Laboratory for Computer Science
Computer Systems Research Division

October 26, 1976
Request for Comments No. 127

A position paper

The Integratable Personal Computer as a focus for research

October 26, 1976 13:22
R. Frankston

The concept of a collection of personal computers connected by a network represents a model for studying cooperation among a large number of cooperating, but not coerced, computers. The individual nodes must be general purpose computers on the lines of today's time-sharing or "utility" systems. In contrast with existing systems there is no central control though for many applications the collection may be viewed as a coherent entity.

NOTE!: The purpose of this RFC is to stimulate discussion rather than giving a complete detailed proposal. If you have comments, suggestions or questions either see me personally or send mail to RMF@MIT-MC.

Table of Contents

What is a personal computer?

We can start with a simple model of a personal computer. It is the computer in everyone's home that can be used for miscellaneous chores like collecting mail, composing mail for other's, keeping track of the address and birthdays of friends, keeping the checking account records, the tax records, the bowling scores, recipes, shopping lists, grudge lists, the heating schedule for the house, the record collection and audio system state and other typical tasks in the average American household. Note a little problem here: how does the concept of a personal computer fit in with the corporate matters of a household? This conflict will be discussed in detail when examining the structure of the personal computer.

It makes little difference to the user of the personal computer whether or not the whole computer is physically owned by him (or her or it) as long as there is an interface that can act as an agent to the rest of the computer system. This concept, and others in this paper are discussed in my Master's Thesis, MAC TR-128 which concentrates on the problems of modelling such a market place and in particular, implementing the necessary accounting interfaces. The key concept is that even if a number of different people are sharing the same base computer for their agents, at some point there must be a distribution of authority, i.e. decision making and knowledge, such that one central administrator cannot dictate all policies and constraints. Rather the constraints arise from a practical need to share and communicate.[1]

At some point we must appeal to intuition for our model. An attraction of the personal computer is that we can fall back on our model of people and their agents in business transactions. Of course, there is the associated danger of taking the intuitive model too literally or misapplying it. Caveat Emptor.

The model I have given above strongly implies that I am interested simply in the problem of placing computers in people's homes. Thisi s an important part of the model in that our thinking should be bold enough to cover the concept of millions of these systems. But this is not goig to be achieved immediately. Initial applications may be in the areas of office automation or military command and control. I believe that there is a basic similarity in these applications in their need for distributed authority and shared information.

It is also necessary to realize that I am not including the microwave oven and other microprocessors as personal computers. These appliances are slaves to a more powerful machine, if they are connected at all. The personal compuer is a level in the hierarchy of machines -- there are a number of more primitive devices.

---

[1] An issue I will avoid is that of the politics of a laissez faire market versus a regulated one. It is a valid discussion in this context but currently we simply do not know enough about the marketplace to regulate it.

As a research direction

Practically, the low cost of computer hardware, versus the cost of computer center management and communications costs will encourage individuals to purchase their own computers. From the point of view of a research group this offers the opportunity to choose a relatively simple model for the components of a distributed system.

One might view the research as a return to the basics. Multics has been a research vehicle pulling in two directions. First there has been the question of how can we provided an environment with controlled sharing, system supported storage management and an open-ended user interface. Secondly how can we provide a very efficient means of utilizing very expensive hardware. We can address these questions in the environment of the collection of personal computers (and service bureaus) but can now emphasize the first aspect -- the support of ideal user interfaces. We of course are still faced with some problems of hardware utilization that even a megaword of memory and megahertz communications lines are not going to solve, but hopefully those considerations will not be as draining of our research resources as they have been.

I have argued that the personal computer model is a practical one. We also need to consider how this vehicle suits our needs in solving specific problems. We can examine some of them:

1. We seem to want a system that is robust. This is a fairly general term. One example of robustness that we have addressed is the problem of continuing to provide service in the face of failure of components. At one level this model performs well in that an individual computer can operate when disconnected from the network because it came into the network as a viable entity on its own. This is not, however, a fair solution to the problem of designing a "robust" system because of the basic assumption that there are interdependencies between the nodes. For example some data is kept at remote sites -- the bank keeps its own records.

