

July 16, 2001

## Soft Gorillas -- part one

Part one discusses how the gorilla playing ground in the Telecosm is different from the one in the Microcosm. Part two will discuss investing in software gorillas in the Telecosm.

[The Gorilla Game](#) is the natural investing outcome of the high tech marketing used during the PC age. The Gorilla Game is predicated on the outcome of TALC (Technology Adoption Life Cycle) and two of the main concepts, [Crossing the Chasm](#) and [Inside the Tornado](#) are the subject matter of two previous marketing books by Moore. There is a basic differences between the Microcosm and the Telecosm that makes it difficult to apply the Gorilla Game to the latter.

The Telecosm is about communications, about computers talking to computers be it in client-server mode, in peer-to-peer mode, in HTML or in XML. But the funny thing is that the original LANs were not designed to allow computers to talk to computers. They were designed to allow inexpensive computers to share expensive resources like the LaserWriter. At one time, the LaserWriter was the most powerful "computer" built by Apple. Initially, AppleTalk printers replaced serial printers in the Apple world. An unexpected result of LANs was that computers could talk to computers and eventually that became thousands of times more important than inexpensive computers sharing expensive resources. I am not ignoring the Internet but please remember that originally the Internet was not designed to connect computers but to "Inter" connect "Net" works. The Internet is a network of networks.

A telephone must, by necessity, be able to talk to all the other telephones in the system, be it in the same city, the same country or anywhere in the world. Very few other products were designed from day one for "connectivity." One car does not have to talk to another car. It can if it has the same operating system. What do I mean? A 12 volt Ford can jump start a 12 volt Mercedes but a 6 volt Volkswagen cannot even though both the Mercedes and the VW are German. In this case, the "operating system" is defined by the voltage the cars use. The original PCs were not designed to talk to each other any more than one typewriter was designed to talk to another one or one desktop calculator was designed to talk to its brothers.

There was no need for an Apple II to talk to an IBM PC or to an Apple /// or to an Osborne. Each company was perfectly at liberty to create a proprietary operating system for his machines with no regard, whatsoever, for what the others might be doing. In the telephone business this was not the case. All telephones had to be able to talk to each other. Each computer manufacturer did his designs without looking at what the competitors were doing. The OS was different. The disk formatting was different. The file formats were different. The printer interfaces were different. Even the character sets were different, most use ASCII and IBM still uses EBCDIC (Extended Binary Coded Decimal Interchange Code). UNIX, DOS and the Mac use three different methods for "New Line." I remember in the old days Apple asking us dealers if the Apple II keyboard should be like a typewriter keyboard or like an IBM terminal key board. I think you get the point! :-))

One of the consequences of the law of unforeseen consequences was that each computer brand ended up with its own architecture. The second unforeseen consequence was that diversity made switching computer brands very complicated. Remember the cigarette ad: "I'd rather fight than switch?" IS managers would rather fight than go through the hassle of switching. The third unforeseen consequence was that Geoffrey Moore coined the expression "Open Proprietary Architecture with High Switching Costs."

Warren Buffett talks about moats to keep out the enemy. He is talking about barriers to entry. Moore talks about high switching costs. These two ideas, even though they lead to similar results, are quite different. Coke's moat is its brand name. Anyone can make sugar water but getting consumers to switch is something else. Would you say: "Give me a rum and Pepsi?" Possible but not likely. On the other hand, if you want to start up your MacWrite on a DOS box, it simply cannot be done. You have to buy new software and that is costly, that is a "High Switching Cost."

Let's say that AT&T has one architecture, say TDMA, and Sprint has another, say CDMA. A TDMA phone owner can talk to a CDMA phone owner and he has no clue as to who is using what and he could care less. Let's say the AT&T user is not satisfied with his cellular service and he wishes to switch. How costly will it be? I remember when carriers paid you to switch. Your cost of switching was negative.

You saved money by switching! Of course, for the carrier, switching is expensive. To tear out TDMA and install CDMA probably means trashing most of what you have and starting anew. The consequence is that the high switching costs in the Telecom don't lie with the end user but inside the value chain.

Why does it matter where the switching costs lie? For the end user, using one or another technology is usually a matter of convenience and the original choice was probably a random event. If all your friend use Macs you are likely to wind up with a Mac as well. Once you are "locked in" it's simply too much hassle to switch. If the service or product is bad enough, or if the grass is that much greener on the other side of the fence, you will switch. But it is not a life and death decision. On the other hand, for AT&T to use the more expensive or less efficient architecture is a question of life and death. In the long run, if they have competition and they don't switch they will die. It is easier for Qualcomm to get AT&T to spend billions of dollars to switch from TDMA to CDMA than it is for Apple to get a person to spend a few thousand dollars to switch from Windoze to Mac. Witness Ericsson and Nokia.

There are hardware gorillas in the Telecom, Qualcomm and CDMA is just one example. But in the fiber optic side of things, it is much more difficult to find gorillas because all systems have to "interoperate." Not only that, in building the physical network you don't have proprietary architectures. A lambda is a lambda no matter how it is generated and how it is groomed and how it is switched. And at the point of connection between networks, the proper translations have to be made should the transport protocols be different. This makes finding hardware gorillas in the Telecom difficult.

Part 2 will discuss how software in the Telecom retains the gorilla properties it had in the Microcosm and I will discuss companies that might be software gorillas. In the mean while, you might want to bone up on the differences between investing in hardware and software as recommended by Moore (buy hardware after the tornado and software in the bowling alley).

Denny

"Demand creates queues. Supply gets rid of them."

[Software Times](#)