From: TPSYS::ERDEN "Bo Erden Production Systems Program Office 17-May-1994 12
52" 17-MAY-1994 12:51:34.00
To: HUMAN::CONKLIN
CC: ERDEN
Subj: U/I: QAR breakdown for the FUJI effort

Peter,

This is the QAR summary that we talked about.

Regards
Bo

FUJI SI Problem Management

Data from Feb 10, 1993 to May 2, 1994

QAR'S BY PRIORITY

Priority Levels	Total # of QAR's	% of QAR's
Showstoppers	25	15%
High	40	24%
Medium	58	35%
Low	42	26%
	165	100%

65 QAR's (or 39%) were showstopper or high priority QAR's.

73 QAR's (or 45%) were integration issues (approx 70% of these were either Showstopper or High)

QAR'S BY PRODUCT

DECNET OSI	19%
J-products	15%
ACMS	15%
RDB	7%
OPENVMS	5%
DEC C/VAX C	5%
FTAM	3%
DCE	3%
MCC	3%
QMA	3%
other* layered products	22%
	100.0%

QAR'S BY TYPE

CODE BUG	35%
INSTALLATION	26%
VERIFICATION	25%
PROD. DEFINITION	8%
DOCUMENTATION	1%
OTHER	5%

Total	100%

PATCHES/KITS APPLIED TO FUJI INTEGRATED PLATFORMS

The following numbers represent patches/patch kits delivered and installed on the Fuji platforms to fix reported QAR's. As each FIPS included a larger number of products, and more integration testing was done on later FIPS deliveries, it is expected to have an increasing number of patches.

FIPS-A	5
FIPS-B	7
FIPS-C	16
FIPS-D	14

Grand Total of Patches or full kits

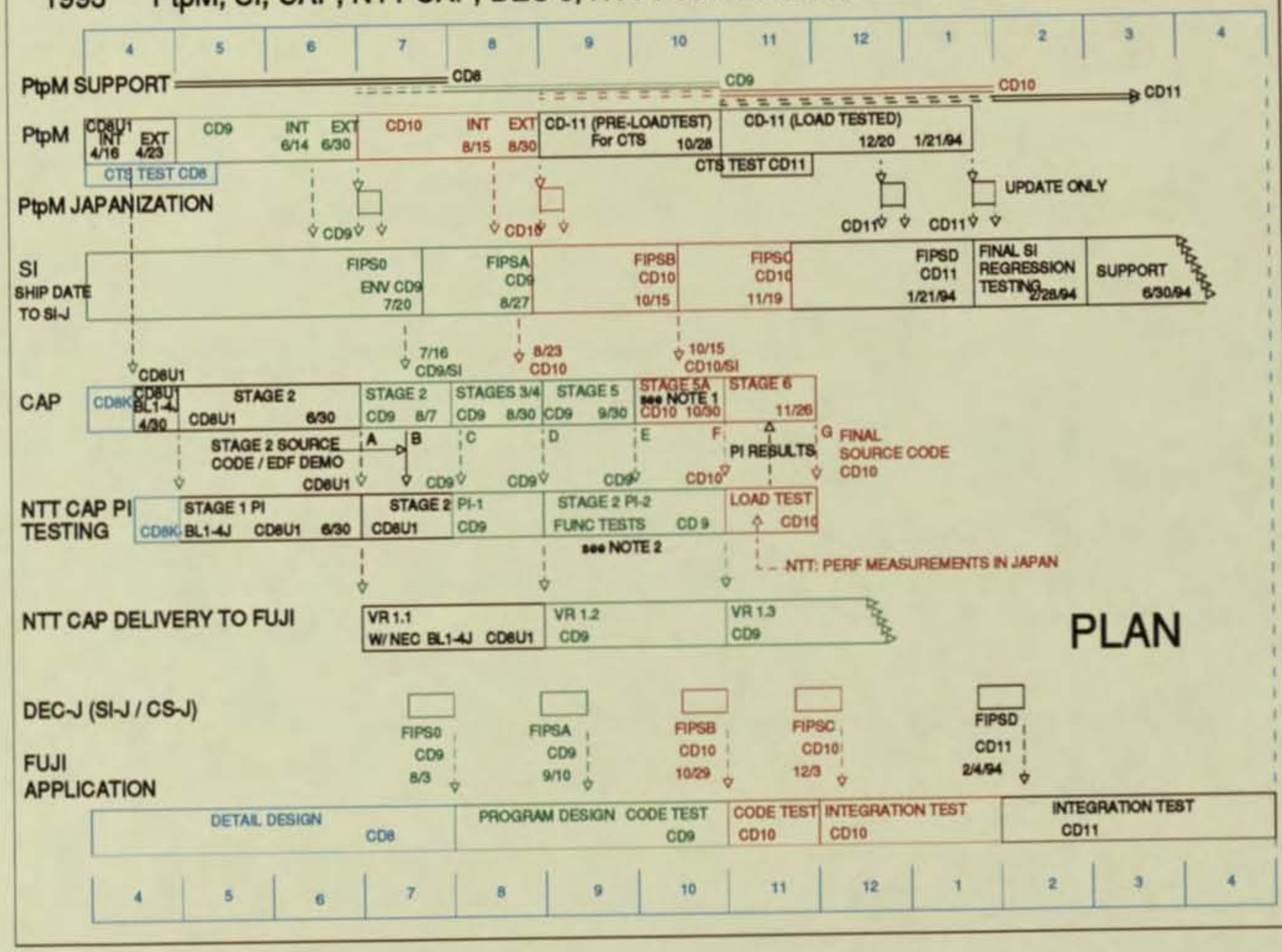
o ACMSxp	: 7
o DECnet OSI	: 2
o J-VMS	: 2
o DASL/ACMS	: 4
o DCE	: 3
o DECnet DNVAPP	: 2
o J-DECSCHEDULER	: 2
o J-DSM	: 1
o OSI Toolkit	: 5
o DCSC	: 2
o J-UCX	: 1
o J-SLS	: 2
o DECSCHEDULER	: 1
o J-DEC PRINTERSERVER	: 1
o J-DEC PRINT	: 1
o J-DECforms	: 1
o DEC Rdb	: 1
o CRL	: 1
o ZKO Debugger	: 1
o DEC C	: 1
o J-DEC Rdb	: 1

	42

Current Definitions Of Priorities For QAR Entry

- o Show Stopper/1 - One of the following conditions must exist for a problem to be classified as a show stopper problem:
  - o The user cannot use the system (i.e.; System Crash, System Loop, User Process Fails Frequently, Severe performance degradation).
  - o The user cannot use the product (i.e.; Regular Product Process Crash, Startup Procedure Failure).
- o High/2 - One of the following conditions must exist for a problem to be classified as a high priority problem:
  - o The user cannot use an important function of the product and has no work around.
  - o The user cannot use an important function of the product and has a work around. However, in order to apply the work around, the user must modify a significant portion of the customer application, making the work around impractical or overly expensive to implement.
- o Medium/3 - The following condition must exist for a problem to be classified as a medium priority problem:
  - o The user cannot use a function of the product and has a work around. The work around is easy to apply and cost effective.
- o Low/4 - The following condition must exist for a problem to be classified as a low priority problem:
  - o The user can use the functions of the product, however, the use of the functions causes some difficulty (i.e.; perceived performance problems, documentation errors, etc.)
- o Zero/5 - The following condition must exist for a problem to be classified as a zero priority problem:
  - o The user can use the functions of the product, however, there may be minor problems with the documentation (misspellings) or the user may have suggestions on the product functions.

1993 PtpM, SI, CAP, NTT CAP, DEC-J, NTT FUJI DELIVERY DEPENDENCIES 1994



4 5 6 7 8 9 10 11 12 1 2 3 4

PtpM SUPPORT CD8 CD9 CD10 CD11

PtpM CD8U1 INT 4/16 EXT 4/23 CD9 INT 6/14 EXT 6/30 CD10 INT 8/15 EXT 8/30 CD-11 (PRE-LOADTEST) For CTS 10/28 CD-11 (LOAD TESTED) 12/20 1/21/94

CTS TEST CD8 CTS TEST CD11

PtpM JAPANIZATION CD9 CD10 CD11 CD11 UPDATE ONLY

SI SHIP DATE TO SI-J FIPS0 ENV CD9 7/20 FIPSA CD9 8/27 FIPSB CD10 10/15 FIPSC CD10 11/19 FIPSD CD11 1/21/94 FINAL SI REGRESSION TESTING 2/28/94 SUPPORT 6/30/94

CD8U1 7/16 CD9/SI 8/23 CD10 10/15 CD10/SI

CAP CD8K CD8U1 BL1-4J 4/30 STAGE 2 CD8U1 6/30 STAGE 2 CD9 8/7 STAGES 3/4 CD9 8/30 STAGE 5 CD9 9/30 STAGE 5A see NOTE 1 CD10 10/30 STAGE 6 11/26

STAGE 2 SOURCE CODE / EDF DEMO A B C D E F G PI RESULTS FINAL SOURCE CODE CD10

CD8U1 CD9 CD9 CD9 CD10

NTT CAP PI TESTING CD8K STAGE 1 PI BL1-4J CD8U1 6/30 STAGE 2 CD8U1 PI-1 CD9 STAGE 2 PI-2 FUNC TESTS CD 9 LOAD TEST CD10

see NOTE 2

NTT: PERF MEASUREMENTS IN JAPAN

NTT CAP DELIVERY TO FUJI VR 1.1 W/NEC BL1-4J CD8U1 VR 1.2 CD9 VR 1.3 CD9

PLAN

DEC-J (SI-J / CS-J) FIPS0 CD9 8/3 FIPSA CD9 9/10 FIPSB CD10 10/29 FIPSC CD10 12/3 FIPSD CD11 2/4/94

FUJI APPLICATION DETAIL DESIGN CD8 PROGRAM DESIGN CODE TEST CD9 CODE TEST CD10 INTEGRATION TEST CD10 INTEGRATION TEST CD11

4 5 6 7 8 9 10 11 12 1 2 3 4

ASSSUMPTIONS:

1. PtpM Support for a given CD stops after delivery to NTT FUJI (only NTT CAP PI testing is supported)
2. CAP assumes all PtpM deliveries will be tested by US SI team prior to use by CAP (level of testing is being defined)
3. PtpM versions must not be used prior to delivery by SI
4. Because of weekend upgrades, a delay in PtpM delivery to CAP of 1-5 days = 1 week CAP delay; of 6-10 days = 2 week and so on.
5. No CAP development on CD11 planned at this time.

NOTE:

1. NTT Performance Tests to be held in U.S.
2. Fuji Functionality is #1 priority.

PtpM Support

= = = = =	Internal only
═════	External

PMF1

REV 1.1 25 June 1993

02/08/93
DIGITAL CONFIDENTIAL

Corporate Direction for Production Systems - Fuji Required Software Environments

SYSTEM INTEGRATED PRODUCTS		LAYERED PRODUCTS	
Stage 1: MIA Platform to support application development platform Delivery: 3/30/93 Appl Development Period: April '93 - August '93			
J-VMS V5.5-2M BL10	[1]	OSI AD Toolkit V1.0	[1,2]
XPG.4 Libraries (BL13/embedded)	[1]	DECtrace for VMS V1.2	[1]
VMS Special Kit (threads)	[1]	DEC DCE Developer's Kit V1.0+(EFT)	[1,2]
DECnet OSI VMS 5.5A(wave2+)	[1,2]	J-VAX Rdb V5.1 VMS (SQL)	[1,2]
FTAM V3.0	[1,2]	QMA (DECmessageQ AS) (BL10)	[1]
VAX DNS V1.1	[1]	DECmcc Director BMS V1.3 (EFT)	[1]
J-DEC TCP/IP Service V2.0	[1,2]	J-VAX COBOL V5.1	[1,2]
		J-PtpM (CD8)	[1,2]
resources.		J-DECwindows Motif V1.1	[1,2]
bug. 2 people		J-DSM V6.0	[1]
in DWest.		J-DASL V3.0	[1]
		Correlation Facility	[1]
		VAX C V3.2-044(required by PtpM)	[1]
		DTSS V010	[1]
		J-Debugger	[1]
PvR		ZKO-Debugger	[1]
J-MR X400 Gateway	[2,3]	J-DEC Mailworks	[2,3]
VAX Packetnet System	[2,4]	DECmcc TCP/IP SNMP AM *	[2,3]
DEC X25 Gateway 100/500	[2,4]	J-DEC C V1.3 (DEC C V1.3)	[2]
WANrouter Software V1.0	[2,4]	DEC FORTRAN V6.0	[2]
DECnet-VAX Extension V5.4	[2]	MIA Toolkit V1.0	[2]
VMS Common Agent OSI/CMIP	[2]	J-DEC GKS/VMS Runtime	[2]
		J-DECforms V1.4	[2]
		DECmcc OSI/CMIP AM V1.0	[2]
VAXcluster Software	[3]	VCS V1.4	[3]
VMS Volume Shadowing	[3]	DECmcc EMS V2.3	[3]
VAX RMS Journaling	[3]	DECscheduler V2.1	[3]
VAX Disk Striping Driver V2.0B	[3]	DEC Rdb Expert for VMS V2.0 $150K budget	[3]
DECperformance Solution V1.1	[3]	J-VAXset V10 (incl ITPU)	[3]
DEC File Optimizer for VMS V1.1	[3]	J-CDD repository for VMS V5.1	[3]
J-DCM	[3]	J-DECdesign V2.0	[3]
VAX DFS	[3]	J-DECwrite V2.0	[3]
VAX DQS	[3]	J-VAX FMS	[3]
VAX SLS V2.3	[3]	VTprint (J-product)	[3]
RSM Client/Server V2.3	[3]	PATHWORKS for DOS 9800 (J-product)	[3]

Notes: [1]	Critical for Fuji March Delivery (incl CAP)
[2]	Conformance item
[3]	Other complimentary software
[4]	Critical for CAP Delivery but not FUJI
*	packaged in DECmcc EMS but a separate conformance item

need V numbers, eg on DFS.

Stage 2: MIA Platform to application and integration testing
Delivery: 8/30/93
Testing Period: July '92 - March '94

J-VMS V5.5-2M Delta
All [3] items noted above
become part of run time environ

Layered Software Delta
STDL Conformance Test Suites
ACMS, DECmessageQ/AS upgrades
- VAX C
- DTSS V010

SI Stage 1 Plan Presentation

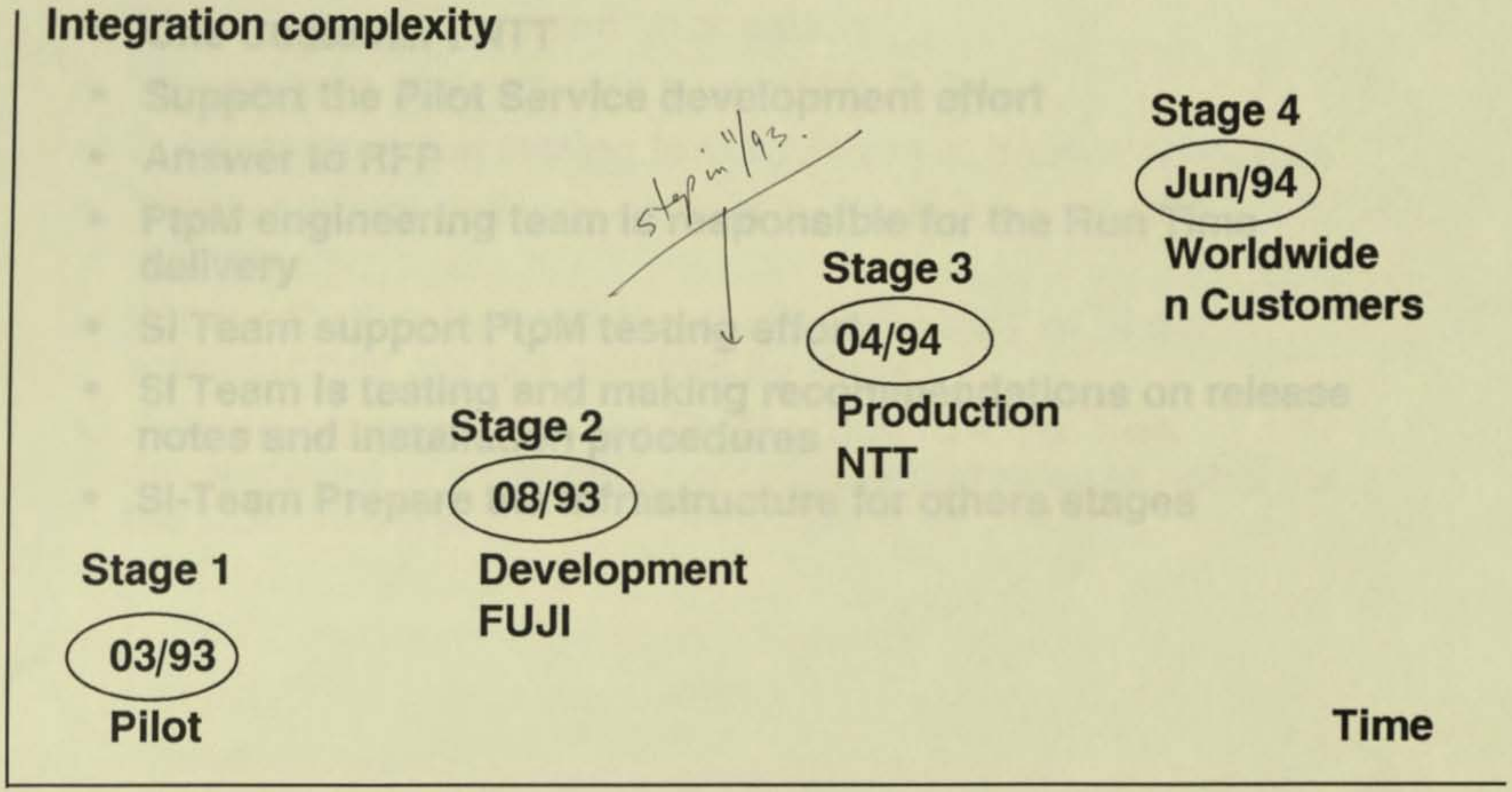
FUJI SI Plan

Agenda

- Overall Plan
- Stage 1 definition
- Goals & Non-Goals
- Assumptions
- Milestones
- Engineering dependencies
- Sand Box schedule
- Stage 2 definition

FUJI SI Plan

Overall Plan


Integration complexity

Stage 1
03/93
Pilot

Stage 2
08/93
Development
FUJI

Stage 3
04/94
Production
NTT

Stage 4
Jun/94
Worldwide
n Customers

Time

step in 11/93.

FUJI SI Plan

Stage 1
Definition

- One Customer : NTT
- Support the Pilot Service development effort
- Answer to RFP
- PtpM engineering team is responsible for the Run Time delivery
- SI Team support PtpM testing effort
- SI Team is testing and making recommendations on release notes and installation procedures
- SI-Team Prepare the infrastructure for others stages

FUJI SI Plan

Stage 1
Goals

- Maintain a configuration information
- Implement a problem mgt system for Fuji SI.
- Focus integration testing to verify fixes to known problems
- Test the install process defined by PtpM for their CD8 delivery
- When schedule permits test the installability of Stage 1 products that are not required by PtpM
- Provide information to the PtpM group to aid in their development of installation procedure and release notes for the CD8 delivery.

Min Perf level.

FUJI SI Plan

Stage 1
Non Goals

- Deliver Fuji platform installation package (done by PtpM).
- Define an installation process/procedure that can be replicated by NTT. (done by PtpM).
- Provide customized release notes (done by PtpM)
- Provide complete Media set for all products (done by PtpM)
- Characterize what has been tested and known problems (done by PtpM + inputs from SI-team)
- Guaranty 100 % functional coverage
- Performance , Stress , Conformance testing
- Multiple platform testing
- Integrated documentation , services , training ,...
- Integrated automated installation procedure
- Stage 2 planning or integration activities.

