The Failure of Personalities to Generalize

Brett D. Fleisch
Computer Science Department
University of California, Riverside
Riverside, CA, 92521-0304

Abstract

IBM's adoption of operating system personalities was one of the most publicized issues in operating systems design. The basic premise of Workplace OS work was: 1) IBM would adopt and improve the CMU Mach 3.0 microkernel for use on PDAs, the desktop, and massively parallel machines, and 2) that several operating system personalities would execute on the microkernel platform concurrently. This architecture would provide users the best worlds as they switch between applications written for different operating systems. IBM would also benefit from significant cost savings by having one common platform for all product lines.

IBM's plans for use of the microkernel and multiple-personalities, as a unifying mechanism for a widely diverse set of hardware products, have failed. Here we examine why IBM's microkernel and multi-personality system was not successful from a technical and business standpoint. We also discuss Power Personal systems, which were introduced during these radical software changes, and then later abandoned.

1 Introduction

We learn great lessons from engineering failures. Petroski[4] discusses some of the largest physical failures such as the 1979 DC-10 crash in Chicago, when an improperly-maintained engine fell off during take-off, the Kansas City Hyatt Regency walkway collapse in 1981, the Tacoma Narrows Bridge failure that was captured on newsreel, and the well-known Challenger disaster. Petroski argues two points concerning these disasters. First, engineering is a human activity, prone to error and failure. Second, progress in engineering, like all human endeavors, comes by taking risks – that is how all engineering failures occur.

IBM is a company with a history of taking engineering risks while also having its share of failures. Brooks[2] discusses several IBM failures in his book which focuses on software engineering and project management. However, less documented IBM failures include the Microchannel Architecture (MCA) and the PS/2, the token ring, Transparent Computing Facility (AIX/TCF), and failure to achieve acceptance of SAA and SNA architectures. More recently IBM proposed Workplace Operating Systems(OS), the portable successor of OS/2 based on the IBM microkernel that supports multiple personalities. IBM envisioned Workplace OS would provide significant cost savings by having one common software platform for all product lines. In this paper, we outline the Workplace OS project and analyze why personalities failed to generalize.

2 System Structure

Figure 1(a) and (b)[3] shows the structures of the planned and the delivered microkernel for Workplace OS. The planned microkernel had multiple personalities. The *dominant personality* shown in Figure 1 was to be the primary personality presented to the user. The *alternate personalities* provide additional operating system services to the user concurrently. Alternate personalities permit execution of different OS applications from those provided by the dominant personality. Figure 1(a) also depicts *personality neutral services* (PNS's) that provide shared services for the multiple personalities that run in user-space. Networking, file systems, scheduling, paging and security services are PNSs. PNSs reduce the size and complexity of each personality, and maximize the amount of shared application code. For example, IBM argued that in some systems such as Windows-NT, extending the operating system scheduler to support real time scheduling would be difficult. In Workplace OS, a real-time scheduler could be easily substituted as a PNS component in user space. Figure 1(b) shows the microkernel that was eventually delivered; multiple personalities were not implemented.

In contrast to Workplace OS, Microsoft's Windows-NT supports five personalities: DOS, Windows, 32-bit Windows, OS/2 version 1.x, and a POSIX-compliant UNIX-like personality[1]. However, the goals for personalities in NT were substantially different from

