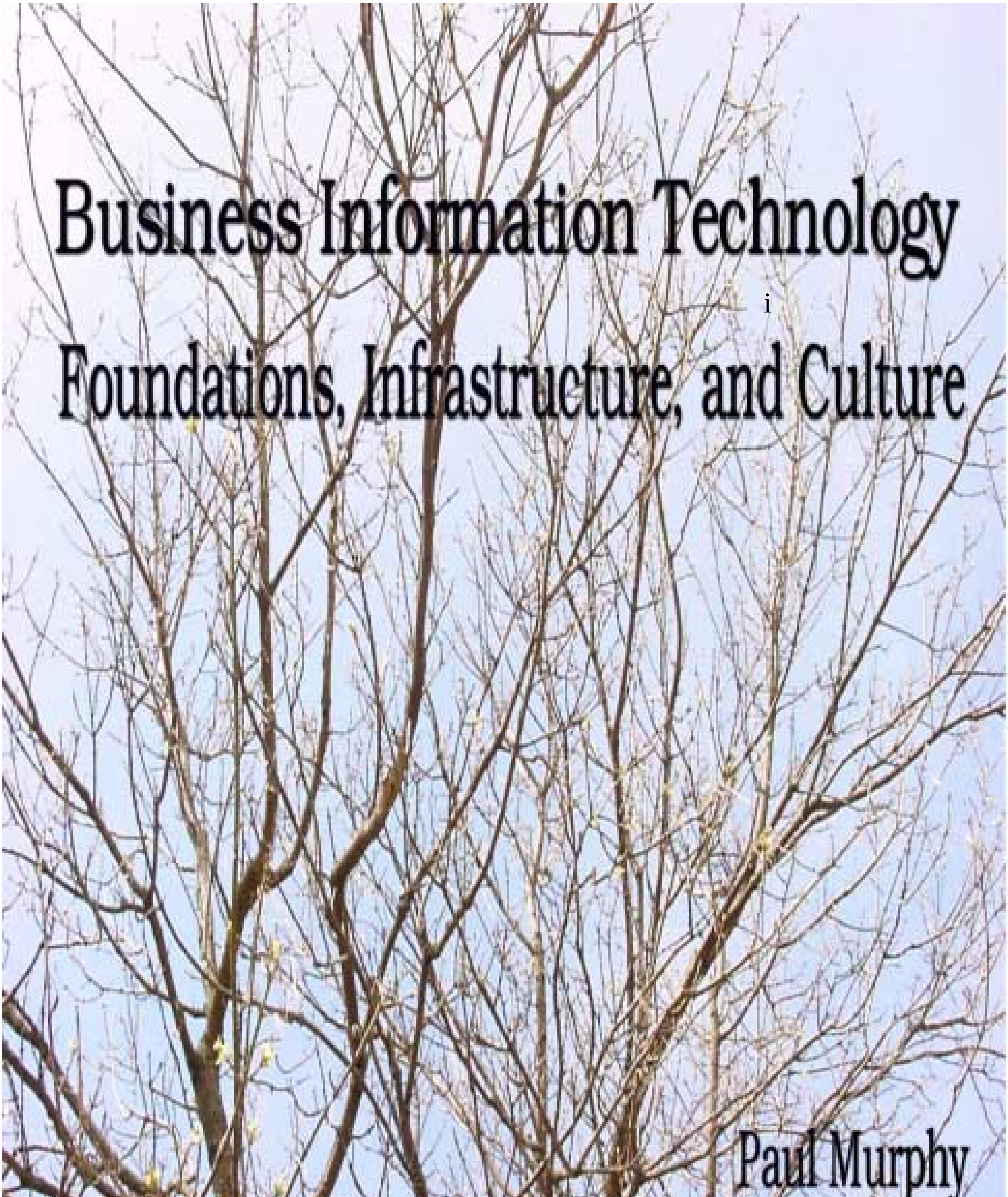
Business Information Technology
Foundations, Infrastructure, and Culture
Paul Murphy

Dedication

This book is dedicated to the kids who inspired it: the Students of Accounting 241, University of Waterloo, Spring 2002.

Acknowledgements

I wish to acknowledge, with thanks, the efforts of those who helped me find mistakes and other problems in the text. In particular John McKeown (Open University, UK) and Jim Taylor of Winnipeg, Canada worked through every chapter. Obviously they bear no responsibility for the remaining errors, but they both found hundreds of them - thanks guys!

Notice

By necessity, given the nature of the thing as a book about information systems technology, this book names names, discusses prices, and gives my opinions on the performance and value of various products. Obviously:

1. all brand names, product names, web site names, and related materials as used in this book are, or may be taken as, trade names, service marks, trademarks, or registered trademarks of their respective owners; and,
2. while every precaution has been taken in the preparation of this book neither the publisher nor the author assumes any legal responsibility for errors or omissions, or for damages arising from the use or distribution of the ideas and/or information, contained herein.

Copyright

Website

The website for this book is on http://www.winface.com/

This is a draft

The current release is a next to final draft. Comments are requested by email to murph@winface.com or via the discussion group on the website.

Table of Contents

Introduction

Overview

This book is designed for people who interact with, but do not want to become, professional information systems staff. It is a management guide to information systems for auditors, executives, and business owners who want to understand what the issues are and how to resolve them.

