Position Statement - How Far Have We Come?

Brian Randell

School of Computing Science
Newcastle University
Newcastle upon Tyne
NE1 7RU, UK
Brian.Randell@ncl.ac.uk

In the late 1960s computers were mainly very big (room size) and expensive, and were mainly used by wealthy organisations for large-scale commercial data processing and scientific calculations, though the first minicomputers were starting to appear in well-funded laboratories. The idea of a personal computer had not yet arrived. Data bases, networks and distributed systems were also yet to come, but work had started on implementing ARPANET, the Internet’s main predecessor, though local area networks were unknown. The term “software” had come into use, though systems software was usually provided “free” with the hardware by the computer manufacturer, and applications software was normally designed specially for particular clients and particular computers. Shifting to a new computer normally implied – because of hardware incompatibilities – users having to abandon or rewrite all existing applications programs.

Since then not just one, but rather many, types of software industry have come into existence, in particular those that design or tailor “bespoke” software for particular clients and environments, and those that produce “of-the-peg” software packages that are sold to thousands or even millions of customers. The first type is a recognisable successor to the software activities of the 1960s. In the second, very different, type of software industry economies of scale, and Darwinian-style evolution, have a large impact on what sorts of software get implemented, and how such implementation is undertaken, e.g. involving getting hundreds or thousands of users involved, willingly or unwillingly, to help with software validation and refinement. Furthermore, this second type of industry is in the main concerned with what economists call a “natural monopoly”, caused by the fact that though its development costs are immense, those required for manufacturing and distribution are comparatively trivial – so that being first to market, and achieving initial market dominance, are highly advantageous.

Any reasonable account of how far we’ve come since the late 1960s (and where we’ve got to) has to treat these two types of software and software industry very differently. The first type of software industry has gone on to attempt ever larger and more complex tasks. But it is still subject to many of the same challenges concerning implementation cost, project schedule, performance and (especially) dependability that so exercised the NATO conference participants. The second now provides a wonderful marketplace of useable and useful software systems, utilities and applications that has utterly transformed society’s utilisation and perception of computers. But technical monoculturalism, allied to the growth of computer networking, has led to this industry and its customers also suffering from all sorts of malicious, indeed criminal, activities that were not in any way foreseen in the discussions at the NATO conferences.

Among the many looming technical developments that were discussed enthusiastically at the NATO conferences, two that now stand out as still of great interest and constituting a considerable challenge are software components and software development environments. A third technical development that was already under way in the late 1960s, though less prominent at the conferences, was that of multiprocessor design – a technical challenge that has been revitalised by the arrival of multi-core processor chips. For all that software and its various industries have developed out of all recognition in the past forty years, I feel that progress in these three topic areas has been distinctly disappointing. In my view, the one major success story regarding components remains Unix’s pipes and filters, software development is fragmented across a ridiculous number of different languages, methodologies, platforms, and “standards”, and Amdahl’s Law still holds. It will be interesting to debate whether there is general agreement on these points, and what the future might bring.