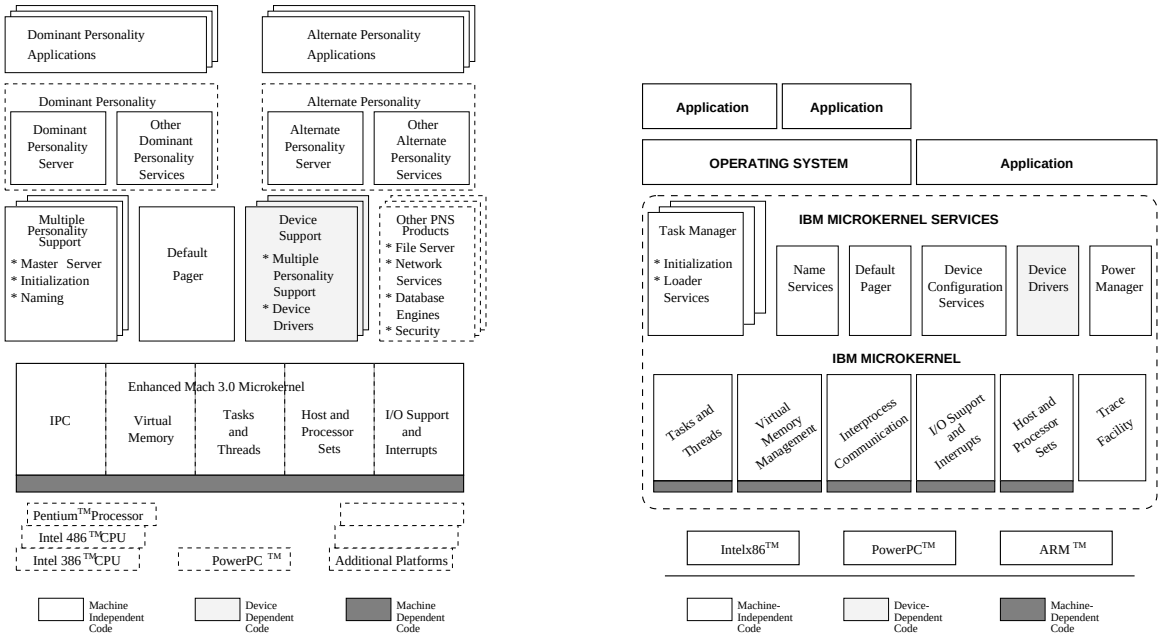

Figure 1: IBM microkernel: planned and delivered structure

Workplace goals. NT made no claims to unify a diverse set of products with a common software platform.

In conjunction with microkernel development, IBM planned to offer workstations based on the Motorola PowerPC which was touted as a more economical RISC machine that would execute personalities compatible with the Intel processor. A Power Personal Systems Division was established with development facilities in Austin, Texas, Boca Raton, Florida and Yamato, Japan. The Division defined the PowerPC systems standard and planned to sell systems that ran all personalities. The PReP (PowerPC Reference Platform) specification was created to specify the structure of the components for PowerPC machines[6]. In addition, IBM planned to push for acceptance of the microkernel as a new standard through the OSF Research Institute where many of the microkernel enhancement ideas originated.

3 Personalities

Personalities provide binary code compatibility for applications and therefore supply two components: the *application program interface* (or API) and the *application binary interface* (or ABI). The API provides the set of functions that the personality exports and thus the API contracts what functionality the personality implements. The ABI dictates the format of the binaries that the personality executes. A UNIX personality need only support the POSIX API for it to be called a UNIX personality. However, an AIX personality must execute native AIX binaries in addition to supporting the AIX API.

There were three key business reasons why personality support was important to IBM. First, DOS, Windows, and Mac programs are currently the dominant software on store shelves; obtaining shelf space for a new incompatible software variant would be difficult, even for IBM. Second, users have come to expect and are not about to discard significant software investments made on application programs, no matter how impressive the new operating system promises to be. Third, compatibility for the PowerPC was a key issue in order for users to adopt it. Success of the Power Personal System Division required software products provide ABI compatibility. Personalities were touted as a means to protect the user's software investment in addition to providing a necessary and familiar API.

4 Difficulties

The task of creating personalities for the microkernel was more challenging than originally envisioned, even for architectures with compatible instruction sets. The problems arose from two areas: structuring and binary compatibility. Both issues were important to Workplace's success, yet after the conceptual design was finished and the actual implementation began, numerous unforeseen problems arose.

Operating system structuring is a fundamental design issue and one of the most important reasons for adopting a microkernel. The IBM microkernel architecture specifies the need for clean separation of shared personality services to avoid duplication of the services in each personality. However, clean separation of shared services required substantial changes to personalities that wasn't well considered early in the design. For example, when IBM tried to integrate OS/2 with Windows, problems arose in that both Windows and OS/2 supported their own memory manager. Unable to modify windows code to use OS/2s memory management (although IBM licensed the source code from MS), IBM resorted to using the Windows memory manager within the OS/2 memory manager. The result was that Windows manipulations of memory could spill over into the OS/2 swap file. Similarly, display drivers needed to be substantially reworked to manage screen space between the two personalities.

