Lawrence Berkeley National Laboratory
Recent Work

Title
SOFTWARE IN RESOURCE SHARING: SCIENTIFIC

Permalink
https://escholarship.org/uc/item/8d45823w

Author
Stevens, David F.

Publication Date
1975-03-01
SOFTWARE IN RESOURCE SHARING: SCIENTIFIC

David F. Stevens

March 1975

Prepared for the U. S. Energy Research and Development Administration under Contract W-7405-ENG-48

For Reference

Not to be taken from this room
DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.
SOFTWARE IN RESOURCE SHARING: SCIENTIFIC*

David F. Stevens
Lawrence Berkeley Laboratory
University of California
Berkeley, California

March, 1975

*Work performed under the auspices of the U. S. Energy Research and Development Administration under Contract W-7405-ENG-48
SOFTWARE IN RESOURCE SHARING: SCIENTIFIC
David F. Stevens

Introduction

Since the first half of this session is split along traditional lines into a "scientific" portion and a "commercial" portion, it might be well for me to take a moment or so to discuss my understanding of the nature of that split. At one time the generally-accepted view was that the two domains were nearly disjoint, with "scientific" computing being dominated by computation and "commercial" computing being dominated by I/O. It is now the generally accepted view that that dichotomy, whether or not it was true in the past, is certainly not true today.

What, then, is the difference? I see it not as a difference in the kind of computing done so much as a difference of philosophy; of methodology, perhaps; of temperament. The computer, after all, is a rather flexible tool and can be adapted to many environments. The "commercial" environment (as seen by someone from the outside) is a tightly-controlled one: only a few people are allowed to write programs; a large portion of the work consists of a (relatively) small number of fixed programs run repetitively, usually to a fairly rigid schedule; there is a "right" answer (which can be verified) to each problem; data files are auditable; in short, it is an environment of order and control.

This contrasts fairly directly with the "scientific" environment (as seen from the inside): everybody in the whole world is allowed to write programs; those (relatively) few programs which are run repetitively undergo constant change, usually existing in several distinct
(and essential) variations; it is often not known even whether a problem has a "right" answer, much less what that answer may be; data is often only statistically significant, and loss is acceptable (if it occurs in no more than reasonable quantities); in short, it is an environment of experiment and perhaps even ferment.

(If I may be permitted an editorial observation: It is this basic difference between an environment of control and one of experiment which is responsible for much of the foofaraw caused by the Audit and 0850: the checkers and the "scientific" doers are looking at the world through different kinds of glasses.)

Since I have begun by defining one of my terms, perhaps I'd better continue by defining the rest. "Software" for my purposes is everything but hardware; it includes programming (both systemic and applied, user and supplier, in-house and out-house (and much of the programming one encounters belongs in the latter)); documentation (and the distribution thereof); and people (and their attitudes, although some of the latter resemble hardware in their rigidity....). "Resources", in order to keep this thing somewhat manageable, will be limited to computer systems and access thereunto. (One should consider the other computer-oriented resources: people, techniques, algorithms, programs, knowledge, data; all of them can be shared without sharing computer systems, but mostly in ways which are already well-known, albeit under-utilized.) "Sharing consists of the simultaneous use of resources by several agencies, whether locally resident or geographically scattered. (And that word simultaneous is essential; non-simultaneous use, or block time, is simply a scheduling problem, i.e. political rather than technical, and therefore of no interest to this audience.)

What I will try to address, then, is how the simultaneous usage of
and access to the computer system affect the programming, the documentation, and the people using and maintaining a wide open shop. I will group my remarks in accordance with the given partition of the meaning of "software": first programming, then documentation, and finally people: in increasing order of recalcitrance. I will attempt to consider these things from the points of view of both the user and the supplier. I have never been a user, however, and so I may be a bit astigmatic from time to time; if so, I am sure that some of you will be eager to prescribe some corrections ....

