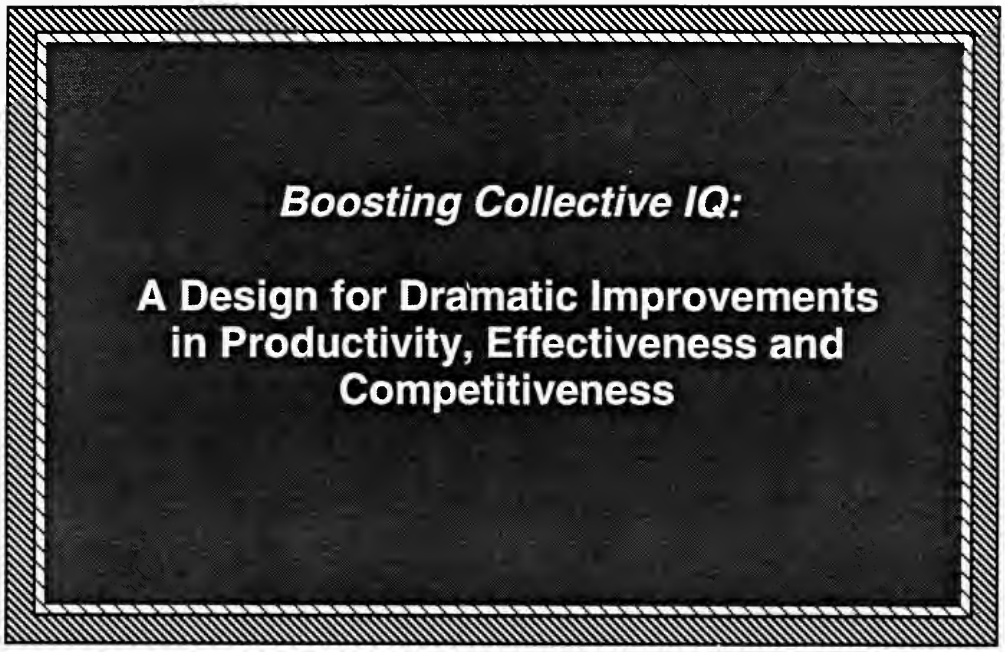
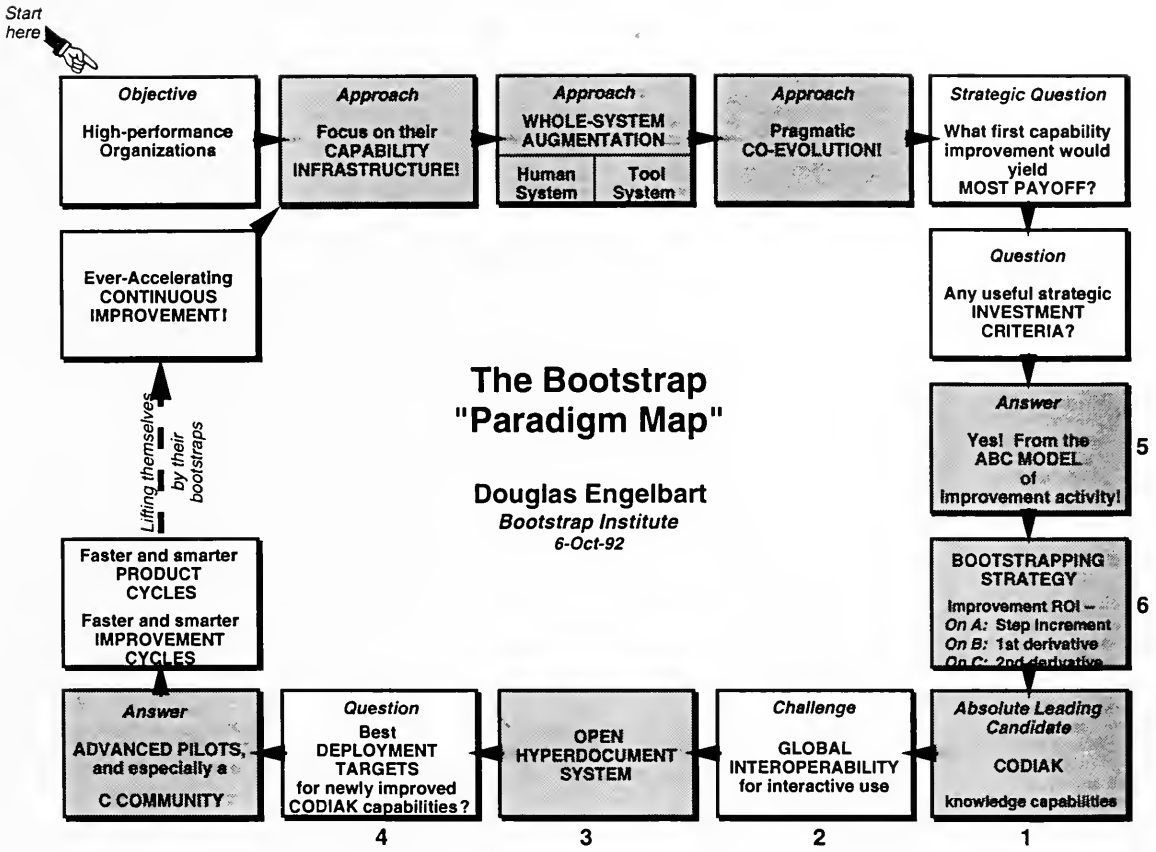



***Expedition Seminar • July 18,1994***

**Bootstrap Institute**
*Dr. Douglas Engelbart*
*Christina Engelbart*

# Contents




*Note: This July 18, 1994 briefing package summarizes a particular subset of these total-strategy elements — the following sections deal with specific elements as indicated.*

*The third enclosed paper (Groupware '92) pretty well covers this whole map.*

1

4

## Objective

***To pursue high-performance organizations.***

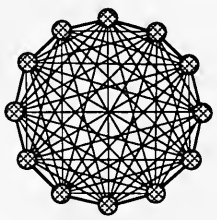
*To instill organizations with ever increasing speed, agility, precision and vision -- extending beyond improvements to their perception and reflexes, to include boosting their reasoning, memory, foresight, learning, and planning abilities as well (Collective IQ).*



Notes

5

## *BEGIN WITH BASICS:* PEOPLE WORKING TOGETHER IN AN ORGANIZATIONAL UNIT



Examples of org units, or ***knowledge domains***:

- an individual
- project team
- department
- functional unit
- task force
- committee
- whole org
- community

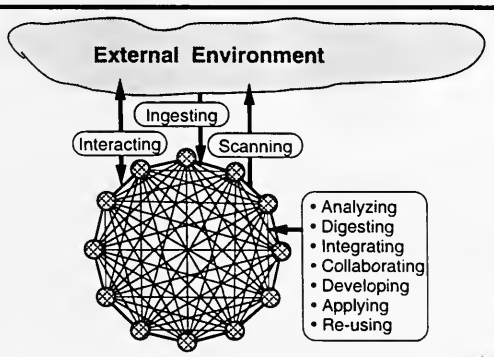
*Note: can be across-multiple organizations*



Notes

6

## EVERY VIABLE ORGANIZATIONAL UNIT REQUIRES BASIC KNOWLEDGE PROCESSES

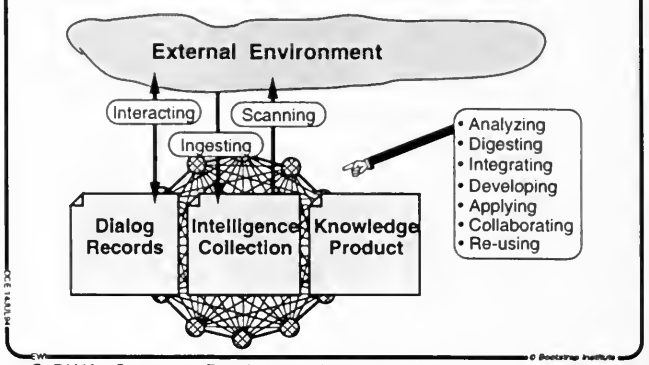





Notes

Notes

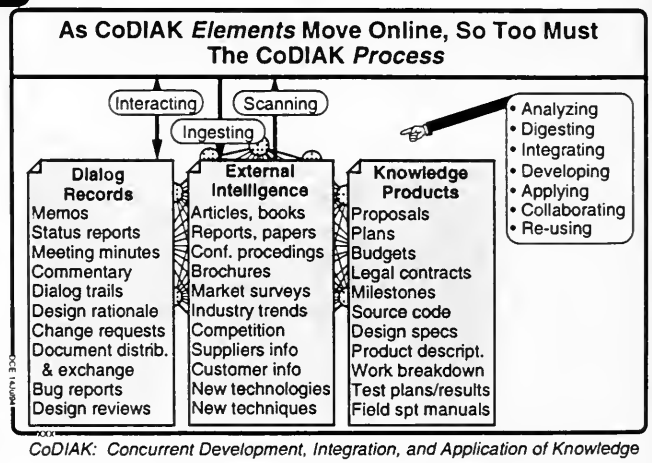
7




*CoDIAK: Concurrent Development, Integration, & Application of Knowledge.*

Notes

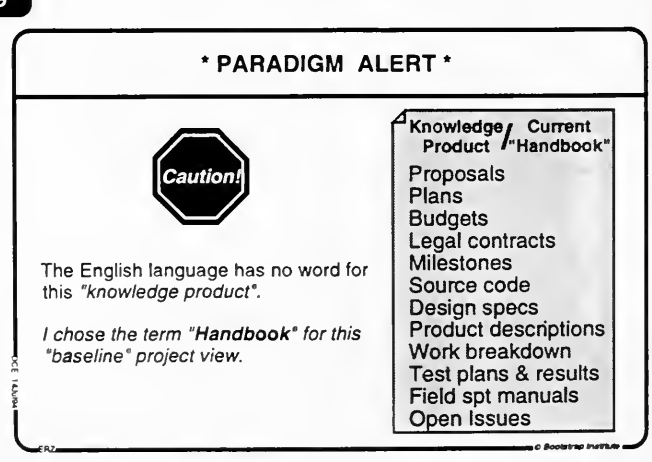
8




*CoDIAK: Concurrent Development, Integration, and Application of Knowledge*

Notes

9



10

## THE IDEAL HANDBOOK WOULD SHOW THE COMPLETE, CURRENT PROJECT STATUS

Goals, Plans, Designs, Budgets, Targets, Commitments, Schedules, Status, Staffing, Organization, Methods, Expectations, Specifications, Work Breakdown Structure, External Reference Data, ...

If kept constantly current and with visible relevance for all, a dynamic Handbook has central importance.

Storing intermediate Handbook states, and a record of the transitional dialog and reasoning, yields a critically valuable *organizational memory.*

DCE 18JUN90

EAJ

## Notes

*Below we use the model of a complex product-development project, in an industrial context, to illustrate the development of our CODIAK concepts.*

*Change the scale and substance, as for almost any complex pursuit, and the CODIAK picture will still emerge as a critical capability to augment.*

*("Handbook" -- the electronic embodiment of a knowledge product.)*

11

## This Basic Organizational Capability Emerges as our Highest-Leverage Target

The **Co**ncurrent **D**evelopment, **I**ntegration, and **A**pplication of **K**nowledge **(CoDIAK)**

*Developing an evolving knowledge base that integrates the concurrent contributions of many distributed participants, operating from the many (nested) knowledge domains involved within and among our enterprises, and concurrently supporting their application of the included knowledge.*

DCE 14JUN94

ERY



## Notes

## Notes

2

12

**Key Challenge:**

Interoperability among
CoDIAK domains

DCE 18Jul94

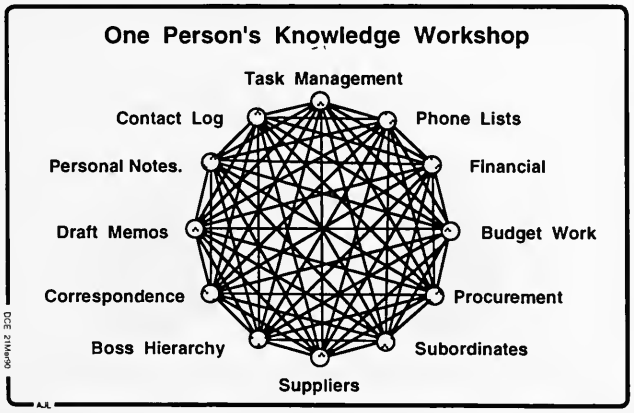
EWP



Notes

13

**EACH FUNCTIONAL DOMAIN IS A CANDIDATE FOR WORKING INTERCHANGE WITH ALL OTHERS**



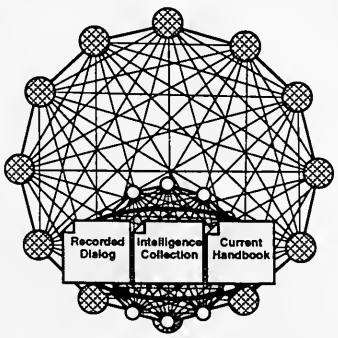

DCE 21May90

AJL

Notes

14

**ORG UNIT'S *CODIAK* PROCESS NESTED WITHIN OTHER ORG EFFORTS**




DCE 21Jul94

ESD



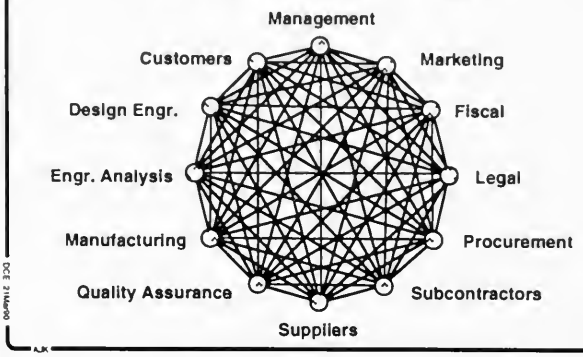
Notes

## Notes

15

### Each Functional Domain is a Candidate For Working Interchange With All Others

*Example: A Manufacturing Organization*



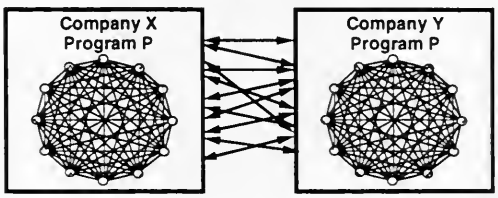

## Notes

*Heavy manufacturing industries have been active in exchange standards for CAD models, and Product Description data -- and also for electronic forms of conventional documents. But there is little appreciation (yet) for what the future, basic CODIAK processes will require.*