As a manager, business owner, or management advisor you will often find yourself forced to make decisions in situations where systems professionals either hold conflicting opinions or seem to have considered only the alternative they're most comfortable with. Whether what's at stake is Sarbanes-Oxley compliance or a strategic business opportunity doesn't matter: it's their field and their professional vocabulary, but it's your decision and you, not they, will wear the result.

Preparing you for situations like these is the most fundamental thing this book is about. Not just helping you past the jargon barrier so their vocabulary doesn't leave you locked out; but getting right to the hard core of the issue: helping you understand the real relationship between their ideas about computing and the business problem your organization is trying to solve.

Read this book carefully and you'll understand that systems decisions aren't usually about bits, bytes, or megahertz, but about people and the way their ideas and interactions affect your organization's ability to do its job.

In the long run business success is determined by how technology is applied, by the things people think they know about technology and how to use it, not by the technology itself.

Talk the talk, but understand the people: that's the hidden lesson from ninety years of business data processing - understand it, and making smarter systems decisions will become part of your professional repetoire.

To the non technical manager a computer tends to look a lot like a computer, but that's not true from the expert's perspective. If Dick and Jane are two systems experts with differing views on a technical issue, those views will reflect what Dick and Jane think they know about computers and therefore the differences in the forces which shaped their opinions.

Dick and Jane may genuinely both be experts, people who deal with computing on a professional basis and whom you'd expect to resolve minor differences by working out the facts, but reality doesn't work that way. Instead they will usually try to focus discussion on the differences, rather than the commonalities, in the technology choices they support.

In most cases these differences are minor in the context of the business, but just as the protagonists in racial or religious wars are usually willing to die to perpetuate differences that are invisible or unimportant to outsiders, so too will Dick and Jane unknowingly sacrifice your business to the unalterable certitudes imposed by their backgrounds. In fact, expecting Dick to work with Jane's ideas, or vice versa, will predictably have roughly the same effect as hiring a Catholic principal for a Protestant school in Belfast.

Things got this way through the problem driven co-evolution of hardware, software, and management methods. In business data processing, as in nature, only the fittest survive - but, again as in nature, fitness is determined by local circumstance. A polar bear is superbly adapted to life in the Arctic, but wouldn't last a week in sub-Saharan Africa.

Speciation, the evolutionary response that adapts some bears to the desert and others to the Arctic, is driven by the problems the organism has to solve to survive and prosper. Once that evolution succeeds, however, the change processes stop; freezing both behavior and perception in place. You can put a polar bear in a southern California zoo and feed it chickens, but it will always scan its wading pool for seals.

The same thing happens among groups of professionals confronting problems or opportunities. They rapidly evolve characteristic methods for using the tools available to solve the problems they see, and then recast their perception of the environment around them in terms of the problem set that drove the group's initial development and specialization.

Generally speaking hardware differences are used to label systems cultures, but the real differentiation comes from management practices and habits of thought; not hardware. At the global level a computer really is just a computer, but the management reflexes that are automatic components of the one right way to run a mainframe data center generally constitute obvious insanity to the Unix systems manager and may be utterly incomprehensible to the Microsoft guru.

If Dick is from one camp and Jane from another, their arguments will usually both be right within the context of the information systems culture they belong to. Dick will genuinely see seals where Jane sees berries and anthills, not because they are antagonistic or incompetent, but because those perceptions reflect their systems culture. In choosing between their positions you need to look beyond their arguments to the cultural coloration through which they filter their perception of both the problem to be addressed and the solutions appropriate to it.

Making systems decisions, choosing between Dick and Jane, is likely to be an every day part of your role as an auditor, a manager, or a business owner. To do it well, you need first of all to be able to communicate effectively with both Dick and Jane. That means much more than learning some acronyms and speaking the words of their language; it means understanding the evolutionary pressures that produced their ideas, their reflexes, and, above all, their verities - the things they never think about but which their data processing culture treats as eternal and unquestionable truths.

That fundamentally is what this book is about: demystifying jargon while helping you understand how Dick and Jane got their opinions, what their cultural assumptions are, and how to understand and balance the values implicit in those assumptions when you have to choose between them.

Structure of the book

An *information systems architecture* consists of:

1. Hardware;
2. Software;
3. Management methods or processes; and,
4. A business or organizational context.

These elements are inter-dependent; they co-evolve as people use change in one as both a cause and a consequence of change in the other three.

Nobody orders Slime Head for dinner - but Orange Roughy is a restaurant favorite
Just as most systems or network engineers aren't engineers, systems and/or network architects aren't usually architects. "Systems Architecture" is just an attractive term that has wide acceptance among systems professionals and no intrinsic meaning beyond what its use tells you about the profession's lack of maturity and self-confidence.

At the broadest level there are four major systems architectures in current use:

1. The IBM mainframe architecture;
2. The application appliance or mini-computer architecture;
3. The Microsoft PC architecture; and,
4. The Unix or Open Source architecture.

Each of these architectures evolved in response to some combination of business and technological change, and each has its own group of practitioners. These groups are distinct in that all have unique ideologies, their own understanding of the one right way to manage computing; their own vocabularies, and their own view of the rightful and necessary role of computing in businesses and other organizations.

Thus when you first meet an IT professional, Dick or Jane, the chances are very good that he or she will be a member of one of these communities - and therefore that the effort you put into understanding these communities will pay off very quickly in your ability to work effectively with Dick or Jane.

