, however, has introduced a measure of uncertainty into that relationship, and the resulting turbulence may provide us with an opportunity to redefine it (the relationship) in a manner profitable to both parties. The purpose of this article is to examine the current state of this uneasy alliance and to suggest how it can be improved.

A parable

That a certain amount of friction has always existed between manufacturer and user is clear from the following story told about the first installation of a large scale system. Since it was, in fact, the first system, it was quite simple. The manufacturer was rather tractable by today's standards: there was only one portion of the system he insisted that the users not diddle -- a particularly critical tree structure. But this, of course, was too much for one of the more subtle graduate students, and he convinced the junior systems programmer (there were only two: it was a small installation) that a small change would surely not be disastrous. The manufacturer discovered the change and closed down the installation, driving out the programmers; also, being rather old-fashioned, he laid a curse on the graduate student: "... dust shalt thou eat all the days of thy life" .... Now this was perhaps over-reaction on the part of the manufacturer, but his attitude is illustrative of current attitudes, and users' behaviour certainly hasn't changed at all.

One result of the long cold-war which manufacturer and user have continued to wage against each other has been a self-reinforcing hardening of attitudes, in which each side has created an uncomplimentary stereotyped image of the other and then, to a certain extent, adopted, as a model for its own behaviour, the stereotype created by the other side.

Our examination of the current state of the interface begins with a consideration of these stereotypes.

The manufacturer as seen by the user

The manufacturer is possessive: the system is his product, and his pride of creation is inordinate. Because it is his creation, he refuses to recognize its shortcomings or failures. Discrepancies between specifications and product are often eliminated by changing the specifications (the product, being created perfect, is beyond change). Also, during those sensitive formative months when the product is developing, it is jealously screened from all contact with the world, lest reality should accidently shape that development. Requests by prospective users for any level of detailed information are promptly and firmly denied.

The manufacturer is myopic: his vision is limited not only in distance, but also in direction (he has tunnel vision). He fails not only to foresee the users' future needs, but also to forsee the uses to which they will put his current products.

The manufacturer is condescending: he knows that he has nothing to learn about computing from the user, and that therefore he is able to create, in a communications vacuum, products of universal utility. What small expertise the user has he acquired from the manufacturer, and it is several years out of date. The user can create nothing of his own worthy of notice by the manufacturer, and those small additions which the manufacturer does adopt are often accepted in the same spirit of amused toleration with which one accepts a mud pie from a four-year-old.... One especially notices that the toleration becomes ever more strained with each succeeding offering.

The manufacturer is overconfident: he overestimates the performance of his products and he underestimates the time it will take him to deliver them. (There are some users who would state this particular case more strongly; they see the manufacturer as misleading: he overstates the

performance of his products and he understates the time it will take him to deliver them. The difference is one of intent. I take the more optimistic view here because if the other is, in fact, the true state of affairs, then there is little hope that any measure of cooperation between user and manufacturer can ever be achieved.) He assumes that they will work as advertised. He assumes that he has solved the problems of the world, and that his products will need little correction, less modification, and no extension.

Finally, the manufacturer is a radical: he subscribes to the Detroit philosophy of planned obsolescence, change for the sake of change. Good programmers are creative people, and he must allow his programmers to express themselves. (That this frequently imposes a distasteful burden upon the users' programmers is unfortunate, of course, but it can't be helped.) Furthermore, it is in his interest to keep the users somewhat off-balance and hence somewhat dependent upon him.

The user as seen by the manufacturer

The user is possessive: the system is his by purchase or lease; possession is nine points of the law. He is unwilling to relinquish one memory cell, one storage cycle, no matter how great the benefit. It must be made to work in accordance with his conception of how a system should work regardless of whether his concepts were designed into it or not. Since the system is his, its past and its future must be his also; his curiosity about these matters is insatiable.

The user is visionary: his estimates of the cost (to the manufacturer) of a "minor" change in direction are unrealistically low; his estimates of the benefits to mankind of the implementation of his pet project are unrealistically high. He ignores side effects.

The user is self-important: his problems are the most important problems in the world, and should be solved first; his solutions are the most promising, and should be tried first. Whatever is in his interest

is in the interest of the whole universe of computing. Features he does not use are barnacles on the ship of progress. His is the one clear voice of truth amid the constant tumult and the shouting.