16

### Close Cooperation Between Large Organizations Puts New Demands on Knowledge-Work Interchange

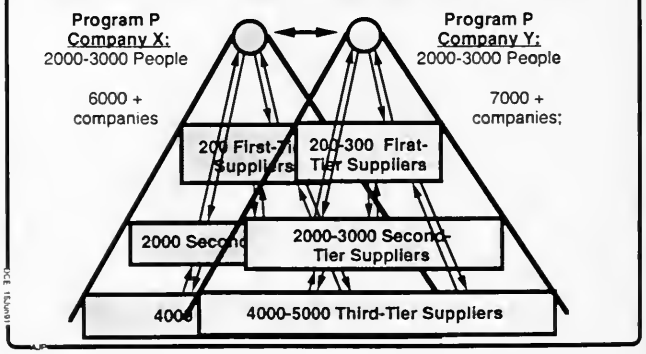
Example: Two Aerospace Companies required to do "Program Teaming"

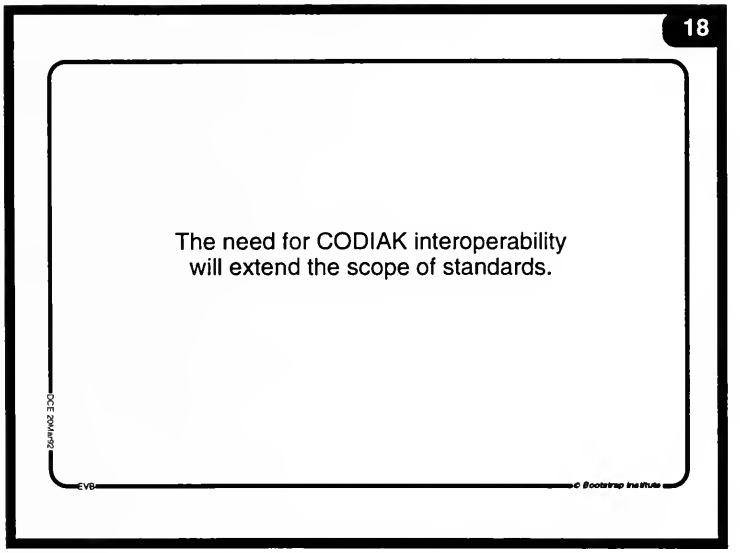



## Notes

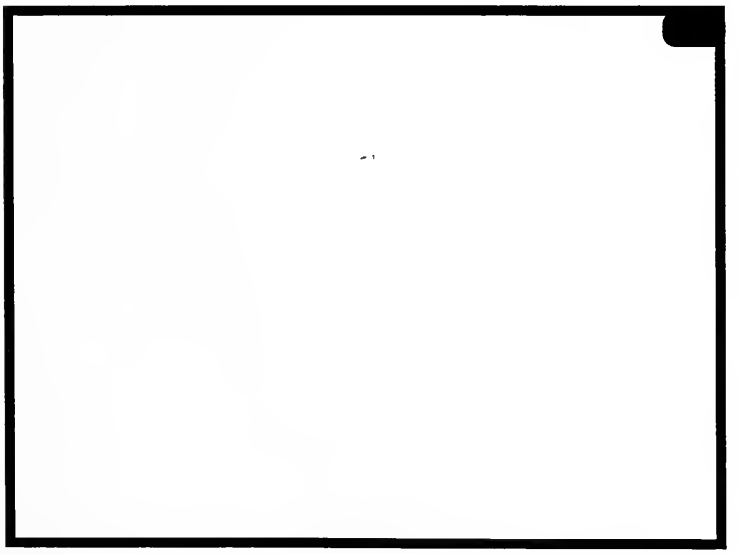
17

### TEAMED AEROSPACE PROGRAM -- IMMENSE DEMAND FOR KNOWLEDGE-WORK EXCHANGE

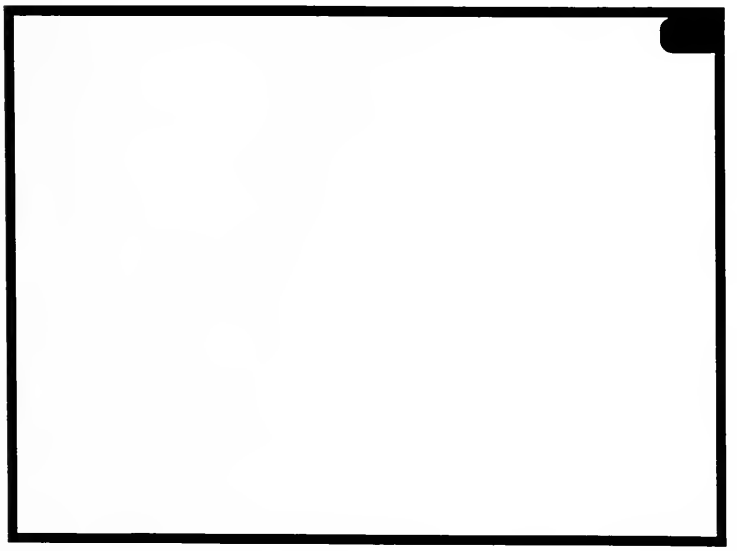





Notes



Notes



Notes

3

19

First beachhead for
improving CoDIAK capability:

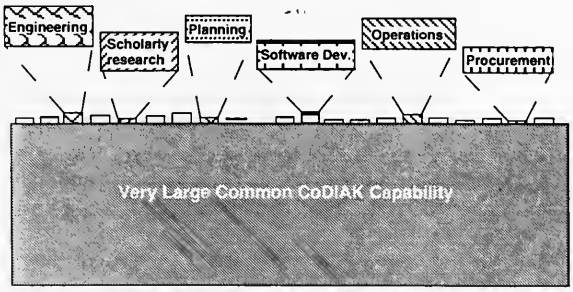
*Open Hyperdocument Systems*

EWO © Bootstrap Institute

## Notes

20

**Common CoDIAK Capabilities Underlie Every Special Knowledge Domain**




*Each adds its own special extensions, all of which must interoperate with the common capability, and with the other knowledge domains.*

EQE © Bootstrap Institute

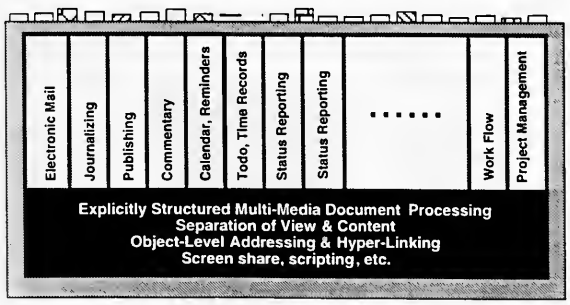
## Notes

*The special-domain list is endless—government at all levels, all business and industry, every institution, every facet of daily life.*

*In the short run, it is far easier for the marketplace to keep investing in isolated, large-system point solutions. But the CoDIAK capability is extremely general, and will inevitably be supported by a* ***generic solution*** *where the subset of special features within each special domain will be surprisingly small. Going after such a general solution sooner will be* ***much more cost-effective*** *than the point-solution alternative, and should yield mucy higher leverage in productivity and performance.*

21

**Open HyperDocument Systems:
An Integrated Knowledge Environment
to Support Very Large Common CODIAK Capability**




XXV © Bootstrap Institute

## Notes

Notes

22

**Examples of OHS Elements:**

- Document Structure
- Viewing/Browsing
- Object-Level Hyper-Linking
- Journal/Library and XDoc Facility
- Architecture Supporting Customizable "Knowledge Workshop"
- Architecture Supporting Shared-Screen Teleconferencing



Notes *Especially an OHS*

23

**AUGMENTING THE CODIAK PROCESS**

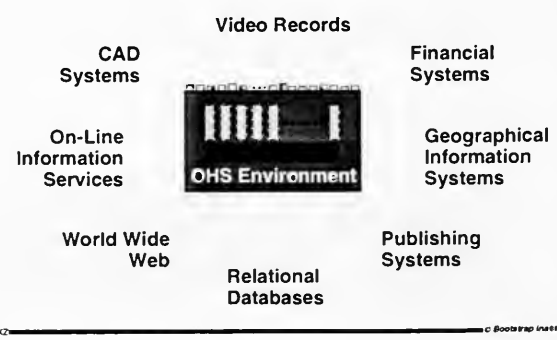
- Open-system, integrated architectures are of critical concern, supporting and spanning across:
  - many different classes of workers
  - many different workstations and application domains
  - organizational units, offices, and organizations
- Exploration must go beyond technology to include associated work methods and organizational structures.
- Exploration must go beyond R&D to include pilots, and strategic deployment within rapidly changing organizations.



Notes

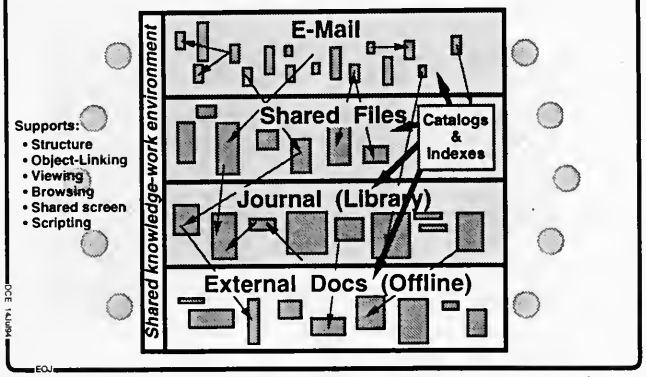
24

**Linkage to and From External Information**



25

## AN OPEN HYPERDOC SYSTEM (OHS): SHARING FILES & SHARING SCREENS

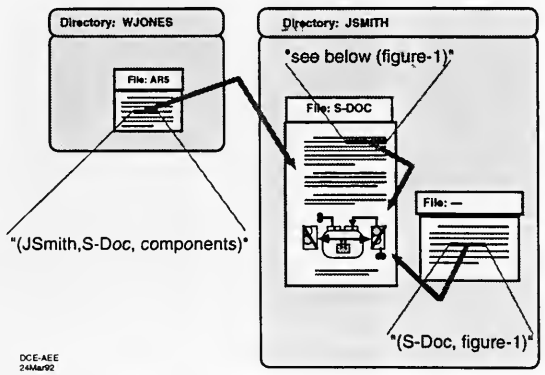



*"Hyperdoc" provides flexible linkages to any object in any multi-media file; "Open" provides vendor-independent access within and across work groups.*

Notes

26

## UNAMBIGUOUS TEXTUAL ADDRESSES ENABLE USE OF IN-FILE CITATION "LINKS"




Notes

27

*Authorship Provisions in AUGMENT DCE 9-Dec-83 17:43-PST OAD,2250,*

**EXAMPLE; Jump online to item 7 in "oad,2250," with flexible views**

Document address

more of the optional vocabulary and skills in a smooth, ward-compatible progression. 6h

**CONTROLLING THE VIEWS** 7

Object address

A user of a book, or of most on-line text systems, is constrained to ing the text as though he had a window through which he sees a fixed atted document. But as described below, our worker can view a s n of text in many ways, depending upon his need of the moment. 7a

**MULTIPLE WINDOWS** 7b

For whatever total screen area is available to the worker, his general performance will be improved significantly if he can flexibly allocate that area into arbitrary-sized windows whose contents can be independently controlled.

DCE 14Jul94
EVR


Notes

## Notes

*The remaining foils show a variety of views which a user could evoke when studying & modifying the structural content of an AUGMENT document.*

## Notes

## Notes

28

VIEWS

<rel. -- OAD,2250,7:cmsy> next view →<:ebt>

7 CONTROLLING THE VIEWS

**7a** A user of a book, or of most on-line text systems, is constrained to viewing the text as though he had a window through which he sees a fixed, formatted document. But as described below, our worker can view a section of text in many ways, depending upon his need of the moment.

***VIEW**: All levels; Numbers On; All line per statement; Blank lines.*

29

VIEWS

<:ebt> next view→<:zg>

7 CONTROLLING THE VIEWS

**7a** A user of a book, or of most on-line text

**7b** MULTIPLE WINDOWS

**7c** WINDOW VIEWS

**7d** USER-SPECIFIED SEQUENCE

***VIEW**: **2** levels; Numbers On; **1** line per statement; Blank lines.*

30

VIEWS

<:ebmg>

7c WINDOW VIEWS
7c1 STRUCTURE CUTOFF. Show only the
7c2 LEVEL CLIPPING. For the designated
7c3 STATEMENT TRUNCATION. For those
7c4 INTER-STATEMENT SEPARATION.
7c5 (Note: The foregoing view controls are
7c6 STATEMENT NUMBERS AND NAMES.
7c7 FROZEN STATEMENTS. A worker may
7c8 USER-SPECIFIED CONTENT FILTERS.

***VIEW**: **3** levels; Numbers on; 1 line per statement; No blank lines; Branch only.*

31

## Structure Provides "Handle" for Any Operation

Move Branch 9b (to follow) 9d

9 MODIFYING THE DOCUMENT
9a Given the array of capabilities described
9b Concurrent use of mouse and keyset also
9b1 Keyset hand strikes "m" and "b" (for
9b2 The mouse hand depresses the
9c A few extra verbs are useful for structure
9d A major source of structure-modification
9e (Note: I just had myself timed for this
9f In our view, interactive computer support

***Handles structural branch of any size.***
*(Can type Stmt Nums or click anywhere on stmt.)*

AFQ

## Notes

32

## Journal/Library Facility

Networking has opened whole new horizons for organizations, but also opened the "floodgates" for information overload. Too much, too hard to manage, and the important knowledge that might have enduring value is buried or lost.