Programming Considerations

The programming considerations are, for the most part, rather well understood. It is sufficient simply to list them; I will be happy to discuss them in more detail during the question period, if desired.

1. Input from multiple sources, output to multiple destinations, with the obvious linkages between them.
2. Reassignment of output from remote to local.
3. Very large, very stable input and output queues.
4. A flexible program storage facility (to minimize the size of job input files).
5. (Desirable) Support for several different RJE disciplines.
6. Extensive job-status enquiry system.
7. (Undesirable but sometimes necessary) Discriminatory scheduling (based upon source of funding for the job).
8. (For suppliers) Support for all possible languages and applications packages.
9. (For users) Avoidance of all machine, system, and implementation dependencies; only the most standard usage of only the most common languages and applications packages.
10. The provision of detailed accounting and billing information. (And here I would like to insert another editorial observation: One of the most prevalent of headaches faced by all federal computer centers is the Tyranny of the CPU Hour. There is a class of auditors which will admit of no other basis for the reporting of computer usage. Perhaps AESOP could undertake the definition of a meaningful measure of system usage to replace the ubiquitous and misleading CPU Hour.)

There is also an intangible which becomes extremely important in the sharing mode, but which, alas, is very poorly understood in the world of computing today: human engineering. The nature of the English language is such that the term "human engineering" admits of two mutually contradictory meanings. One of these meanings has been evident in the vast majority of the computing systems now in use: namely, "the engineering of humans (to make them more understandable to machines)". It is long past time for the other meaning to move into prominence: i.e. "the engineering of systems to make them tolerant of the diversity and perversity of human understanding".

Nowhere is this need more apparent than at a remote site, with nothing but a telephone line for company. Messages are all too often designed to technical restraints (fixed, usually small, length), using technical language, and abbreviated beyond recognition (the ominous ING and OCU of UT200 infamy, for instance, or the IBM 5-digit-numbers-complete-with-codebook). They are read, and acted upon, however, by people who may be only semi-literate with respect to computing, and who have no direct access to expert advice.

The importance of human engineering varies as the square of the perceived distance between the user and the supplier (where the user is doing
the perceiving). All of the strictly technical programming problems associated with resource sharing have been solved many times over; nevertheless many sharing relationships remain extremely frangible today because the human engineering problems have not even been addressed, much less solved.

Documentation Considerations

One way in which the supplier can act to ease the sense of isolation and frustration which besets the user, especially the new user, is via suitable documentation. "Ah yes", you say; "so what's new about that?" Very little, actually. The criteria for suitability of documentation are the same whether the user be remote or local; a conventional approach which is satisfactory for the needs of the local users, however, may founder when applied to a far-flung remote user community because of the distances involved: distances in background and experience, in terminology and jargon, as well as in space. Anything you can do to lessen those distances will be well-received!

1. Catalog (index)
   a. A librarian with an eidetic memory is no substitute for a catalog
   b. Since a catalog is a document (or several) it must satisfy all the criteria listed below; in a dynamic environment it is absolutely essential that the catalog be updated regularly and frequently.
   c. Instant availability; ease of reference.
   d. Every installation invents its own dialect of the appropriate technical jargon; see that all local variants are indexed.

2. Accessibility
a. If I can't get it when I need it, it might as well never have been written.

b. I don't always know what I'm going to attempt tomorrow; therefore I can't always order tomorrow's needs today.

c. Total absence of red tape.

d. The solution: on-line documentation, printable on demand at every terminal.

3. Completeness

a. Begin at the beginning; most remote users don't have the benefit of your exposure to local tradition.

b. The "leap of faith" is a pre-requisite for a certain level of religious commitment: you have no right to ask it of your users; fill in the details.

c. Truncated documents are more inaccurate than truncated numbers.

d. A missing document is an offense against humanity.

4. Accuracy

a. Someone will believe what you say, however blatant the error.

b. Try all examples; verify all verifiable statements, especially cross references.

c. Differentiate clearly between fact and opinion.