A second major problem was the need to support incompatible executable code. The Microsoft solution in Windows-NT uses *virtual machines* that permits Intel code to be executed on a virtual Intel x86-based machine. IBM established a goal for all Power Personal systems, no matter what the operating system port, to run DOS/Windows binaries. This support required IBM provide two components: an API remapper and a binary translator. Specifically, at the API level, IBM supports WABI for API remapping, as well as developing Windows capability in OS/2. At the emulation level, IBM has developed an instruction set translator that compiles blocks of 80x86 code into blocks of PowerPc code on-the-fly and performs optimizations in the background. OS/2 for PowerPC had a split personality: an OS/2 personality and this DOS/Windows instruction set translator.

A third major problem was support for AIX, IBM's UNIX product. Early in the project, IBM announced plans to port the monolithic AIX to the PowerPC. The AIX port from the RS6000 Power architecture to PowerPC was easy since the PowerPC executes all Power instructions. However, an important goal of Workplace was to unify IBM's diverse product line. Therefore, AIX must operate on Workplace and industry observers regarded existence of an AIX personality essential for Workplace to be successful. On the other hand, support for an AIX personality on Workplace was a challenging problem because both OS/2 and the microkernel were Little Endian but AIX was Big Endian. An IBM press release indicated that IBM planned to address this problem by assigning an IBM Fellow to "crack the problem" along with a small team[1] of IBM's best research minds. However, supporting both OS/2 and AIX personalities simultaneously would require software emulation code to switch byte ordering on-the-fly which would substantially degrade AIX performance.

5 Project History

In January 1991 the project was conceived. The first presentation of IBM's new operating systems strategy was given to internal management with a chart referred to as the Grand Unification Theory of Operating Systems (or GUTS, for short). GUTS outlined how one microkernel would unify several operating systems with common "subsystems". At the end of 1991, a small team from Boca Raton, Florida and Austin, Texas had been formed to begin work on a version of the Mach Microkernel to support OS/2, the lead personality. In the summer of 1992, the prototype was underway and there was good progress. IBM successfully demo ed OS/2, DOS, DOS/Windows, and Unix running on the Mach microkernel at the Fall Comdex in 1992.

Soon after IBM announced plans to develop OS/2, DOS, and Unix as microkernel personalities for both PowerPC and Intel architectures[5]. At Comdex in 1993 IBM Chairman Louis Gerstner announced that the microkernel would not replace AIX. Instead, Gerstner told AIX customers that they would be able to migrate to Workplace OS, later. Subsequently, intense work on Workplace followed. IBM divided five major personality projects across separate divisions. Each division was required to support their own OS personality on the microkernel. In addition, a microkernel business unit was established to market the microkernel and create University relationships.

In May 1994 the division director of RISC Systems software announced plans to study an AIX personality for Workplace. A small internal research team of less than ten members was assembled and led by an IBM Research Fellow. The announcement was accompanied by information that a significant problem with development of the AIX personality was that of byte-ordering. IBM also reminded customers that monolithic AIX runs perfectly well on the PowerPC and that IBM needed time to address this difficult endian problem.

Seven months later in January 1995, IBM announced the AIX personality effort would be halted and an AIX personality for Workplace would not be built. Instead in February, IBM announced that it would offer a non-AIX personality for Workplace. The

[1] less than ten

new UNIX personality was intended for users that might otherwise find themselves rebooting between microkernel OS/2 and AIX.

In October 1995 IBM finally announced the general availability of Version 1 of the microkernel for the PowerPC. In the first year of release, IBM had several commercial vendors and Universities that adopted the microkernel including Digital Equipment Corporation, LG Electronics (Goldstar), Komatsu, Trusted Information Systems, and Bell-Northern Research. In addition, Universities such as Carnegie-Mellon University, Notre Dame, Oregon Graduate Institute, University of California at Irvine and Riverside, University of Texas, Arlington, Helsinki University of Technology, Tokyo University, and Cornell University were using IBM's microkernel for their research.

Later in October, media reports began to circulate that the PowerPC 620, which was the basis for the new improved desktop PowerPcs, was bug-ridden. Shortly after this news, IBM cancelled the Workplace project and folded the Power Personal Division. The latest and last release of the stillborn microkernel, version 2.0, was distributed to microkernel adopters early the following year. The final release supported the Motorola PowerPC, Intel x86, and the ARM(*Advanced RISC Machine*) embedded processor.