FUJI SI Plan

Stage 1
Assumptions

- Time Constraint Project
- Quick and easy access to engineering groups (DRI's)
- Content of Stage 1 is defined and closed

FUJI SI Plan

Stage 1
Milestones

(FOB Maynard)

- Jan 25 : Project start
- Jan 29 : Team in place & 1St Draft Plan ready for review
- Feb 01 : Sand Box PSIP1 in TWO
- Feb 05 : Platform ready in NIO
- Feb 08 : Publish Plan
- Feb 08 : Sand Box PSIP1.1 in TWO
- Feb 09 : Sand Box PSIP1.2 in TWO
- → Feb 10 : Finish Install PSIP2 on Red & Green lab systems
- Feb 26 : Sand Box update with PSIP2
- Mar 05 : Finish Integration of PSIP2
- Mar 05 : Installation of PSIP3 completed
- Mar 26 : Finish Integration testing for PSIP3

FUJI SI Plan

Stage 1
Engineering dependencies

- Follow Sand Box Schedule
- Feb 12 : Need to receive PSIP2 (test suites + Kit)
  - DCE , DECmcc , DECtrace
- Feb 19 : Need to receive PSIP2 (test suites + KIt)
  - RDB , DECforms
- Mar 02 : Need to receive PSIP3 lot 1 (test suites + Kit)
  - ZKO Debugger , CRL , DECset , VAX C , Cobol
- Mar 04 : Need to receive PSIP3 lot 2 (test suites + Kit)
  - DTSS , RMS Journaling , DCE
- Mar 09 : Need to receive PSIP3 lot 3 (test suites + Kit)
  - DNS , DECdtm , QMA

FUJI SI Plan

Stage 1
Sand Box schedule

- Feb 12 : DEcnet Osi Wave 2 + PSIP1.2
- Feb 19 : DCE , DECmcc , DECtrace PSIP1.2
- Feb 26 : RDB , DECforms PSIP1.2
- Mar 5 : CRL , ZKO Debugger,DEC C, Cobol, ,DECset PSIP2
- Mar 12 : DNS, DECmessage Q PSIP2
- Mar 19 : DTSS, RMS journaling , DECdtm, X25Gateway,Vax Packet system PSIP2
- Mar 26 : Contingency

FUJI SI Plan

Stage 2
Definition

- One Customer : NTT
- Support the development effort
- Goals :
  - Deliver FUJI development platform (not automated)
  - Support of March delivery
  - Deliver a support plan for readiness for Stage 2
  - Architecture for worldwide PSIP defined (Production System Integrated Platform)
    - » Process
    - » Model for software integration
    - » Model for documentation integration
    - » Model for entry criteria and component engineering group responsibilities
  - Study DEC past experience with integration efforts (NAS , DECstep, SITP,...)
  - Define areas of responsibility (DEC-J , PSPO , SI , ENg,..)

Production Systems:

Meeting the Technical Challenge

Peter F. Conklin
Technical Director, Operating Systems

January 20, 1993

digital

Production Systems Strategy

- Focus on software core competencies
  - Distribution, Networking, Open Systems Infrastructure, Standards, Multivendor Integration
- Multiplatform offerings
  - VMS, OSF, NT
- Integrated software
  - Building blocks for open systems
- ≤ 2 year window of opportunity for Digital

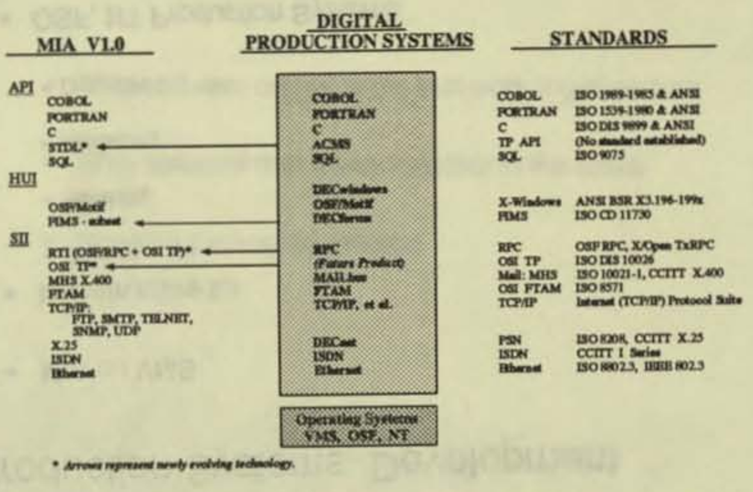
Production Systems Development

- MIA on VMS
- Infrastructure for
  - Security
  - Reliability
  - Availability
  - Distribution
- OSF, NT Production Systems

Multivendor Integration Architecture MIA

- Customer-driven initiative
- Infrastructure for Open Production Systems
- Moving into Standards Bodies, Consortia (CODASYL, X/Open, ETIS, INRIA, NMF)
- Foundation for Digital's open, distributed systems
- Digital Corporate Commitment

MIA, Production Systems, and Standards



COBOL	ISO 1989-1985 & ANSI
FORTRAN	ISO 1539-1980 & ANSI
C	ISO DIS 9899 & ANSI
TP API	(No standard established)
SQL	ISO 9075
X-Windows	ANSI BSR X3.196-199x
FIMS	ISO CD 11730
RPC	OSF RPC, X/Open TxRPC
OSI TP	ISO DIS 10026
Mail: MHS	ISO 10021-1, CCITT X.400
OSI FTAM	ISO 8571
TCP/IP	Internet (TCP/IP) Protocol Suite
PSN	ISO 8208, CCITT X.25
ISDN	CCITT I Series
Ethernet	ISO 8802.3, IEEE 802.3

* *Arrows represent newly evolving technology.*

Digital MIA Commitment

- NTT Fuji Project : $220M 1992-1996
  - $110M to Digital 1992-1996
  - $165M NOR to Digital over 12 years
- First major MIA project
- Awarded to Digital by NTT (November 24, 1992) (IBM, NEC also bid)

The Technical Challenge

- Meeting delivery commitments to NTT: March 1993; August 1993; April 1994; September 1994
- Creating infrastructure for development and deployment of distributed, open systems
- Large-scale software integration
- Coordinated release of software
- Quality production software

Fuji (<u>L</u>isting <u>M</u>aintenance <u>S</u>ystem, LMS) Schedule

• Software delivery began	- 12/92
• Procurement begins	- 3/93
• System delivery to NTT	- 8/93
• Fuji parallel running begins	- 4/94
• Fuji in Production	- 9/94

Fuji: Preliminary Configuration

- Hardware
  - 3 VAX 10000-630s in a cluster
  - 11 client sites (revised in Fuji Design Review, 1993.1.15)
  - Office environment
- Software: ≈ 60 components
  - 6 Version 1.0 products
  - 8 major enhancements
  - 500 GB database requirement (partitioned)
  - DECnet incompatibilities
  - Local language Systems

Fuji LSW

MIA - Compliant Software: 27 Components

APIs:	**5 components**	**(Application Program Interface)**
SII:	**15 components**	**(System Interconnection Interface)**
IIII:	**6 components**	**(Interenvironment Information Interchange Interface)**
HUI:	**1 component**	**(Human Interface)**

Non-MIA Software: 31 components

Fuji LSW
MIA-Compliant Components

APIs: **J-ACMS V4.0 DEC C V1.0 COBOL 5.0 FORTRAN 6.0 Rdb 4.2, 5.1**

SII:

DECnet	**DCE V1.0**
DECnet-VAX Extension V5.4	**WANrouter Softer V1.0**
X25 Gateway 100/500	**OSI AD Toolkit V1.0**
OSI/TP V1.0	**Packetnet System V4.3**
DECmcc TCP/IP SNMP AM	**J-MR X400 Gateway V2.2**
J-DECmailworks V1.1	**OSI CMIP**
J-DEC TCP/IP Service V2.0	**Digital leased line**
ISDN	**RTI**

IIII:

DECforms V2.0	**MIA Toolkit**	**VMS DEBUG**
3.5" 2DD Disk	**Mag tape**	**Schema DDL**

HUI: **J-DECwindows Motif V1.1**

Fuji Bid

J-ACMS V4.0	DEC C V1.0	DEC COBOL V5.0	DEC FORTRAN V6.0	Rdb 4.2, 5.1
DECnet	DCE V1.0	DECnet-VAX Extension V5.4	WANrouter Softer V1.0	X.25 Gateway 100/500
OSI AD Toolkit V1.0	OSI/TP V1.0	Packetnet System V4.3	DECmcc TCP/IP SNMP AM	J-MR X400 Gateway V2.2
J-DEC Mailworks V1.1	OSI CMIP	J-DEC TCP/IP Service V2.0	Digital leased line	ISDN
RTI	DECforms V2.0	MIA Toolkit	VMS DEBUG	3.5-inch 2DD Disk
Mag tape	Schema DDL	J-DECwindows Motif V1.1	J-VMS V5.5-2M	VAXcluster software
VMS Volume Shadowing	VMS RMS Journaling	DECperformance Solution (DC/PA/CPAC)	VAX SLS	VAX Disk Striping Driver
VAX Disk Striping Driver	DECscheduler	DEC File Optimizer	VAX DFS	VAX DQS
VAX Data Distributor	DECtrace for VMS 1.2	DEC Rdb Expert for VMS 2.0	DB Stress V1.1	RSM Server
RSM Client	DEC MCC EMS	J-VAXset V1.0	J-DECdesign (Plat)	J-DECdesign (Your)
J-DEC GKS/VMS (runtime) V5.0	J-CDD/Repository for VMS V5.1	DECmessageQ AS	J-ALL-IN-1	J-VAXnotes
J-DASL/DSM	J-DCM	Anet+	DECwrite V1.1	DECwrite V2.0

Note: Box encloses "MIA Components". Other components required for complete system.

Fuji Work by 8/93

- New Product

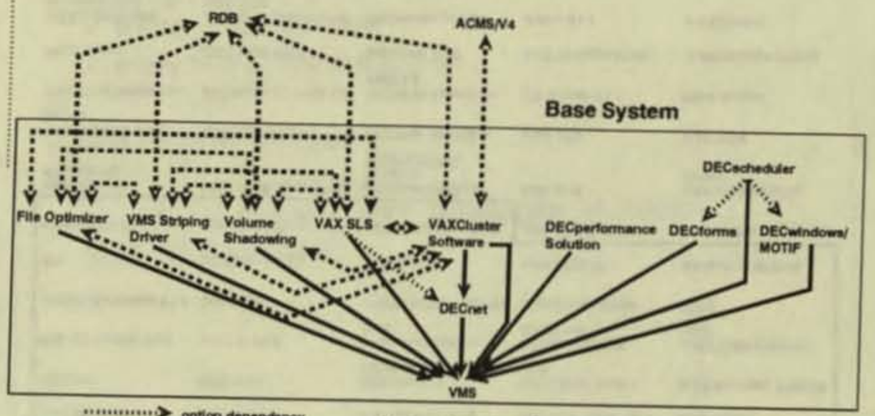
- New Version

J-ACMS V4.0	DEC C V1.0	DEC COBOL V5.0	DEC FORTRAN V6.0	Rdb 4.2, 5.1
DECnet	DCE V1.0	DECnet-VAX Extension V5.4	WANrouter Softer V1.0	X.25 Gateway 100/500
OSI AD Toolkit V1.0	OSI/TP V1.0	Packetnet System V4.3	DECmcc TCP/IP SNMP AM	J-MR X400 Gateway V2.2
J-DEC Mailworks V1.1	OSI CMIP	J-DEC TCP/IP Service V2.0	Digital leased line	ISDN
RTI	DECforms V2.0	MIA Toolkit	VMS DEBUG	3.5-inch 2DD Disk
Mag tape	Schema DDL	J-DECwindows Motif V1.1	J-VMS V5.5-2M	VAXcluster software
VMS Volume Shadowing	VMS RMS Journaling	DECperformance Solution (DC/PA/CPAC)	VAX SLS	VAX Disk Striping Driver
VAX Disk Striping Driver	DECscheduler	DEC File Optimizer	VAX DFS	VAX DQS
VAX Data Distributor	DECtrace for VMS 1.2	DEC Rdb Expert for VMS 2.0	DB Stress V1.1	RSM Server
RSM Client	DEC MCC EMS	J-VAXset V1.0	J-DECdesign (Plat)	J-DECdesign (Your)
J-DEC GKS/VMS (runtime) V5.0	J-CDD/Repository for VMS V5.1	DECmessageQ AS	J-ALL-IN-1	J-VAXnotes
J-DASL/DSM	J-DCM	Anet+	DECwrite V1.1	DECwrite V2.0

Note: Box encloses "MIA Components". Other components required for complete system.

Fuji / CAP RFP Configuration

Base System & Major LSW



option dependency

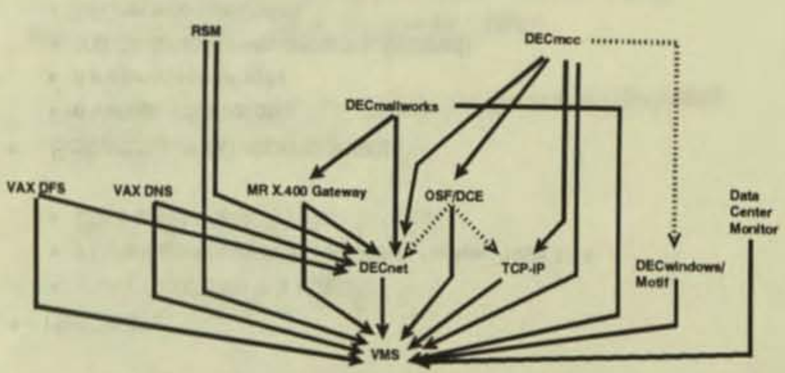
dependency

Dependencies & Interoperability

interoperability reqmnt

Fuji RFP Configuration

Communications & Management Infrastructure



option dependency

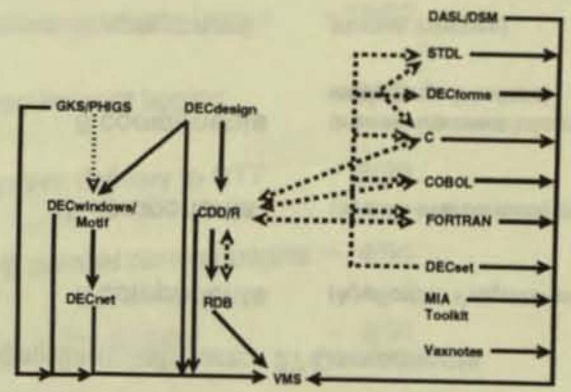
dependency

Dependencies & Interoperability

interoperability reqmnt

Fuji RFP Configuration

Development Environment



option dependency

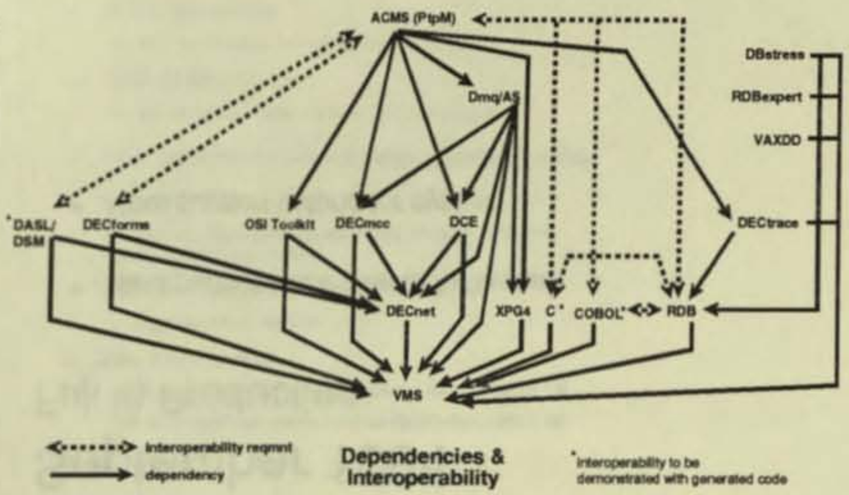
dependency

Dependencies & Interoperability

interoperability reqmnt

CAP/Fuji RFP Configuration

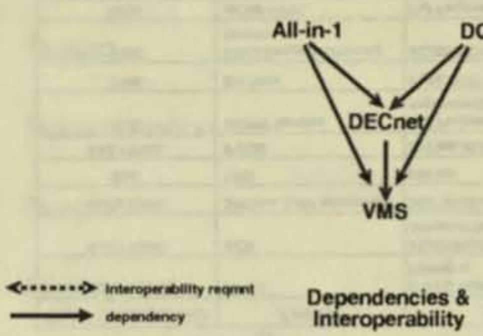
TP Runtime Environment



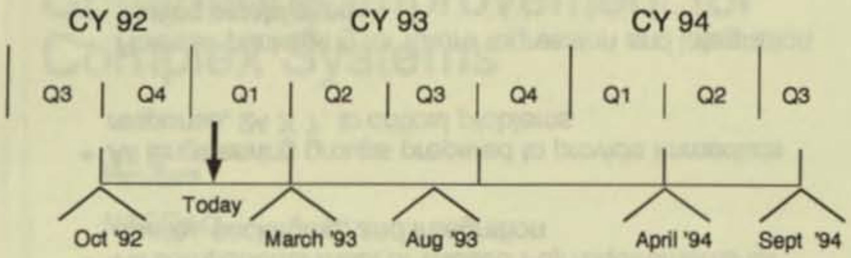
*interoperability to be demonstrated with generated code

CAP/ Fuji RFP Configuration

Office Environment



Fuji Timeline



October 92: NTT requires MIA compliance for bids

March 93: Development environment delivery to JRDC

August 93: System Delivery to NTT for final testing

April 94: NTT deploys Fuji application for parallel running

September 94: Fuji in Production

March 1993

Development Environment Delivery to JRDC

- All required MIA extensions incorporated and fully functional within components
- All components logically integrated (packaged)
- All components functionally tested
- Component quality, packaging, and integration tested
- Support commitment from all engineering groups (24 hour or less response time)
- Baseline packaging for future regression and integration testing established

August 1993

System Delivery to NTT for Final Testing

- All components logically integrated (packaged) and functionally tested
- All components meet or exceed Fuji requirements for quality, packaging, and integration
- All Engineering groups prepared to provide immediate response, 24 X 7, to critical problems
- Release packaging for future regression and integration testing established

April 1994

Fuji Deployment

- Parallel running with NTT's existing LMS system
- Any and all outstanding vendor declarations resolved

September 1994

Fuji in Production

- Meet performance criteria for system
- Meet support criteria for system

Known March Delivery Problems
Tip of the Iceberg

Digital Organization(s)	Problem	Software / Projects Affected
ISE/ TNSG	XPG.4	C RTL, PtpM, DECMessageQ Advanced Services, VMS Pakcaging
NAC / TNSG	DCE	ACMS-STDL, Threads, DECmcc, DECMessageQ Advanced Services
NAC / TNSG	DECnet - LSW mismatches	DEC MAILworks, FTAM
NAC	DNS	DECnet
NAC / TNSG	FTAM	DECnet, DEC MAILworks, Fuji Program
NAC	TCP/IP Services	Fuji Program, CAP (Customer procurement requirement)
TNSG	DECmcc	PtpM, DECMessageQ Advanced Services, CMIP
TNSG	DECMessageQ Advanced Services	ACMS-STDL
TNSG	ACMS-STDL	Fuji Program, CAP, CTS
TNSG / NAC	DEC MAILworks	DECnet, Fuji Project, CAP
VMS	VMS Common Agent	Fuji Program (Customer procurement requirement)
VMS / SDT	VMS Debugger, CF	PtpM, Fuji

Problems Being Worked On

- DCE Performance
  - Poor performance, no optimization of local RPC
- XPG.4 Coordination
  - Need to support many localities; ownership issues
- VMS DEBUG internationalization, correlation facility
- Threads Coordination
  - Threads built on different code drops; thread optimization; thread-safe code needed
- DECnet Coordination: wave2 and wave3
  - e.g. FTAM, MHS, OSI-TP
- Data management
  - 462 GB requirement (revised Fuji Design Review, 1993.1.15)
  - 14 DB 11.5GB each (revised Fuji Design Review, 1993.1.15)

Further Requirements:

Consistency in

- Testing
- Documentation
- Support
- Maintenance
- Packaging

Continuous Improvement for Complex Systems

Reuters

NASDAQ

Bombay Stock Exchange

Du Pont

GTE Cellular

Hong Kong Jockey Club

Engineering Process Improvement

Engineering needs to develop a System Model at the Applications (Total System) Level

(Un)Availability definition includes any time that the total system is running below the minimum application threshold in any dimension, including during "scheduled downtime"

Typical Complex Mission Critical system dimensions:

Performance — Availability — Database Size

Challenge to Digital

- Ship software components that work together now
- Ship software that works together in the future
- Develop packaging strategy

Resolution

- Focus on deliveries for NTT March 1993, August 1993
- Resolve identified problems
- Identify remaining problems, and resolve them
- Create infrastructure for development and deployment of distributed, open systems
- Large-scale software integration
- Coordinated release of software
- Quality release of software
- Plan future coordination to get ahead of industry trends

Working Group Mission from Production Systems BOD

Deliver Integrated Systems
with known characteristics which meet
Fuji's Mission Critical Attributes
by August 1993.