That's why the first four chapters of this book are organized as tours of these communities with:

1. An overview of the history behind today's realities focusing on the problems that shaped the technology and the management ideas that go with it;
2. A tour of a working data center in which we look at costs, roles, qualifications, technologies, and management practices; and,
3. A case study exemplifying the culture at work..

Critical technical or other information

Inset boxes like this one are found throughout the text. Usually the information provided either modifies and expands on information in the core narrative or defines terms used in the main text.

These boxes are not asides. The structure of the book is designed to make learning a lot of technical terms easier by providing the context in which the technologies evolved - but terms and examples still need to be presented and that's what most of these boxes do. For example:

Systems and "The Systems Men"[a]

The word "system" is often used to label computers or software but "a system" is really a multi-step process or procedure executed in pursuit of some goal. Thus a printing *system* for a daily newspaper can include people, organization, and a sixty million dollar printing plant. Systems do not need to incorporate technology or be successful in achieving their goals - it is proceduralization: the defined, organized, repeatable, nature of the processes involved, that makes something a system.

"The systems men," an association of data processing professionals that started in the 1920s (formalized in 1947 as The Systems And Procedures Association of America inc.) focussed on leveraging their way into the executive suite by deploying organization wide systems built around procedures appropriate to the data processing equipment they controlled.

a. See: T. Haigh, *Inventing Information Systems* (Business History Review; 75, spring 2001, pp 15-61).

The topics discussed in Chapter Five are extremal, either so different that side by side explanation is called for, or so similar across all of the major architectures that separate explanation would be a waste of everyone's time.

Included are sections discussing the major goals, technologies and related issues associated with:

1. Clustering
2. Information integrity
3. Networking and Security
4. Relational Database Management Systems
5. Total Cost of Ownership
6. Major Business Applications; and,
7. E-Commerce.

A note from the author

Various reviewers have told me that this book is boring. It is, and I'm sorry about that; unfortunately I don't know how to fix the problem. All that IBM stuff in the first chapter, for example, is of little interest to people whose focus is on making the current technology work. Bear in mind, however, that your role as a manager or an advisor to managers, isn't likely to involve making technology work. What it will involve is making decisions: does the company back Dick's ideas or Jane's? bet on this technology or that one? promote Zina or Ivan?

For those decisions you need first to speak the language, that's a requirement for communications, but, more importantly, you need to understand where they're coming from - and that's what all that boring computing history is about. Dick's opinions, like Jane's, evolved in response to external pressures and events; understand those and you'll understand Dick's professional persona well enough to make your decisions. Fail to understand them, and you're left to base your judgement on other people's opinion, the cut of their clothes, their position in the company, or other externalities that really have little or nothing to do with the merits of their arguments.

So yes, some sections are long and boring; but please think of it as a kind of Murphy's mixture - it may taste bad, but it's good for you.

Chapter 1
Data Processing and the IBM Mainframe

About this grouping

The IBM mainframe *information systems architecture* consists of:

1. IBM System 360 compatible and successor hardware;
2. JCL, DOS 360, IMS, CICS and successor software;
3. Data center management methods or processes; and,
4. A transactions oriented business or organizational context.

As of late 2003 there were about 18,000 mainframe data centers in operation worldwide. Some amount to billion dollar businesses but, on average, each spends an estimated 43 million (USD) per year; the vast majority of it on staff, desktop support, and facility services. Taken together these data centers cost their owners nearly eight hundred *Billion* dollars to operate.

Originally data processing centers focused on developing reports summarizing business data well after that data was generated by normal business processes. Although reductions in that time lag have allowed many data center operations to get closer and closer to real time business support, the general focus remains on a class of tasks that can fairly be described as automated clerking.

As part of that after-the-fact focus on clerical automation most data centers were initially created to reduce processing costs in Finance, and cost justified mainly on the basis of either displaced clerical wages or increased accuracy in clerical tasks. As a result the Information Systems Director has traditionally reported through the Finance department and focused on cost cutting.

Many companies now also have Chief Information Officers (CIOs) and Chief Technology Officers (CTOs) to, or through, whom the data center reports to a steering committee of some kind. In general, however, today's mainframe data centers remain separate from the rest of the of the business and are usually still

To address the first kind of problem IBM strengthened the memory partitioning process, first in the JCL and later by imposing hardware barriers to break available storage into small chunks that could then be allocated within JCL.

Since this approach extended the independence of multiple batch jobs running on the same machine, it also offered a way to share the hardware among several users because each partition looked like a real computer to the programmer.

This approach to managing the conflicting demands multiple users put on a machine first became known as virtualization[1] and eventually as partitioning. In its virtualization guise it formed the foundation for IBM's attempt to build a multi-user operating system on top of CP and PR/SP that became known as VM; in its partitioning guise it became the foundation for a critical mainframe "best practice" - hardware partitioning.

Hardware partitioning attacked the problem of the million dollar phone bill by enforcing memory partitioning at the hardware microcode level. Equally importantly, however, it met the data center manager's need to isolate test and development work from production work without forcing him to buy another multi-million dollar machine.

Today VM[2], a virtual machine environment usually layered on top of hardware partitioning, is IBM's major interactive user environment for the mainframe. In VM, individual users are given software managed private partitions within a hardware partition or dedicated machine. Thus data center management can enforce process resource utilization priorities and control within the VM user configuration and each user gets access to a virtual mainframe with which he, or she, can do just about anything that could be done with an actual private machine - including loading a "guest" operating system such as VM itself, Linux, or MVS.

