

Future issues of this column will focus on the net effects of Net-centric computing (NCC) on you. Possible topics include the changing nature of software engineering and the Web, using virtual private networks to lower the cost of remote access for geographically dispersed organizations, and the implications of high-speed access technology such as xDSL (Digital Subscriber Line) to NCC. There is certainly no lack of issues related to NCC, so drop me a line if you have a particular topic in mind!

The Era of Net-Centric Computing

Scott Tilley



“In the future, network computers will be purchased and used with the same enthusiasm as home exercise equipment.”

Scott Adams, *The Dilbert Future: Thriving on Stupidity in the 21st Century* (New York, NY: HarperBusiness, 1997).

Is Mr. Adams right? Maybe. But it's important to make a distinction between network computers (NC) and Net-centric computing (NCC). An NC is just one type of “thin client.” NCC is more than just thin clients; it is an emerging phenomenon whose effects will be profound and far reaching. Almost anyone involved in computer science, information technology (IT), or software engineering will be affected.

So what is NCC? The underlying principle behind NCC is a distributed environment where applications and data are downloaded from network servers as needed. This is in stark contrast to the current use of powerful personal computers (PCs) that rely primarily on local resources. In some respects, NCC resembles an earlier computing era of mainframes and dumb terminals. However, there are important differences. NCC relies on portable applications that run on multiple architectures (“write once, run anywhere”), high bandwidth (for downloading applications on demand), and low-cost thin clients such as the NC, the NetPC, and Windows-based terminals (WBT).

Thin clients

An NC uses Java for local processing. It was initially proposed by the “gang of four” (IBM, Netscape, Oracle, and Sun) as an alternative to the Microsoft/Intel duopoly. The NC vision was for a computer that did not run Microsoft Windows software, and it could use processors other than Intel's Pentium chips. This vision has at least one flaw: People still want to access their legacy applications (primarily Windows programs) and data.

The NetPC was an interim solution proposed by Microsoft and Intel to counter the NC. It is essentially a stripped-down PC with a sealed case. The selling point of the NetPC is a reduction in total cost of ownership because end users are unable to add or remove new hardware or software. It can be centrally administered and it can run Java applications if needed. In this sense, some consider it to be a better network computer than the NC itself.

The thin client that seems most likely to succeed is the WBT. A WBT is the thinnest of the thin clients, relying completely on a central server for applications and data. The WBT acts only as a display device, much like an X Station does on Unix. Early versions of WBTs are in fact current computers running Citrix Systems' WinFrame client. This application lets users on Windows, Macs, or Unix machines access a centralized computer running a modified multi-user version of Microsoft Windows NT Server. This technology (code-named “Hydra”) is being rolled back into Windows NT 5 Server as the “Windows Terminal Server.” I think it will prove successful because it lets organizations leverage their current IT infrastructure. The productive life of older 386- and 486-based PCs can be extended while newer machines are purchased incrementally.

Effects in the office

Irrespective of which type of thin client you adopt, how will NCC affect you in the office? Since almost everyone is affected by software these days (like it or not!), you likely fit into at least one of the following three categories: user, developer, or administrator. For some users, the relief at not having to maintain a PC means they can instead concentrate on their primary tasks. Other users may chafe at the limitations that NCC brings with it. The removal of “personal” from PC means that users will no longer be able to significantly alter their desktop environments.

For developers, NCC offers an opportunity to greatly increase their customer base: An application written in an NCC-aware programming

language, such as Java, means writing code once and having it immediately accessible on multiple platforms. It also means a different development environment, a new deployment model (renting applications versus buying), and new concerns about security. In a Net-centric world, security is not just for system administrators. For most developers, security is a quality attribute that is treated as an add-on to the system. For NCC, it needs to be treated as a first-class concern.

For administrators, NCC means a potential reduction in the cost and complexity of managing IT resources. The total cost of ownership issue has been cited as one of the motivating factors behind NCC, but so far little real data is available to suggest that NCC will be cheaper than today's methods. It may in fact be more expensive because of the increased complexity of a heterogeneous and distributed environment.

Effects elsewhere

Outside of the office, the effects of NCC may prove even more significant. For software engineering, NCC offers a fundamentally new way of thinking about software. Basic issues such as version control need to be re-evaluated. For example, if a software application is being delivered to the user (and continually updated) using push technology such as Microsoft's CDF or Marimba's Castanet, what does it mean to say "the current version"? If the application is being monitored and updated in the manner of superdistribution, this may make software versions based on millisecond differences a reality.

Still not convinced this "NCC thing" will really affect you? Here's another quote from Mr. Adams's book:

On the off chance that you are not familiar with the NC versus PC debate, allow me to provide some background. The NC is blah, blah, blah, Java, blah, blah, trying to screw Microsoft, blah, blah, no hard disk, blah, blah, Larry Ellison.

The "blahs" are Mr. Adams's, not mine. Larry Ellison is the CEO of Oracle and a big proponent of NCC in general, and of NCs in particular. Since the PC industry is driving many of the innovations in both academia and industry these days, the NC versus PC debate in the context of NCC will very likely affect you whether you follow the debate or not. There is currently a tremendous amount of discussion about NCs versus other types of thin clients. Time will tell which type will be the most popular, but history has shown that it doesn't usually pay to bet against the Redmond juggernaut.

About the author

Scott Tilley is a visiting scientist at the SEI. He works with the Product Line Practice Initiative in the Reengineering Center, focusing on transitioning best practices in legacy-system reengineering in a disciplined manner. Before taking on this role, he was on leave from IBM and a member of the Rigi project in the Department of Computer Science at the University of Victoria. Tilley is the author of a 1993 book on home computing and has over 50 publications. He has a Ph.D. from the University of Victoria. He can be reached at stilley@sei.cmu.edu.

The views expressed in this article are the author's only and do not represent directly or imply any official position or view of the Software Engineering Institute or Carnegie Mellon University. This article is intended to stimulate further discussion about this topic.

Copyright © 1998 by Carnegie Mellon University.

The Software Engineering Institute (SEI) is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University.

SM Capability Maturity Model, IDEAL, Personal Software Process, PSP, Team Software Process, and TSP are service marks of Carnegie Mellon University.

® Capability Maturity Modeling, CERT, and CMM are registered in the U.S. Patent and Trademark Office.