E

TANDEMCOMPUTERS

Why Do Computers Stop and What Can Be Done About It?

Greg:
Finally, a Duplexed (2-sided copy) = the
Thanks for tolerating
previous ~~2 P~~ illegible
[illegible]

Jim Gray

Technical Report 85.7
June 1985
PN87614

Why Do Computers Stop and What Can Be Done About It?

Jim Gray

June 1985

Tandem Technical report 85.7

Tandem TR 85.7

Why Do Computers Stop and What Can Be Done About It?

Jim Gray

June, 1985
Revised November, 1985

ABSTRACT

An analysis of the failure statistics of a commercially available fault-tolerant system shows that administration and software are the major contributors to failure. Various approachs to software fault-tolerance are then discussed -- notably process-pairs, transactions and reliable storage. It is pointed out that faults in production software are often soft (transient) and that a transaction mechanism combined with persistent process-pairs provides fault-tolerant execution -- the key to software fault-tolerance.

DISCLAIMER

This paper is not an "official" Tandem statement on fault-tolerance. Rather, it expresses the author's research on the topic.

An early version of this paper appeared in the proceedings of the German Association for Computing Machinery Conference on Office Automation, Erlangen, Oct. 2-4, 1985.

TABLE OF CONTENTS

Introduction

Computer applications such as patient monitoring, process control, online transaction processing, and electronic mail require high availability.

The anatomy of a typical large system failure is interesting: Assuming, as is usually the case, that an operations or software fault caused the outage, Figure 1 shows a time line of the outage. It takes a few minutes for someone to realize that there is a problem and that a restart is the only obvious solution. It takes the operator about 5 minutes to snapshot the system state for later analysis. Then the restart can begin. For a large system, the operating system takes a few minutes to get started. Then the database and data communications systems begin their restart. The database restart completes within few minutes but it may take an hour to restart a large terminal network. Once the network is up, the users take a while to refocus on the tasks they had been performing. After restart, much work has been saved for the system to perform -- so the transient load presented at restart is the peak load. This affects system sizing.

Conventional well-managed transaction processing systems fail about once every two weeks [Mourad], [Burman]. The ninety minute outage outlined above translates to 99.6% availability for such systems. 99.6% availability "sounds" wonderful, but hospital patients, steel mills, and electronic mail users do not share this view -- a 1.5 hour outage every ten days is unacceptable. Especially since outages usually come at times of peak demand [Mourad].

These applications require systems which virtually never fail -- parts of the system may fail but the rest of the system must tolerate failures and continue delivering service. This paper reports on the structure and success of such a system -- the Tandem NonStop system. It has MTBF measured in years -- more than two orders of magnitude better than conventional designs.

```
Minutes

  +   0    Problem occurs
  |   3    Operator decides problem needs dump/resart
  +   8    Operator completes dump
  |  12    OS restart complete, start DB/DC restart
  +  17    DB restart complete (assume no tape handling)
  |
  +  30    Network restart continuing
  |
  +  40    Network restart continuing
  |
  +  50    Network restart continuing
  |
  +  60    Network restart continuing
  |
  +  70    DC restart complete, begin user restart
  |
  +  80
  |
  +  90    User restart complete
```

Figure 1. A time line showing how a simple fault mushrooms into a 90 minute system outage.

Hardware Availability by Modular Redundancy

Reliability and availability are different: Availability is doing the right thing within the specified response time. Reliability is not doing the wrong thing.

Expected reliability is proportional to the Mean Time Between Failures (MTBF). A failure has some Mean Time To Repair (MTTR). Availability can be expressed as a probability that the system will be available:

$$\text{Availability} = \frac{\text{MTBF}}{\text{MTBF+MTTR}}$$

In distributed systems, some parts may be available while others are not. In these situations, one weights the availability of all the devices (e.g. if 90% of the database is available to 90% of the terminals, then the system is .9x.9 = 81% available.)

The key to providing high availability is to modularize the system so that modules are the unit of failure and replacement. Spare modules are configured to give the appearance of instantaneous repair -- if MTTR is tiny, then the failure is "seen" as a delay rather than a failure. For example, geographically distibuted terminal networks frequently have one terminal in a hundred broken. Hence, the system is limited to 99% availability (because terminal availability is 99%). Since terminal and communications line failures are largely independent, one can provide very good "site" availability by placing two terminals with two communications lines at each site. In essence, the second ATM provides instantaneous repair and hence very h availability. Moreover, they increase transaction throughput at locations with heavy traffic. This approach is taken by several high availability Automated Teller Machine (ATM) networks.

This example demonstrates the concept: modularity and redundancy allows one module of the system to fail without affecting the availability of the system as a whole because redundancy leads to small MTTR. This combination of modularity and redundancy is the key to providing continuous service even if some components fail.

Von Neumann was the first to analytically study the use of redundancy to construct available (highly reliable) systems from unreliable components [von Neumann]. In his model, a redundancy 20,000 was needed to get a system MTBF of 100 years. Certainly, his components were less reliable than transistors, he was thinking of human neurons or vacuum tubes. Still, it is not obvious why von Neumann's machines required a redundancy factor of 20,000 while current electronic systems use a factor of 2 to achieve very high availability. The key difference is that von Neumann's model lacked modularity, a failure in any bundle of wires anywhere, implied a total system failure.

VonNeumann's model had redundancy without modularity. In contrast, modern computer systems are constructed in a modular fashion -- a failure within a module only affects that module. In addition each module is constructed to be fail-fast -- the module either functions properly or stops [Schlichting]. Combining redundancy with modularity allows one to use a redundancy of two rather than 20,000. Quite an economy!

To give an example, modern discs are rated for an MTBF above 10,000 hours -- a hard fault once a year. Many systems duplex pairs of such discs, storing the same information on both of them, and using independent paths and controllers for the discs. Postulating a very leisurely MTTR of 24 hours and assuming independent failure modes, the MTBF of this pair (the mean time to a double failure within a 24 hour window) is over 1000 years. In practice, failures are not quite independent, but the MTTR is less than 24 hours and so one observes such high availability.

Generalizing this discussion, fault-tolerant hardware can be constructed as follows:

- * Hierarchically decompose the system into modules.
- * Design the modules to have MTBF in excess of a year.
- * Make each module fail-fast -- either it does the right thing or stops.
- * Detect module faults promptly by having the module signal failure or by requiring it to periodically send an I AM ALIVE message or reset a watchdog timer.

- Configure extra modules which can pick up the load of failed modules. Takeover time, including the detection of the module failure, should be seconds. This gives an apparent module MTBF measured in millennia.

The resulting systems have hardware MTBF measured in decades or centuries.

This gives fault-tolerant hardware. Unfortunately, it says nothing about tolerating the major sources of failure: software and operations. Later we show how these same ideas can be applied to gain software fault-tolerance.

An Analysis of Failures of a Fault-Tolerant System

There have been many studies of why computer systems fail. To my knowledge, none have focused on a commercial fault-tolerant system. The statistics for fault-tolerant systems are quite a bit different from those for conventional mainframes [Mourad]. Briefly, the MTBF of hardware, software and operations is more than 500 times higher than those reported for conventional computing systems -- fault-tolerance works. On the other hand, the ratios among the sources of failure are about the same as those for conventional systems. Administration and software dominate, hardware and environment are minor contributors to total system outages.

Tandem Computers Inc. makes a line of fault-tolerant systems [Bartlett] [Borr 81, 84]. I analyzed the causes of system failure reported to Tandem over a seven month period. The sample set covered more than 2000 systems and represents over 10,000,000 system hours or over 1300 system years. Based on interviews with a sample of customers, I believe these reports cover about 50% of all total system failures. There is under-reporting of failures caused by customers or by environment. Almost all failures caused by the vendor are reported.

During the measured period, 166 failures were reported including one fire and one flood. Overall, this gives a system MTBF of 7.8 years reported and 3.8 years MTBF if the systematic under-reporting is taken into consideration. This is still well above the 1 week MTBF typical of conventional designs.

By interviewing four large customers who keep careful books on system outages, I got a more accurate picture of their operation. They averaged a 4 year MTBF (consistent with 7.8 years with 50% reporting). In addition, their failure statistics had under-reporting in the expected areas of environment and operations. Rather than skew the data by multiplying all MTBF numbers by .5, I will present the analysis as though the reports were accurate.

About one third of the failures were "infant mortality" failures -- a product having a recurring problem. All these fault clusters are related to a new software or hardware product still having the bugs shaken out. If one subtracts out systems having "infant" failures or non-duplexed-disc failures, then the remaining failures, 107 in all, make an interesting analysis (see table 1).

First, the system MTBF rises from 7.8 years to over 11 years.

System administration, which includes operator actions, system configuration, and system maintenance was the main source of failures -- 42%. Software and hardware maintenance was the largest category. High availability systems allow users to add software and hardware and to do preventative maintenance while the system is operating. By and large, online maintenance works VERY well. It extends system availability by two orders of magnitude. But occasionally, once every 52 years by my figures, something goes wrong. This number is somewhat speculative -- if a system failed while it was undergoing online maintenance or while hardware or software was being added, I ascribed the failure to maintenance. Sometimes it was clear that the maintenance person typed the wrong command or unplugged the wrong module, thereby introducing a double failure. Usually, the evidence was circumstantial. The notion that mere humans make a single critical mistake every few decades amazed me -- clearly these people are very careful and the design tolerates some human faults.

Software faults were a major source of system outages -- 25% in all. Tandem supplies about 4 million lines of code to the customer. Despite careful efforts, bugs are present in this software. In addition, customers write quite a bit of software. Application software faults are probably under-reported here. I guess that only 30% are reported. If that is true, application programs contribute 12% to outages and software rises to 30% of the total.

Next come environmental failures. Total communications failures (losing all lines to the local exchange) happened three times, in addition, there was a fire and a flood. No outages caused by cooling or air conditioning were reported. Power outages are a major source of failures among customers who do not have emergency backup power (North American urban power typically has a 2 month MTBF). Tandem systems tolerate over 4 hours of lost power without losing any data or communications state (the MTTR is almost zero), so customers do not generally report minor power outages (less than 1 hour) to us.

Given that power outages are under-reported, the smallest contributor to system outages was hardware, mostly discs and communications controllers. The measured set included over 20,000 discs -- over 100,000,000 disc hours. We saw 19 duplexed disc failures, but if one subtracts out the infant mortality failures then there were only 7 duplexed disc failures. In either case, one gets an MTBF in excess of 5 million hours for the duplexed pair and their controllers. This approximates the 1000 year MTBF calculated in the earlier section.

Implications of the Analysis of MTBF

The implications of these statistics are clear: the key to high-availability is tolerating operations and software faults.

Commercial fault-tolerant systems are measured to have a 73 year hardware MTBF (table 1). I believe there was 75% reporting of outages caused by hardware. Calculating from device MTBF, there were about 50,000 hardware faults in the sample set. Less than one in a thousand resulted in a double failure or an interruption of service. Hardware fault-tolerance works!

In the future, hardware will be even more reliable due to better design, increased levels of integration, and reduced numbers of connectors.

By contrast, the trend for software and system administration is not positive. Systems are getting more complex. In this study, administrators reported 41 critical mistakes in over 1300 years of operation. This gives an operations MTBF of 31 years! Operators certainly made many more mistakes, but most were not fatal. These administrators are clearly very careful and use good practices.

The top priority for improving system availability is to reduce administrative mistakes by making self-configured systems with minimal maintenance and minimal operator interaction. Interfaces that ask the operator for information or ask him to perform some function must be simple, consistent and operator fault-tolerant.

The same discussion applies to system maintenance. Maintenance interfaces must be simplified. Installation of new equipment must have fault-tolerant procedures and the maintenance interfaces must simplified or eliminated. To give a concrete example, Tandem's newest discs have no special customer engineering training (installation is "obvious") and they have no scheduled maintenance.

A secondary implication of the statistics is actually a contradiction:

* New and changing systems have higher failure rates. Infant products contributed one third of all outages. Maintenance caused one third of the remaining outages. A way to improve availability is to install proven hardware and software, and then leave it alone. As the adage says, "If it's not broken, don't fix it".

* On the other hand, a Tandem study found that a high percentage of outages were caused by "known" hardware or software bugs, which had fixes available, but the fixes were not yet installed in the failing system. This suggests that one should install software and hardware fixes as soon as possible.

There is a contradiction here: never change it and change it ASAP! By consensus, the risk of change is too great. Most installations are slow to install changes, they rely on fault-tolerance to protect them until the next major release. After all, it worked yesterday, so it will probably work tomorrow.

Here one must separate software and hardware maintenance. Software fixes outnumber hardware fixes by several orders of magnitude. I believe this causes the difference in strategy between hardware and software maintenance. One cannot forego hardware preventative maintenance -- our studies show that it may be good in the short term but it is disasterous in the long term. One must install hardware fixes in a timely fashion. If possible, preventative maintenance should be scheduled to minimize the impact of a possible mistake. Software appears to be different. The same study recommends installing a software fix only if the bug is causing outages. Otherwise, the study recommends waiting for a major software release, and carefully testing it in the target environment prior to installation. Adams comes to similar conclusions [Adams], he points out that for most bugs, the chance of "rediscovery" is very slim indeed.

The statistics also suggest that if availability is a major goal, then avoid products which are immature and still suffering infant mortality. It is fine to be on the leading edge of technology, but avoid the bleeding edge of technology.

The last implication of the statistics is that software fault-tolerance is important. Software fault-tolerance is the topic of the rest of the paper.

Fault-tolerant Execution

Based on the analysis above, software accounts for over 25% of system outages. This is quite good -- a MTBF of 50 years! The volume of Tandem's software is 4 million lines and growing at about 20% per year. Work continues on improving coding practices and code testing but there is little hope of getting ALL the bugs out of all the software. Conservatively, I guess one bug per thousand lines of code remains after a program goes through design reviews, quality assurance, and beta testing. That suggests the system has several thousand bugs. But somehow, these bugs cause very few system failures because the system tolerates software faults.

The keys to this software fault-tolerance are:

- * Software modularity through processes and messages.

- * Fault containment through fail-fast software modules.

- * Process-pairs to tolerate hardware and transient software faults.

- * Transaction mechanism to provide data and message integrity.

- * Transaction mechanism combined with process-pairs to ease exception handling and tolerate software faults.

This section expands on each of these points.

Software modularity through processes and messages

As with hardware, the key to software fault-tolerance is to hierarchically decompose large systems into modules, each module being a unit of service and a unit of failure. A failure of a module does not propagate beyond the module.

There is considerable controversy about how to modularize software. Starting with Burroughs' Esbol and continuing through languages like Mesa and Ada, compiler writers have assumed perfect hardware and contended that they can provide good fault isolation through static compile-time type checking. In contrast, operating systems designers have advocated run-time checking combined with the process as the unit of protection and failure.

Although compiler checking and exception handling provided by programming languages are real assets, history seems to have favored the run-time checks plus the process approach to fault containment. It has the virtue of simplicity -- if a process or its processor misbehaves, stop it. The process provides a clean unit of modularity, service, fault containment, and failure.

Fault containment through fail-fast software modules.

The process approach to fault isolation advocates that the process software module be fail-fast, it should either function correctly or it should detect the fault, signal failure and stop operating.

Processes are made fail-fast by defensive programming. They check all their inputs, intermediate results, outputs and data structures as a matter of course. If any error is detected, they signal a failure and stop. In the terminology of [Cristian], fail-fast software has small fault detection latency.

The process achieves fault containment by sharing no state with other processes; rather, its only contact with other processes is via messages carried by a kernel message system.

Software faults are soft -- the Bohrbug/Heisenbug hypothesis

Before developing the next step in fault-tolerance, process-pairs, we need to have a software failure model. It is well known that most hardware faults are soft -- that is, most hardware faults are transient. Memory error correction and checksums plus retransmission for communication are standard ways of dealing with transient hardware faults. These techniques are variously estimated to boost hardware MTBF by a factor of 5 to 100.

I conjecture that there is a similar phenomenon in software -- most production software faults are soft. If the program state is reinitialized and the failed operation retried, the operation will usually not fail the second time.

If you consider an industrial software system which has gone through structured design, design reviews, quality assurance, alpha test, beta test, and months or years of production, then most of the "hard" software bugs, ones that always fail on retry, are gone. The residual bugs are rare cases, typically related to strange hardware conditions (rare or transient device fault), limit conditions (out of storage, counter overflow, lost interrupt, etc,), or race conditions (forgetting to request a semaphore).

In these cases, resetting the program to a quiescent state and reexecuting it will quite likely work, because now the environment is slightly different. After all, it worked a minute ago!

The assertion that most production software bugs are soft -- Heisenbugs that go away when you look at them -- is well known to systems programmers. Bohrbugs, like the Bohr atom, are solid, easily detected by standard techniques, and hence boring. But Heisenbugs may elude a bugcatcher for years of execution. Indeed, the bugcatcher may perturb the situation just enough to make the Heisenbug disappear. This is analogous to the Heisenberg Uncertainty Principle in Physics.

I have tried to quantify the chances of tolerating a Heisenbug by reexecution. This is difficult. A poll yields nothing quantitative. The one experiment I did went as follows: The spooler error log of several dozen systems was examined. The spooler is constructed as a collection of fail-fast processes. When one of the processes detects a fault, it stops and lets its brother continue the operation. The brother does a software retry. If the brother also fails, then the bug is a Bohrbug rather than a Heisenbug. In the measured period, one out of 132 software faults was a Bohrbug, the remainder were Heisenbugs.

A related study is reported in [Mourad]. In MVS/XA functional recovery routines try to recover from software and hardware faults. If a software fault is recoverable, it is a Heisenbug. In that study, about 90% of the software faults in system software had functional recovery routines (FRRs). Those routines had a 76% success rate in continuing system execution. That is, MVS FRRs extend the system software MTBF by a factor of 4.

It would be nice to quantify this phenomenon further. As it is, systems designers know from experience that they can exploit the Heisenbug hypothesis to improve software fault-tolerance.

