Vertex Contract in Hindsight Computer driven control system design philosphy

A contract for a complex software-driven system should include:

1. No firmware in EPROMs*
2. Spare CPU boards, interface boards, supplied as part of the contract.
3. Acceptance tests should include demonstrations that the system comes up and runs correctly with any mixture of primary and spare boards.
4. All programming should be done in high level languages
5. High-level documentation - not just "self documentation" through progammer's comments.
6. assigning an engineer to monitor the contract - keeping up to date on the system design and the documentation. This engineer would be in charge of acceptance testing and would make sure that everything is received, including all necessary vendor documentation, licenses, etc.
7.System design should include provision for self test and diagnostics

* Note: EPROMs are probably a bad idea in any system that is subject to change. EPROMs seem attractive in that they can't lose the the program. But changing them requires a separate programming facility/ability: a prom burner, cables, floppies containing the drivers for the burner, a pc, floppies containing the software development system, personal and corporate memory of how to use all this. Keeping track of the separate components. The system should be and remain always fully integrated. We attempted to maintain an programming facility for the old 8080-based pointing system but failed; when the program finally needed changing, it was done at the machine language level using the binary listings and a bare prom burner.

A complicated system should contain its own development system, i.e. it should be possible to program and reprogram the system from its own operating console or terminal. That terminal should be as generic as possible - a plain vanilla pc with standard hard disks - maybe with a 2nd back-up hard disk, a printer, etc. If a remote processor or processors are required in the system, they should all be accessible (controllable, loadable, readable) from the central terminal. controller/promming station.

Ideal System:
1. Digital and Analog I/O modules that are
    a. driven by industry standard interfaces: bus or com (longer lasting standards). What com? RS232, Rs422, Ethernet..?

Should these modules be driven by local computer or only remote computer?

A general and flexible approach for our pointing system would be to have all the sensors and effectors connected to an upstairs unit that is driven by a fast Ethernet connection. This upstairs unit should, as much as possible, have a simple straightforward functionality (that could be explained in 1 minute), hardware design that could be realized with off-the-shelf components that are themselves as generic as possible. In short, the upstairs unit should be simple enough (in scope and specification) that a new engineer could design and build a new upstairs unit in a matter of a couple of weeks.

Why Ethernet? The requirement here is sufficient speed and longevity. RS232 has had the longest history but is too slow. Ethernet seems a good bet.

The upstairs unit will be driven by a pointing computer which should be as generic as possible - With Ethernet inputs, as well as Ethernet outputs, the pointing computer can reside either upstairs or downstairs or back and forth, as needed for normal operation and maintenance.

The pointing program would best exist in one version only, including diagnostic and maintenance features.

.

If this computer is sufficiently generic (a pc) the “pointing computer” can exist mainly as a program which will run