The user is implacable: he is unforgiving, vindictive, and possessed of a very long memory. The slightest delay is intolerable, the tiniest imperfection is inexcusable, the least hint of change in specifications is totally unacceptable. The user is an expert in the creative interpretation of reference documentation, and unrelenting in his efforts to secure the realization of his inferences. He is no less skillful in the creation of iron-clad contractual commitments out of preliminary, tentative, incomplete, and inaccurate data.

Finally, the user is an archconservative: he resists progress at every level if it involves change. ("The old ways are the good ways.") Technological improvements are complications to be mistrusted: old, familiar, trusted tools are not to be abandoned, no matter how much more potent their replacements are. The game is never worth the candle.

Who is which?

It should be emphasized that these stereotypes have broad applicability: the term "manufacturer" is not limited to "hardware manufacturer" any more than the term "user" is limited to "occasional Fortran or Basic user". In fact, most people involved with computing alternate between these roles, sometimes with bewildering rapidity in the course of a single conversation.

The lines are most firmly drawn, however, and the stereotypes most firmly entrenched, at two levels (both of which exist in almost every shop): the interface between the outside vendor and the systems programmers, and the interface between the systems and applications programmers. One result of this situation might be that systems programmers, having a foot in each camp, serve as conciliatory agents. They might, when dealing with the outside vendor, remember that they, too, have delivered late;

that their own products have been released with live bugs; that their own documentation has been incomprehensible or non-existent. Remembering these things, they might react with sympathy and understanding. They might, when dealing with the applications programmers, remember that they, too, have resisted change; that their own programs have failed to run on new releases; that the documentation they have received has been incomprehensible or non-existent. Remembering these things, they might react with sympathy and understanding.

But of course they do not. They remember, certainly, but instead of acting as filters they act as amplifiers. The abuse hurled at them by the applications programmers is hurled with increased vigor at the vendor; the scorn heaped upon them by the vendor is heaped even higher upon the applications programmers. The process is iterative and with each iteration the stereotypes become ever more firmly entrenched and reality approximates the stereotypes ever more closely.

Changing the image

It is clear from the above that the relationship between user and manufacturer has become a combative one -- highly ritualized, it is true, but combative none the less. It will continue to be combative until the pejorative stereotypes have been replaced by more cooperative models, for the user as well as for the manufacturer. The substitution need not be instantaneous, but it must be undertaken by both parties or it will be ineffective, incomplete, and quite temporary. The models suggested below may not be optimum, but they are steps in the right direction.

The attitude a manufacturer adopts towards his creations should be somewhat akin to parental pride, remembering that good parents recognize and treat illness and injury. Good parents foster the development of their children (and much development takes place outside the home), and recognize that they have limitations beyond which they should not be pushed.

The users, on the other hand, should recognize that children are often recalcitrant, and usually require some training before they are capable of behaving properly away from home.

User and manufacturer need to adopt standards of communication not involving ritual attack and defense. The manufacturer should seek the users' advice before he fixes the specifications of a new product; he should recognize that users have a legitimate need for advance information. Users should remember that advance information is often inaccurate or misleading, and should accept that risk when they accept the information. Dialogue concerning bugs should be undertaken with the object of improving the product, not as exercises in invective (by the user) or skillful evasion of responsibility (by the manufacturer).

Both manufacturer and user need to broaden their perspectives: the manufacturer to recognize that the users harbor a wealth of experience and expertise which could be tapped, if the manufacturer would make the effort; the users to recognize that the manufacturer must be responsive to other users with other priorities. Both user and manufacturer should try to look at things from the other point of view before criticizing.

Finally, user and manufacturer should work together to define and regulate the development of the product. The user should recognize that improvement is rarely achieved without some sacrifice, that progress exacts its toll. The manufacturer should recognize that change must be justified, and that his estimate of the cost of the change should include the cost incurred by the users in adopting it. The user must recognize that some growth (change) is inevitable; the manufacturer must realize that not all progress (change) is desirable. User and manufacturer should agree upon which changes should be provided by the manufacturer and which should be left to the user. The manufacturer must expect the user to make some changes; those areas where changes are likely should be designed to simplify their installation. The user who makes such changes must be willing to accept the non-conformity he has thus created.

In brief, both the user and the manufacturer need to grow up a little. The present struggle is fun, but it is childish. There is much loose talk about the coming-of-age of the computer (which is a far different thing from the coming age-of-the-computer): the computer cannot achieve maturity until those who direct it do.

(This work was done under the auspices of the U.S. Atomic Energy Commission.)

TECHNICAL INFORMATION DIVISION
LAWRENCE RADIATION LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720