Process-pairs for fault-tolerant execution

One might think that fail-fast modules would produce a reliable but unavailable system -- modules are stopping all the time. But, as with fault-tolerant hardware, configuring extra software modules gives a MTTR of milliseconds in case a process fails due to hardware failure or a software Heisenbug. If modules have a MTBF of a year, then dual processes give very acceptable MTBF for the pair. Process triples do not improve MTBF because other parts of the system (e.g., operators) have orders of magnitude worse MTBF. So, in practice fault-tolerant processes are generically called process-pairs. There are several approaches to designing process-pairs:

Lockstep: In this design, the primary and backup processes synchronously execute the same instruction stream on independent processors [Kim]. If one of the processors fails, the other simply continues the computation. This approach gives good tolerance to hardware failures but gives no tolerance of Heisenbugs. Both streams will execute any programming bug in lockstep and will fail in exactly the same way.

State Checkpointing: In this scheme, communication sessions are used to connect a requestor to a process-pair. The primary process in a pair does the computation and sends state changes and reply messages to its backup prior each major event. If the primary process stops, the session switches to the backup process which continues the conversation with the requestor. Session sequence numbers are used to detect duplicate and lost messages, and to resend the reply if a duplicate request arrives [Bartlett]. Experience shows that checkpointing process-pairs give excellent fault-tolerance (see table 1), but that programming checkpoints is difficult. The trend is away from this approach and towards the Delta or Persistent approaches described below.

Automatic Checkpointing: This scheme is much like state checkpoints except that the kernel automatically manages the checkpointing, relieving the programmer of this chore. As described in [Borg], all messages to and from a process are saved by the message kernel for the backup process. At takeover, these messages are replayed to the backup to roll it forward to the primary process' state. When substantial computation or storage is required in the backup, the primary state is copied to the backup so that the message log and replay can be discarded. This scheme seems to send more data than the state checkpointing scheme and hence seems to have high execution cost.

Delta Checkpointing: This is an evolution of state checkpointing. Logical rather than physical updates are sent to the backup [Borr 84]. Adoption of this scheme by Tandem cut message traffic in half and message bytes by a factor of 3 overall [Enright]. Deltas have the virtue of performance as well as making the coupling between the primary and backup state logical rather than physical. This means that a bug in the primary process is less likely to corrupt the backup's state.

Persistence: In persistent process-pairs, if the primary process fails, the backup wakes up in the null state with amnesia about what was happening at the time of the primary failure. Only the opening and closing of sessions is checkpointed to the backup. These are called stable processes by [Lampson]. Persistent processes are the simplest to program and have low overhead. The only problem with persistent processes is that they do not hide failures! If the primary process fails, the database or devices it manages are left in a mess and the requestor notices that the backup process has amnesia. We need a simple way to resynchronize these processes to have a common state. As explained below, transactions provide such a resynchronization mechanism.

Summarizing the pros and cons of these approaches:

* Lockstep processes don't tolerate Heisenbugs.

* State checkpoints give fault-tolerance but are hard to program.

* Automatic checkpoints seem to be inefficient -- they send a lot of data to the backup.

* Delta checkpoints have good performance but are hard to program.

* Persistent processes lose state in case of failure.

We argue next that transactions combined with persistent processes are simple to program and give excellent fault-tolerance.

Transactions for data integrity

A transaction is a group of operations, be they database updates, messages, or external actions of the computer, which form a consistent transformation of the state.

Transactions should have the ACID property [Haeder]:

Atomicity: Either all or none of the actions of the transaction should "happen". Either it commits or aborts.

Consistency: Each transaction should see a correct picture of the state, even if concurrent transactions are updating the state.

Integrity: The transaction should be a correct state transformation.

Durability: Once a transaction commits, all its effects must be preserved, even if there is a failure.

The programmer's interface to transactions is quite simple: he starts a transaction by asserting the BeginTransaction verb, and ends it by asserting the EndTransaction or AbortTransaction verb. The system does the rest.

The classical implementation of transactions uses locks to guarantee consistency and a log or audit trail to insure atomicity and durability. Borr shows how this concept generalizes to a distributed fault-tolerant system [Borr 81, 84].

Transactions relieve the application programmer of handling many error conditions. If things get too complicated, the programmer (or the system) calls AbortTransaction which cleans up the state by resetting everything back to the beginning of the transaction.

Transactions for simple fault-tolerant execution

Transactions provide reliable execution and data availability (recall reliability means not doing the wrong thing, availability means doing the right thing and on time). Transactions do not directly provide high system availability. If hardware fails or if there is a software fault, most transaction processing systems stop and go through a system restart -- the 90 minute outage described in the introduction.

It is possible to combine process-pairs and transactions to get fault-tolerant execution and hence avoid most such outages.

As argued above, process-pairs tolerate hardware faults and software Heisenbugs. But most kinds of process-pairs are difficult to implement. The "easy" process-pairs, persistent process-pairs, have amnesia when the primary fails and the backup takes over. Persistent process-pairs leave the network and the database in an unknown state when the backup takes over.

The key observation is that the transaction mechanism knows how to UNDO all the changes of incomplete transactions. So we can simply abort all uncommitted transactions associated with a failed persistent process and then restart these transactions from their input messages. This cleans up the database and system states, resetting them to the point at which the transaction began.

So, persistent process-pairs plus transactions give a simple execution model which continues execution even if there are hardware faults or Heisenbugs. This is the key to the Encompass data management system's fault-tolerance [Borr 81]. The programmer writes fail-fast modules in conventional languages (Cobol, Pascal, Fortran) and the transaction mechanism plus persistent process-pairs makes his program robust.

Unfortunately, people implementing the operating system kernel, the transaction mechanism itself and some device drivers still have to write "conventional" process-pairs, but application programmers do not. One reason Tandem has integrated the transaction mechanism with the operating system is to make the transaction mechanism available to as much software as possible [Borr 81].

Fault-tolerant Communication

Communications lines are the most unreliable part of a distributed computer system. Partly because they are so numerous and partly because they have poor MTBF. The operations aspects of managing them, diagnosing failures and tracking the repair process are a real headache [Gray].

At the hardware level, fault-tolerant communication is obtained by having multiple data paths with independent failure modes.

At the software level, the concept of session is introduced. A session has simple semantics: a sequence of messages is sent via the session. If the communication path fails, an alternate path is tried. If all paths are lost, the session endpoints are told of the failure. Timeout and message sequence numbers are used to detect lost or duplicate messages. All this is transparent above the session layer.

Sessions are the thing that make process-pairs work: the session switches to the backup of the process-pair when the primary process fails [Bartlett]. Session sequence numbers (called SyncIDs by Bartlett) resynchronize the communication state between the sender and receiver and make requests/replies idempotent.

Transactions interact with sessions as follows: if a transaction aborts, the session sequence number is logically reset to the sequence number at the beginning of the transaction and all intervening messages are canceled. If a transaction commits, the messages on the session will be reliably delivered EXACTLY once [Spector].

Fault-tolerant Storage

The basic form of fault-tolerant storage is replication of a file on two media with independent failure characteristics -- for example two different disc spindles or, better yet, a disc and a tape. If one file has an MTBF of a year then two files will have a millennia MTBF and three copies will have about the same MTBF -- as the Tandem system failure statistics show, other factors will dominate at that point.

Remote replication is an exception to this argument. If one can afford it, storing a replica in a remote location gives good improvements to availability. Remote replicas will have different administrators, different hardware, and different environment. Only the software will be the same. Based on the analysis in Table 1, this will protect against 75% of the failures (all the non-software failures). Since it also gives excellent protection against Heisenbugs, remote replication guards against most software faults.

There are many ways to remotely replicate data, one can have exact replicas, can have the updates to the replica done as soon as possible or even have periodic updates. [Gray] describes representative systems which took different approaches to long-haul replication.

Transactions provide the ACID properties for storage -- Atomicity, Consistency, Integrity and Durability [Haeder]. The transaction journal plus an archive copy of the data provide a replica of the data on media with independent failure modes. If the primary copy fails, a new copy can be reconstructed from the archive copy by applying all updates committed since the archive copy was made. This is Durability of data.

In addition, transactions coordinate a set of updates to the data, assuring that all or none of them apply. This allows one to correctly update complex data structures without concern for failures. The transaction mechanism will undo the changes if something goes wrong. This is Atomicity.

A third technique for fault-tolerant storage is partitioning the data among discs or nodes and hence limiting the scope of a failure. If the data is geographically partitioned, local users can access local data even if the communication net or remote nodes are down. Again, [Gray] gives examples of systems which partition data for better availability.

Summary

Computer systems fail for a variety of reasons. Large computer systems of conventional design fail once every few weeks due to software, operations mistakes, or hardware. Large fault-tolerant systems are measured to have an MTBF at orders of magnitude higher -- years rather than weeks.

The techniques for fault-tolerant hardware are well documented. They are quite successful. Even in a high availability system, hardware is a minor contributor to system outages.

By applying the concepts of fault-tolerant hardware to software construction, software MTBF can be raised by several orders of magnitude. These concepts include: modularity, defensive programming, process-pairs, and tolerating soft faults -- Heisenbugs.

Transactions plus persistent process-pairs give fault-tolerant execution. Transactions plus resumable communications sessions give fault-tolerant communications. Transactions plus data replication give fault-tolerant storage. In addition, transaction atomicity coordinates the changes of the database, the communications net, and the executing processes. This allows easy construction of high availability software.

Dealing with system configuration, operations, and maintenance remains an unsolved problem. Administration and maintenance people are doing a much better job than we have reason to expect. We can't hope for better people. The only hope is to simplify and reduce human intervention in these aspects of the system.

Acknowledgments

The following people helped in the analysis of the Tandem system failure statistics: Robert Bradley, Jim Enright, Cathy Fitzgerald, Sheryl Hamlin, Pat Helland, Dean Judd, Steve Logsdon, Franco Putzolu, Carl Niehaus, Harald Sammer, and Duane Wolfe. In presenting the analysis, I had to make several outrageous assumptions and "integrate" contradictory stories from different observers of the same events. For that, I must take full responsibility. Robert Bradley, Gary Gilbert, Bob Horst, Dave Kinkade, Carl Niehaus, Carol Minor, Franco Putzolu, and Bob White made several comments that clarified the presentation. Special thanks are due to Joel Bartlett and especially Flaviu Cristian who tried very hard to make me be more accurate and precise.

References

[Adams] Adams, E., "Optimizing Preventative Service of Software Products", IBM J. Res. and Dev., Vol. 28, No. 1, Jan. 1984.

[Bartlett] Bartlett, J.,"A NonStop Kernel," Proceedings of the Eighth Symposium on Operating System Principles, pp. 22-29, Dec. 1981.

[Borg] Borg, A., Baumbach, J., Glazer, S., "A Message System Supporting Fault-tolerance", ACM OS Review, Vol. 17, No. 5, 1984.

[Borr 81] Borr, A., "Transaction Monitoring in ENCOMPASS," Proc. 7Th VLDB, September 1981. Also Tandem Computers TR 81.2.

[Borr 84] Borr, A., "Robustness to Crash in a Distributed Database: A Non Shared-Memory Multi-processor Approach," Proc. 9th VLDB, Sept. 1984. Also Tandem Computers TR 84.2.

[Burman] Burman, M. "Aspects of a High Volume Production Online Banking System", Proc. Int. Workshop on High Performance Transaction Systems, Asilomar, Sept. 1985.

[Cristian] Cristian, F., "Exception Handling and Software Fault Tolerance", IEEE Trans. on Computers, Vol. c-31, No. 6, 1982.

[Enright] Enright, J. "DP2 Performance Analysis", Tandem memo, 1985.

[Gray] Gray, J., Anderton, M., "Distributed Database Systems -- Four Case Studies", to appear in IEEE TODS, also Tandem TR 85.5.

[Haeder] Haeder, T., Reuter, A., "Principals of Transaction-Oriented Database Recovery", ACM Computing Surveys, Vol. 15, No. 4, Dec. 1983.

[Kim] Kim, W., "Highly Available Systems for Database Applications", ACM Computing Surveys, Vol. 16, No. 1, March 1984

[Lampson] Lampson, B.W. ed, Lecture Notes in Computer Science Vol. 106, Chapter 11, Springer Verlag, 1982.

[Mourad] Mourad, S. and Andrews, D., "The Reliability of the IBM/XA Operating System", Digest of 15th Annual Int. Sym. on Fault-Tolerant Computing, June 1985. IEEE Computer Society Press.

[Schlichting] Schlichting, R.D., Schneider, F.B., "Fail-Stop Processors, an Approach to Designing Fault-Tolerant Computing Systems", ACM TOCS, Vol. 1, No. 3, Aug. 1983.

[Spector] "Multiprocessing Architectures for Local Computer Networks", PhD Thesis, STAN-CS-81-874, Stanford 1981.

[von Neumann] von Neumann, J. "Probabilistic Logics and the Synthesis of Reliable Organisms From Unreliable Components", Automata Studies, Princeton University Press, 1956.

[von Neumann] von Neumann, J. "Probabilistic Logics and the Synthesis of Reliable Organisms From Unreliable Components", Automata Studies, Princeton University Press, 1956.

f

Peter

```
*****************
* d i g i t a l *
*****************
```

TO: see "TO" DISTRIBUTION

DATE: MON 21 DEC 1981 11:50 AM EST
FROM: BRUCE DELAGI
DEPT: STRATEGIC ENG.
EXT: 223-4887
LOC/MAIL STOP: ML12-3/A62

SUBJECT: DOOMSDAY REPORT

all here

The following report will be coming out in three sections. Please watch for them.

"TO" DISTRIBUTION:

PETER CHRISTY
RICHARD FIORENTINO
BUD HYLER
ROY RODGERS
BOB SUPNIK
SULTAN ZIA

BOB DALEY
DON GAUBATZ
RAY OCHESTER
STEVE ROTHMAN
JOHN TANK

ENG STAFF:
CARL GIBSON
JOSEPH REILLY
HERB SHANZER
STEVE TEICHER

ATTACHED: DOCUMENT;156

Zeb

Copy PSD Staff

Js.

Several of the people who recently played the Doomsday Game asked me to put together a summary of the outcome. I've attached single sheet resumes of what was generated.

The feedback was:

1. The excercise was good/interesting/valuable;

2. Using the "results" in our plans would (unfortunately) be purely a matter of chance;

3. Baseline data on industry size, partitioning, shares, and growth rates were missing, (and a bit more competitive detail in the way of 1ØK reports, etc. would be nice to have);

4. The DEC cases needed a starting point (base case) to make the time spent playing them more productive;

5. It should be played again (in separate sessions) by all of PEG, GVPC, OC, PG managers, manufacturing planning people, and key technical contributers.

Some personal observations I'd add.

1. We seem very short on a fact-based common understanding of the long range actions of our strategic competitors.

2. We easily revert to a DEC-centered view of the real world threats and opportunities.

3. We demonstrate an almost touching loyalty to the concept of the central processor and operating system as the centerpiece of corporate strategy.

I'd hope that future playing of the game would change this situation a bit. To do this, together with my co-designers, I'd like to:

1. Arrange a PEG/GVPC/PGM game in February or March;

2. Put the finishing touches on the competitive base-case packages;

3. Force the DEC simulations to be more realistic and more responsive to projected moves by our competition;

4. Find a way to use the game outcomes to affect our plans, values, and behaviors. (I'm at a loss on this one given the current DEC climate - at least as I perceive it).

IBM

- o Occupy 9Ø% of the F1Ø,ØØØ desks by integrating corporate information/communication service functions & making aggressive use of technology in the low cost manufacture of personal computers.

- o Maintain the dominent position in the central EDP environment by shifting revenue toward software & services; reducing hardware prices to meet (say) Fujitsu; adjusting the user/OS/cpu boundary to provide new capabilities customers will value.

- o Avoid the kind of stable interface definitions that would allow Fujitsu to make the best use of their resources (Watch Europe for the first indications of how this struggle will evolve).

- o Example moves:

  1982 - SNA for Display Writer, Personal Computer & S/38;
  - OA architecture: "document" interchange standards voice/image/text/data.

  1986 - Retread the typewriter salesforce for OA component sales;
  - New PC based on S/38; S/38 LAN;
  - Integrated voice/data PBX (with SBS linkage);
  - Database interface for System/38 tying to 37Ø core clusters;
  - All channels viable from independent merchandizers thru F5ØØ direct sales account control.

  199Ø - Provide information-supplier services for private viewdata (with state-of-the art viewdata terminals).

AT&T

- o Be the single source for all small business (and consumer) information services.

- o Provide commodity access to information and information processing (leveraging off intrinsic strengths in interconnect, switching and the ability to efficently bill and collect for an aggregate of small transactions).

- o Do this with the quality and reliability so long associated with Ma Bell.

- o Example moves:

  1982 - First level network data services: message store/forward;
  - "Phone replacement" for the home/small business: adds good quality keyboard microprocessor, 64KB memory, small screen.

  1986 - New generation departmental/small business PBX supporting inter-enterprise electronic mail (with limited electronic invoicing/billing);
  - Third party applications distributed by AT&T for home and small business software (UNIX & CP/M foundation);
  - Information access and (electronically based) retailing services;
  - Merchandizer channels, phone stores, interconnect suppliers.

  1990 - Image based electronic retailing and entertainment services.

CONVERGENT TECHNOLOGY

o Exploit fast time-to-market of commodity hardware for cluster servers and workstations internals.

o Use CTOS and its continual enhancement as an application base to lock in large volume OEM sales channels (e.g. NCR/Burroughs in the office) for applications, service, and sales.

o Focus on the establishment of a CLUSTER architecture (to provide future hardware cluster enhancement sales).

o Risks - being outpaced on CTOS enhancements and;
  - being reverse engineered out of the picture;
  - hooking up with inadaquate, second-rate OEM's.

o Example Moves:

1982 - \$10K - \$20K RAM font & "1/2 page" resolution workstations;
- \$4K each instance of WPS, sort/merge, ISAM, forms, PASCAL.

1986 - ship 32-bit (386) full page resolution workstation with telephone management;
- Use commodity disks and cluster peripherals as manufactured in Japan;
- System software enhancements for file structures, communications,.....

1990 - 486 based workstation with a buyout high performance flatscreen CRT;
- Audio/video/data local area net.

NIPPON ELECTRIC COMPANY

o Capture the desk by first providing total communication service and then a total information service for the office (and later for the home).

o Accomplish this with volume manufacture of medium performance personal computer phone replacements that support popular operating systems for applications development.

o And the PBX communications center of a small firm (or department) - complemented by (fiberoptic/CATV) local area nets as needed.

o Use multiple, geographically knowledgeable channels of distribution to provide familar brand names and pertinent applications know-how (e.g. Xerox -or Prime, Sanyo, interconnect distributers).

o Do all this from a strong position in base technologies: semiconductors, communications, video, speech recognition/synthesis.

o Example moves:

1982 - 8086/8088/286 commodity general purpose processors with proprietary processors for graphics and speech;
- digital PBX fiberoptics and wireless LAN;
- modular PC desk unit integrating video, voice, data (with a CP/M operating system);
- establish North American partnership with Xerox;
- sell a minimum (& minimal) PC as low as $1K entry cost (retailed under the Sanyo brand name).

1986 - Intel 386 for a new generation PCC and 8" media for an information server greater than 100MB;
- Provide the (network OS) infrastructure for an information services network for business and the home user;
- Home PC (with voice/video/data) at stereo prices.

1990 - VLSI supercomputer and optical storage;
- knowledge base expert information systems;
- highly parallel departmental computing.

HP

- Focus on specific industries to lead on selected application areas (e.g. manufacturing) rather than on the traditional undifferentiated DEC-DG-Prime minicomputer market

- Build on strengths in (design, packaging, manufacturing, sales, and service of) integrated desk-top computers to hold significant market share in $500 to $30,000 systems and personal computers.