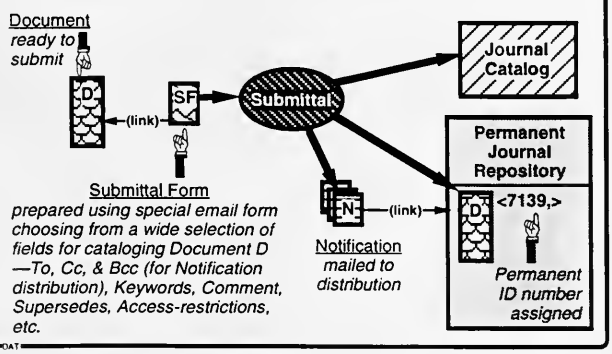
Try providing an integrated library-like system. Just prepare a submittal form for the message or document, and an automated "clerk" assigns a catalog number, stores the item, notifies recipients with a link for easy retrieval, notifies of supercessions, catalogs it for future searching, manages document collections, ...

EVR © Bootstrap Institute

## Notes

33

## A Journal System Provides "Hypertext Publishing" for Durable Recorded Dialog and Exchange




DAT

## Notes

Note: OHS should support multiple journals, distributed across multiple servers within and across organizations.

## Notes

34

### A JOURNAL SUPPORTS RECORDED DIALOG

| Function | Examples |
|---|---|
| • Formalized email | Memos,notices, thinkpieces, status reports |
| • Online conferencing, BBoard | Ongoing discussion topics, items of interest |
| • Online publishing, "Digital libraries" | Plans, reports, papers, reference material |
| • Document exchange | "Here's the latest draft for review -- note major changes to <sect-f> |
| • Document review | "Inconsistencies in <2a> and <5d>..." |
| • Doc management | Storing intermediate states of proj docs |
| • Doc accountability | Versions tracked, signatures verifiable |
| • Intelligence collection | Market research, news releases, findings |
| • More ... | |

## Notes

*An integrated external-document (XDOC) system was planned from the beginning to be a part of what we now call the CODIAK capability.*

*It remains a basic element in expected future Hyperdocument Systems.*

35

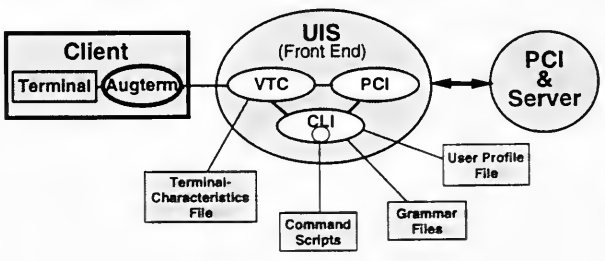
### CONTROLLING EXTERNAL DOCUMENTS IS AN IMPORTANT COLLABORATIVE FUNCTION

- Books, clippings, articles, etc.
- Catalogued and indexed via same tools as for "internal," on-line documents.
- Important purpose is to support citation links to external material.
- Another purpose is the common one of facilitating retrieval and access.

## Notes

36

### OHS Architecture Provides "Knowledge Workshop" for Many Classes of User



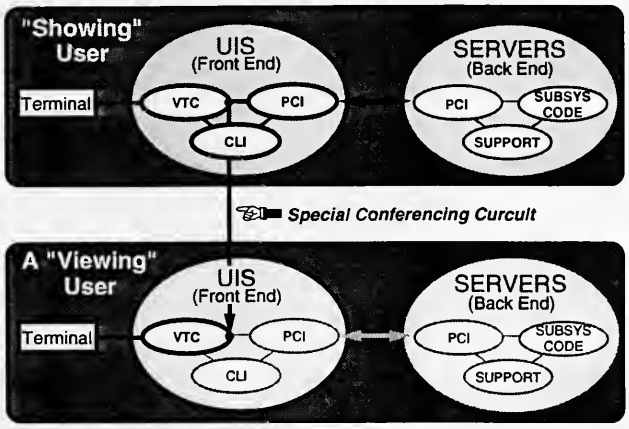

**VTC:** *Virtual-Terminal Controller*

**CLI:** *Command-Language Interpreter*

**PCI:** *Procedure-Call Interface*

37

## SHARED-SCREEN CONFERENCING




DCE 14Apr92 AJC

## Notes

*Use of the Conference Subsystem enables the setup and subsequent control of shared-screen conferencing. Control may be passed to any participant. The controlling user may employ any subsystem and operate upon any file to which the initiating, "showing" user has access. A participant may use any terminal equipment or workstation which the VTC module and a specific "characteristic" file provides for.*

38

## A Shared Screen Feature Has Many Uses

| Function | Examples |
|---|---|
| • Joint review and/or edit | "Let's finalize this wording, then sign." |
| • Collaborative planning | "Let's outline the pros and cons." |
| • Guided tours of the knowledge domain | "Can you show me those figures?" |
| • Online coaching | "Can you show me how you compiled that code?" |
| • Meeting support; remote briefing support | Preparation, formulating/displaying (dynamic) agenda and group notes, presenting/retrieving docs, full remote participation... |
| • More ... | |

DCE 14Apr92

ELM



## Notes

39

## OHS: TO SERVE "ALL DOCUMENT NEEDS" WITHIN VERY LARGE PROJECTS

Large, small: formal, legal documents or informal working notes.

"OHS E-mail" to convey a general-purpose "hyperdocument" of any size.

Requirements, specifications, design details, status reports, work breakdown structures, change orders.

References, instructions, policy, glossary, RFP, bids, work orders, "even" source code.

DCE 21May92

DAB

## Notes

Notes

40

## Essential Elements of OHS
Document-centered hypermedia groupware environment

| | |
|---|---|
| *Hyperdoc* | • Explicitly structured multimedia documents<br>• View control of objects' form, sequence, and content<br>• Object addressability -- global and human-understandable<br>• "Hyper" linkage -- reference; hot; implicit; indirect; back<br>• Link addresses -- human-readable and -interpretable<br>• Hard-copy print options to show object addresses and links |
| *Groupware* | • Shared filing, with version and access control<br>• Personal signature encryption<br>• Hyperdocument email; Hyperdocument Workflow<br>• Hyperdocument "Journal" System (library)<br>• External document control ("XDoc")<br>• Shared-window teleconferencing |
| *Architecture* | • Global and individual vocabulary control<br>• Many look-and-feel interface choices (incl. high-performance)<br>• Inter-linkage between OHS and other systems<br>• Interoperable, integrated, extensible, evolvable, scalable |



EWT 

Notes

41

## Potential OHS Applications: To Support and Integrate CoDIAK-Intensive Work

- Program Management
- Collaborative Planning & Tracking
- Command and Control
- Concurrent Engineering
- Software Engineering (CASE)
- Acquisitions Support (CALS)
- CA~ Integrated Architectures
- Contracts Management
- World Wide Web
- Records Management
- Digital Libraries
- Tele-Commuting
- Online Document Delivery
- Groupware, CSCW
- Enterprise Integration
- Total Quality Management
- Continuous Process Improvement
- Re-Engineering
- Organizational Learning
- Distance Learning
- Technology Transfer
- etc.....

***Supporting improvement programs offers special strategic leverage (the essence of bootstrapping).***



EWS 

Notes

42

**OHS Alliance**

## Some OHS Attributes and Benefits

**OHS is ...**

- An integrated, extensible environment with tools, not just tools
- Document centered, not tool centered
- Interoperable across multiple tools, users and communities

**OHS Provides ...**

- End-to-end integration of all information and associated processes from proposal to field support and reuse
- Effective capture of knowledge assets (decision trails, dialogs) for corporate memory

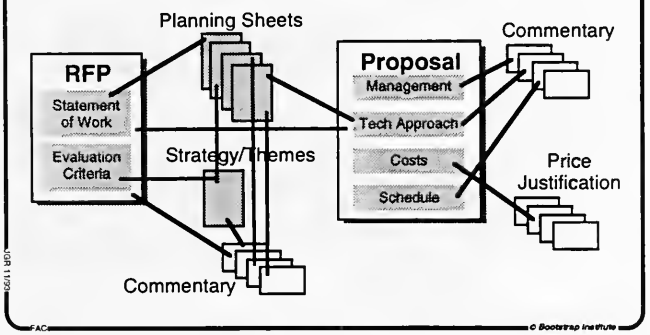


AI 

43

OHS Alliance

## Proposal Management



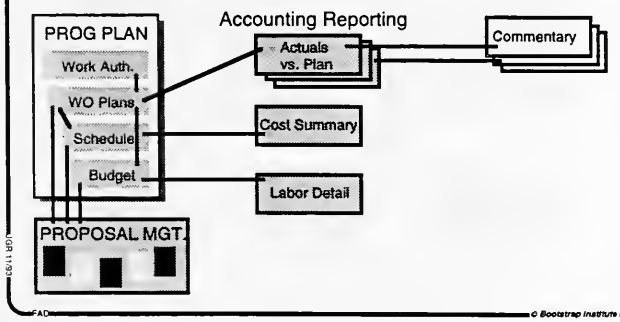



Notes

44

OHS Alliance

## Program Business Management



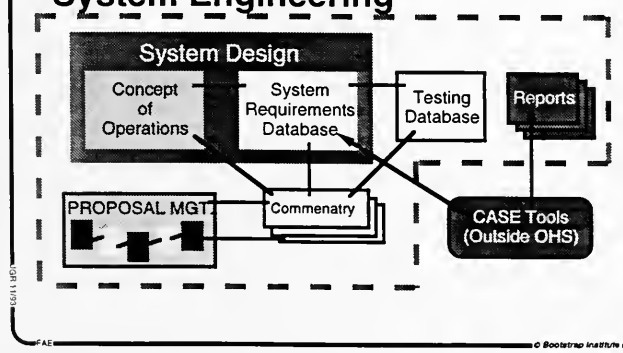



Notes

45

OHS Alliance

## System Engineering





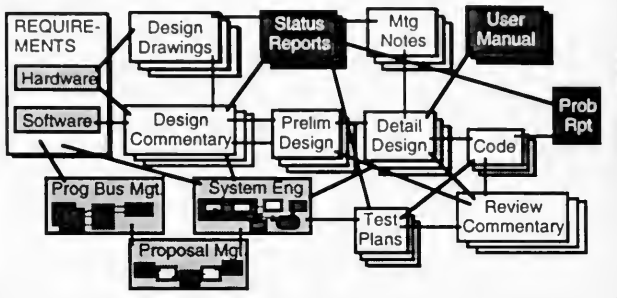

Notes

Notes

46

OHS Alliance

## Program Development & Maintenance

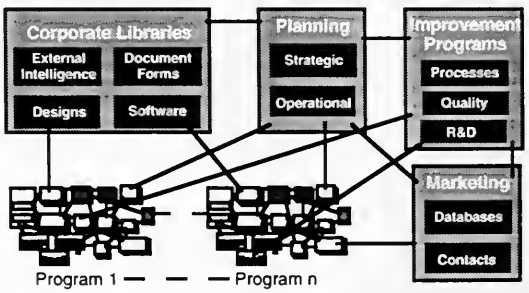



Notes

47

OHS Alliance

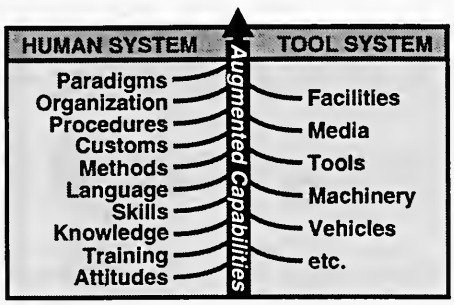
## Corporate Knowledge Base




Notes

48

## Our Capability Infrastructures Grew Via Co-Evolution of Human & Tool Systems






Notes

49

## HUMAN-SYSTEM CONSIDERATIONS FOR CODIAK SUPPORT

There is much more to be learned about the rigorous use of an OHS in a wide-area, distributed CODIAK process.

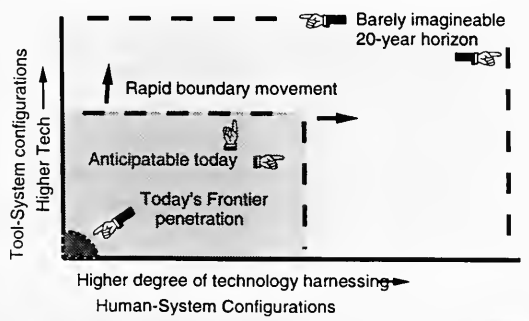
The human-system elements -- all the methods, procedures, conventions, skills, etc. -- must be highly developed in close association with the continuing evolution of OHS requirements.



Notes

50

## Much-Simplified Representation of Organizational Futures' Frontier






Notes

Notes

51

**Re: Your Improvement Plans — Prevailing Paradigms Strongly Affect Answers To:**

What degree of improvement is considered possible?

What degree of change can be contemplated — in tools, knowledge, skills, roles, methods, conventions, ...?

What degree of dependence can be placed on "The Marketplace" to provide the necessary change support?

Who among the current players in the marketplace are expected to play the critical roles of exploration and transition mapping?

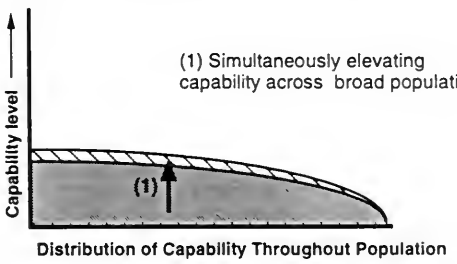


D.C. Engelbart

Notes

52

**Mode (1) For Increasing Regional or Organizational Capability**






D.C. Engelbart

Notes

53

**HUMAN-SYSTEM EVOLUTION NEEDS SUPPORT FOR GRADES OF USER PROFICIENCY!**

- Yes, "easy to learn" for beginners; but this will decrease in importance as the user-population continues to mature.
- Evolution will be severely inhibited if experienced, heavy users can not extend their capability with enhanced vocabulary and procedural proficiency.