---

1. Note that "virtualiation" continues to refer to division of a machine in some usages but has also acquired the opposite meaning in other circumstances. Thus Sun's N1 virtualization software allows users to treat a large number of independent computers as a single set of resources.

2. For a fascinating look at the history and the conflicts between the batch oriented majority within IBM and the attempts to create an interactive environment that led to VM/CMS see <a href=http://pucc.princeton.edu/~melinda/25paper.pdf><u>VM and the VM Community: Past, Present, and Future</u></A> by Melinda Varian.

Data communications and user desktop maintenance and support are contracted to IBM Global Services.

Until late 1998 all offices used IBM 3278 terminals and RJE (remote job entry) on 3274 style controllers[1] to connect to mainframe services for the two most mission critical applications clusters: Customer Claims (query and processing) and Sales Management (order and quotation processing). At that time the company rolled out new IBM Netvista desktops to all offices and instituted a seven year agreement with IBM Global Services to provision, manage, and support the company's 15,000 desktops along with just over 1,700 NT/Windows 2000 servers located outside the data center.

The original Netvista PCs had SNA boards enabling the machines to emulate a 3278 style terminal. In the current round of desktop upgrades these boards are not being replaced because the latest generation mainframes now use TCP/IP (the internet standard networking protocol) and 327X emulation can therefore be handled entirely in software. Most applications requiring RJE or a 3279 terminal have, furthermore, had customized Windows 2000 clients developed to replace the old character interfaces.

The Windows 2000 or NT servers located outside the data center handle Microsoft Exchange, Lotus Notes (now IBM Domino), and departmental file and print sharing.

The data center itself started with a System 370 in 1973, was most recently rebuilt to house four S/390 processors in 1997, and is currently transitioning to a pair of IBM zSeries 2064-216 mainframes in a two way sysplex (clustered) configuration.

The five year capital lease on the new IBM mainframe gear includes[2]:

---

1. The 327X terminal/controller combination introduced with the System 370 in 1972 offered page mode, glass terminal, keyboard data entry and display. The controllers, really separate mini-computers, operated the terminals and managed things like restricting screen entry on specified fields to specified formats (e.g. "must be an integer").

   Page mode terminals transmitted and received information one page at a time, not one character or line at a time. Thus users typically filled out an on-screen form, and then sent the whole form -a bit like a page submit on a web form- for validation and processing.

2. IBM does not publish detailed pricing information for mainframes. The mainframe pricing information used here is from the tech-news.com website which tracks list pricing offered US customers.

concurrent user processes over a thirty day period without a hiccup - sure, that was run on a much larger machine (72 dual CPUs at 1200Mhz), but the principle applies to any Unix, including Linux.

In your context here there's no technology related reason not to combine several applications - like email, boot services, and file and print support, on each Linux machine.

If you planned things out carefully, you could probably use Linux to shutdown as many as two thirds of those 218 servers while cutting operating costs on the remainder - but you certainly can't consolidate everything to a single z800; it doesn't have the horsepower or the network connections, and, most importantly, it operates in a way that's backwards to what you want.

A more senior IT manager might have walked out of the meeting at this point; but the presence of the VP finance, to whom the IT director nominally reports, and a couple of senior user side managers probably intimidated IT's "*Manager: Consolidation Services*." So instead of responding with a bluff based on appeal to authority, he responded to this further provocation with an appeal to ignorance.

"It's a server Dammit!," he thundered, "you can't compare a PC to a mainframe; even the smallest mainframe is hundreds of times more powerful." He went on quite a bit, but the main burden was that I should be kicked out of the room never to darken their collective doorways again - and, of course, a single zSeries engine could easily replace hundreds and hundreds and hundreds of PC servers.

That was a challenge I didn't want to ignore - and I had remembered to bring one of my favorite overheads on this along. "The z900" I said, "is the biggest of the big iron at five million bucks for a 16 CPU, 64 GB unit, so lets compare that to a mid range Sun machine running Unix and then go from there to the baby mainframe- or z800 versus the PC.

"Bear in mind," I said, "that IBM hasn't participated in a public benchmark for the mainframe since 1998 - when a S/390 lost badly to a lightly loaded Sun 10000 on a SAP benchmark".[1]

---

1. See: http://www.sun.com/servers/highend/10000/performance/mips/

More Terminology
A LAN is a Local Area Network - meaning a network restricted to a single work site whether that's an office or a campus. By extension a WAN is a wide area network joining several LANs[a].
"T1" is an North American telco designation for a single cable provisioned to carry up to 24 concurrent telephone conversations but usually applied to equivalent rate data communications. Since a phone conversation has a digital equivalent rate of 56Kbits per second and requires an 8KBS control channel, a T1 provides for transmission at the rate of 1.536Mbs (=24 x 64) or about 192K characters per second.
North American data transmissions ratings go all the way up to OC/3 - about 173M characters per second. The comparable European designations start at E1 =40 phone line equivalents or about 2.5Mbits/sec.
ISDN (integrated services, digital network) puts a digital signal in place of the analog phone signal and comes in increments of 56Kbits/sec. but can use the 8Kbs control link as well. Thus an ISDN link described as "2b+c" has two 56K channels and a shared 16K control channel to deliver 128Kbs in throughput.
ADSL/SDSL are upgraded versions of ISDN using simpler wiring but broader spectrum spreading to achieve much higher throughput - usually in the range of 1MBS to 8MBS depending on the user's needs and budget.

a. The internet is a not a WAN. As discussed in Chapter 5, the core internet technology, TCP/IP, does not differentiate LANS and WANS.