   What the distributed personal computer model gives us is not a complete solution to the robustness problem, but rather a framework in which to examine it. The question is not whether we can make a system that is robust under all possible attacks. It is whether we can get a clear understanding of the issues involved. For example the user may keep his own copy of his financial records, but only update them monthly to correspond to the bank's interest payments. This is similar to how people manage to know the balance of their accounts, at least approximately, without having to go to the bank each time.

2. One justification for distribution is the economic advantage offered by growth without complete interconnection. A system of personal computers has obvious potential for growth in that individual computers can be added to the network in much the same manner as telephones. The main concern is that adding additional nodes does not require knowledge on the part of all other nodes and that conversely does not increase the burden on a bottlenecking facility.

3. As noted earlier, a force leading to the creation of personal computers is not only the requirement of distribution of authority but its inevitability, for both economic and personal reasons. The self-motivated growth can be termed distribute initiative and provides a lever to increase the effectiveness of a design framework that can provide for coordinating such systems.

Given that this model may be suitable for addressing problems of interest to CSR, we still must address the issue of why this is a suitable research vehicle for CSR. Before answering this, we can look with some detail at the personal computer and collections of personal systems.

Attributes of the personal computer

Since the theme of this paper is the integration of personal computers we can start by considering the attributes of the personal computer itself. This cannot be done in isolation because the requirements of the community of such systems and the need to share and community dictate to a large degree the necessary structure of such a machine so any analysis must iterate between the two points of view. The concentration here will be on the personal computer as a typical node in a network.

The traditional view of the personal computer has been that of either a pitifully dumb computer whose sole virtue is that one can put one's hands around it and say "mine, all mine!" or else it has been that of an IBM 7094 class machine or even a Lisp machine[1]. But this view is naive from many points of view. The economic arguments that lead to the view that everyone will have his/her/its own personal computer do not mean a one computer for each person or even for each appliance. When we have much sharing of function or information it is difficult to separate out whose computer is where or what.

Even given that one has one's own computer and does not want to offer a "time-sharing" service, there is still no excuse for a primitive environment. There is still a need for a sophisticated environment for running software dictated not only by the need to make programming simpler, but by the requirement that there be enough of a common environment such that the support can be taken for granted when supplying prewritten software. Self protection is enough of a justification for protection and access control mechanisms but these computers will be running vendor's software. The first level of protection is against accidental interference such that a bumbling programmer cannot provide software to erase the file it is maintaining for you. The threat of malicious software is great in an environment where a simple program can potentially get any data from anyone.

I contend that at minimal a Multics level system is necessary in the personal computer. The environmental constraints of a personal computer system are much greater than those of a Multics installation where the sophistication of the personnel can be taken for granted, or at least the user is capable of affording such and taking responsibility for its lack. Facilities such as the ability to perform multiple tasks simultaneously, manage storage and extend the base hardware are necessary.

But to say "Multics level" doesn't mean a Multics as it is. We can consider the current Multics base machine to be a storage management and programming machine. It might be more useful for the lowest level to normally be a database interface. I am exploring this in more detail in a paper I am trying to write. In that paper I am considering a model in which there is a local database management system which can also access data in other systems, but without incurring undue overhead. The bank example I gave earlier demonstrates a method for allowing the user to have

---

[1] Reference to Winston and Horn article in datamation

an approximate image of the account status without having to continually access the bank's database.

Before leaving the topic of the personal computer I should comment on issue of simpler personal computers. These must be expected but I would think of them as essentially peripherals of the local "smart" computer. Thus there may be many computers in devices and appliances that cannot support the full protocol, but they interface to the larger network. At present terminals only interface to the ARPA Network via a friendly host.

Attributes of collections of personal systems

We can consider a collection of personal computer on two levels. On a low level there is the explicit communication. This is largely an issue of protocols. It is also an issue of marketplace structure as in TR-128.