- Use position in personal and low cost computing systems to seek growth opportunities in new application areas (e.g. office)

- Rely on strong marketing competance and a quality product image to maintain and build HP brand loyalty as a value in itself

- Capitalize on (highly divisionalized) corporate structure to capture diversified targets of opportunity

Example moves:

1982 - Build a unique operating system for a new 32-bit architecture of the 80's. First product is 0.3Mips at $40K (This is such a self-destructive act that only previous history makes it credible);
- Include strong new CAD/CAM applicantion packages on the new system;
- Provide consulting services for manufacturing automation and production control systems;
- Introduce $10-15K low-end workstations a low cost integrated terminal and low-end disks: 0.25MB floppy and a 10-30 MB hard disk.

1986 - Add 32-bit applications in the laboratory and medical environments and expand revenue from consulting services;
- Provide systems integration for robotics/factory systems;
- Support industry standard communication links for the factory, the laboratory, and the hospital (CATV/broadband, line-of-sight interfacility links, full function PBX's);
- Introduce high-function CAD/CAM workstation;
- Offer integrated 32-bit office system with low cost terminal/workstation;
- Start to lower prices as investments in HP's internal manufacturing systems pay off.

1990 - Provide HP robotics/artificial intelligence systems in the factory. These systems provide voice interaction with factory operators.

DEC COMMODITY PRODUCTS

- Change from OWNER to PROPOGATOR of standards - and seller of cost competitive equipment based on these standards.

- Accelerate acquisition and development of high volume manufacturing technologies in semiconductors, interconnects, (storage) media, assembly, and material flow.

- Establish components division (separate assets, headset change).

- Sell components internally at transfer cost, externally to world market.

- Example Actions:

  1986 - Ship single board VAX; introduce single package VAX;
  - Develop one chip VAX - work for lead architecture of the 90's;
  - Sell low-end high volume disk on DEC and non-DEC systems;
  - Provide are extensive software/applications library and sell software on non-DEC VAX systems;
  - Generally price hardware/systems low but price standards (and leadership products) high.

  1990 - Develop and propogate standards for voice & vision;
  - Announce a single board 10Mips VAX;
  - Start development of a system on a chip;
  - Announce architecture of the 90's;
  - Sell a one chip, 2 MIPS VAX;
  - Acquire an optical disk design and manufacture it.

DEC PROPRIETARY SYSTEMS

- Serve markets that require integrated products (and services) to solve complex problems.

- Stay one level closer to end-customer needs than commodity manufactures - provide tools that enable significantly higher customer productivity (particularly in sub-solution integration).

- Always have at least one leadership (flagship) product: exploit a quality image superior to commodities.

- Example Actions:

  1982 - proprietary layered environments (e.g. OSM, CTAB);
  - leadership development tools (nets, files, languages);
  - parameter driven application programs.

  1986 - Scorpio/CT32, Venus leadership processors;
  - leadership cluster environments (and ADA);
  - user-oriented documentation;
  - human interface design tools.

DEC-UNCONSTRAINED

- o Provide an integrated computing and communications environment for leadership user productivity over a range of products from $500 to $10,000,000
  (by buying/licensing/commissioning more actively in the support of an integrated architecture).

- o Position a DEC Information Architecture (speech/voice/data/text,...) at personal and departmental levels.

- o Compete for the desk but co-exist with IBM at the Corporate EDP level and with AT&T at the Corporate communications level.

- o Example Actions:

  1982 - Subset the VAX Architecture for public use;
  - Focus internal design and manufacture on $10K-250K systems;
  - Commission 32-bit commodity workstation product in Japan using DEC VAX architectures;
  - Commission a Japanese HyperVAX;
  - Buy PBX (Mitel) for resale in integrated computing and communications systems;
  - Do all the office automation - even pursuing a service-oriented business linking OA, EDP, and communications.

  1986 - Provide only 5 1/4" (45MB) and 14" (1GB) storage;
  - Announce an integrated communication and information architecture (and its lead products);
  - Develop a retail channel to provide VT/CT's as viewdata terminals (?).

  1990 - Selected end-user expert systems.

```
*****************
* d i g i t a l *
*****************
```

At.

TO: see "TO" DISTRIBUTION

cc: see "CC" DISTRIBUTION

DATE: THU 29 OCT 1981 3:38 PM XXX
FROM: RICK CORBEN
DEPT: CORP PRODUCT MGMT
EXT: 223-3123
LOC/MAIL STOP: ML12-1/T39/T39

SUBJECT: CORPORATE PRODUCT STRATEGY EXERCISE

DATE: TUESDAY, NOVEMBER 3 THROUGH THURSDAY, NOVEMBER 5

TIME: 8:15 - 5:00

LOCATION: ML12-1 (RICK CORBEN'S OFFICE)

OBJECTIVE: Complete enough of the DECsite "configuration brochures" to serve as input for the Distributed System Architecture Task Force.

BACKGROUND: You should have read the write-up of DECsite/Engineering (Boeing), and there will be one for DECsite/Division Headquarters (Merrimack) shortly. During our three days, we will do three additional sites.

TENTATIVE SCHEDULE:

TUESDAY: FACTORY (GM OR GE?)
WEDNESDAY: BANKING (BANKERS TRUST)
THURSDAY: SMALL BUSINESS (WES MELLING TO PROVIDE)

Wes Melling will make a start-off presentation on the factory environment. Don Jenkins and Vijay Thakur will join us Tuesday morning to provide additional expertise. Jack Mileski and Gene Hodges will join us on Wednesday morning. They've studied Bankers Trust extensively. Jim Willis (COEM) will join us Thursday morning for the small business discussion. Wes Melling is preparing a presentation on a real-world small business.

HUMAN ENGINEERING:

We might move the Thursday meeting to Peter Conklin's home as a change of pace.

My office is comfortable enough for our group, but there are not too many spare seats. Please do not bring guests without checking with me.

PROBLEM DEFINITION:

The following description from the appendix to DECsite/Engineering may help those of you who will be participating for the first time.

NATURE OF EXERCISE

Goal & Constraints

1. Typical customer site configuration that we want to sell in FY'86 based on Corporate Product Strategy.

2. Concentrate on building within worldwide networks.

3. Consider CI and other technologies in addition to NI.

4. Our output should be a basis for a "Product Brochure" (Look through a customer's eyes.)

5. Concentrate on product features/attributes that are not owned by a single Program Office.

Characteristics of DECsite

1. Component Nodes - Hardware/Software
2. Interconnects
3. Price
4. Applications being used
5. Uniqueness - customer benefits unique in the marketplace from DEC
6. Performance expectations
7. Maintenance philosophy
8. Growth pattern
9. Relation to other communications facilities (PBX, SNA, X.25, DECNET)
10. Relation to other competitor's systems
11. Conformance to standards (External/internal)
12. Connection of foreign entities to environment
13. System management philosophy

/jdm
RC1.S5.29

"TO" DISTRIBUTION:

JOHN ADAMS
RICHARD FIORENTINO
GLENN REYER
JOHN ANDERSON
LAURA MELLING
*PETER F CONKLIN
JOHN O'KEEFE

"CC" DISTRIBUTION:

GENE HODGES
PEG:
DON JENKINS
THAKUR @MK12
JACK MILESKI
JIM WILLIS

TRIP REPORT: BOEING CORP.

JAN 21-22, 1981

Nancy Neale and Harry Hersh
User Research Group
Corporate Research

INTRODUCTION

Three organizationally autonomous sites were visited at Boeing: 1. ATAD (Advanced Technology And Development), part of Boeing Computer Services; 2. The Wind Tunnel Facility, run by Boeing Computer Services; and 3. Interactive Graphics Data Analysis Group, part of Boeing Commercial Airplane Company. Each of these organizations has developed workstations for internal use. Thus, in addition to information concerning the environment and user requirements, detailed information was obtained concerning hardware, software, and operational aspects of these systems.

SUMMARY ISSUES

The following major issues emerged from this two-day visit to engineering and computer organizations at Boeing in Seattle:

1. Graphics is a given. Across organizations, both video and hardcopy graphics, fully integrated with data analysis software, was assumed.

2. The demand for color graphics is elastic.

3. Ease of use is defined as a user's ability to do something "useful" at the workstation in the first 10 minutes.

4. End-user requirements were the FOCUS of the development processes. These requirements were assessed through continuous dialogues between users and developers.

5. "Latent computer demand" was defined as the ongoing phenomenon of evolving new uses identified by the users once workstations were available and actively in use.

6. Reliability is considered critical, with a vendor's reputation and ability to support and service systems a major factor in product choice.

7. In each case, cost justification based on productivity evaluations was cited in dollars. This emphasis was conspicuous.

8. Demand for workstations evolved from financial and/or time penalties of overloading mainframes with many small jobs.

9. Security is an important issue addressed mainly through physical control (e.g., locking floppies in a desk drawer).

10. DIGITAL was considered a major vendor for supplying quality processors at a reasonable price (i.e., high, but worth the cost). DIGITAL was criticized several times for poor quality/performance peripherals at ridiculously high costs to customers.

11. Timing of the capital budget process was an important factor in equipment acquisition. Equipment could only be purchased at one point in the 12 month cycle.

12. In all three instances workstations primarily replaced previous batch and/or time-sharing operations.

13. Internally developed training and documentation was available for all workstation users.

ATAD (ADVANCED TECHNOLOGY AND DEVELOPMENT) GROUP

BCS views itself as an OEM to the Boeing organization. Historically, the main business of BCS has been selling time on IBM and CDC mainframes, both internally and externally. This group has developed, and is currently marketing within Boeing, BITS (Boeing Intelligent Terminal System). BITS is an integrated workstation concept for the automated office enviroment.

ENVIRONMENT/USERS

BITS was characterized by the developers as end-user driven. Rather than proceeding from hardware to system software to end-user applications, BITS began by assessing users' needs within Boeing. Target users included:

Clerks
Secretaries
CTS users
Programmers
Executives and managers
Professionals

Their experience has been that a majority of BITS usage is by professionals and clerks who are knowledgable about computers in general. (Financial managers are included in the professional category.) Some managers use BITS, but they are in the minority.

Consistent with their emphasis on the development/evolution of BITS being end-user driven, users are actively encouraged to exercise the Enhancement/Problem Report system. Their experience has been that problem reports have decreased exponentially (with an unspecified exponent) since the introduction of the system.

SOFTWARE FEATURES

The BITS operating system is interpretive and PASCAL based, with all software above the interpreter being portable. The complete set of software modules consists of approximately 200K lines of PASCAL code. Using judicious make-buy decisions, the total software system was developed over two years by 4-5 designers.

Much of the software (e.g., editors, word processor) are emulations of previously existing software. Thus, even though there is a single menu-oriented interface, the remainder of the software modules does not appear to be truly integrated.

The BITS system is targeted at BOTH programmers and non-programmers; at sophisticated and non-sophisticated users. The result is quite a bit of duality in the system. For example,

1. The "universal keyboard" appears to be based on a simple merging of every key on the WANG keyboard and every key on the VT100 keyboard, including SETUP and XON (no scroll).

2. The keyboard contains an "X" key which, when pressed, bypasses the menu and allows the user to execute a command.

3. Both word processing (copied directly from WANG) and text processing (a la RUNOFF) exist on the system.

4. The business graphics system is parametrically driven. Parameter changes can occur either by running an easy-to-use interactive routine, or by directly editing the parameter file.

The following software packages currently run on BITS:

1. Languages: PASCAL, FORTAN 77, BASIC (compiler only), Assemblers for LSI-11, Z80, and 8085.

2. A Screen editor: menu oriented

3. 2 line editors that emulate CTS and KIT. The emulations allow users to modify the command names to suit their own applications or preferences.

4. Word processing: a direct copy of WANG.

5. Text processing: Command oriented like RUNOFF, but (they claim) easier to use.

6. A communications package for both synchronous and asynch communications. They currently support BNA, Boeing's version of SNA.

7. A business graphics package (Not for CAD). The graphics package is integrated with the plotter for automatic hardcopy graphic output (e.g., for automatic viewgraph production). Multiple fonts are available here and for word and text processing; limited only by device resolution. Similarly, the software supports hard- and soft-copy color graphics on configurations containing appropriate devices.

8. Comprehensive data entry utility (50K lines of code, which represents 25% of the total code written.)

9. Teltext and Viewdata are available to users.

10. Applications packaqes are available in the following areas:

    Finance
    Enaineerinq
    Office graphics
    Operations
    Program (i.e., software) development
    Product assurance

    At the current time, twelve RX02 diskettes are distributed with the typical BITS workstation.

11. Several CAI packaqes are available for use with the applications packaqes. They claimed that the on-line documentation was limited by the amount of local storage available.

HARDWARE FEATURES

The base level hardware is variable, with a present average cost less than $12K for a single user system, optionally tied to a host (CDC, IBM). BITS started on an 11/70, and the ATAD group was interested in determining whether an 11/23 could support the functions.

Generically, the hardware consists of the following:

- o A processor
- o 64K bytes of memory
- o Two 500K byte 8" floppies (IBM 3740 format)
- o A keyboard (Standardized)
- o CRT display, 24x88 min; must support low resolution graphics: "The system is tailored to different levels of graphic needs." Color display at the high-end.
- o At least two RS232 ports
- o Support for dot matrix (graphic) and daisy wheel (LQP) printers, 4-color plotter, hard disks (Winchesters up to 20M), digitizers, voice recognition and response units.

The following hardware systems are currently being used:

- o TERAK 8510 (has an 11/2 in it)

  This system contains a 320x240 resolution CRT, and is the most widely used hardware within the BITS program. (Currently 95% of terminals used are TERAK.)

- o DEC 11/23 (with VT100)

- o DEC PDT 11/150

- o ONTEL OP-1/70 (8085 based)

- o RAMTEK C214 (Z80 based)

- o TI I/90

ISSUES FOR BITS AND BCS

The following issues emerged from the discussions:

1. The system must be easy to use and easy to maintain. Users familiar with the time-sharing software (e.g., CTS editor) can be up and running on BITS in 10 minutes. Novices require one to three days of classroom training.

2. An important distinction was made between managerial and professional use of the workstation: we were told that professionals (primarily engineers and financial analysts) need functionality specific to their work requirements. Managers review data and play "what if" games. Managers want instant recall of information; they never update, just look.

3. We were told that managers do not want keyboards: not because of the typing issue, but because they do not do data entry. A special BITS workstation was constructed for a high level manager. The "workstation" consisted of a large color display in a walnut case resting on a lazy-susan. Input was accomplished through a radio controlled (i.e., portable) keypad. The executive would instruct his secretary to run a program or obtain some data. The executive then used the keypad, in conjunction with a simple menu on the screen, to review the information.

4. "Corporate data" resides on a central mainframe; only personal and local data are stored at the workstation. Individual departments are driving the need for localized computing capability.

5. Currently, the workstations only communicate with the mainframes, not with each other. (There is a striking parallel here between the physical constraints in the computer network and the information control in their organizational network.) The result is a time-sharing system with varying amounts of local intelligence. (This is consistent with their primary business goal of selling time and services on their mainframes.) They are, however, very interested in the development of local area networks, and mentioned specifically their interest in the development of the Ethernet.

6. Workstations were introduced through justification of need via productivity analysis. Their conclusions indicate significant incremental savings: e.g., 10 BITS workstations showed a savings of $635,594 (which was considered to be a conservative estimate).

7. Their view of the market:

   o User is looking for integration - won't just buy word processing.

   o They see the need for a wide range of processors: from 8 bit processors for simple office applications to 64 bit processors for precision engineering applications.

   o Near-term cost of the workstation must be under $10K.

   o Color is liked for its highlighting effect, but they won't pay a whole lot extra for it. They do not consider color in the mainstream today.

8. System noise level must be acceptable for an office environment.

9. On the 11/23, they liked the performance and the memory capacity. They did not like the packaging (too bulky), the power requirements (restricts portability), or the noise (too noisy for an office environment).

10. Security: Their networks have encryption capability. Their response to software security issues was that BITS is a single user system, and users can remove their floppies and lock them in a desk drawer. Access to the software modules is controlled through their general menu interface.

11. Teleconferencing is a long-term goal they're working toward: electronic mail and the "automated office" are in their future. Their attitude toward software security, however, suggests that they may not have started to address these issues seriously.

12. The keyboard must be universal. (This follows from their requirement to satisfy both programming and non-programming users.) The VT100 keyboard is not acceptable: not enought function keys.

13. They want to meet both sophisticated and unsophisticated users' requirements with modular, compact hardware. They'll buy hardware from a vendor but not the software.

14. Terak is not a good supplier for their commercial marketing needs. They would like to buy all their hardware from a primary source.

15. Their universal software needs require merged text and graphics.

16. RL01 disks are too big (physically) and too expensive: They can obtain a disk from another vendor (Charles River) that contains 4 times the capacity for the same price and in the same size package.

17. Same price and package size problems were mentioned for the RX02.

WIND TUNNEL FACILITY

The BCS operates the Data Analysis Modules (workstations) at Boeing's wind tunnel facility. Volumes of data are collected from various physical measurements during a single simulated flight. Historically, these data have been analyzed with batch routines. Most recently BCS has introduced its Data Analysis Module, which makes the data analysis interactive.

ENVIRONMENT/USERS

This single wind tunnel facility consists of a subsonic and a supersonic wind tunnel, and serves design engineers from the various groups of the corporation. The wind tunnels are in use 24 hours per day, 7 days per week. The several workstations are housed in a single large room adjacent to the quite noisy wind tunnel. The process is as follows:

1. An engineer produces a structural design.
2. Contact with the Wind tunnel is made and a scale model is built with the requisite sensors built in.
3. The model is placed in wind tunnel.
4. The test is run and data collected with a PDP-11/70.
5. Parameter changes and additional tests may be run.
6. The data are then available for analysis within the Wind Tunnel facility.
7. Wind tunnel personnel take primary responsibility for carrying out data analysis as specified by the designer. Designers may perform the actual analysis if they want.
8. Access to data is tightly controlled. No data is released to a design group until it is officially certified.

SOFTWARE

The software for the Data Analysis Module consists mainly of the existing batch-oriented data analysis routines which have been modified for interactive use. Although we were not given a demonstration of the system, comments from the wind tunnel personnel, other Boeing people, and local DEC sales people suggested that the user interface was somewhat difficult to use.

Causes of difficulty in usage were: (1) unspecified features of the user interface design; and (2) the fact that individual customers (i.e., designers) used the wind tunnel facitily very infrequently, and thus never gained familiarity with the software syntax.

HARDWARE

Each Data Analysis Module contained an 11/70 and an RP06 disk. Attached to the processor was an Evans & Sutherland graphics system, and an HP terminal.

ISSUES

1. This is a dedicated workstation - No possibility for generic functions at this point in time.
2. Control of data was an important issue. It had to be used at their facility: there was no remote access to the data. Some control issues were political.
3. System reliability was important. The facility must be up and running continuously. Any downtime is a major problem.
4. Ease-of-use is considered important, so that the staff isn't continuously tied up with the engineering users.
5. Graphics is very important. It was an integral part of the migration path from batch processing to workstation, from pre-specified sets of batch data plots to interactive graphic data analysis.

INTERACTIVE GRAPHICS DATA ANALYSIS GROUP

This group represents technical staff responsible for airplane structural analysis. They have developed an integrated network of (group) workstations for aircraft structural analysis. The manager is singularly powerful within the corporation, and was a major factor in the development of the IGDA concept.

ENVIRONMENT/USERS

IGDA systems are currently at 21 geographically dispersed sites in the greater Seattle area. These sites resemble group work areas more than individual work stations, providing an environment supportive of cooperative problem solving. Each sites is physically bounded and is overseen by a site coordinator who functions as a combination system manager, local expert, and security agent.