As part of the new agreement AS/400 peripherals purchased from IBM during the lease period are being returned for credit on the new gear, thereby reducing the list price to a total of $4,213,896.72, pre-tax.

The new machine is a fully configured iSeries Model 890-0198 and includes:

1. 32 x 1.3GHz Power4 processors;
2. 192 GB of memory;
3. 6.5TB of disk storage (10K RPM, 17.54GB disks, RAID 5 in 22 enclosures);
4. Continuation of the company's IBM licenses, including OS/400 V5R2, DB2, RPG, and dozens of other tools or utilities;

You can, I tell them, drive a 2 inch nail with a sledgehammer if you want to - but it doesn't work the other way. Try to apply a leadership based approach in an all Windows environment and you'll find yourself doing the equivalent of trying to drive railway spikes with an 8 ounce hammer.

It's a difficult concept and acceptance isn't helped by the fact that most places they've seen Unix used have been disasters - usually precisely because IT management tried to treat it as just another constrained resource to be closely managed.

That brings us to infrastructure deployment options and costs. For this, I tell them that I'm going to compare the effect a Windows client-server approach with normal hierarchical systems management would have on the company to the effect of going with Unix, smart displays, and a leadership based approach to IT management.

Somewhere in Paul Strassman's book: The Politics of information Management (New Information Economics Press, New Canaan, Conn., 1995) he makes a throwaway comment to the effect that IBM lost the competition with Microsoft in part because they believed that the client-server revolution would require massive servers and so ramped up mainframe production instead of focusing on PCs and developing OS/2.

IBM was right of course --and they would have known, having abandoned client-server as unworkable during research on the Future Systems project of 1967-72. The consequence is that, despite the Windows myth of personal desktop device control, Cutter would have to implement very tight, centralized, control of desktop PCs to make a Windows architecture system work - thus leaving users with responsibility for, but no control over, desktop gear.

On the Unix side those desktop PCs get shown the door in favor of smart displays running server based applications - that means both architectures centralize processing, but Unix does it on one machine with a simple TCP/IP network while Windows managers need to co-ordinate rackmounts full of little servers with fairly cantankerous desktop gear and complex network switching schemes.

Smart displays offer both advanced graphics and stability -unlike PCs which have high failure rates and require frequent reboots and/or replacement, smart displays offer 300,000 hour mean time to failure rates, usually last five to eight years, and don't change no matter what happens to the applications or servers.

printer manager accessible by users on network attached personal computers. That, however, was enough to turn them into a billion dollar company by mid 1990.

The response IBM and Microsoft co-developed was a networking technology known as NetBEUI (NetBios Enhanced User Interface) which extended the PC's BIOS, stored in ROM and run at boot time, to make it "network aware" and so laid the foundations for a PC network architecture known as SMB/NetBIOS that is still in use in most corporate Microsoft Windows installations today.[1]

File and print sharing through a server led to demand for more shared services - particularly shared databases and, from this, the PC version[2] of client-server, started to evolve.

At about the same time, however, there was a growing rift between Microsoft and IBM. The success of GUIs on Atari, Macintosh, and Unix computers had demonstrated their potential and, of course, both IBM and Microsoft wanted to own the inevitable PC version.

As a result Microsoft, which had developed a partial windowing environment modelled on a GEM application for the Tandy 2000 (the only 1983/4 Intel machine with the power to run anything remotely like a GUI, in this case using a 12Mhz 80186) but dead ended it, revived that project and issued a partially graphical interface for DOS 3.1 called Windows 2.1[3] in December of 1987 - just over three years after announcing it and four years after abandoning Windows 1.0.

---

1. Windows 2000 is designed to work with TCP/P instead of Server Message Blocks (SMB) and NetBios but users tend to turn on SMB networking in order to perpetuate previous practice. Windows/XP deals with this by routing SMB over TCP/IP instead of using the NetBIOS but some users continue to install the older technologies.
2. The original IBM PC, the model 5100, had been designed as a client computer to be attached to the relational database engine, or future system, in 1972. At that time IBM cancelled the program in favor of the 370 architecture. By the time that the intended server was released as the System 38 in 1978/9, IBM researchers had deprecated the client-server plan as unworkably complex and so the System 38 was released with ordinary terminals instead of the client computers envisaged in the original system design.
3. Competitors included: VIsiON, from Visicorp in 1983, GEM (from DR) in mid 1984, TOPView, from iBM in early 1985, and DESQview from Quarterdeck Office Systems in mid 1985.

Case study: Happy Valley Tax Authority[1]

Note: The "I" in this scenario is that of a hapless systems consultant who didn't do his homework before setting off to meet the client. These events take place in mid 2000.

The Happy Valley Tax Authority, its staff and mandates, are fabrications but the situation presented, and the remedies offered, reflect the author's unhappy experience with real-world clients facing similar problems. This tax authority is imaginary, but the conditions, decisions, and outcomes described are broadly based on real events.