Table 1 summarizes assumptions versus outcomes associated with IBM microkernel personalities. Sources indicate that $2 billion dollars was spent on the Workplace project making it one of the most expensive operating system failures in history. Approximately one year after cancellation of Workplace OS, the IBM Boca Raton, Florida facility was closed.

6 Project Management

It is difficult to assess the quality of project management from an outside perspective. However, the combination of effective project management and high quality software engineering is essential for technical innovations to be successfully committed to products and is essential to the success of large software projects. Therefore many software companies in Japan and Europe have aspired to meet the international standard ISO 9000-3 which is the software-specific version of ISO 9001. The ISO standard proscribes specific *management activities* that guarantee quality during the software production cycle. The rationale for ISO standards argues that even great inventions will not benefit companies that have chaotic processes.

Despite the compelling need, we speculate[2] that software engineering and ISO standards may have taken a back seat in Workplace design which focused on significant new innovations to support cross-platform adapatability. However, effective project management was essential to Workplace since there were over 400 microkernel programmers[6] and 1500 OS/2 programmers[7] geographically distributed in different divisions. In contrast, Microsoft employed 10 people on Windows-NT and later another 40 or 50[1].

7 Lessons

To summarize, we offer the following lessons from analyzing Workplace:

1. It's easier to create a plan than a working operating system with multiple personalities. Overly ambitious (GUTSy) goals lead to failure.

2. IBM underestimated the difficulty in creating personalities. Each personality required extensive restructuring to support shared PNSs. These divisions were not always easy to delineate or implement as common subsystems. PNSs require that personality designers communicate effectively to reach common agreement on goals and implementation strategies for shared services.[3] Also, co-existence of ABI-compatible personalities with different "endianness" presented insurmountable problems.

3. IBM considered personalities late when compared to the microkernel where considerable effort on functionality, efficiency, and portability was placed early in the design. In contrast, in Windows-NT, personalities were considered early in the design and there was no emphasis on generalizing the NT microkernel for all products.

4. It was poor judgement for IBM to require all divisions to support the microkernel until more research had been conducted on its applicability across the diverse product lines, the applicability across existing software products, and to have one prototype with all *essential* personalities. In addition, IBM should have marketed personality-based PowerPCs *after* having the essential personalities prototyped. Associating the success of Power Personals with the success of personality development was unwise.

[2]IBM did not officially comment or provide assistance concerning this part of the evaluation

[3]Brooks [2] p.16-19 has a good discussion of the problems that arise on large software projects when there is a need to communicate between parties.

Personality Assumptions	Personality Outcomes
UNIX, OS/2, OS/400, Windows would run side-by-side on the microkernel as personalities.	OS/2 and Windows-NT ported to PowerPC without IBM microkernel.
PowerPC price/performance would attract customers along with a multi-personality operating environment.	Delays in introduction of software and hardware reduced performance advantage of PowerPC. Without personalities, PowerPC incompatibilities outweigh benefits.
IBM invites Apple to adopt the microkernel for Mac OS.	Apple refuses microkernel adoption and states that the microkernel has excessive resource requirements. About one month later IBM announces a marketing study indicating there is no customer demand for Mac OS on the PowerPC.
IBM announces a study of a Workplace AIX personality. IBM would address this problem by assigning an IBM Fellow to "crack the problem" along with IBM's best research minds.	AIX personality abandoned in January 1995 and IBM denies any original plans to support an AIX personality. IBM cited success of monolithic AIX on PowerPC and continued work on OS/2. Privately, some IBM executives admitted Workplace was dead.
In February 1995 IBM announces non-AIX personality for the microkernel described by IBM as a variant of AIX with a non-AIX API.	A new version of UNIX is not welcomed. The media expresses concerns whether Workplace will be successful.
In March 1995 IBM clings to late summer release date for OS/2.	In June 1995 IBM ships PowerPCs with monolithic AIX and Windows-NT. In July 1995 IBM quietly announces it will have Mac OS on PowerPCs next year.
In October 1995 reports circulate that the PowerPC 620 is bug-ridden.	IBM announces end of Power Personal Division and the end of the microkernel strategy.

Table 1: Assumptions and Outcomes in Personality Development

5. Both IBM and Microsoft used different approaches to preserve the user's software investment. Workplace announces an architecture-centric view to customers. IBM marketed a solution looking for adopters that was not a working solution. Microsoft, in contrast, has a user-centric view hiding details of the underlying approach and instead focusing on end-users. Microsoft appeals to the customers concern in NT with what applications run on the machine, not what is "under the covers" and how it works.