The workstations were forcibly introduced by the manager (who had enough political muscle to accomplish this major change very quickly). His tactic was to take away the project leaders' dollars and people, forcing them to come "kicking and screaming" to use the workstations. Once used, however, the IGDA concept was positively received.

IGDA users (1050 currently) tend to be fairly sophisticated. Most users are engineers and technical support people. Few are managerial-level users. Average usage ran 1/2 to 1 hour daily.

SOFTWARE

The 11/70's all run under IAS; network nodes are connected through DECNET V.3.0 and a hyperchannel. Neither software nor data are truly distributed. Data files exist on one of the nodes of the network: in order to work with data from a particular node, one specifies the node name and file name, and indicates whether to "put" or "get" the data. Everything else is transparent. Most application code is written in FORTRAN.

The user interface is uniform across the 21 sites, and consists of a menu tree, containing 11 levels. The menu design uses the joystick on the Vector General Display system as the selection device. The user interface was also uniform across the various application modules. Major software components include:

- o General Graphics Package
- o General Contour Package
- o Dynamic Visualization Package
- o Viewfoil Generation (i.e., overhead slide production), incorporating graphics, charts, and text.
- o Work Schedule Chart Program (Pert, Gantt, etc.)
- o Other technology staff specific software packages:

  1. Aerodynamcis (e.g., world wind models)
  2. Flight controls (e.g., aeroelastic correction and trim)
  3. Noise technology (e.g.,multiple-microphone jet noise source location computation)
  4. Propulsion (e.g., fuel tank simulation)
  5. Structures (e.g., finite element model)
  6. Systems (e.g., dynamic mechanical simulation)
  7. weights (e.g., Loading envelope of airplane weight)
  8. Military Airplane Co. (e.g., wing section graphical display)

All these technology staff specific packages use the IGDA graphics capabilities. This commonality contributes to the uniformity of the user interface.

HARDWARE

The IGDA hardware complement was uniform across the 21 sites. Included are:

- o PDP 11/70 with 3/4M of memory running IAS.
- o three RP06's. Users are responsible for mounting, dismounting, and storing their own disk packs.
- o A 9-track magtape and a 7-track magtape (for compatability)
- o Vector General 4-color display workstation and a B/W workstation. Each station contained a QWERTY keyboard, separate keypad of function keys, and a 3-dimensional joystick.
- o A RAMTEK high resolution color graphics system is currently being evaluated.

- o Two Tektronix graphics terminals
- o Four dumb terminals (Beehive)
- o An HP intelligent terminal
- o Lineprinter
- o Gould 5200 electrostatic plotter
- o Communications lines: 4800 baud, 9600 baud, and/or Hyperchannel (50 Mbaud). The 21 sites were configured in a manner very similar to DIGITAL's Engineering Network. In addition, the network ties into a CDC Cyber, which, in turn, connects to a Cray I, an IBM mainframe, and another backup Cyber.
- o Optional equipment: Card reader, Talos digitizer (4' x 6'), Calcomp 4-color drum plotter.

ISSUES

1. Productivity justifications were strongly emphasized. The current technical staff of 4000 perform more work than did the staff of 5000 in 1972. They claim average increased engineering productivity of 25 times, and a decrease in analysis flow time between 1/10 and 1/100 of previous rates. An important result is the increase in design/analysis cycles per unit of time: a critical factor in building safer airplanes.

2. Estimated cost of all sites and equipment to date is $15 million. The calculated return on this investment is $58 million: "That's worth a little more than one 747 (worth $43 million) that we've returned to the company."

3. Price sensitivity is not an issue here due to the manager's political clout. Hardware acquisition is directly a function of perceived needs.

4. They claim that their overhead penalties were too costly for the amount of work they were doing on the mainframes. This became a key motivation for the move to the IGDA workstation.

5. The developers felt they were sensitive to user needs:

   - o Uniformity of hardware and software across sites contributed greatly to the system's ease-of-use. All sites contain identical equipment and capability so that engineers can be "at home wherever they go".

- o A user survey was conducted to determine which dumb terminal to purchase. Several different terminals were brought in, and users were allowed to try them out and express their opinions.

- o A subset of users requested HP terminals, a terminal type not included in the original hardware configuration. These terminals were eventually installed, not only at the requesting sites, but at EVERY IGDA site in order to maintain consistency.

6. Once data has been initially read into the system, a user need never worry about the details of the data format. The software hides all these "irrelevant" details.

7. They feel color is useful (e.g., for disambiguating complex graphs). Color is only found, however, on the Vector General Graphics Systems.

8. They are looking for microprocessor based communication and database interfaces.

9. Security is provided by physical control:

   - o All equipment is used only at sites, overseen by site coordinator
   - o Engineers can lock up their disks at the sites.
   - o All files are "protected" by passwords!
   - o All privileges are given to every user.
   - o No terminals exist outside the site. "As soon as you put a terminal on a desk you must assume a hostile environment."
   - o workstations communicate only with other workstations; no other communication lines are allowed.

10. Several training courses are provided by a full-time, in-house trainer. The basic users one week training course (half lab, half lecture) requires a working knowledge of FORTRAN. An eight-hour cook-book presentation is offered to managers. Ongoing instruction and help are available from the site coordinator, who goes through three months of intensive on-the-job training.

11. They have written their own documentation, and rewrote and expanded the HELP facilities under IAS.

12. This group exibited considerable loyalty to DIGITAL, including being active in DECUS. Much of their loyalty can be attributed to their previous DEC equipment purchases ($3 million this year alone).

13. They are a strong supporter of IAS, and claim it's the only decent operating system produced by DIGITAL!

14. They ran their own benchmarks on VAX vs. 11/70. Their initial result was that they saw no price/performance improvement in the 11/780 over the 11/70. A second benchmark showed a marginal improvement in the VAX over the 11/70. They clearly had no interest in the 11/750, and are waiting for the higher performance VAX's to appear.

15. Even with the Vector General systems off-loading much of the graphics processing, many of their programs are I/O bound. They want disks that are larger and faster than RP06's, at a reasonable price.

16. They stated quite clearly that DIGITAL's price structure on peripheral's is way out of line.

17. Why doesn't DIGITAL sell a little hardware box that would accept vector information (from a vector display) and output raster scan data (for a dot matrix plotter).

18. "The 68000 is a screamer"!

ACKNOWLEDGEMENTS

Our thanks to the Engineering Systems Group for this opportunity to interview a PBS-like DEC customer. In particular we'd like to thank:

ESG Product Planning

Barbara Chapin
Bob Meese

DEC Boeing Sales Force

Jeanne Collins
Jim Kirsila
John Macauley
Jamie Milne
Denise Seacombe

```
+---------------------------+
|   |   |   |   |   |   |   |
| d | i | g | i | t | a | l |
|   |   |   |   |   |   |   |
+---------------------------+
```

interoffice
memorandum

TO: PEG

Date: 23 SEP 81
From: Rick Corben
Dept: Corp. Product Management
Ext: 223-3123
MS: ML12-1/T39

SUBJ: RE: EXERCISE IN CORPORATE PRODUCT STRATEGY

Attached is the full write-up of the Exercise on Corporate Product Strategy (specifically Boeing) that I presented at our meeting on Tuesday in Bedford.

The appendix on the last page lists the objectives of the overall exercise. The cover memo lists some of my concerns at the time that I did the write-up. Specific comments and suggestions would be helpful in improving the vision of Boeing.

/jdm
RC1.S4.41

Attachments

```
+---------------------------+
|   |   |   |   |   |   |   |
| d | i | g | i | t | a | l |
|   |   |   |   |   |   |   |
+---------------------------+
```

interoffice
memorandum

TO: DISTRIBUTION

CC: B. Delagi
T. Yang

Date: 4 SEP 81
From: Rick Corben
Dept: Corp. Product Management
Ext: 223-3123
MS: ML12-1/T39

SUBJ: OUR EXERCISE ON THE CORPORATE PRODUCT STRATEGY

Attached please find the write-up of our exercise on the Corporate Product Strategy. Please review and send suggested revisions. I think there is a lot of room for improvement! Indeed, the document raises a multitude of questions including:

1. There is no name for the Corporate Product Strategy. For this write-up, I called it "DECsite". The name is sufficiently hideous that no one should be confused into believing that we intend to market anything like it or that we have concrete development plans to produce anything like it. It was an exercise only!

2. Did we generate a realistic configuration and a realistic price even for a wealthy customer such as Boeing? We hypothezied a relatively homogeneous population of engineers. Doesn't the real world have a greater diversity of engineers, technicians, technical writers, etc.? Does everyone really need a SUVAX? One of the real advantages of the DECsite concept is its ability to conform to the diversity of real world environments. Could we make our picture of Boeing more realistic? Would we end up with a more appealing price?

   Incidentally, do we really need two secretaries in each department. If the SUVAXs manage telephones, calendars, mail, etc., isn't one secretary enough?

3. The most cost-sensitive part of our hypothetical configuration was the SUVAX workstations. They represent most of the hardware cost. Their strategic significance and the need for a really competitive price is obvious.

4. The distribution of function between the SUVAX workstation and the Computation Server will be an essential element in explaining the benefits of our hypothetical configuration. If the concept is not well-defined and implemented, we end up with a SUVAX which is little more than an over-priced Apple or

with a Nautilus which is a left-over from the bygone era of time-sharing.

5. The clear incompatibility between over current software pricing mechanisms and our vision of future computing environments is frightening. Strategic focus on this problem is as important as our NI development plans!

6. Should the fileserver really be J-11 rather than Scorpio-based?

7. Our configuration used a CI for each Venus pair. I assume that we really would mix systems and CIs for greater availability.

8. Is it realistic to assume the use of DECmail in a large corporation with large numbers of IBM systems? Will the entire DECsite concept be destroyed by too many standards that we cannot control?

9. We really should be more specific about the engineering applications running in the network.

10. There are roughly 1,000 nodes on the single NI in the building. Perhaps, we had better switch to multiple NI's.

11. If the Nautilus conputation server is only 2 to 3 times faster than the Scorpio, will anyone actually use it?

Follow-up sessions are being planned to work on some of the other customer environments of interest.

/jam
RCI.S4.41
Attachment

DISTRIBUTION:

JOHN ADAMS
PETER CONKLIN
ED LAZAR
RICH LEWAN

WES MELLING
JOHN O'KEEFE
GEORGE THISSELL

DECsite/ENGINEERING IN FY'86

CUSTOMER:

The customer is a large Engineering organization within a major Fortune 500 manufacturer. It is composed of 70 separate departments. Each department is assumed to have two secretaries and one manager. The engineers require substantial computational support for functions such as finite element analysis. The customer is a sophisticated and experienced computer user. The primary objective is maximizing the productivity of the engineers.

In addition to the DEC equipment at the site, there is a large H-Series IBM machine with substantial on-line disk storage and 3850 honeycomb store. A large CDC computational engine also is available. The customer began using Digital VAX systems eight years ago and has been building a portfolio of engineerng applications since then. The customer decided on a DECsite/Engineering environment because of the unique coherence of the computing capabilities it provides and the richness of the available software.

CONFIGURATION:

The total configuration is easiest to understand by considering four levels:

Equipment at individual desks (fig. desk)
Equipment in each engineering department (fig.dept.)
Equipment shared by several departments (fig. shared)
Central facility (fig. central)

Each is shown in the attached charts. Every manager, secretary, and engineer has a personal computer on his desk. It is the only terminal on the desk. The manager and secretaries do not have significant computational requirements so they use economical systems in the CT200 family. The engineers make heavy use of dynamic graphics so they require the power of a SUVAX workstation. Depending on the needs of the individual engineer, different SUVAX configurations were selected but, the average price was only $35,000.

A DECfileserver typically provides an additional Gigabyte of on-line storage for the computers in the department. It contains the department's shared documents (specifications, standards, etc.). Also, the files stored on the individual workstations are "backed-up" to the server automatically so that users do not have to worry about protecting their valuable work against equipment failure.

The individual systems are connected to the file server via an NI coaxial cable (also known as Ethernet cable) which runs through the building. NI was selected because it is inexpensive, yet provides the high-speed (10 Megabits/second) required for transferring files. Many shared devices (e.g., printers and other DEC computers) as well as some non-DEC devices also are connected to the same NI. Physical connection to the NI is easy, and it has the bandwidth necessary to support the large number of devices within the building.

Since some devices have enough capacity to serve more than one department, they are shared by several. For example, a DEC EP1 electronic printer outputs 10-12 pages per minute. In order to have printing capability conveniently located, one unit is shared by two departments. Thus, there are 35 units in the building. Also, shared by two departments is the VAX Nautilus "computation server". This is a powerful VAX processor (2.5 times a 11/780). It is totally compatible with the SUVAX workstations on each engineer's desk. But since the workstations are occupied running the dynamic display, they send any significant computation to the Nautilus server.

The managers and even some of the engineers may require access to a large IBM computer in the building or at other locations owned by the customer. Five departments share a DEC gateway/SNA. This device, connected to the NI, provides the necessary communication to IBM machines. For instance, a manager seated at his CT250 can query a database on an IBM system and extract data and reports. To the IBM machine, the CT appears to be a 3270-class terminal. Also, the IBM machine in the building has a large archival storage facility. Thus, many files are sent there when they are not in need of frequent updates.

Another shared device is the DECNET/X.25 Router. Three departments share one. It connects to X.25 public packet networks which is useful for accessing a number of commercial data services (e.g., scientific and legal abstract services). Also, the same units provide DECNET communications to the customer's DEC computers at other locations for exchange of files, electronic mail, etc.

Finally, the customer's engineers produce large volumes of documentatioin and specifications. High quality printing and photocomposition capability is required at several locations in the building. For every 10 departments, there is a suitable unit supplied by Xerox or another vendor. It is connected to the NI via a DECgateway/printer which translates the standard communication protocols on the DEC computers into the commands necessary to control the non-DEC printer.

There is one large central computing facility in the building. It is the only location with a computer operations staff. (All of the other locations are either unattended or operated directly by end-users.) The large IBM E-Series mainframe is located here along with a large system manufactured by CDC. There are five large DEC "database servers". Each is composed of a pair of Venus (3.5 times an 11/780) processors and 16 Gigabytes of on-line storage. The processors are totally compatible with SUVAX workstations and the Nautilus computers. They provide rapid access to a number of large, shared databases. Each processor pair has a redundant pair of HSC-50 intelligent disk controllers managing ten RA82 (1.2 GB) disks. The processors and HSC-50s are connected with a CI high-speed interconnect to achieve a high-availability VAX system cluster.

In addition to database service, one of the VAX clusters also runs the DECNET Network Management Facility. This product provides continuous status and performance information about the network in the building. One or more of the clusters also runs MUX200 software to interconnect the CDC computer with the rest of the network.

EQUIPMENT ON INDIVIDUAL DESKS

Fig. Desk

ENGINEER'S WORKSTATION

Graphics - Oriented SUVAX System with AZTEC (21F + 21R)

Engineering Applications

Graphics Mgmt System

Programming Software
- FORTRAN
- PASCAL (ADA)
- Small Talk
- Step

Database Software
- Datatrieve
- CDD
- Local RDMS
- FMS
- Design by Forms

Communication Software
- DECNET
- SNA Access
- X.25 Access
- Xerox Access

Office Aids
- Word Processing
- Mail
- Calendar
- File Cabinet
- Telephone Mgmt.
- Directory
- Desk Calendar
- Digicalc
- Prof. Time Acctg.
- Mini PERT/COST

Presentation Aids Package

Standard VMS Utilities

MANAGER'S WORKSTATION

CT250 with 10 Mb Mini-Wini and floppy

Graphics Mgmt System

Programming Software
- BASIC
- PASCAL (ADA)
- Small Talk

Database Software
- Datatrieve
- Local RDMS
- FMS
- Design by Forms

Communication Software
- DECNET
- SNA Access (includes 3270 emulation)
- X.25 Access
- Xerox Access

Office Aids
- Word Processing
- Mail
- Calendar
- File Cabinet
- Telephone Mgmt.
- Directory
- Desk Calendar
- Digicalc
- PERT/COST
- Budgeting Package

SECRETARY'S WORKSTATION

CT220 with Floppies

Database Software
- Datatrieve
- Local RDMS
- FMS

Communication Software
- DECNET
- SNA Access (includes 3270 emulation)
- X.25 Access
- Xerox Access

Office Aids
- Word Processing
- Mail
- Calendar
- File Cabinet
- Telephone Mgmt.
- Directory
- Desk Calendar
- Digicalc

In-House Publication Package

EQUIPMENT SHARED BY SEVERAL DEPARTMENTS

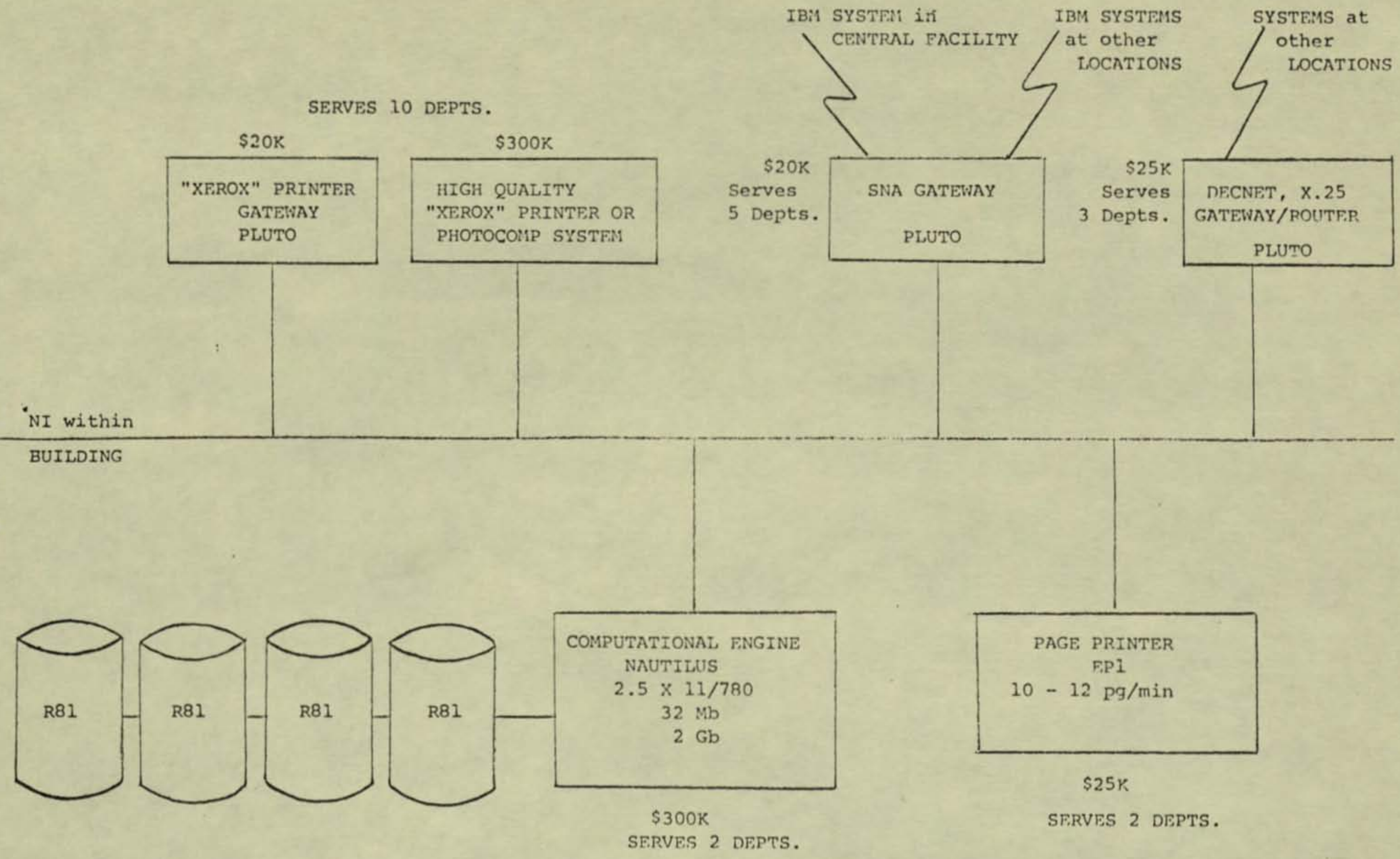

IBM SYSTEM in CENTRAL FACILITY
IBM SYSTEMS at other LOCATIONS
SYSTEMS at other LOCATIONS
SERVES 10 DEPTS.
$20K
$300K
"XEROX" PRINTER GATEWAY PLUTO
HIGH QUALITY "XEROX" PRINTER OR PHOTOCOMP SYSTEM
$20K Serves 5 Depts.
SNA GATEWAY PLUTO
$25K Serves 3 Depts.
DECNET, X.25 GATEWAY/ROUTER PLUTO
NI within BUILDING
R81
R81
R81
R81
COMPUTATIONAL ENGINE NAUTILUS 2.5 X 11/780 32 Mb 2 Gb
$300K SERVES 2 DEPTS.
PAGE PRINTER EP1 10 - 12 pg/min
$25K SERVES 2 DEPTS.