Background

The Happy Valley Tax Authority was set up as a regional co- operative to administer tax programs for local governments along an eighty mile stretch of highway. At the time of incorporation none of the players would agree to use the largest municipality's name for the joint effort and so the tax co-operative was named for a local tourist attraction: the Happy Valley Ranch. Although now rebuilt as a federally funded national heritage site, the ranch house had been built in the 1890s as a second generation cattle baron's imitation of an English country manor, acquired a rather different cultural status during prohibition, and been razed to the ground in an uprising of the local moral majority in 1957.

In the twelve years since start-up, the tax authority has acquired duties that go beyond simple property tax assessment and collection. One town has a hotel room tax, another provides school tax credits for couples with two or more children, while a third has an industrial land development program with both rebate and tax relief schemes to attract tenants. Today the authority collects 32 different levies from about 45,000 taxpayers; administers eight rebate, direct support, or tax relief programs, and collects tolls on one road bridge and two park entrances.

About eight months ago a town councilman with strong connections to the firm I usually work with got his council to hire the firm to do an operational audit of the authority's effectiveness and assess what value the town was getting for its continued support of the authority's mandate.

1. An earlier version of this parable appeared on Linuxworld.com in January, 2002.

Plan 9 was developed at Bell Labs (the research division of Lucent Technologies). In July 2000 the 3rd Edition was released as Open Source, and this is was quickly made available by Vita Nuova in the form of a bootable CD. The 4th Edition was released by Bell Labs in April 2002 and Vita Nuova is now shipping 4th Edition Boxed Sets and CDs. The developers of plan 9 include some of the original Unix designers including Rob Pike and Dennis Richie.

To the end user, Plan 9 bears a superficial resemblance to Unix, but the underlying operating system is very different. In particular, Plan 9 was designed as a network operating system, as opposed to Unix where networking support was added as an afterthought. Some of the standard Unix tools are available, but they have generally been either enhanced or completely rewritten.

Plan 9 has been ported to a number of architectures, including Intel x86, Sparc, Alpha, Power PC, and Arm[1].

Plan 9 reflects the future of Unix and Sun appears to be currently migrating Solaris, the technically most advanced commercial Unix variant, to full Plan 9 equivalence by concurrently expanding its OS vision outwards through network manageable resource allocation and inward to multi-CPU SMP on a chip.

Controls

The most important thing to note about the operation of a successful Unix data center is the relative invisibility of hierarchal control. Most people work in teams that form for specific projects and disband just as quickly. Management facilitates, it does not direct. Most new work is user initiated, annual budgets are more or less fixed but re-allocations are done on the fly. Most of the CoBiT[2] controls simply do not apply. Most of the process documentation expected by the traditional auditor has no purpose and does not exist.

---

1. http://www.vitanuova.com/plan9/origin.html. The name, meaning extended life, is from a passage in Dante's Inferno. The language of many jokes and puns in Plan 9 (a Unix tradition dating from McIlloy's prior work with Ken Iverson on APL), looks like Italian but is actually Fruilian.

2. See: http://www.isaca.org/ for details

4. Operations desk change orders had near real-time effect on the information provided at departure desks, on buses, and on-board the aircraft; and,
5. The airline could generally expect good regulatory co- operation on minor flight plan change,

then operations should be able to maintain an overall schedule optimized in terms of passenger needs while giving up a minimum in cycle time, fuel, or other operational penalties.

Overall optimization for passengers does not, of course, mean individual optimization. The occasional passenger may find herself temporarily re-routed to Alaska or stranded in Saskatoon, but the system would continually adjust itself to produce the best possible result for the majority of passengers the majority of the time.

As it happens that's also the best possible result for the airline because minimizing passenger waiting times and ground travel distances is usually the same as keeping the fastest gear, i.e. the airplanes, busiest earning money.

The critical design question is therefore clear: can the scheduling problem be formulated in such a way as to be sufficiently inclusive to generate usable results and yet solve in near real-time? - particularly if we define the latter as "generally less than one minute"?

Formulating the scheduling problem requires considerable expertise and an immense amount of data - both of which should be available. Since there is no compelling reason to believe that the problem cannot be properly formulated we'll assume that it can be and concentrate on options for solving it.

The actual problem size is difficult to predict, inclusion of passenger concerns will add less complexity than might be expected because many passenger constraints are linearly dependent - meaning that a full linear program might have 100 million rows and 200 million columns but the subset of interest will usually be at least an order of magnitude smaller on each dimension.

There are some givens in solving this. For example, the use of the Informix database with Tuxedo is a given in view of the reliability requirements for the transactions environment and the consequent need to keep the two data centers fully synchronized. Since this requirement also amounts to a Solaris specification for the primary transactions processing and database hosting jobs, the real architectural issue for the solver lies between:

up to switch jobs (applications) rather than processor tasks and a machine failure would therefore more commonly trigger an application restart than a simple processor switch.

Transaction Processing Monitors

In the "TPM", or transaction processing monitor, all arriving service requests are first captured by the TPM software and then passed to two or more servers - but only one response is passed back to the requestor. As a result the two or more servers involved are kept current during normal operations and activity logged on the running machine during a server outage can be replayed on recovery to bring the two systems back to a synchronized state.

Although this approach is now largely obsolete, both IBM's CICS, which acquired some TPM characteristics as early as 1970, and Tuxedo, now owned by Bea Systems, exemplify this approach.

TPM's are extremely reliable and very effective in environments where older applications and/or technologies need to be used and the data center either does not have control of the application source or does not have the skills and other resources needed to update the technology.