54

## PURSUING HIGH-PERFORMANCE AUGMENTATION SHOULD START WITH SMALL GROUPS

Rather than large groups because: shorter evolutionary cycles; more economical scale of experiments; more "cultural mobility."

Rather than individuals because: exploring high-performance augmented collaboration is too promising to be omitted.

**"High-Performance Augmented Teams"**

DCE 6JAN90

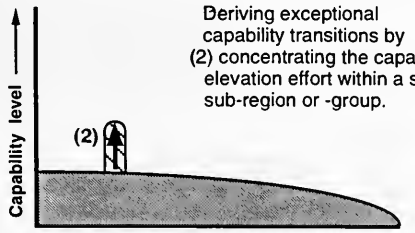
AIA

## Notes

*A very important type of future "intentional pilots" will be for specially recruited, equipped and trained "high-performance teams." Hard to picture any other way to accellerate evolution toward the future high-performance organizations.*

55

## Mode (2) For Increasing Regional or Organizational Capability




Deriving exceptional capability transitions by (2) concentrating the capability-elevation effort within a selected sub-region or -group.

DCE 16JUN94

XX13

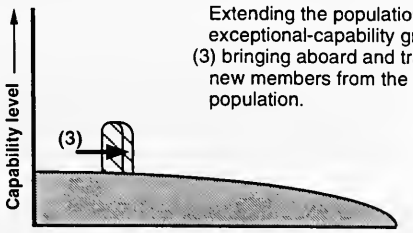


D.C. Engelbart

## Notes

56

## Mode (3) For Increasing Regional or Organizational Capability




Extending the population of an exceptional-capability group by (3) bringing aboard and training new members from the larger population.

DCE 16JUN94

XX19



D.C. Engelbart

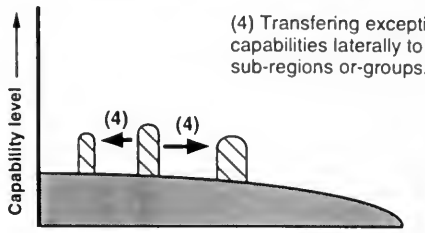
## Notes

Notes

57

## Mode (4) For Increasing Regional or Organizational Capability

(4) Transfering exceptional new capabilities laterally to other sub-regions or-groups.





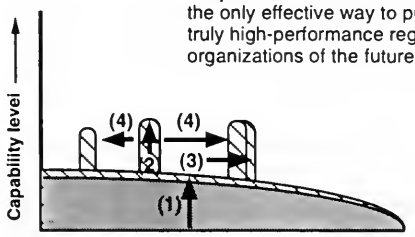

D.C. Engelbart

Notes

58

## Concurrent Multi-Modal Increase In Regional or Organizational Capability

Needed: A complete, coherent ***"Improvement Infrastructure"*** -- the only effective way to pursue the truly high-performance regions and organizations of the future.






D.C. Engelbart

Notes

59

## *Bootstrapping Option:* Continuous Improvement of the Improvement Process

**Question:**

Which set of capabilities might we focus on that, when improved significantly, would make us much more effective at ***all four*** improvement modes — (1), (2), (3) and (4)?

**Our answer: CoDIAK**
(a heretofore un-named set of capabilities — the Concurrent Development, Integration and Application of Knowledge.



D.C. Engelbart

60

## Current Marketplace Is Inadequate to Serve The Radically Expanding Frontier

Neither the user organization nor the groupware vendors are actively exploring the regions out into that frontier.

so, vendors don't really know what to produce;

and user organizations don't really know what they need.

Who is responsible, then to explore and chart the way?



**Notes**

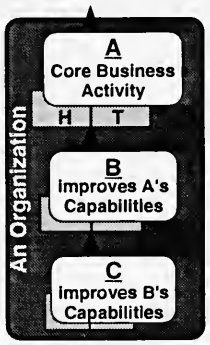
**Notes**

**Notes**

5

61

## ABC's of Organizational Improvement




**A Activity:**
*Product R&D, mfg, marketing, sales, accounting, etc. Ex: aerospace -- producing planes; congress -- passing legislation; medicine -- AIDS research.*

**B Activity:**
*Improving the organization's ability to perform A work. Ex: introducing email or CAD systems; upgrading quality processes.*

**C Activity:**
*Improving the organization's ability to perform B work. Ex: introducing better ways to address needs, or run pilots.*

EOF © Bootstrap Institute

*How an organization explicitly improves its improvement capability.*

Notes

62

## Prevailing B Practices Will Not Lead Directly to High-Performance Organizations!

The current means of developing and integrating improvements are not adequate for the scale and rate of change faced today.

- assessing needs and possibilities
- surveying and evaluating options
- selecting/developing, integrating, testing and applying
- co-evolving technical and non-technical requirements
- identifying suitable pilot groups
- running and evaluating the pilots
- learning how much to introduce, how quickly
- how to overcome cultural barriers
- how to quickly incorporate lessons learned

Need more effective ways of introducing dramatic improvements into rapidly shifting organizational targets.

ESB © Bootstrap Institute

Notes

63

## A COMMON GENERIC SET OF CAPABILITIES EMERGES FOR A, B, AND C

They all must:

- Identify needs and opportunities;
- Design and deploy solutions;
- Incorporate lessons learned.

All of which depends heavily upon the collaborative development of complex, integrated knowledge.

END © Bootstrap Institute

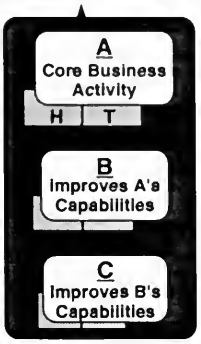
Notes

## Notes

*This is of basic importance -- suggesting investing in a permanent C-Activity towards continuous improvement in B-Capability. (Within an effective investment strategy, of course.)*

64

**HERE IS A USEFUL WAY TO CHARACTERIZE THE GOALS OF B AND C ACTIVITIES**




**B Work:**
*Reduce **product-cycle** time -- to make faster, smarter, higher-quality **A** Activities*

**C Work:**
*Reduce **improvement-cycle** time -- to make faster, smarter, higher-quality **B** Activities*

## Notes

## Notes

65

**Strategic Issues re: Investing Effectively in Regional or Organizational Improvement**

Invest how much towards capability improvement,

- into which capabilities,
- toward what targeted improvement levels,
- in which sub-regions or sub-organizations,
- in what sequence, and at what rate?

**These issues are addressed by the "Bootstrap Strategy" toward highest performance levels and maximum ROI.**

DCE 27Jul93 XX17 © Bootstrap Institute

D.C. Engelbart

Notes

66

If the scale and pervasiveness of change is to be as great as seems likely,

then a great deal more attention will have to be given to organizational evolution than we have ever before considered necessary!

New approaches will be required, with ample organizational support at very high levels.

DCE 22Oct92 AKH

Notes

67

Assume that the computer-communications revolution is only in its early stages, and that future changes will be **very pervasive** and **very significant!**

Then a serious future problem for every large organization (business or gov't) becomes how to accomodate **very complex** and **increasingly rapid** changes:

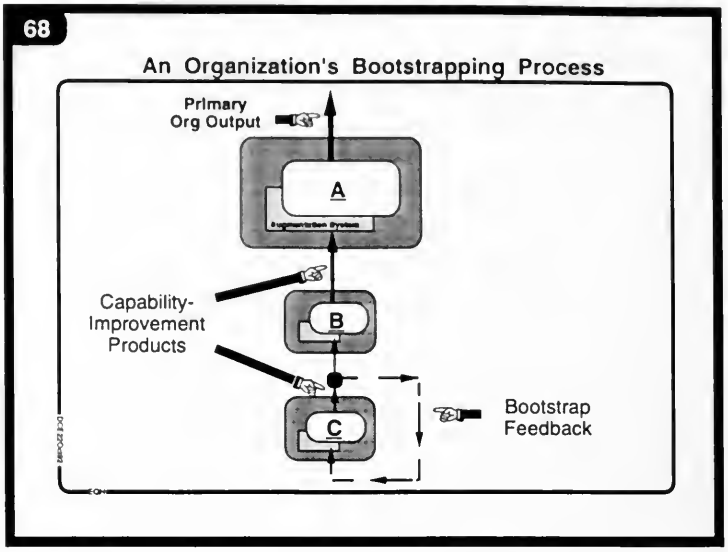
- in the organization's external operating environment;
- in the organization's internal operating environment.

DCE 22Oct92 AKF

Notes

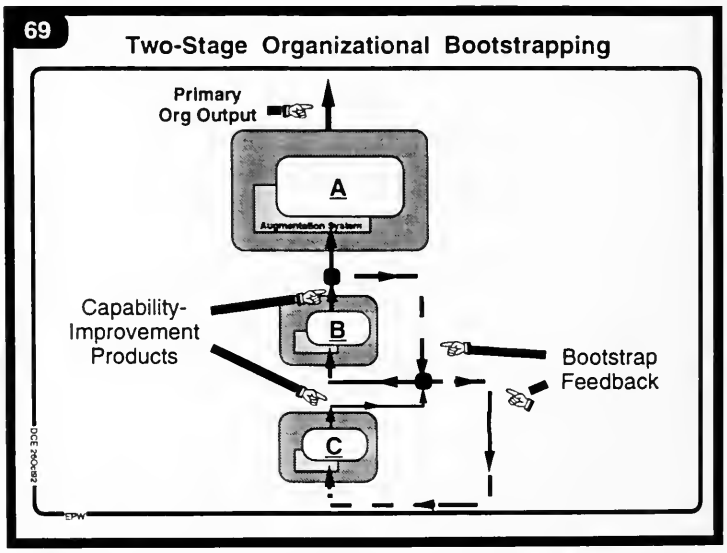
Notes

68



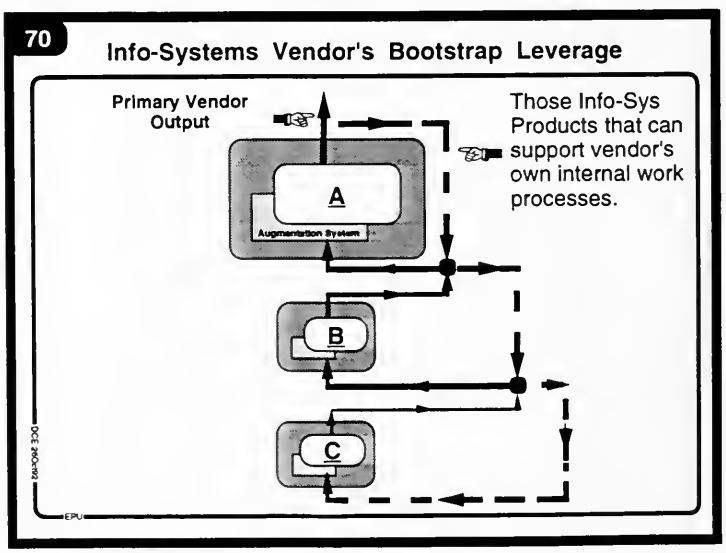

Notes

69

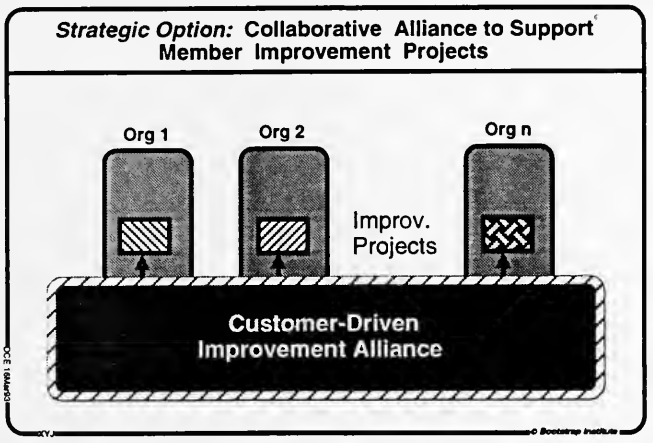



Notes

70

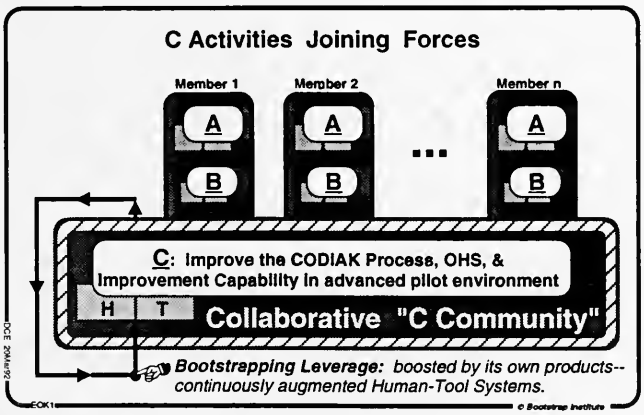


71




Notes

72




Notes

73

The Bootstrap Strategy is based upon the hypothesis that the constructive capabilities of individuals and their organizations can be further augmented to much higher levels than we experience today.

The Bootstrap Strategy offers a pragmatic way to check that out.