5. Timeliness

a. Last year's manual is less accurate than last year's calendar.

b. Advance notice should arrive before the event.

c. Documentation changes should precede system changes.

6. Clarity

a. Clarity is in the eye of the beholder; some of your beholders view you through a wire, dimly.

b. Ambiguity is the life of poetry; it is the death of
technical prose: if your style admits of two interpretations, your users will find three.

c. Define your terms (especially terms which conflict with or supersede those in common usage at other installations).

d. Match the medium to the message. (Among the possible media are (using LBL jargon) broadcast messages, the dayfile, the billboard, BKYNEWS, the Newsletter, an on-line write-up, a document obtainable from the librarian.)

e. Match the prose to the peruser. Write basic documents in basic terms; more sophisticated works can build upon earlier ones.

f. Make sure that important facts are visible. (Changes are important facts.)

g. Clarity is inversely proportional to prolixity.

Upon casting my eye back over this collection of maxims I realize that there is very little there that is unique to the sharing game. What must be kept in mind is that the remote users depend extremely heavily upon the documentation provided by the supplier: they do not have the option of getting the straight scoop in person. It can be exceedingly frustrating dangling at the end of the wire; accurate, up-to-the-minute documentation, instantaneously available, can help to alleviate that frustration.

People Considerations

There are two aspects to the people question, as there are to many of the questions which have been examined in this meeting. There was a time when I was moderately competent in the second; I have never been competent in the first, and I suppose that's why I tend to deplore its
increasing importance (even though I recognize that very development as a sign of maturity) .... By now most of you have lost the thread of my remarks, and that's probably just as well ....

The two aspects are, of course, the political and the technical. The political aspect is concerned with such questions as

Should we?

What if it works? (A very different question from "what if it fails"!)

What does _______ think about it?

What will (s)he think about it tomorrow?

The technical aspect is much simpler; there is only one question:

Can we make it work?

The answer to the last question is clearly "yes". (And here I offer my last editorial observation: All technical problems with respect to resource sharing between any two sites could be resolved almost immediately in response to a single political imperative: Your next computer acquisition depends upon demonstrated competence, as either user or supplier, in computer resource sharing.) As I indicated before, however, I am not interested in pursuing the political questions; that leaves us with only the single technical question and its implications: i.e.

What are the human implications of a successful effort in resource sharing?

Once again I find it easiest to express myself by means of a few reasonably self-explanatory maxims:

1. Stamp out chauvinism!

Pogo once said "We have met the enemy and he is us"; in this case the enemy is the very concept of "us" in contradistinction to "them". Until the supplier's whole staff is concerned about all of "our users" -- until the user thinks of the
supplier as "our computer center" -- the universe is subject to divisive, and destructive, partitioning.

2. Provide communication channels -- in all directions.
   Tell the users what you have in mind; let them tell you what they want and need. Encourage them to get together. (They may find cooperation rather a strain, inasmuch as they may be competing for the same contracts, but they do have common problems.)

3. Be prepared to spend a lot of time on the telephone.
   Handling complaints by telephone is a rare skill; learn it!

4. You cannot have too many (qualified) consultants.

5. High-class, full-time consultants are as plentiful as unicorns.
   (High-class, part-time consultants, however, are relatively abundant.)

Some Exegetical Remarks In Conclusion

At a conference of this sort one hears too many talks on the same subject; I have just spared you three of them: You have heard neither the talk you expected, nor the one Don Peck had in mind when he asked me to speak, nor the one I intended to write. Neither have you heard anything very startling or original. You have heard a distillation of several years' experience in large-scale resource-sharing: successful large-scale resource-sharing. The implication, I hope, is clear; nevertheless I will make it explicit: There are no hidden (technical) software problems to be encountered in resource sharing in an open shop environment; there are, in fact no significant differences (from the supplier's point of view) at all, except those that naturally arise from a wider customer base. The one thought that I want you to take away from this session is that if you really want to do it, it ain't such a big deal after all.
LEGAL NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.