The more modern rendition of the same basic ideas involves database activity replication. In replication, users access an application that, in turn, accesses a database management system, or DBMS. The DBMS then replicates any transactions that affect stored data on a second server. Since the servers need only be connected via a standard TCP/IP network they can be in different countries, cities or states without affecting the second machine's ability to pick up the load anytime the first fails.

Since the first DBMS manages synchronization, SQL interpretation, query optimization, and transaction serialization, the load on the second server is far smaller than on the first. As a result people who have two or more large scale database applications usually split them between two servers using crossover replication to ensure continuity. Consider, for example, two application suites, say an ERP and an HRIS running on servers A and B respectively. Thus server A will have the primary ERP database and the HRIS replicant while server B will have the primary HRIS database and the ERP replicant. Should a primary host fail, the other server will then carry both loads using the surviving and replicant databases until the database management systems resync when the failed primary is brought back on line..

From a human perspective the IP addressing approach translates the human “winface.com” to the machine’s “216.94.64.226” and the rest of the protocol’s

Subnets and netmasks

Technically an IP address carries two meanings: it defines a network and it identifies hosts on that network. Thus a.b.c.0 identifies a network of up to 255 hosts (devices) all of whose IP addresses begin with a.b.c.

Suppose you have two workgroups, either physically separate or just people with different needs and agendas. Traffic is to flow from one of these groups to the other via a router but, within each group, all traffic is to flow to all machines - which will then inspect packets to see if they’re addressed to themselves.

Getting this to work requires creation of a subnet - breaking a network class like a.b.c.0 into two or more pieces.

Subnets are defined by reference to the underlying binary interpretation of the four numbers in an IP address. “To mask” means to apply a logical “AND” to the binary digits revealed when both the “subnet mask” and the base address of the IP class are expressed in binary.

Thus the 240 in the subnet mask 255.255.255.240 translates as “11110000” and has the literal meaning that only the last four bits of the address class expressed in binary define local hosts. Packets in which those bits are not set are therefore accepted by the router for transmission to other nets or subnets and packets in which those bits are set only circulate locally.

In practice this often gets quite esoteric - the “right answers” on the latest (January, 2003) CISCO Admin exam are technically incorrect[a] but most people don’t need to know the details, just that it can be done and what it does.

a. See: http://www.experts-exchange.com/Hardware/Routers/Q_20430870.html

operations are invisible.

From a machine perspective packets are routed to ethernet addresses using domain names and IP addresses only as direction finders. The actual linkage information needed to make this work is disseminated across the internet using router software acting under protocols like RIP2 - Routing Information Protocol V2- and within local networks using ARP - Address Resolution Protocol.

In 1969 the more widely accepted solutions to the problem of co-ordinating data usage among applications and development teams were accreted around the imposition of tight process control, requirements for additional documentation, the use of "structured" programming, and extensive data and process flow diagramming..

Procedural and non-procedural languages

The COBOL code fragment

```
PROCEDURE DIVISION.
BEGIN.
   SORT WorkFile ON ASCENDING KEY WSCORE
        INPUT PROCEDURE IS GetScore
        GIVING ScoresFile.
   STOP RUN.
```

embeds a call to a procedure called GetScore which reads each record, moves over 17 spaces from the left of the record, and writes the next four characters out as rows in WorkFile for later sorting and eventual output as ScoresFile.

In principle "procedural" languages like COBOL are *imperative* - specifying both what to do and how to do it. By extension "non-procedural" languages like SQL are *declarative* and only specify what to do, not how to do it.

In practice most "non procedural" languages, including SQL, use procedures and the real difference lies in the absence or presence of locally defined code like the example above to read and structure the input data before use.

Be aware, however, that many data processing professionals regard this distinction as important and meaningful so it's usually best to just smile and move on when they use the terminology.

Using a format adapted from that developed for the analysis of industrial processes by time and motion experts in the nineteen twenties and thirties, these diagramming methods generally linked data, user processes, and ultimately COBOL paragraphs using highly stylized pictographs.

By the mid seventies one of these, a dialect known as "entity relationship modelling," had become well established as the leading methodology. Ten years later the PC's ability to do on-screen graphics allowed E-R diagramming to

The biggest difference was in security servers, where Linux systems cost $91,000 over five years and Windows systems cost $70,000, IDC said. Next came print jobs, where a Linux server cost $107,000 over five years and a Windows server cost $87,000. In file sharing, Linux cost $114,000 to Windows' $99,000. In Web site jobs, Linux was less expensive at $31,000 to Windows' $32,000[1]

Bear in mind that the Linux software is free - and that no one has yet found a significant security bug in its file and print services, firewalls or email services.

Example Four

Total Cost of Application Ownership (the Tolly Group[2])

This study by the Tolly Group compares the cost of application delivery using two architectures:

1. Traditional client server with rackmounts in the data center and Windows 2000 desktop PC clients; and,
2. Traditional client server with rackmounts in the data center and Windows 2000 desktop PC clients running Citrix ICA clients.

Under the Citrix ICA deployment strategy the desktop client runs as a terminal to a Windows 2000 instance emulated on a server. Notice that this turns the desktop PC into a terminal while requiring extra server resources by duplicating the PC's computational and data storage functions on the server.

Citrix technology started out as means of showing Windows desktops on Unix smart displays, but later became a means of using Windows desktop machines ranging in scale from tablets running Microsoft Windows CE (known as thin clients) to high end PCs running Windows 2003/XP (known as fat clients) to remotely access simulations of other Windows desktop machines running on Windows servers.