*What are we waiting for?*

Notes

74

## OHS Alliance

***Objectives***
- Develop evolving OHS prototype series
- Co-evolve best practices for
  - knowledge environment design
  - Collaborative processes, conventions, roles
  - training, coaching, and deployment strategies
- Collaborate on advanced pilots ("frontier outposts")
- Share cost, risks, and lessons learned
- Stimulate commercialization

***Approach***
- Stakeholder participation
- Use results for Alliance's own work accelerates learning
- "Grow" results from lessons learned, projected scenarios and ongoing intelligence ingestion
- Facilitate commercialization through vendors

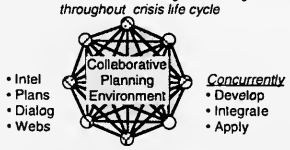


## Notes

75

## BI ARPA Project for Command Control

*"Any time, any where"*
*Distributed JTF Planning & Tracking throughout crisis life cycle*




**OBJECTIVES:**
- Deliver quantum-leap boost to JTF "Collective IQ" via integrated multi-media groupware "Open Hyperdoc System" tools & methods(OHS)
- Deliver similar capabilities to ATD Team, and to the nation at large

**KEY IDEAS:**
- Document-based, flexible browsing, editing, linking, structuring, coordinating, tracking, version control
- Integrated, evolvable infrastructure
- Co-evolution of tools & practices
- Support high-performance teams

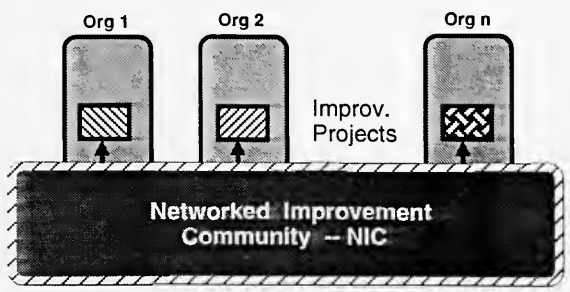
**FY94 MILESTONES:**
- Add GUI to first-generation OHS
- Transfer OHS to UNIX server and extend OHS with API & reach-thru
- Demonstrate OHS applicability (tools & practices)
- Conduct Workshop for ATD Team



## Notes

76

## AN IMPORTANT NEED: COLLABORATIVE COMMUNITY TO SUPPORT MEMBER IMPROVEMENT PROJECTS

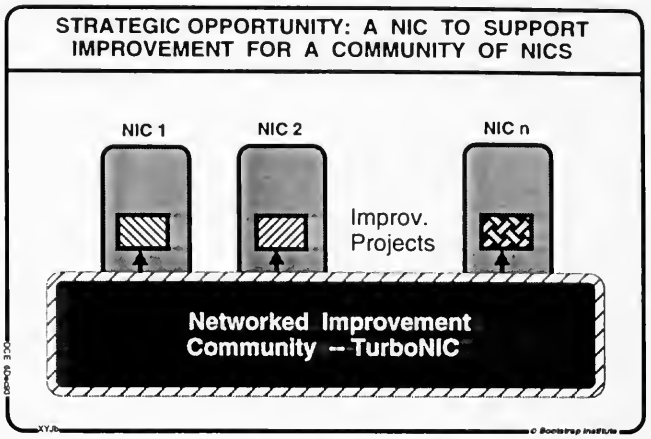





## Notes

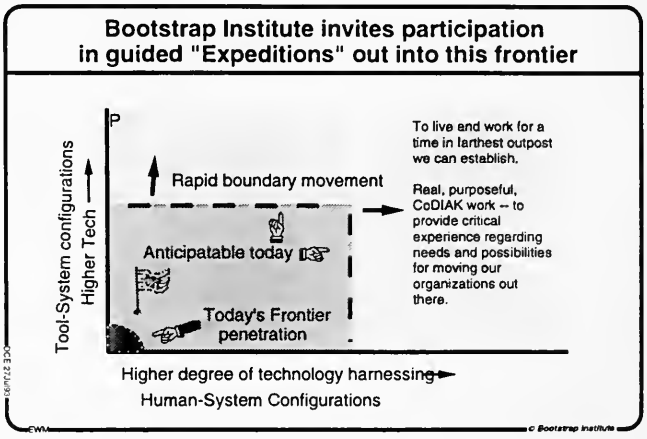
Notes

77




Notes

78




Notes


8

## Selected Papers

- Authorship Provisions in Augment
- Working Together
- Toward High-Performance Organizations: A Strategic Role for Groupware

Also published in ***Computer Supported Cooperative Work: A book of Readings***, Irene Greif [Editor], Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988, pp. 107-126. Also published in *Groupware: Software for Computer-Supported Cooperative Work*, D. Marca & G. Bock [Editors], IEEE, 1992.



# AUTHORSHIP PROVISIONS IN AUGMENT

**Douglas C. Engelbart**
*Tymshare, Inc., Cupertino, CA*
*Journal (OAD,2250,)*
*December 9, 1983* 1

*Note: Published in "COMPCON '84 Digest, " Proceedings of the 1984 COMPCON Conference, San Francisco, CA, February 27 - March 1, pp. 465-472.*

## ABSTRACT 2

AUGMENT is a text processing system marketed by Tymshare for a multi-user, network environment. In AUGMENT's frontend is a User Interface System that facilitates flexible evolution of command languages and provides optional command recognition features. Exceptionally fast and flexible control of interactive operations is enabled by concurrent action of mouse and optional one-handed chord keyset. Files are hierarchically structured, and textual address expressions can flexibly specify any text entity in any file. The screen may be divided into arbitrary, rectangular windows, allowing cross-file editing between windows. Many options exist for controlling the "view" of a file's text in a window, e.g.: level clipping, paragraph truncation, and content filtering. Structural study and modification of on-line documents are especially facilitated. A Journal system and "Shared Screen Teleconferencing" support collaboration among authors and their colleagues. Graphic illustrations may be embedded in the same file with text. 2a

## INTRODUCTION 3

AUGMENT was designed for augmenting human intellectual capabilities. It was targeted particularly toward the core work of professionals engaged in "tough knowledge work" -- e.g., planning, analyzing, and designing in complex problem domains. And special attention was paid to augmenting group collaboration among workers pursuing common goals. 3a

Authorship has received a great deal of attention in AUGMENT's evolution, as one of the central human activities to be augmented. An important set of provisions within AUGMENT -- in its architecture, design principles, and specific features -- is directly aimed toward bringing high performance to the authorship activities of knowledge workers. For the purposes of this paper, we thus speak interchangeably of "knowledge worker" and "author." 3b

We recognize explicitly that highly skilled workers in any field, and knowledge work is no exception, are those with good command of their tools. Our basic design goal was to provide a set of tools that would not themselves limit the

capabilities of the people using them. A system designed to encourage more skilled workers will always enable higher human performance than one designed to support less skilled workers. 3c

In this regard, our design goal was to provide as much capability as possible for each level of system usage skill, and a continuous evolution path between skill levels. We believe firmly that knowledge workers are motivated to grow in knowledge and skill and that provisions in system design should support this. As the rest of the paper reveals, this approach translates into a rich set of AUGMENT provisions, aimed at providing speed and flexibility for skilled workers in organizing and pursuing their core knowledge work -- in which "authorship" is a primary activity. 3d

An explicit sub-goal in AUGMENT's development was to "augment" the development, production and control of complex technical documentation -- through the whole cycle of gathering information, planning, creating, collaborating, reviewing, editing, controlling versions, designing layout, and producing the final documents. 3e

This paper concentrates upon the development phase of this cycle. AUGMENT has well-developed tools to support the later, production phase, but their discussion is not included here. 3f

Studying another's work provides a well-recognized challenge, but one of the toughest jobs is to study one's own work during its development: to see what it really says about Issue X; to see if it does provide for Concept Y; to see if it is reasonably organized and structured -- and to do these over a body of material before it is "polished", i.e., before it is well structured, coherently worded, non-redundant and consistently termed. 3g

## SOME BACKGROUND 4

### HISTORY 4a

AUGMENT is an integrated system of knowledge-worker tools that is marketed by Tymshare's Office Automation Division. The system was developed at SRI International over an extended period under the sponsorship of NASA, DARPA, and RADC. Commercial rights were transferred to Tymshare in 1978 (where the system has since been renamed from NLS to AUGMENT) and its evolution continued. A short history of AUGMENT's development may be found in <Ref-1>, along with a summary of system characteristics and features. The general R&D philosophy and the design principles behind AUGMENT'S development are laid out in <Ref-2>. 4a1

The system evolved on time-shared, mainframe computers, and in a packet-switched network environment. In 1970 our computer was the second to be

attached to the ARPANET, and since 1978 we have also operated extensively in the TYMNET environment. We have benefited directly from both the time-sharing and the network environments in matters that are important to the authorship process -- especially in dealing with large documents and multi-party documentation activities. In 1976-77 we conducted some applied studies for the Air Force, as reported in <Ref-3> and <Ref-4>, which concentrated upon this latter application. 4a2

## RELEVANT ARCHITECTURAL FEATURES 4b

Perhaps AUGMENT's most unique architectural feature is its User Interface System (UIS), a special software module, which handles the human/computer interfaces to all interactive programs. It takes care of all command-language dialog and connection protocols, and provides a framework for building a coherent and integrated user environment while supporting flexible evolution on both sides: on the user's side, with evolution of command function and terminology; and on the technology side, with evolving hardware and software. (Design details are outlined in <Ref-5>; rationale and utilization in <Ref-6>.) 4b1

The UIS provides a reach-through service to non-AUGMENT systems, and can optionally translate back and forth to a foreign program's command language. It also supports the shared-screen, remote collaboration capability discussed below. 4b2

AUGMENT's architecture provides for open-ended expansion and flexible evolution of system functionality and worker command languages. 4b3

It is assumed that for any class of knowledge workers, specialized application systems developed by other parties, perhaps running on other computers, will provide services worth integrating. The "author class" of worker should be no exception. Continuing evolution toward the "author workshop of the future" will certainly depend upon some such features in workshop architecture. 4b4

It provides adaptation for different terminal characteristics, enabling application programers to work as though with a virtual terminal. 4b5

## FILE CHARACTERISTICS 4c

AUGMENT employs explicitly structured files, with hierarchically organized nodes; each node can contain either or all of: up to 2,000 characters of text, a graphic structure, or other forms of useful data (e.g., digitized speech). The worker has a definite model in mind for the structuring of any file that he works with; in composing and modifying it he can organize and modify structure using the same verbs as for working with text strings (e.g. Insert, Replace, Move, Copy, Delete), with appropriate structural-entity nouns (e.g., Statement, Branch, Group, Plex). For any existing hierarchical structure, he

has many flexible alternatives for addressing its entities, modifying its organization, jumping around within it, and viewing it in a most beneficial manner. 4c1

(Note: AUGMENT workers generally use the term "statement" to refer to a file node, which is natural enough since the terminology became established before we added the graphic capability. Now an AUGMENT "statement" can contain either or both a text statement and a graphic diagram.) 4c2

## CONTROLLING THE TOOLS 5

Many of AUGMENT's unique author-support provisions address basic operations common to almost every task, things done over and over again. These operations, executed with speed and flexibility, provide for composing and modifying one's working material, and for studying what is there over a wide range of substantive levels -- from a single text passage to a collection of end-product draft documents and their associated set of working notes, reference material, and recorded-message dialog (assuming all to be on line). 5a

In the early stages of our program at SRI, we did a great deal of detailed work on what we called the "control interface" -- how users control the functional application of their tools. These details can be very important to "low-level" interactions which are done hundreds of times during a working day. Some of these details are quite relevant to bringing high performance to the authorship process. 5b

AUGMENT commands are expressed with verbs, nouns, and appropriate qualifier words; every command word is designated by entering one or more characters. The UIS recognizes the command word from these characters according to the command-recognition options designated in each individual's "profile file." Users seem to migrate fairly rapidly to "expert" recognition modes, where a minimum number of characters will elicit recognition of command words. The fully spelled-out command words are presented in the Command Feedback Window as soon as they are recognized. The Backspace Key will cause backup, one command word at a time. 5c

Of the system requirements behind our choice of this noun-verb command form, two are particularly relevant here: (1) The "vocabulary" of the functions of the tools, and of the entities they operate upon, must be as extensible as is a natural language; (2) Textual lists of commands must conveniently lend themselves to writing, documenting, and executing as "macro" commands. 5d

Screen selection is done with a mouse. If the command's noun is a single, defined text or structure entity, e.g., a "word", then there is only one selection needed (e.g., to pick any character in the designated word). 5e

Besides using a standard keyboard for character entry, an AUGMENT user may optionally use a five-key, one-hand, chord keyset. Remarkably little practice is required in order to enter alphabetic characters, one hand-stroke per character. With less than five hours practice, a person can begin profitably working in a two-handed, concurrent mode -- operating the mouse with one hand and simultaneously entering command characters and short literal strings with the other hand. 5f

Here is an example of a low-level action which reveals some basic characteristics of high-performance execution. It is a very simple situation, but representative of what is met over and over and over again in doing hard knowledge work. The worker is composing or modifying something in one area of the screen, when his eye catches a one-character typo in another area. For a skilled AUGMENT worker, the typo could be corrected in less time than it would take someone to point it out to him -- with three quick strokes of the keyset hand during a casual flick of the mouse hand, and an absolute minimum of visual and mental attention taken from the other ongoing task. 5g

Fast, flexible, graceful, low effort -- these are important to all high-frequency, low-level, knowledge-work actions. This same kind of speed and flexibility are achieved by skilled AUGMENT workers in executing all of the other functional features described below. Description of mouse and keyset, and their concurrent employment, may be found in <Ref-7>. 5h

## ADDRESSING THE WORKING MATERIALS 6

There is a consistent set of addressing features that a worker may use in any command to designate a particular structural node or some element of text or graphics attached to that node. It adds appreciably to the power and flexibility of the system commands to have a rich, universally applicable vocabulary for directly addressing particular entities within the working files. Below are some examples. 6a

### EXPLICIT STATEMENT ADDRESSES 6b

There are four "handles" by which a given statement may be directly addressed: 6b1

***Structural Statement Number***. This designates the current "structural location" of the statement. It is assigned by the system, depending upon where the worker installs or moves a statement within an existing structure, or how that structure might have been re-organized subsequently. It is usually expressed as an alternating sequence of number-letter fields -- e.g. "1", "1a", "1a1", "1a2", and "1b". At a worker's option, these same statement numbers could be shown as "1", "1.1", "1.1.1", "1.1.2", or "1.2", but this bulkier alternative is seldom chosen. 6b2