More complicated is implicit interaction, or linkage between systems when it is not obvious what the interactions are. To bring back the bank example, whether the account information is kept locally or remotely is of no interest to the user except in terms of response time resulting from communications costs and robustness.

The division between explicit and implicit sharing is not sharp. There is the dual question of how the user is to view the system conceptually, and how the services are going to be actually provided. It is convenient if the policies mesh exactly. This has been the goal of pricing policies which attempt to reflect the cost of providing the service but may not be suitable in all cases. However, if the operations of the system are visible because of intrinsic delays or the inability of the computer to determine which John Smith, then it is convenient and appropriate to reflect this to the user.

In an earlier section I gave the examples of making aspects of robustness and naming visible to the user. But we don't want to make the user aware of all that is going on. The analogy is with the phone company. The phone user is generally aware of where the called party is, but not of how the call is routed. On the other hand call forwarding hides the geographic knowledge and WATS lines introduce some knowledge of routing.

The goal of the network is not so much as to hide the fact that there is distribution as much as it is the attempt to simplify dealings with this fact. To bring back the bank example, the user should be able to tell the local datamanagement system where the account is kept by giving the name of the bank, and perhaps an account number. And cease to worry a beyond noticing that a request for the exact, as opposed to approximate balance incurs an extra delay and might require the bank be open for business.

One particular issue I have been looking at is the problem of distributing a database in such an environment. In keeping with the philosophy stated above, the database manager maintains the user's local database which may contain copies of data the user is actively working with, but which normally reside at other nodes.

Some specific research topics

As a research vehicle the personal computer enables the Computer Systems Research Division to make use of its expertise in operating systems and its developing expertise in networks. It represents a continuation of ongoing research projects in extending the bounds of a single computer system to a collection of such systems.

I just touched upon the attributes of personal computers and collections of such machines to indicate the some of the starting points for research. Some specific research topics include:

1. How can the software be constructed for such a machine? It can be discussed within the context of running such software on an existing system such as Multics. This can be done using the full capability of Multics or can assume a more restricted environment.

2. Instead of asking how the software can be written we may address a fundamental issue of the hardware/system architecture. The experience with dataflow architectures, multilevel operating systems and microprocessors makes it feasible to actually design and construct such a system.

3. If information management is the normal purpose of such systems, then database facilities should be available as a standard facility. It might be considered to be at the level that the directory structure fits into Multics. Should such an interface be equivalent to the low level interfaces to a relational system or should they be in the form of query language requests or should they be at an even more primitive level as in the current Multics system.

4. Information management is not limit to a single node but involves using information that may reside at multiple nodes. How is the information to be managed. How visible should this management be to the user?

5. How do the concepts of access control extend to such a collection of systems.

6. How can the functions of finding and accessing data be provided across such a dynamically changing network?

In addition to these individual research topics CSR as a whole is involved in setting up the framework in which the individual projects can fit. It is not enough to design the architecture of an individual machine if the properties of such a machine are in conflict with their integration. It is in this area of creating a coherent framework that a group effort required and it is the richness of this area that makes it suitable as a research vehicle.

Conclusion

I have given a model of the a collection of integratable personal computers as a model within which we can explore general problems of distributed systems. I have argued that such a model is suitable for the problems that CSR has traditionally tried to address in the area of applying computer systems technology at a level that is applicable to what we conceive of as general applications area. I have attempt to justify the appropriateness of this level of analysis as a means of addressing practical problems but at the same time not restricting the research to specialized problems of more limited scope.

As a research vehicle it provides a coherent set of goals for the group while at the the same providing a rich set of topics for individual researchers (i.e. grad students). It also has the property of having a rich potential for demonstrable results. The specification of the properties of such a system (individual machines or collections) is a valuable result. But even if agreement cannot be reached on the overall properties the separate results of such research are generally applicable to the design of individual computers as well as to systems of cooperating machines.

References

A. Clark, D; LCS/CSR/RFC-124; ARPA/ONR Proposal for Research in Distributed Computing.

B. Frankston, Robert M; MAC TR-128; The Computer Utility as a Marketplace for Computer Based Services.