6. University research and prototypes could have saved IBM a substantial amount of money and exposed difficulties in generalizing concepts. Instead, funding was directed primarily to OSF and away from Universities who were viewed as microkernel adopters and personality developers rather than designers, implementers, participants and evaluators of the overall Workplace OS[4].

7. Sound software engineering practices and ISO techniques may have helped. Coordinated project management was a key issue for success of the project.

The Workplace project organizers seemed to lack good technical judgement. In [4]p.121, Petroski states:

> *The first and most indispensible design tool is judgment. It is engineering and design judgment that not only gets projects started in the right direction, but also keeps a critical eye on their progress and execution. Engineering judgment, by whatever name it may be called, is what from the very beginning of a conceptual design indentifies the key ele-*

[4]Privately, IBM continues to express doubts concerning the value of conducting University funded research favoring its own internal research labs.

ments to go to make up an analytical experimental model for exploration and development. It is judgement that separates the significant from the insignificant details, and it is judgment that catches analysis going astray....Judgement, in short, is what avoids mistakes, what catches errors, what detects flaws, and what anticipates and obviates failures. The single most important source of judgement lies in learning from one's mistakes and those of others.

We often learn *more* from designs that have failed than designs that have been successful. Operating systems conferences and systems-oriented journals would best serve designers by discussing design failures more often. In short, IBM's failure in Workplace is a joint academic failure; academics have placed overly heavy emphasis on reporting sucesses in operating system work rather than failure where we learn the most.

The blame for Workplace cannot be entirely attributed to not having sufficient academic forums for discussing failures. It was IBM's responsibility to learn from lessons of the past; the IBM/360 project and the failures reported by Brooks[2] seem to have repeated themselves in Workplace. How could IBM's previous failures seemingly repeat themselves in this new project? Petroski's analysis[4]p.166 is revealing:

> *The relationship between success and failure in design constitutes one of the fundamental paradoxes of engineering. The accumulation of successful experience tends to embolden designers to attempt ever more daring and ambitious projects, which seem almost invariably to culminate in colossal failure that takes everyone by surprise. In the wake of failure, on the other hand, there is generally a renewed conservatism that leads to new and untried design concepts that prove ironically to be eminently successful precisely because the design process proceeds cautiously from fundamentals and takes little for granted. As the new form evolves and matures, however, the cautions attendant upon its introduction tends to be forgotten, and a new period of optimism and hubris ensues.*

Engineering is an imprecise science because it relies on judgement. Human error is a major cause of design errors throughout history. Leaning from past errors, with case histories such as this analysis of Workplace OS, can greatly assist in preventing future errors.

8 Conclusion

IBM's failure is a story of overly grandiose GUTSy ambitions, ineffective project management, and failure of personalities to generalize beyond their tested previous scope. In this paper, we described the Workplace OS project and how it proceeded from grandiose vision to grandiose failure. We presented lessons we learned from examining and analyzing Workplace. We observed that good judgement is essential to successful designs and that failure-based paradigms sharpen our judgement. We offered insight concerning how past engineering failures operate in a cycle of success and failure and it is from failures we can learn the most.

Acknowledgements

I would like to thank Tom Payne for several valuable discussions which helped clarify these issues. Jeff Mogul suggested looking at Petroski's work for engineering insights.

References

[1] H. Custer. *Inside Windows NT.* Microsoft Press, Bellevue, WA, USA, 1993.

[2] J. Frederick P. Brooks. *The Mythical Man-Month: Essays on Software Engineering.* Addison-Wesley, 1982.

[3] IBM. IBM microkernel: Overview and Programming Guide Release 2.0, March 1996.

[4] H. Petroski. *Design Paradigms: Case Histories of Error and Judgement in Engineering.* Cambridge Press, 1994.

[5] Staff. Windows NT and Workplace OS:plug it in. *Byte Magazine*, 19(1):166, Jan. 1994.

[6] T. Thompson and B. Ryan. Apple, IBM Bring PowerPC To the DeskTop, Apr. 1994.

[7] N. Y. Times. IBM Set To Introduce its Latest Os/2 Software, September 25, 1996.