***Statement Identifier, or SID.*** This is a unique integer, assigned in sequential order by the system as each statement is first inserted, and which stays with a statement no matter how much its content may be altered or where it may be moved in its file structure. To make it uniquely recognizable for what it is, a SID is always displayed, printed, or designated with a prefixed "0" -- e.g., "012", "0417", etc. SIDs are particularly useful for referencing passages in a document while it is evolving. 6b3

***A Worker-Assigned Statement Name (or label).*** For any statement or part of the file structure, an author can designate as "name delimiters" a pair of characters that indicate to the system when the first word of a statement is to be treated as a name for that statement. For instance, if "(" and ")" are set by the author as name delimiters for a specified part of the file, any parenthesized first word in a statement would be recognized by the system as that statement's name. 6b4

(Note: It is optional whether to have any of the above three identifiers displayed or printed with the statements' text.) 6b5

***A Direct Screen Selection.*** When a statement to be designated is displayed in a window, usually the best way to "address" it is to use the mouse to position the cursor anywhere on the statement and depress the mouse's "Select" key (indicated below by "<Select>"). This mode is generally used for text manipulation -- selecting characters, words, numbers, visibles, invisibles, etc. (any of the text entities which have been made system recognizable). 6b6

## MARKERS 6c

As one "holds a place" in a book by leaving a temporary place marker in it, an author can place "markers" at arbitrary locations within an AUGMENT file. When placing a marker, he attaches it to a specific character in the text and gives it a name or label. Marker names are local to each file. Simple commands provide for displaying where one's markers are located and what their names are, for deleting or moving a marker, or for installing a new one. 6c1

A marker name may be included in an address expression, to provide another way of designating an address. A marker name can designate not only a particular statement, but a specific character within that statement. For example, "Copy Word #x (to follow word) <Select>" would designate that a word located somewhere in the file and marked with an "x" is to be copied to follow the cursor-selected word. There are many unique ways in which markers may be employed by an author who has integrated their artful use into her working methodology. 6c2

As a comparative example of some of the foregoing addressing forms, consider a statement whose SID is "069", whose statement number is "3b5", that has

statement-name delimiters designated for it as "NULL" and ":", that starts with the text "Capacity: For every ...", and that has a marker named "x" positioned on one of its characters. A command to move this statement could optionally be expressed as: 6c3

"Move Statement <Select> ...", 6c3a

"Move Statement 3b5 ...", 6c3b

"Move Statement 069 ...", 6c3c

"Move Statement Capacity ...", or 6c3d

"Move Statement #x ...". 6c3e

## RELATIVE-ADDRESS EXTENSIONS 6d

A sequence of characters may be appended to the address of a given statement to specify an address of a position "relative" to that statement. A major class of these designations deals with relative structural location, such as: Up a level, Down a level, Successor at same level, Predecessor at same level, Head at this level, Tail at this level, and End statement at last and lowest position in this branch. A period (".") in the address string indicates that relative addressing is beginning, and each of these relative-location designators is indicated with a directly mnemonic, one-letter designation. 6d1

For example, "Move Statement 0609 (to follow statement) 4b.dt" would move Statement 0609 to follow the tail statement of the substructure one level down from Statement 4b -- or, to conceptualize the associated address-location pathway, "go to 4b, then Down a level and to the Tail". 6d2

## EMBEDDED CITATION LINKS 6e

A special use of address expressions is within an explicit text entity that we call a "Citation Link" (or "Link" for short). Links are used as textual citations to some specific file item within the workshop domain. A link is delimited by parentheses or angle brackets and contains a valid address string whose path leads to the cited file entity. For example, "(0306)" or "(4b.dt)" are valid links. Also, the reference items at the end of this paper are statements named "Ref-1", "Ref-2", etc., and as such can be cited with links "<Ref-1>", "<Ref-2>", etc. An AUGMENT reader may travel via such a link directly to the referenced bibliographic citation. 6e1

A special feature in AUGMENT's link provisions is the use of "indirect link referencing". In path-following terms, including ".l" in an address string stipulates, "scan forward from this point to the next link, and follow that link to

its target." For example, to follow the path prescribed by link "(4b.l)", one would "go to 4b, then find the first link in that statement and follow the path that it specifies." This latter path in turn could prescribe use of another link, etc. There is no intrinsic limit to the number of these indirect links that may be employed in a given path -- only a natural caution against such a path looping back upon itself. 6e2

As an example, note that "<Ref-1>" is a link to the statement named "Ref-1", a bibliographic citation at the end of this paper. In that citation, there is a link to the original source document of the referenced publication, permanently stored in the AUGMENT Journal as Item 71279 (the Journal is described below). The point to be made here is that with the link "<Ref-1.l>", I can reference the original source document -- and a Jump Link command would "take me there." 6e3

## TEXT AND CONTENT ADDRESSING 6f

Other addressing options include scanning for a content match, and/or stepping backward and forward a given number of characters or words (or other text entities). For instance, the foregoing link could have involved a bit more smarts in designating which link to follow: e.g., the path for '(4b "*D" .l)' would be "to 4b, scan for first occurrence of "*D", then follow the next link found in that statement." 6f1

## OTHER-FILE ADDRESSING 6g

By preceding an in-file address string with a file address, and separating the two strings with a comma, one obtains a composite address designating a given entity within a given file. Extending this principle lets one prefix the file name with a directory name in which the file is to be found; and further, one can prefix this with a host-computer name. 6g1

For example, '(Office-5, Program-Documentation, Sequence-Doc, Specifications "Journal")' specifies the path: to the Office-5 host computer, to its Program-Documentation file directory, to its Sequence-Doc file, to its statement named "Specifications", and then scan to the location of the text "Journal". 6g2

If a person were working on the Office-5 host, he would only have to specify '(Program-Documentation, Sequence-Doc, Specifications "Journal")'. If he were already working within a file with its "link default" set to the Program-Documentation directory, he would only have to specify '(Sequence-Doc, Specifications "Journal")'. And if he were already working within the Sequence-Doc file, he would only have to specify '(Specifications "Journal")'. And if he were planning to reference items relative to the Statement named "Specifications" very often, he could affix a marker (e.g., named "s") to its front and would then only have to specify '(#s "Journal")'. 6g3

Or, suppose he were working in another file in a different directory on Office-5 and wanted to reference items relative to that same "far off" statement with special ease: in some temporary place in that file he could install a statement named "Ref" (for example) containing the textual link, "(Program-Documentation, Sequence-Doc, Specifications)". He could then cite the above reference with the link, '(Ref.1 "Journal")'. This path description is: go to the statement in this file named "Ref", take the first link that you find there (traveling across intervening directories and files and statements), and beginning in the statement on the other end of that link, scan forward to the string "Journal". 6g4

This is only a cursory treatment, but should illustrate well enough what is meant by "a rich and flexible addressing vocabulary." As with other high-performance features in AUGMENT, a beginner is not forced to become involved in the larger vocabulary in order to do useful work (with productivity on at least a par with some other, restricted-vocabulary system). But an AUGMENT worker interested in higher performance can steadily pick up more of the optional vocabulary and skills in a smooth, upward-compatible progression. 6h

## CONTROLLING THE VIEWS 7

A user of a book, or of most on-line text systems, is constrained to viewing the text as though he had a window through which he sees a fixed, formatted document. But as described below, our worker can view a section of text in many ways, depending upon his need of the moment. 7a

### MULTIPLE WINDOWS 7b

For whatever total screen area is available to the worker, his general performance will be improved significantly if he can flexibly allocate that area into arbitrary-sized windows whose contents can be independently controlled. AUGMENT has long provided this basic capability, along with the provision that material from any accessible file may be shown in any window, and also that screen-select copying or moving can be done across the different windows. 7b1

(Note: Cross-file editing can be done at any time, between any two legally accessible files. If one or the other file's material or destination is not being displayed in any of the windows, one may always opt to employ a textual address expression instead of a <Select> within any editing command.) 7b2

User-adjustable parameters are used to control the view presented on the display. Adjusting one's view parameters is a constantly used AUGMENT feature that has solidly proved its value. To facilitate their quick and flexible

use, the view-specification actions evolved into cryptic, single-character codes, called "viewspecs." The syntax of all Jump commands (used for traveling) includes the option of designating new viewspecs, and a special combination of mouse buttons enables quick, concurrent, keyset action to change the viewspecs for a given window. Here are a few of the frequently used view controls: 7b3

## WINDOW VIEWS 7c

***Structure Cutoff.*** Show only the statements that lie "below" this statement in the structure (i.e., this "branch"); or show only those following statements that are at this level or deeper; or show all of the following statements that will fit in this window. 7c1

***Level Clipping.*** For the designated structure cutoff, show only the statements down to a specified level. Lower-level statements are "clipped" from the view; the worker can thus view just a selected number of the upper levels of his document/file. 7c2

***Statement Truncation.*** For those statements brought into view (as selected by other view specifications), show only their first n lines. Truncation to one line is often used, along with level clipping, in order to get an effective overview. 7c3

***Inter-Statement Separation.*** For viewing ease -- blank lines can be optionally installed between statements. 7c4

(Note: The foregoing view controls are extremely helpful when studying and modifying a document's structural organization.) 7c5

***Statement Numbers and Names.*** Optionally, for a given window, show the Statement Number (or the SID) of each statement -- with an option for showing them at either the right or at the left margin. Independently, the showing of statement names may be turned on or off. 7c6

***Frozen Statements.*** A worker may select a number of statements, in random order, and designate them as "frozen." One of the view-specification options is to have the frozen statements appear at the top of the frame, with the rest of that window left for normal viewing and editing. The frozen statements may be edited, or even cross-edited between any other displayed (or addressable) statements. 7c7

***User-Specified Content Filters.*** A simple content-analysis language may be used in a "Set Content Pattern" command, which compiles a little content-checking program. One of the view-specification options will cause the system to display only those statements which satisfy both the structure and level conditions imposed by other viewspecs, and which also pass the content-analysis test applied by this program. Where desired, very sophisticated

content-analysis programs may be written, using a full-blown programming language, and placed on call for any user. 7c8

## USER-SPECIFIED SEQUENCE GENERATORS 7d

In the foregoing, a "view" is created by beginning at a designated location in a document (file) and selecting certain of the the "following" statements for display, according to the viewing parameters -- possibly suppressing statements that don't pass the test of a content-analysis program. This is essentially a "parameterized sequence generator," and provides very useful options for selectively viewing statements within a document; however, it works only by selectively discarding statements from a sequence provided in standard order. 7d1

Application programmers can provide alternate sequence-generator programs, which any user can invoke in a straightforward manner. In such a case, the apparent structure being presented to the user could be generated from a sequence of candidate statements according to any rules one may invent -- and the actual views could be further controlled by the above-described viewspecs for level clipping, truncation, content filtering, etc. 7d2

Perhaps the most commonly used, special sequence generator is one that provides an "Include" feature, where specially tagged links embedded in the text will cause their cited passages to be "included" in place of the Include-Link statements, as though they were part of this file. This provision enables arbitrary assemblage of text and formatting directives, from a wide collection of files, to represent a virtual, one-document, super file. For instance, the whole assemblage could be passed to the formatter, by means of a single user action, to generate a composite, photo-typeset document. 7d3

## TRAVELING THROUGH THE WORKING FILES 8

An important provision in AUGMENT enables an author to freely "travel around" in his on-line file space to reach a particular "view point" of his choice -- i.e., the position within a file from which the system develops the desired form of "view" according to the currently invoked view specifications. 8a

Traveling from one view point to another is accomplished by Jump commands, of which the simplest perhaps is a direct Jump to a statement designated by a screen selection. Then, for a worker grown used to employing address strings, a next form would be a Jump on an embedded link, or to a statement designated by a typed-in address string -- using any combination of the addressing elements and viewspecs described above. For example, the link "<4b:mI>" points to the Statement 4b, while invoking viewspecs "m" and "I" which cause the statements' SIDs to be displayed. The link "<Ref-1.l:i;LL>" points to the document referenced by the link in the statement named "Ref-1",

invoking viewspec "i" for user content filtering, and sets the filter to "LL" to show only those statements beginning with a lower-case letter. The applications are effectively endless. 8b

## MODIFYING THE DOCUMENT STRUCTURES 9

Given the array of capabilities described above, it is very simple also to provide for very flexible manipulation of the file structure. For operating on a small, basic set of structure-entity nouns, essentially the same basic verbs may be used as for text manipulation -- i.e. Insert, Delete, Move, Copy, Replace, and Transpose are quite sufficient for most cases. For instance, "Move Branch 2b (to follow) 3c" immediately moves Statement 2b and all of its substatements to follow Statement 3c -- and their statement numbers are automatically changed from 2b, 2b1, etc., to 3d, 3d1, etc. 9a

A few extra verbs are useful for structure manipulation. For instance, a "Break" command will break a given statement off at a designated point in its text string, and establish the rest of the text as a new, separate statement. And an "Append" command does the reverse -- i.e., it appends the text of one or more existing statements to the end of a designated statement. 9b

A major source of structure-modification capability derives from the associated "studying" capabilities. For example, if an author can view a file (document) with specifications that show him only one line each of just those statements in the top two levels, he gets an overview of the high-level organization that helps immensely to study his current structure or outline. 9c

Concurrent use of mouse and keyset also provide considerable gains in speed and flexibility for studying and modifying document structure. For example, if when studying the overview described in the previous paragraph, the author perceives that Statement 2b really belongs in Section 3, following Statement 3c, he can execute the necessary move command in a very quick, deft manner: 9d

Keyset hand strikes "m" and "b" (for Move Branch), while the mouse hand is positioning the cursor anywhere in the text line of Statement 2b. [Two chord strokes.] 9d1

The mouse hand depresses the <Select> button on the mouse while the cursor is on Statement 2b, then moves to Statement 3c and depresses it again, and then depresses it again to say, "OK, do it." [Three button pushes, synchronized with the mouse movement as it made two selections on easy, window-wide, whole-line targets.] 9d2