Fig. shared

RC 8/25/81

"TYPICAL" DEPARTMENT (70 in BUILDING)

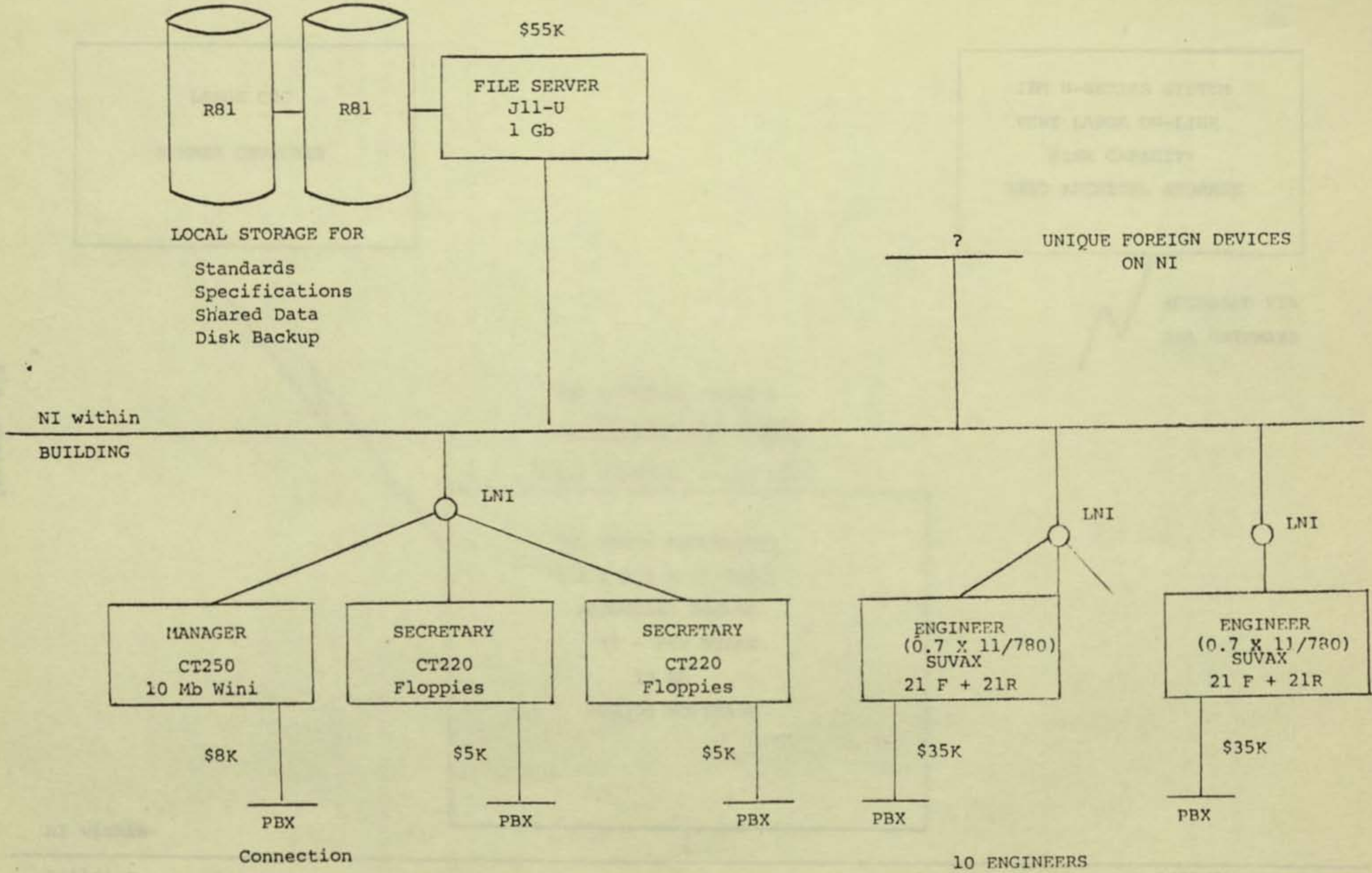
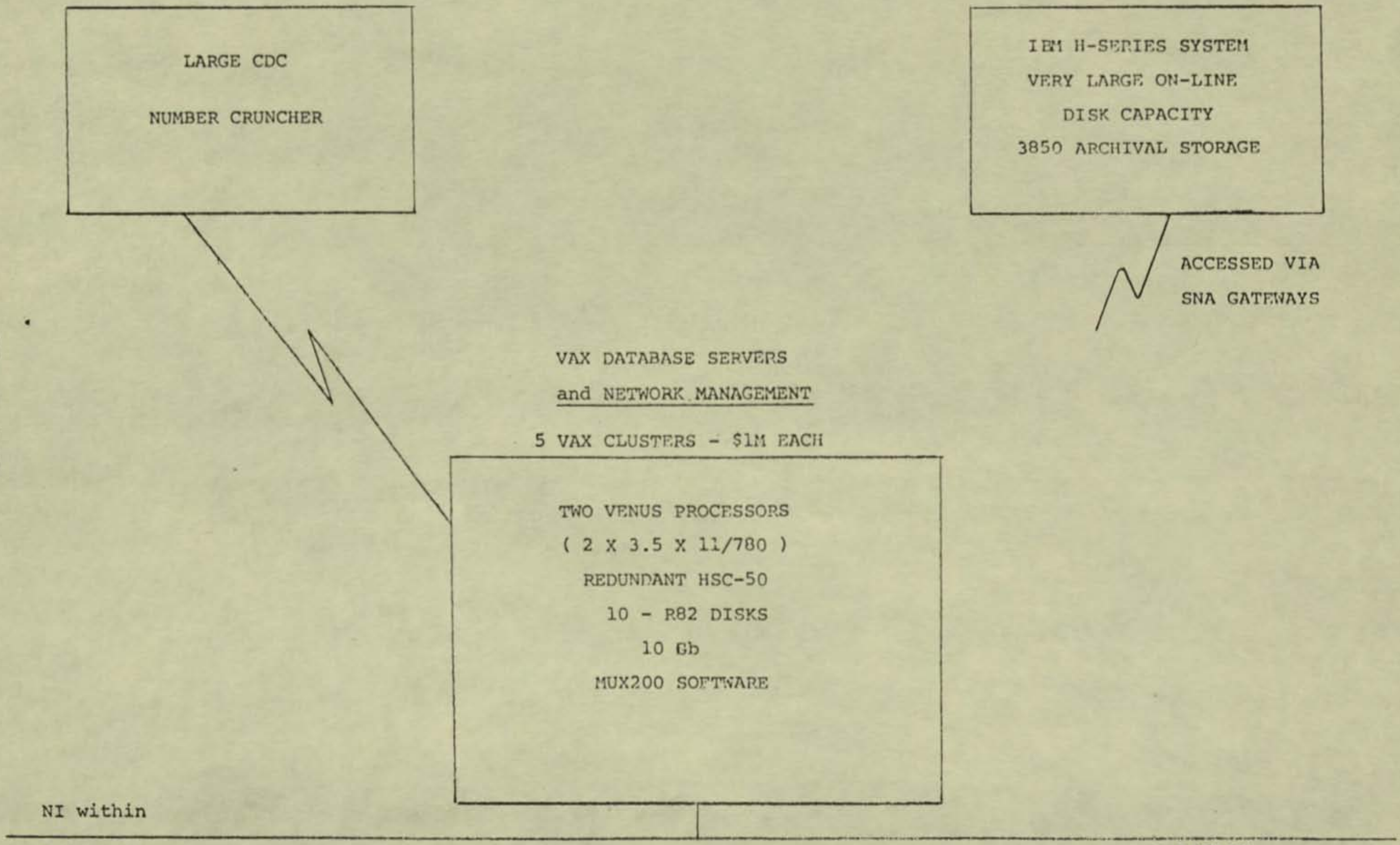


Fig. dept.

RC 8/25/81

CENTRAL FACILITY



LARGE CDC

NUMBER CRUNCHER

IBM H-SERIES SYSTEM

VERY LARGE ON-LINE

DISK CAPACITY

3850 ARCHIVAL STORAGE

ACCESSED VIA

SNA GATEWAYS

VAX DATABASE SERVERS

and NETWORK MANAGEMENT

5 VAX CLUSTERS - $1M EACH

TWO VENUS PROCESSORS

(2 X 3.5 X 11/780)

REDUNDANT HSC-50

10 - R82 DISKS

10 Gb

MUX200 SOFTWARE

NI within

Building

Fig. central

RC 8/25/81

PRICE:

For purposes of pricing, we can consider a single department and allocate its share of all the equipment used by multiple departments. (Note: This estimate does not include the cost of the IBM and CDC computing equipment.)

NO. OF UNITS	DESCRIPTION	UNIT PRICE $K	DEPT. COST $K	PERCENT OF TOTAL
1.0	Manager's CT250	8	8	1%
2.0	Secretary's CT220	5	10	1
10.0	Engineer's SUVAX	35	350	34
1.0	J11-V File Server (1 GB)	55	55	5
0.5	Nautilus Computation Server (w/2Gb)	300	150	15
0.5	EP1 Page Printer	25	13	1
0.33	DECNET/X.25 Gateway/Router	25	9	1
0.20	DECgateway/SNA	20	4	-
0.10	"Xerox" Printer and Gateway	320	32	3
0.08	Venus Database Server	1000	84	8
	Miscellaneous NI and LNI cabling costs	-	4	-
	Hardware within Building		719	70%
	Layered Software Price		308	30%
	Total Price for Department		1,027	
	Price per Engineer = $103K			

The prices are approximate and based on best current estimates. There is no meaningful mechanism today for pricing the software at a DECsite with 700 SUVAXs so the assumption was simply made that layered software would constitute 30% of the total system price.

Since each of the personal computers has a telephone management system with an auto-answer modem, every user can dial into his personal computer from a home terminal. (There is no need for a terminal concentrator in this configuration.) The cost of home terminals has not been included but could go as high as $80K to provide a CT250 for each of the ten engineers.

This is a very "rich" configuration in terms of both capability and price. The total spending on DEC hardware and software is close to $72 million dollars. While this is very high, the capital per engineer requiring this rich set of capabilities is only $103,000 (excluding non-DEC gear). This customer makes little distinction among the 700 engineers. Each receives an extraordinary array of hardware and software tools. Many other customers have employee populations with greater diversity of talent, assigned task, and computing requirements. They would choose a DECsite configuration of greater diversity. For instance, there probably would be heavier use of the lower cost CT systems. Of course, the option of matching the component system to the needs of each individual user is one of the great strengths of DECsite.

PERFORMANCE EXPECTATIONS:

The DECsite Simulation Program is available for customers who need to estimate the performance of proposed DECsite configurations. It asks questions about the workload on each of the systems in the network and then estimates the response time and throughput which can be expected. Using this tool, customers can design the DECsite configuration which meets their needs.

MAINTENANCE PHILOSOPHY:

The DECsite Network Management Facility (NMF) runs continuously on one of the VAX processors in the building. It monitors network status, reports outages, and provides performance statistics. Since there are many non-DEC devices on the NI, the NMF is able to diagnose problems to the failing unit on the NI so that appropriate maintenance personnel can be called.

All of the DEC equipment can be diagnosed remotely from Digital Diagnostic Centers. The VAX systems (including SUVAXs) are connected directly to the building's PBX so they can be diagnosed by direct telephone connection. Other DEC products can be diagnosed the same way or through the NMF over the NI.

SYSTEM MANAGEMENT PHILOSOPHY:

The DECsite configuration operates with a minimum of personnel. Most of the equipment runs unattended (e.g., file servers, gateways, etc.) or operated by the end-users (e.g., EP1 print server). The DEC equipment in the central facility can run unattended especially at night and over weekends. However, an operator usually is in attendance during the day to respond to NMF reports, mount tapes, etc.

System programming requirements also are minimal. Since the system software is composed of standard DEC products, the major activity is installing new releases; and this function is supported by a standard feature in the NMF for keeping track of and installing (via the NI) the software on each system in the network.

Thus, the data processing organization is free to concentrate on system planning, special application development, and user support. They are not burdened with simple user questions on system usage, since each DEC machine has a powerful on-line help and self-instruction facility. Similarly, the staff works only on relatively complex applications since simple ones can be readily implemented by the users themselves. This translates into highly productive computing achieved with minimal staff -- an ever more important factor as trained staff gets harder to find and more expensive to retain.

GROWTH PATTERN:

The customer did not install the entire configuration on June 30, 1986. Rather, it grew as a series of incremental acquisitions starting with the customer's first VAX in 1978.

The customer was able to match the precise growth pattern of his organization and computing needs. The system best-suited to an individual's needs was placed at that individual's desk. The shared facilities were determined on the basis of each department's actual requirements. As computational needs increase in the future, the number of computation servers can be increased. The same applies to file, database, printing, and other requirements.

UNIQUE CAPABILITIES:

DECsite provides the customer with coherent computing:

Comprehensive - DECsite offers the broadest possible range of computing from a small personal computer for a few thousand dollars to large VAX clusters for several million. Software is available for office automation, traditional commercial data processing, and engineering and scientific applications. DECsite also interconnects to the customer's non-DEC equipment.

Compatible - User productivity in DECsite installations is significantly enhanced by the compatibility of the DEC software. Re-programming and data conversion are not required to move from one machine in the network to another. Data manipulated by one program (e.g., Datatrieve) is immediately available and useable by others (e.g., Digicalc, Mail, etc.). This uniform, graceful treatment of data means that users can apply the individual DEC software products in a simple, intuitive fashion. Users can think about their information and decision needs instead of trying to memorize a list of strange quirks associated with each product.

Comprehensible - The large number of software products available on each DEC workstation brings with it a danger. How can ordinary users learn to work efficiently with so many separate products? Of course, each product has on-line help and self-instruction facilities. But users need to do real work, not study the features of a computer program. DECsite products avoid the long learning-time and rapid forgetting-time syndrome by having a common, unified command interface. The rules for operating one program have the same style as those for all the others. Experience with one program makes it easy to learn another. In combination with compatible data-handling, the DECsite products offer an unparalleled enhancement to the productivity of system users.

COMPETITION:

Each of the component pieces of a DECsite configuration is a highly competitive, quality product in its own right. With broad-based competition from IBM, Japan, and the packagers of semicomputer chips and with the diversity of markets in which DEC operates, none of our products is without significant competition. Each must be built on fundamental strengths such as VAX architecture, competitive packaging, cost-effective storage, etc.

The real uniqueness offered by DECsite is its sheer breadth and coherence. Here Digital is without competition. No one else can offer the same capabilities in 1986.

IBM is tied to 370 (batch) software architecture because of PCM competition. Integrating these traditional products with the smaller, more interactive style of distributing computing forces design compromises that keep IBM products second-rate. With Japan focusing on cost-cutting imitation of IBM and Apple, they are not likely to be leaders in the development of the new computing style exemplified by DECsite.

The repackagers of Intel and other chips do not have the breadth of hardware or software to compete with DECsite. Xerox may know the office, but its follow-through on programs outside reprographics is questionable and their ability to compete in the data processing arena is well-established!

Thus, DECsite does not relieve any competitive pressure on individual products. But it gives DEC a unique set of capabilities and a unique market image based on the quality of our overall architecture and vision. That is a traditional strength which has always worked well for us. Many customers buy a 32-bit VAX system not for what they need today but for what they might need in the future. So too, customers will favor products that "play" in a DECsite environment not only for what they can do by themselves but also for what they can do when combined with other DEC products.

APPENDIX

Goal & Constraints

1. Typical customer site configuration that we want to sell in FY'86 based on Corporate Product Strategy.

2. Concentrate on building within worldwide networks.

3. Consider CI and other technologies in addition to NI.

4. Our output should be a basis for a "Product Brochure" (Look through a customer's eyes.)

5. Concentrate on product features/attributes that are not owned by a single Program Office.

Characteristics of DECworld

1. Component Nodes - Hardware/Software
2. Interconnects
3. Price
4. Applications being used
5. Uniqueness - customer benefits unique in the marketplace from DEC
6. Performance expectations
7. Maintenance philosophy
8. Growth pattern
9. Relation to other communications facilities (PBX, SNA, X.25, DECNET)
10. Relation to other competitor's systems
11. Conformance to standards (External/internal)
12. Connection of foreign entities to environment
13. System management philosophy

Critical Environments for Study??

1. Laboratory/Research/Engineering
2. Manufacturing factory
3. Computer room
4. Small business
5. Office in large organizations
6. Embedded applications

Participants - Customers

International Harvester - Loren Gilmore

General Motors -Dale Larson and Chip Lackey

E.I. duPont deNemour - George Keenan

General Electric - Jim Fitzpatrick

U.S. Steel - Jeff Edmundson

Georgia Pacific - Gary Brannan

Simpson-Sears - Maurice Anderson

British Leyland - Mike Colin

Renault - Claude DuCrocq

Bayer-AG - Walter Neveling

ADP - Jim Watson

Pratt & Whitney - Arthur Simonian

2. Systems Design Task

The second workshop again provided a loosely structured, interesting task to wrap up the seminar. For this workshop, the customers were divided into four groups of three each, and the groupings were primarily homogeneous. For example, one group was comprised of GM,. Renault, and British Leyland customers; another was DuPont, U.S. Steel, and Bayer AG.

The objective of the task was to examine specific applications and their relation to the total communication network as perceived by the customers within five years. Monitor Mahoney provided each group with a 4' x 5' piece of posterboard, glue, and paper cutouts in shapes generated by an IBM template. The shapes represented various hardware components of a computer system. The four groups, based upon the priorities of applications established in Workshop I, were assigned the task of designing, respectively, (1) distribution/warehouse management, (2) an inspection and test, (3) a wet make (process manufacturing), and (4) a dry make (discrete manufacturing) system. The groups were encouraged to label the hardware with regard to any special features or dimensions required, and were encouraged to show how the specific station communicated with local and remote hosts. The groups were given 45 minutes to complete the tasks.

The resulting systems are portrayed here (see following). Each group presented their system for approximately ten minutes. We will not attempt to re-create presentations here, but point out key features. Further explanation of the enclosed diagrams can be derived from enclosed notes (see notes) or by contacting Bill Mahoney or Wes Melling of MDC.

Warehouse Management by: Simpsons-Sears, Georgia Pacific, ADP

- o Full credit/accounting/order entry communicated via CRT throughout system - checks done on line at warehouse.

- o Color required to highlight bad credit, full orders, generally to assist lower grade employee complete transactions (need transaction processor).