This odd arrangement can offer value in two kinds of situations:

1. If the desktop application is extremely unstable, then moving it into a server co-located with the support organization can reduce support costs by reducing the time needed for service personnel to access and reboot the machine; and,

---

1. http://news.com.com/2100-1001-975938.html?tag=fd_top

2. http://www.precisiongroup.com/TCA1Whitepaper.pdf

What made the difference, in terms of the software and the conflicts generated, was the inclusion of the financial systems in the optimization package. From an optimization perspective this made sense because a lot of the data needed for inventory management and production cost optimization is naturally found in the financial systems. Adding direct links to the GL, Payroll, and Purchasing applications thus greatly simplified overall processing.

Within most companies, however, the sales, warehousing, and production people see themselves as the people who make the money. As a result MRPII systems are usually relatively easy to implement because people who see themselves contributing directly to the organization's success are generally happy to work with others seen as contributing equally.

In contrast, most production people see Finance people as freeloaders who just make, and try to enforce, rules while treating production people as layoff targets.

Thus getting production people to work with Finance on the implementation of an ERP system can be very difficult because the production people aren't usually willing to the cross the us-vs.-them cultural barrier that needs to be bridged between the producers and the bookkeepers.

Nevertheless ERP was the natural outgrowth of MRPII and nothing could stop the logic of its adoption once the hardware and software conditions for its arrival were in place.

On the hardware and cost side what made the difference was Digital's 1977 introduction of the VAX 780; IBM's later response with the revival of the future systems project as the System 38; and the 1980 introduction of the VAX 11/750.

The System 38 was not widely accepted, but the VAX was a major hit with academics and spin-off software developers. With low cos,t 32bit memory addressing, and good floating point performance, the Vax allowed developers to include on-line optimization in schedule and related MRP processing.

Thus when the small and cheap, but remarkably powerful, Vax 750 became available in late 1980, many developers were ready with advanced shop floor and warehouse management systems that could be linked together to form MRPII systems.

The consequent rapid growth in small system MRPII installations drove big company data centers to try to re-assert control by moving to consolidated ERP systems despite the technical and organizational complexities of doing so.

Is it paranoid to self-report as Windows/IIS while actually using Linux with Apache?

Consider this often corroborated story by John Lettice on theregister.co.uk (a Wintel oriented news site) for May 5/03:

Opera Software has accused Microsoft of deliberately engineering the MSN home page in order to make it look as if the Opera browser has a serious flaw in it. And the Norwegian company has published the results of an investigation which it says proves this.

Opera's techies downloaded the page using wget, in three different formats, identifying as Opera 7, MSIE and Netscape 7.01. The files sent to each browser are different, which is not necessarily suspicious, and the one sent to Opera7 has less content and is bigger than the one sent to IE. But that is not necessarily suspicious either.

Where it does get suspicious is when you look at the style sheets MSN sends to the browsers. The culprit, says Opera, is a 30 pixel value set on the margin property in the Opera style sheet. This instructs Opera to move list elements 30 pixels to the left of the parent, which means content moves off the side of its container, which means it looks like Opera is broken.

Opera tried to test whether or not this was deliberate by changing identification to the non-existent browser Oprah. This returns the IE style sheet, which works perfectly well in Opera. In Opera's view MSN is therefore looking specifically for "Opera" in the User-Agent string and sending it a broken style sheet. That, of course, could still be a mistake, as it's perfectly logical to send IE as the default if the browser can't be identified. But as there was no need for MSN to design an Opera-specific style sheet in the first place, one wonders.

** The Reg, incidentally, is regularly baffled by being unable to find stuff on Microsoft TechNet using Opera, because yet again we've forgotten that for some bizarre reason, lots of results from there in IE can equal no results at all in Opera. We've no idea whether this is a plot or not, either. Or indeed whether somebody might have fixed it by now.*

HTML pages presenting information about the product and an HTML order form enabling the customer to provide shipping, quantity, and payment information.

Numerics

A

B

C

G

H

I

R

S

T

U

Business Information Technology

This book is designed for people who interact with, but do not want to become, professional information systems staff. It is a management guide to information systems for auditors, executives, and business owners who want to understand the issues and know how to resolve them.

Most of the time these decisions call for an understanding of how and why various players support various options and what the consequences of going along with them are likely to be. As a result this book is as much a field guide to the players as it is a guide to their technologies.

As a manager, business owner, or management advisor you will often find yourself having to make tough decisions in situations where systems professionals people either hold conflicting opinions or seem to have considered only the alternative they're most comfortable with. Whether what's at stake is Sarbanes-Oxley compliance or a strategic business opportunity doesn't matter: it's their field and their professional vocabulary, but it's your decision and you, not they, will wear the result.

Preparing you for situations like these is the most fundamental thing this book is about. Not just helping you past the jargon barrier so their vocabulary doesn't leave you locked out; but getting right to the hard core of the issue: helping you understand the real relationship between their ideas about computing and the business problem your organization is trying to solve.

Read this book carefully and you'll understand that systems decisions aren't usually about bits, bytes, or megahertz, but about people and the way their ideas and interactions affect your organization's ability to do its job.

Talk the talk, but understand the people: that's the hidden lesson from ninty years of business data processing - understand it, and making smarter systems decisions will become part of your professional repetoire.

ISBN 0-9689004-0-2

52995

US $39.95
CDN $65.95