(Note: I just had myself timed for this above operation -- an unhurried 2.5 seconds.) 9e

In our view, interactive computer support offers an author a priceless opportunity to get away from the geometric bondage inflicted by pages, margins, and lines -- things which have very little if any bearing upon the content and organization of one's text. In terms of value to the authoring process, we differ sharply from those who advocate a "What you see is what you get" working mode during the development of a document's content and organization. For this kind of work, experienced users of the foregoing kind of flexible facility for addressing, viewing, and manipulating structured documents, would consider a "What you see ..." mode as a relative handicap. 9f

## SUPPORTING MULTI-PARTY COLLABORATION 10

The support that advanced technology can provide for close collaboration among knowledge workers is a very important and much under-rated possibility. For multiple-author activities, collaborative support is an important aspect of system capability. Some years ago, we introduced the following provisions into AUGMENT. (A more complete, overview treatment of these is given in <Ref-8>.) 10a

***Electronic Mail.*** Its primary attributes of speed, automatic distribution, and computer-to-computer directness are well recognized -- and are generally accepted now as important to the effectiveness of knowledge workers. AUGMENT Mail has features that are beyond what most electronic mail systems offer, and which provide unique benefit to the authorship process. 10b

AUGMENT's mail system allows one to "send" complete, structured documents as well as small messages. In an authorship environment, an important role for "electronic mail" is for the control and distribution of documents -- where small, throw-away messages are considered to be but a special class of document. An author should be able to bundle up any combination of text and graphics, in the forms that he has been using for studying and manipulating them -- and send the bundle to other workers. In AUGMENT, such a bundle is just like any other file structure, and can be studied and manipulated, incorporated into other files (documents), saved or deleted. 10b1

***Recorded Mail -- AUGMENT'S Journal System.*** When mailing a document, an AUGMENT worker may optionally specify that it be installed as a "recorded" item. In this case, before distributing the item, the system will make a permanent record if it, as a file in a specified Journal collection. And, just as though it had been published, this recorded Journal item cannot later be changed. The system assigns a straightforward accession identifier (a simple number), and any authorized worker is henceforth guaranteed access to that Journal item by specifying the name of the Journal-collection and the Journal-item number -- e.g., as specified in the link "<OAD,2237,>". 10c

A given journal may be set up to serve multiple hosts and is much like a special library. It has its collection of documents, and AUGMENT provides associated support processes for entry, cataloging, retrieval, and access. 10c1

Together with the linking capability described above, a Journal system provides an extremely effective form of "recorded dialog." Cross-reference links between a succession of Journal items produces an inter-linked network of collaborative contributions -- plans, outlines, document drafts, schedules, short comments, detailed critiques, reference material, etc. The on-line worker can follow these links very easily and, using multiple windows and flexible viewing options, can make very effective use of such records. 10c2

For instance, consider a detailed commentary directed toward a "preliminary design" document recorded in a given Journal collection. The author writing the commentary could view the design document in one window and his developing commentary document in another. He can easily establish links in his commentary to cite any passage in the design document -- e.g., a statement, a term in the statement, or a diagram. Then this author would submit his commentary into the Journal, perhaps specifying a list of colleagues for "distribution." Each listed user would automatically receive a mail item announcing this new Journal entry, giving subject, author, date, etc., and the all-important link to the new Journal file containing the commentary. Any such recipient can subsequently study both the commentary and its cited planning document in a similar, multi-window, link-assisted manner. 10c3

Furthermore, this second reader could develop and submit his own recorded commentary, which because of the citation power of AUGMENT links could be as short and to the point as: "Frankly, John, I think your comment in (DDD,xxx,aa) is a mistake! Didn't you notice the earlier assumption in (DDD,xxx,bb)? Maybe you should go back to Tom's earlier requirements document -- especially at (EEE,yy,cc)." (Here, "DDD" and "EEE" represent Journal names, "xxx", "yyy", and "zzz" represent Journal item numbers, and "aa", "bb", and "cc" represent addresses pointing to specific passages in those Journal files.) 10c4

In official parlance, "retrieval" is the finding out about the existence of a relevant piece of information, whereas "access" is the subsequent process of gaining possession of the information. For users of AUGMENT's Journal system, retrieval is immensely facilitated by the widespread use of citation links. When one can follow them as easily as can a practiced AUGMENT worker, these links provide extremely effective retrieval support. We have supplemented this with some simple, automatically generated catalog files, which made a rather nice balance. Access is provided by direct Jump on a reference link if the file is on line; if it isn't, AUGMENT asks the worker if she wants it retrieved, and a simple affirmative response automatically launches a request for the system operator to retrieve the file from its archive tape, after which the worker is notified of its availability via electronic mail. 10c5

A private document can be submitted into a Journal. In this case, only those workers listed at Journal-entry time can get access to the central copy. Such a private item would not be listed or indexed in the "public" catalogs. 10c6

We have used the Journal system very heavily since 1970 to support AUGMENT's development activity; many customers have employed it heavily since 1975. There are about 100,000 entries recorded in the original Journal now (I don't know about other, newer AUGMENT Journal collections). We found that as workers became at home in this environment, they were increasingly free about submitting their items to the "public." It became evident that the scientific tradition of active and open interchange has some solid relevance to the collaborative processes in our smaller, "colleague communities." Time and again a worker would come across others' dialog and be able to contribute some valuable information (sometimes a one-sentence comment with a critical citation link). Often the payoff went the other way: the new party found immediate value in an old piece of recorded dialog. 10c7

***Shared-Screen Teleconferencing.*** Consider a case where two people sit down to work together at a terminal, where they can both see the screen(s), and where either one can take over the controls. This is being done countless times every day throughout the country, in different combinations of expert-expert, expert-novice, novice-coach, etc. When talking together on their telephones, two or more distantly separated AUGMENT users can collaborate in a manner very similar to this. 10d

Suppose that two workers, Smith and Jones, want to set up and operate in a Shared-Screen Conferencing mode. Smith is in Princeton, working on host Office-4, and Jones is in San Francisco, working on host Office-12 -- and both of these host computers are connected to the same network. Assumedly they are in telephone contact when they decide to work in this shared-screen mode to collaborate on Smith's current job. 10d1

Jones will enter the command "Share (display with user) SMITH! On host OF12! Viewing (other display)!!" 10d2

Smith will enter the command "Share (display with user) JONES! On host OF4! Showing (this display)!!" 10d3

To give these commands, each person only entered the characters shown in upper case (entry case actually irrelevant), plus the digits, plus an "OK Key" action where each exclamation point is shown. 10d4

Whatever tool that Jones is currently using will continue responding to his controlling actions, as evidenced by various feedback and portrayal actions in the windows on his screen. Smith's screen image will clear, and be replaced with a replica of Jones' screen image -- multiple windows and all. For the duration of the shared-screen session, Smith's screen image will continue to replicate what is shown on Jones' screen. 10d5

There are provisions for passing control back and forth between workers. For instance, Jones can pass control to Smith so that Smith can show him some material or method of work. There are also provisions for the subsequent entry and departure of other conference participants. 10d6

## EMBEDDING THE GRAPHIC ILLUSTRATIONS 11

For complete support of document development, it is important to provide integrated means for developing, viewing, and manipulating graphical portrayals. These portrayals should be part of the working files from the very start, to be studied, passed about in mail, shared in Conferencing mode, edited, captioned, labelled, and moved about within the document structure. Furthermore, active, relevant citation links pointing to these graphical constructs would be installed in and followed from textual passages throughout the associated set of documents (including Mail and Journal documents). 11a

AUGMENT's architecture and file structure were designed for this end, and a good bit of the associated implementation is in place. 11b

A graphical data structure can be attached to any given file node, and there are basic capabilities for composing, studying, and modifying graphical diagrams. When formatting for a suitably equipped photo-typesetting device, there are formatting directives to designate the position and scale for placing these diagrams on a page. An AUGMENT file with integrated text and graphics can thus be mapped automatically onto a high-quality document whose pages contain both text and line drawings. 11c

Our goal here was for what we call an "illustrative graphics" capability -- basic to which is a command that, when directed toward any conventional "plotter" file, will translate it into a diagram attached to a designated node. In this way we can make use of graphic constructs developed within almost any applications system, most of which have provision for outputting "conventional" plotter files. 11d

The most important next step is to adapt a bit-mapped display as an AUGMENT workstation, so the integrated text and graphics can be viewed and manipulated on the same screen. Heretofore, to do graphic work, an author has had to attach a Tektronix 4014 storage-tube display to the special printer/graphic port of her AUGMENT workstation. This has made use of AUGMENT graphics slow and expensive enough to limit the number of user groups who have developed the integrated use of mixed text and graphics. 11e

## CONCLUSION 12

AUGMENT's unique provisions stemmed for the most part from the conceptual framework within which AUGMENT was developed. For instance, consider the pervasive and significant changes in the environment in which humans will be doing their knowledge work. Note that the habits, methods, conventions, intuitions, etc., that comprise the "ways" in which we think, work and collaborate, are for the most part products of many centuries of cultural evolution -- in a radically different environment. With a radically different environment, this constant process of cultural evolution can be expected to take some radical turns. 12a

The AUGMENT developmental framework assumed that many of these "ways" are candidates now for change in directions that heretofore would not have been beneficial. The AUGMENT system emerged as a first step in considering a few such changes, which perhaps can improve human capability for doing knowledge work because their new "ways" will enable us more effectively to harness the new tools toward more effective basic capability. (This is very different from trying to "automate" our old "ways" of doing things.) 12b

As an example, consider the "What You See Is What You Get" (WYSIWYG) syndrome. It is a highly touted feature for many vendors. It provides a definite advantage for the final process of converting a computer-held document to a nicely formatted hard copy. But what does it do for authorship? Well, in our framework, it has a negative impact. We were happy to abandon those constraints of lines and pages and other formatting geometry which did not contribute to matters of content and structure. We have chosen instead to provide the authorship process with structured files, flexible addressing, flexible window-size viewing, level and truncation viewspecs, etc. -- things that would be awkward or impossible to provide in a WYSIWYG environment. This provides the authorship phase with flexibility and power for studying and manipulating content and structure that we wouldn't consider trading off for WYSIWYG. Save it for the production phase. 12c

Here is another bit of culture that deserves re-examination. Consider the dictum, "Easy to learn, and natural to use." Or, "User friendly." The question is, for whom are you judging that things will be easy, or natural, or friendly? For designers of craft-work tool systems, very different perceptions of this issue are warranted between a system for the occasional, weekend do-it-yourself person and a system to be heavily used day after day by professionals. The AUGMENT User Interface System enables us easily to configure either kind of a tool collection. 12d

This paper describes part of what is provided to professional knowledge workers who do a significant amount of authorship work. We observe no more difficulty in their learning how to employ this relatively large collection of tools than one would expect for professional woodworkers in their learning about the relatively large collection of chisels and other tools of their trade. 12e

It is a basic part of our framework that, to augment human knowledge workers, attention must be given not only to tools, but to methods and skills as well. Because of space limitations, the scope of this paper was restricted to a summary of those tool provisions within AUGMENT that especially facilitate the authorship process. A full description of "How to use AUGMENT to ..." would definitely need to include methods of work that effectively harness these tool provisions, and the special kinds of skills that yield unique payoff in executing these methods. This is true for every tool system, of course, but it seems especially true in this case because many AUGMENT provisions do not fit into the general cultural background of our authorship process. 12f

Perhaps the best way for very brief summarization of what AUGMENT's users feel about its unique features is simply to say that those who leave its working environment really miss them. 12g

## REFERENCES 13

Ref-1: **"Toward Integrated, Evolutionary Office Automation Systems,"** Douglas C. Engelbart, *Proceedings of the Joint Engineering Management Conference*, Denver, CO, October 16-18 1978, pp. 63-68. (AUGMENT,71279,). 13a

Ref-2: **"The Augmented Knowledge Workshop,"** Douglas C. Engelbart, Richard W. Watson, and James C. Norton, *AFIPS Conference Proceedings,* Vol. 42, National Computer Conference, June 4-8, 1973, pp. 9-21. (AUGMENT,14724,). 13b

Ref-3: **"Document Production and Control Systems,"** Elizabeth K. Michael, Dirk H. van Nouhuys, Beverly R. Boli, Raphael Rom, and Ann C. Weinberg, *Phase One report of Document Production and Control Systems Design Study*, by the Augmentation Research Center, SRI International, for AF Rome Air Development Center, Contract F30602-76-C-003, March 1, 1977. (AUGMENT,37730,). 13c

Ref-4: **"A Model Document Production System,"** Beverly R. Boli, Harvey G. Lehtman, Elizabeth K. Michael, Raphael Rom, Dirk H. van Nouhuys, and Nina Zolotow, *Phase Two report of Document Production and Control Systems Design Study*, by the Augmentation Research Center, SRI International, for AF Rome Air Development Center, Contract F30602-76-C-003, July 30, 1977. (AUGMENT,29000,). 13d

Ref-5: **"Toward High-Performance Knowledge Workers,"** Douglas C. Engelbart, *OAC '82 Digest*, Proceedings of the AFIPS Office Automation Conference, San Francisco, CA, April 5-7 1982, pp. 279-290. (AUGMENT,81010,). 13e

Ref-6: **"User Interface Design Issues for a Large Interactive System,"** Richard W. Watson, *AFIPS Conference Proceedings*, Vol. 45, AFIPS Press, Montvale, NJ, 1976, pp. 357-364. (AUGMENT,27171,). 13f

Ref-7: **"Design Considerations for Knowledge Workshop Terminals,"** Douglas C. Engelbart, *AFIPS Conference Proceedings*, Vol. 42, National Computer Conference, June 4-8, 1973, pp. 221-227. (AUGMENT,14851,). 13g

Ref-8: **"Collaboration Support Provisions in AUGMENT,"** Douglas C. Engelbart, *OAC '84 Digest*, Proceedings of the 1984 AFIPS Office Automation Conference, Los Angeles, CA, February 20-22, pp. 51-58. (OAD,2221,). 13h

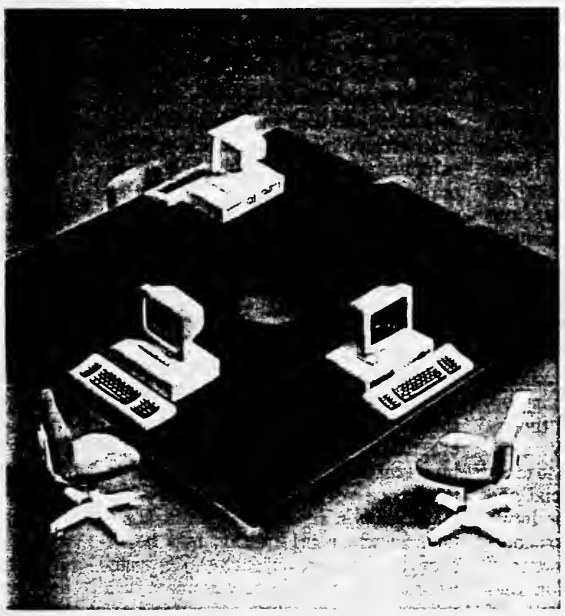
# Working Together

*The "human system" and the "tool system" are equally important in computer-supported cooperative work*

*Douglas Engelbart and Harvey Lehtman*

The emergence of the personal computer as a major presence in the 1970s and 1980s led to tremendous increases in personal productivity and creativity. It also caused setbacks in the development of tools aimed at increasing organizational effectiveness—tools developed on the older timesharing systems.