- o Lower right cart is a "picker" - human operated or automated - color screen right on the cart, doing transactions - possible wireless application.

Inspection & Test by: International Harvester, GM, Pratt & Whitney

- Upper left is central engineering drawings bank, billions of bytes storate required. Graphics, color application.
- Ultra-high-speed comm lines needed between work station (lower left, right, center) and drawings bank through local, central hosts.
- Flow: Testor at work station suspects defect, uses color graphics terminal to draw print from central bank via central host and local plant host. Uses color to highlight print area where defect suspected. Records through terminal data on defective part or assembly, sends to local host while drawing stored locally or in terminal.
- Other applications as listed at station (lower right).

Process Manufacturing by: DuPont, Bayer AG, U.S. Steel

- Key is interconnect of plant hosts; USA is fully interconnected - Europe serially connected (cost makes central cross too expensive).
- RJE is comm architecture of today; SNA future.
- Micro's handle I/O processing at stations along "dataway".
- Terminals for maintenance, status checks, process accounting, etc. Color, graphics utilized. Hard copy required.

Press Plant by: GM, Renault, British Leyland

- o Automotive process stamping out floor panels.

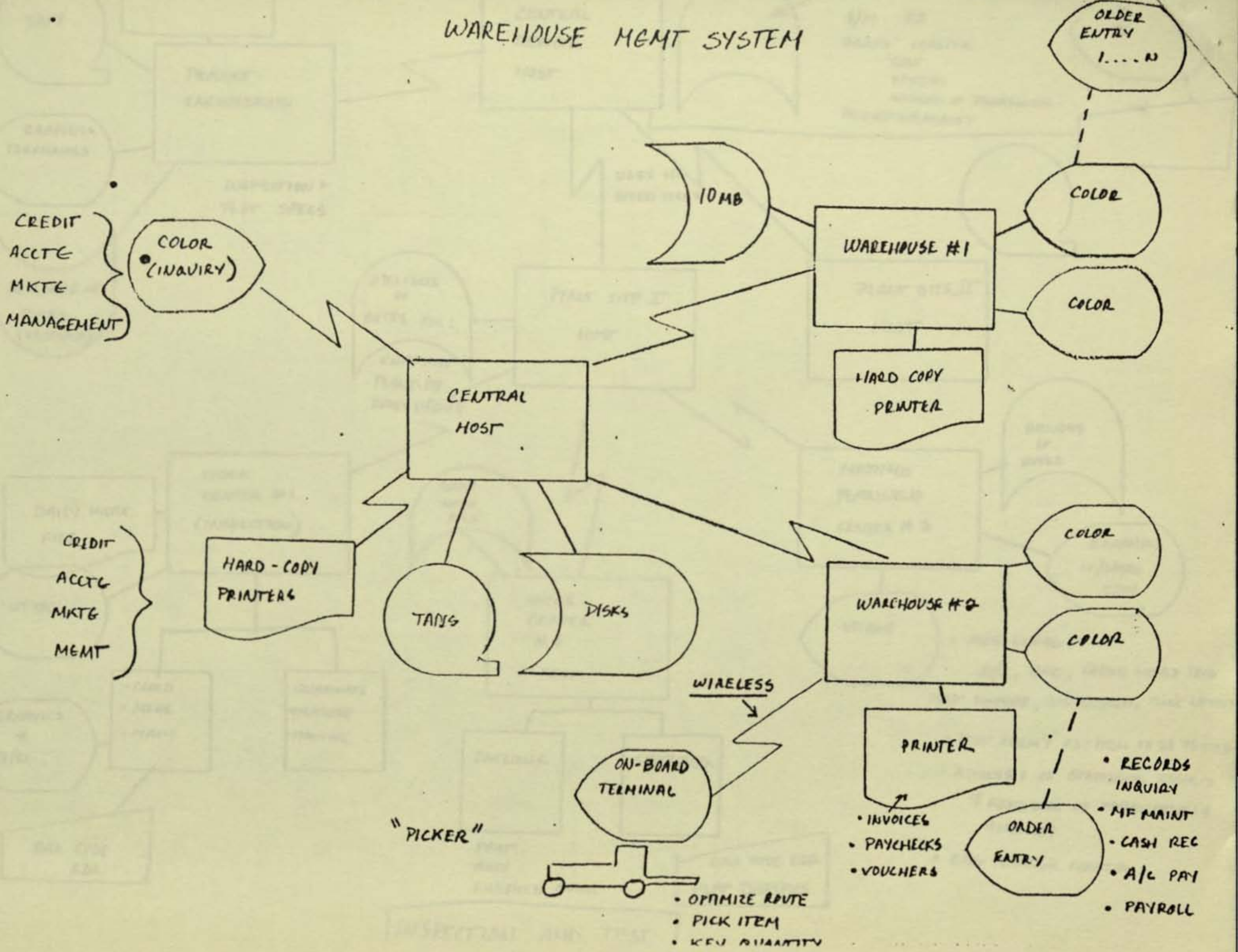
- o Fully automated through computer controlled robots which draw and stamp steel. No humans within "comm lines" block. Humans outside at terminals check status through sensors (micros) and t.v. monitors. Keyboard and CRT terminals within robot space are for maintenance personnel for troubleshooting and status only. Another possible color, graphics application.

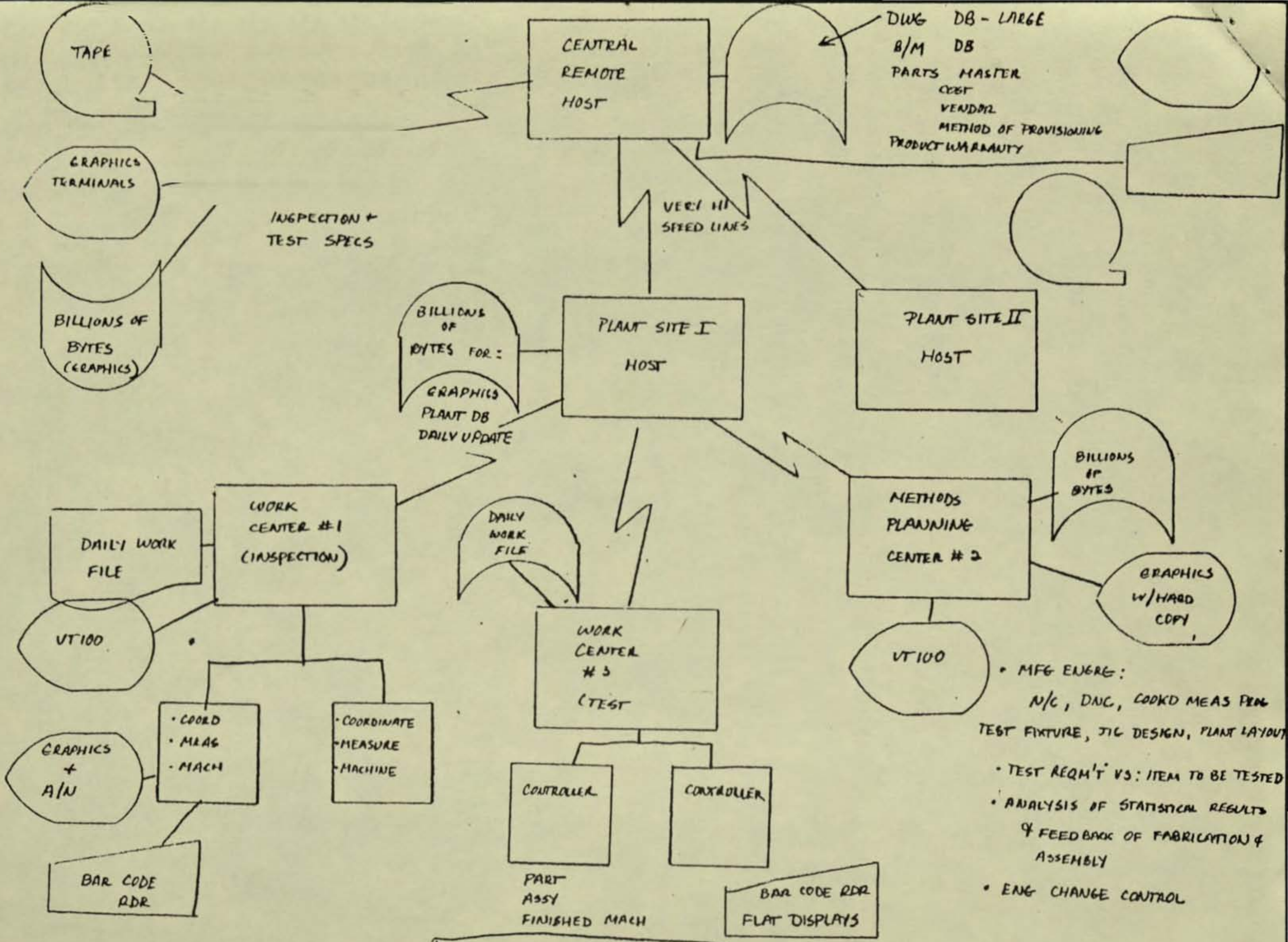
- o Dual "11/8Ø" processors control robot network. Message: We are good at utilizing 16-bit architecture, so please give us a bigger 11 vs. conversion to 32 bits.

- o Storage of completed panels connotes "picker"-type apparatus as described in warehouse system, but more likely pure robot.

Photographs of the completed systems are within the package. Response to the workshops on the part of customers was very positive; they very much enjoyed being left alone to interact and complete their tasks. One issue did arise - the five year horizon; customers wanted to know if they would have to wait five years for the terminal we described. Art Williams, Terminals Group, deftly explained that in the proposed modular terminal, Digital was taking a large step toward producing a terminals family whose technology would remain viable for far beyond just five years; this contrasted with the Company's past development of really point products in the terminal area, and therefore, while five years might be pessimistic in terms of the FCS of the first members of the modular terminal family, it was a good estimate of the timing of developing a solid family which would last ten years and beyond.

WAREHOUSE MGMT SYSTEM





- MFG ENGRG:

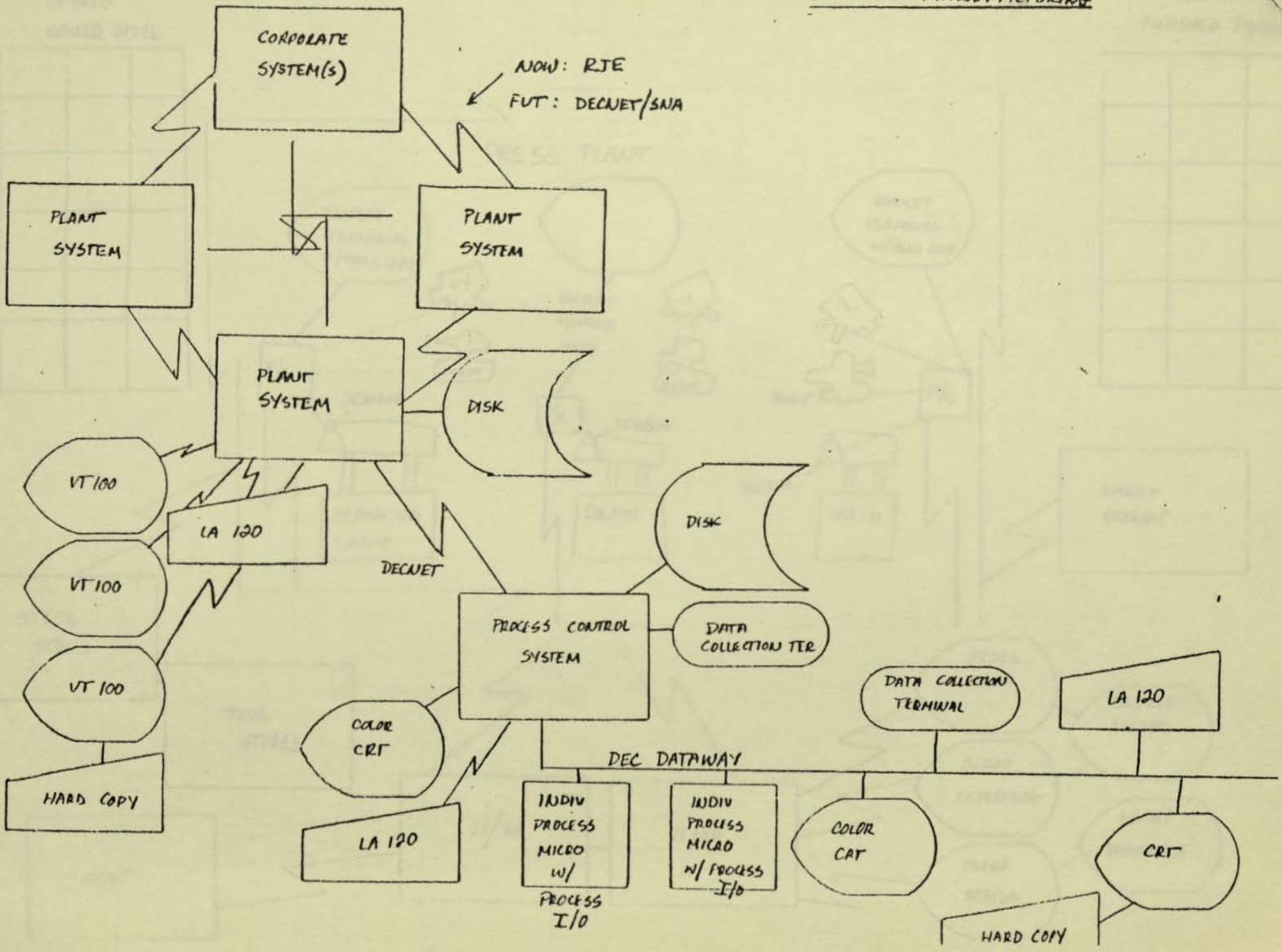
  N/C, DNC, COORD MEAS PROG

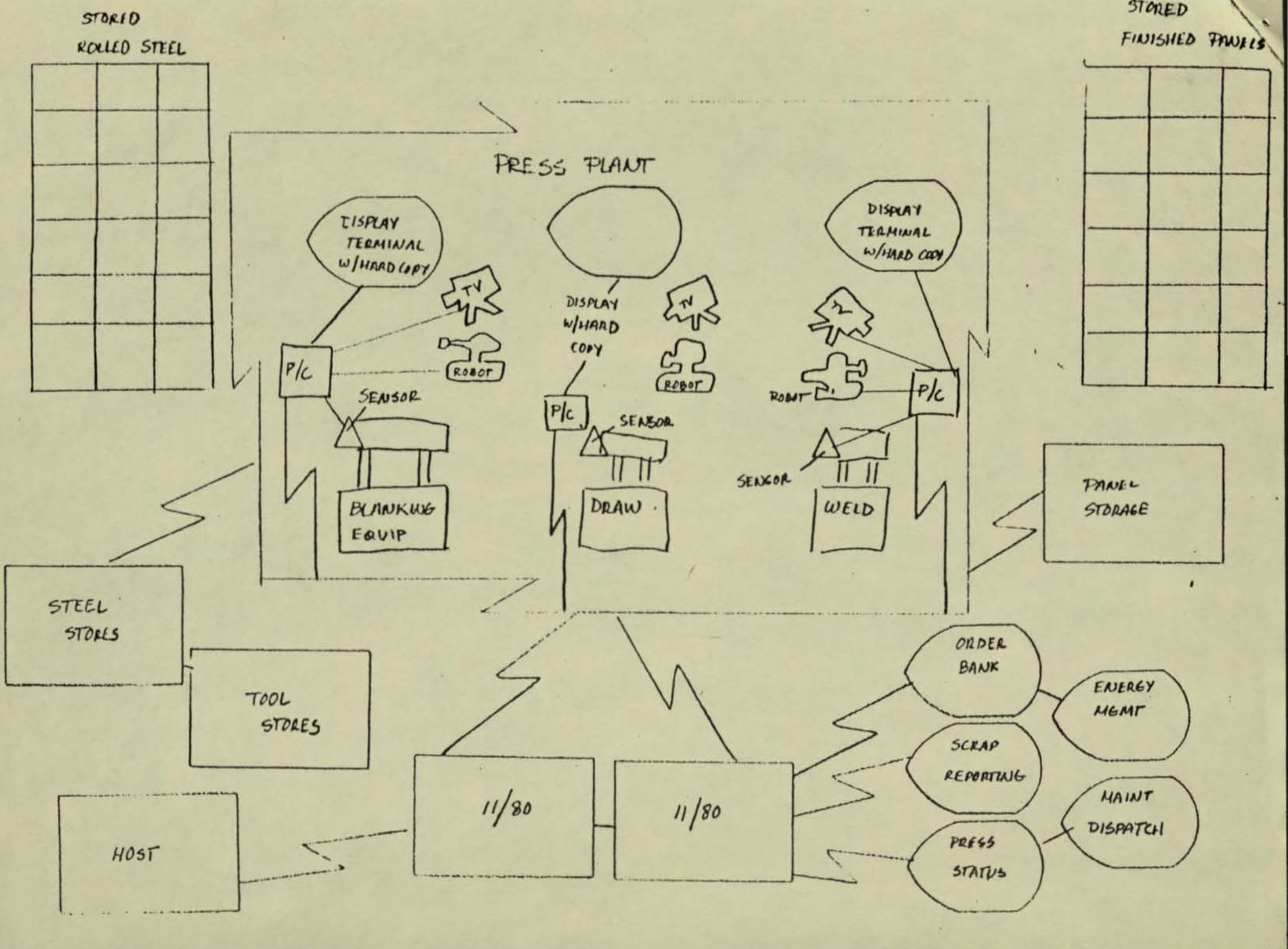
  TEST FIXTURE, JIG DESIGN, PLANT LAYOUT

- TEST REQM'T VS: ITEM TO BE TESTED
- ANALYSIS OF STATISTICAL RESULTS & FEEDBACK OF FABRICATION & ASSEMBLY
- ENG CHANGE CONTROL

INSPECTION AND TEST

PROCESS MANUFACTURING




STORED ROLLED STEEL
STORED FINISHED PANELS
PRESS PLANT
DISPLAY TERMINAL W/HARD COPY
TV
ROBOT
P/C
SENSOR
BLANKING EQUIP
DISPLAY W/HARD COPY
TV
ROBOT
P/C
SENSOR
DRAW
DISPLAY TERMINAL W/HARD COPY
TV
ROBOT
P/C
SENSOR
WELD
PANEL STORAGE
STEEL STORES
TOOL STORES
HOST
11/80
11/80
ORDER BANK
ENERGY MGMT
SCRAP REPORTING
PRESS STATUS
MAINT DISPATCH

.PLICATION ATURE	Mfg Eng'g. Planning ①	Mach + Assy ②	Mat. Handling ③	Insp Test ④	Mfg Control Mgmt ⑤	Comm Net ⑥
T	H	L Flat	H Flat	H Flat	H	H
LOR	M	L	L	M	H	H
APHICS	H	H	L	M	H	H
OGRAMMABLE (Remote)	H	H	H	H	H	Lα H
ECIAL KEYBOARD	H	H	H	H	H	H
RINTER	H	H	H	L	H	H
AR CODE	L	H	H	H	L	L
ARD READER	L	L	L	L	L	L
ADGE READER	L	H	H	H	H	L
IGHT PEN	M	L	L	L	L	H
ORTABLE DATA ENTRY	L	M	H	H	L	L
UGGEDIZED	L	H	H	H	H	L
APPLICATION SFTWRE	H	H	H	H	H	H
DATA ENTRY LANG	L	L	L	L	H	H
LOCAL COMM (Intra Plant)	H	H	H	H	H	H
REMOTE COMM (Intra Plant)	H	L	L	L	L	H
ocal Storage	L	H	H	H	L	H
oice Input	L	L	H	H	H	H
Wireless	L	M	H	H	H	L