MEMORANDUM

To: Seth Steinberg, Chris Herot, Andy Lippman
From: Nicholas Negroponte
Subject: 6.0
Date: Oct. 4, 1976

At the risk of initiating the notion and then proliferation of committees within our operation, I would like to ask the three of you to act as a MAGIC 6.0 committee, to follow design, and create 6.0. My meeting with Seth, et al has scared me into this in the following way. You will note that it has nothing to do with confidence in technical expertise.

Remember Mike Miller. He spent months designing away, chatting with Sather, reading books, and never using his rented typewriter. The result was a better educated Mike and no system. In contrast, I sense (no offense) that we have the opposite happening right now. Namely, people are coding away, and we will get a system. But, is it what we want and are expecting?

It is unreal and foolhardy to expect three UROPers and Seth part-time to generate the system with which we plan to live for the next five to eight years, without help in the planning and conceptual stages. For example, I think it is myopic to consider the user-processor operating system either a "toy" or "non-operating system". It is a graphical operating system, perhaps. Meanwhile, the i/o processor is a file-processor, not a time-sharing system. Seth's analogue of the AI Lab is correct: they have minis strung out and serviced by a HOST central system. In contrast, we have a smart disk-controller servicing Maxis. This is an extreme example, but to be considered seriously.

Yes, I understand perfectly well that the smarts necessary to be an intelligent file processor just need a little more (memory, speed, and code) to do things without burdening the user-processors. But how little? We don't know. We do know that when copying a file from disk to tape, it will not pass from disk to i/o-proc to user-proc, back to i/o-proc finally to tape. But where does this end? The compiler can live in both places. My feeling is that (at least at first) it should live in the user processor, using the printer, for example, via the shared bus, not the i/o-processor. This is the point of most contention and we can argue it out. Hence your committee.

I am not asking you as a group to take over what Seth is doing so well. I am looking for a way to maintain "the year of no surprises." You have carte blanche on this system; you will be living it more than I
I recommend that you meet like once a week to discuss conceptual matters, not programming details. For example:

1. What is a graphical operating system?
2. What if ea ch user-processor had a 2314 and the shared disks were 1000 megabytes of something or other?
3. If we do make a "terminal system," one in each office, do they go into a special kind of user-processor?
4. What is the implication of sending RAMTEK-like stuff back and forth?

(OVER)

Would you please meet (without me) a couple of times between now and the 20th of October. The purpose of your meetings should be to compose something akin to a policy document for 6.0: its intentions, its characterization, its schedule (for 16 and 32 bit machines). Please write this up for Machinations that week. Subsequently, let's meet in a larger group, (even if "larger" only means the addition of me).

At the risk of being repetitious, in your arguments for one approach versus another, please do not include: it is easy to do, it has already been done, the code is the same, etc. Let's start doing things because they are the right things to do. I am very worried that "rightness" coming out of a MULTICS model, but at least that is better than the mechanics of production.

Finally, if you think it is appropriate or helpful, I am happy to ask one or group of EE faculty to review our designs.