To some extent, the personal computer was a reaction to the overloaded and frustrating timesharing systems of the day. In emphasizing the power of the individual, the personal computer revolution turned its back on those tools that led to the empowering of both co-located and distributed work groups collaborating simultaneously and over time on common knowledge work.



The introduction of local- and wide-area networks into the personal computer environment and the development of mail systems are leading toward some of the directions explored on the earlier systems. However, some of the experiences of those earlier pioneering systems should be considered anew in evolving newer collaborative environments.

Computer Supported Cooperative Work (CSCW) deals with the study and development of systems that encourage organizational collaboration. Most groupware products fall under this classification. CSCW projects can be classified into three categories: tools for augmenting collaboration and problem solving within a group geographically co-located in real time (e.g., CoLab at Xerox Palo Alto Research Center); real-time tools for collaboration among people who are geographically distributed; and tools for asynchronous collaboration among teams distributed geographically.

In our work at the Augmentation Research Center (ARC) at the Stanford Research Institute (SRI) International beginning in the mid-1960s, we developed a system called NLS (On-Line System) and tools that supported these forms of collaboration. However, we placed the greatest emphasis on collaboration among people doing their work in an asynchronous, geographically distributed manner.

Our original goal at ARC was to "augment" individuals doing knowledge work. (See the text box "The NLS/Augment Architecture" on page 247.) In fact, some of the tools, techniques, and artifacts we developed then have become widely used in personal computer environments. These include full-screen windowed editing systems, mouse-controlled cursors, hypertextual linking of documents, and consistent user interactions across all aspects of a system. As timesharing systems and then wide-area networks (such

*continued*

ILLUSTRATION: ROBERT TINNEY © 1988

as the ARPANET) were introduced, the domain we attempted to augment widened to include groups collaborating in the same place, as well as over distances bridged by the networks and over time bridged by tools for creating a recorded dialogue among the collaborators.

One of the key strategies at ARC was the notion of bootstrapping: making use of available technology to create tools, techniques, and methodologies for knowledge workers in general, and the ARC group in particular, to use in further development of the tools. We served as the developers of the technologies, as well as the subjects for the analysis and evaluation of the augmentation system we had been developing. Many of the surface features of the system appeared in fancier dress as bit-mapped graphical hardware that became available first at Xerox, then later, much more widely, at Apple.

While it was exciting to see bits and pieces of the original NLS, now called the Augment system, appear commercially over the years, many elements of the system's conceptual core have only recently been recognized: outline editors (for easy manipulation of ideas); hypertextual linking capabilities fully integrated into the system; a system of recorded group dialogue that transcends most mail systems; user programmability and customizability of the system; and, most important, tools for augmenting not just individual knowledge workers but also teams of people both coresident and distributed over the world interacting through a networked environment.

We thought that success in creating tools for collaborative knowledge work was essential to the necessary evolution of work groups in increasingly knowledge-rich societies and to increasing organizational effectiveness. Until the recent growing interest in CSCW, most developers limited their analyses to technical issues and ignored the social and organizational implications of the introduction of their tools; such considerations were, however, key to our work.

There is growing recognition that some of the barriers to acceptance of fully integrated systems for augmenting groups of knowledge workers may be more significantly social, not solely technical. The availability of rapidly evolving new technologies implies the need for concomitant evolution in the ways in which work is done in local and geographically distributed groups.

ARC experienced this phenomenon continuously. The bootstrapping approach, so important to the continuing evolution of the system, caused us to constantly undercut our world. As soon as we became used to ways of doing things, we replaced platforms to which we were just becoming accustomed. We needed to learn new roles, change attitudes, and adopt different methods because of growth in the technological system we ourselves produced.

We brought in psychologists and social scientists to serve as observers and facilitators. They were as important to our team as the hardware and software developers. The resistance to change, which we soon realized was an essential part of introducing new technologies into established organizational settings, and the psychological and organizational tensions created by that resistance were apparent in ourselves. We were required to observe ourselves in order to create appropriate methodologies and procedures to go along with our evolving computer technologies.

*We brought in psychologists and sociologists to serve as observers.*

Our lab was concerned with *augmentation*, not automation. The choice of this term was significant. Aspects other than introducing new technological tools into the workspace (e.g., conventions, methods, and roles) are at least as important to the success of any CSCW system. The elegant tools available now and in the future—superlative graphics, artificial intelligence services, and so on—only make sense in an integrated workshop of tools in which information may be exchanged. The tools in such an integrated workshop need to be conceptually and procedurally consistent.

We expect that as tools are introduced and used, a co-evolution will occur between the tools and the people using them. Thus, WYSIWYG systems eased the acceptance of computer systems by nontechnically oriented users; however, these systems produce a map of what you would see on paper as opposed to a hyperdocument with structural links evolving over time. We are now seeing the increasing acceptance of other presentation metaphors (such as Apple's HyperCard and Owl International's Guide) incorporating some of the nonlinear linking capabilities that were present in Augment.

The architecture and character of Augment were directly oriented toward augmenting the capability of humans to deal with tough knowledge work and to process effectively the large volumes of information with which knowledge workers must deal. A subgoal was to support active collaboration among groups of workers. To gain experience with the issues and needs associated with this support, we developed and operated the Network Information Center (NIC) for the original ARPANET user and researcher community.

### Creating a Collaborative System

The following elements are necessary ingredients in a system designed to support collaboration in a community of knowledge workers. The sequence represents an explicit progression that begins with tested techniques whose "cultural shock" and financial investment are relatively low; it proceeds through paced, open-ended evolution with time, experience, and perceived payoff toward tools and techniques that involve a greater investment in both financial and social areas.

• *Collaborative dialogue.* Computer tools for the composition of messages and for their subsequent reviewing, cross-referencing, modification, transmission, storage, indexing, and full-text retrieval are a necessary part of a CSCW system. A "message" in such a system can be of any length. It can contain formalized citations pointing to specific passages in prior messages, so that a group of related messages becomes a network of recorded-dialogue contributions.

There should also be automatic message delivery; full cataloging and indexing; on-line accessibility both to message notification and to the full text of all messages; and open-ended storage of the dialogue records. These services enable a community of people who are distributed in space and time to maintain effective, recorded, collaborative dialogue in a manner that qualitatively differs from most ordinary electronic-mail systems.

With Augment, real-time remote dialogue (teleconferencing) was supported by a "shared screen" facility through which users could "link up" their displays; each party to the link sees a common display view. Any party to the link is able to point to or control or execute

*continued*

# The NLS/Augment Architecture

The On-Line System, or NLS, was designed to support members working in varied disciplines, including software engineers, managers, and social scientists. There were core tools used by all these knowledge workers, as well as specialized tools developed for particular requirements. All the tools shared the commonality of design principles that we thought essential to the success of what we termed a knowledge workshop. Early development began in 1963 and proceeded until 1976. (See photo A.)

The physical environment on which Augmentation Research Center (ARC) members (and collaborators across the country) worked evolved along with our system and externally available technologies. Back when the project started, display technologies were extremely primitive: Most people were still using punched cards and paper tape. Few computer users had direct access to a computer.

### A Revolutionary Console

In that context, the NLS terminals were especially revolutionary. The display consoles were equipped with typewriter-like keyboards, a five-finger keyset for one-handed character input, and a mouse, invented in our lab, for cursor control (see photo B).

The keyset was useful for most members of ARC, as commands were generally recognizable by single-character mnemonics, with appropriate feedback provided by the system. Most team members became proficient at one-hand text input, leaving the other hand available for cursor control by means of the mouse as they moved through the information space on their terminal screens.

Initially, screens were generated on small CRTs in our machine room and transmitted via closed-circuit television to the ARC workstations. Later on, as character-based displays became commercially available, we created external boxes to those terminals for attaching mice and keysets and controlling the cursor and screen updates in the manner required by our essentially nonlinear system devices, which were developed principally as "glass teletypewriters."

Those boxes, or line processors, were eventually made available to users over the ARPANET so they could experience the display-based version of NLS. However, because of the initially limited availability of displays, we also created a typewriter version of the system (TNLS), which had a complete mapping of the display NLS (DNLS) interface and permitted ready access to information across the country through the then more cost-effective typewriter terminals.

NLS was the core workshop software application system. It centered around the composition, modification, and study of structured textual material. Graphics were available in a primitive manner on the early terminals; the later line-processor-based systems made graphics available on additional, external graphics displays.

The type of bit-mapped graphics systems and hard-copy printers readily available today were not available to us at the time, although later evolutions of our file-system content architecture could accommodate graphical entities as data nodes. Moreover, there were important areas associated with the text domain that needed exploration.

### A Hierarchical Structure

The underlying NLS document architecture was hierarchically structured: the structure of a file was separated from its content. Originally, content nodes were strictly textual in nature; eventually, each structural node referred to a property list of content nodes of varying types, including other hierarchies (i.e., text, graphics, code, and so on).

The structure made for rapid navigation through the information space created by a file or collection of files. Its complexity was hidden from novice users (who didn't need to know about its implementation and, in fact, could ignore the hierarchy if they wished as they created linear documents in the NLS editor).

However, more sophisticated users

*continued*

**Photo A:** *A 1967 augmented meeting. This configuration is similar to more current sustems, such as Xerox PARC's CoLab.*

could address any point in any file throughout a network via a link—a syntactic address, which could be embedded anywhere in other files. These links were essential to the first implementation of the sort of system later called hypertext by Ted Nelson. (See the October BYTE.)

The basic node in an NLS file was a statement, most often used to represent a paragraph in text, a line of code in a program. The user could impose filters on the content or structure through tools either built into the system (view specifications) or installed through a user-programming facility. Thus, users could look at a particular number of lines of those statements at a particular level in a file. This facility was similar to those in so-called idea processors, such as Living Videotext's More. Associated with each statement was the date and time of its last edit as well as the identifier of the community member who created or edited it. Document filters over authors and time could also be installed.

Because of the collaborative nature of the development of NLS, there were tools and conventions for group authorship. Only one person could have write-access to a file at a time. Other team members could have read access to the file, minus the edits currently being made. A lock was placed on a file being written; if another team member accessed the file or attempted to write on it, that person would be told who had the file locked.

**A Variety of Tools**

NLS had tools for moving through the information space, using the mouse to select locations on the screen or the addressing capability (using the link syntax) to specify locations not directly accessible from the screen. You could jump to locations related to structural entities (successor, predecessor, and so forth), or you could jump via links by pointing to a textual link in a file or typing one in when prompted. Users could have up to eight windows on a screen with different files or different parts of the same files visible. Material could be copied across windows.

Programmers had access to a number of languages we created: Tree Meta, a compiler-compiler, was used to bootstrap us onto different machines (XDS 940, PDP-10, PDP-11, and DEC 20) and to create the other compilers and assemblers we used. L10 was a block-structured language with pattern-matching and string-construction facilities. The same pattern-matching syntax was used by less sophisticated users to generate filters in the core workshop. The Command Meta Language (CML) was used to create user interfaces that were independent of terminal type (display or typewriter) and individual user preferences. CML grammars were interpreted. Contextual entries into syntactic and semantic help systems were generated from the CML grammars. The Output Processor interpreted a comprehensive document-formatting language.

Programmers could look at procedures on the display and, encountering a reference to another procedure, jump to it. If it was not within the currently open file, the jump took place indirectly through a procedure catalog automatically generated by the automated program librarian.

The program librarian operated over system databases at night (or whenever it was invoked). If a code file had been modified, it would be automatically compiled; if all compilations took place without error (errors were recorded in other NLS files), a new system would be linked and created. The catalog was sorted alphabetically and, in addition to links to the files containing the procedures, included comments and calling sequences that were extracted from the procedure.

Programmers could view and modify procedures, compile them independently into their own address spaces, and automatically "replace" the existing versions of the procedures in the system to try out variations. Users could install (automatically when entering the system) alternative versions of standard system procedures. A symbolic debugger could be called up in a separate window, and breakpoints could be set by pointing at procedure names in the source-code file with the mouse.

We had tools for creating recorded dialogues with other users: Our Journal provided the usual message-passing facilities available on other timesharing and networked systems. However, we

**Photo B:** *A display NLS workstation with video overlay. Note the chord keyset input device used as a supplement to the keyboard. (The mouse may be seen in the video overlay on the screen.)*

could also submit larger documents or parts of them for permanent storage and retrieval or for the information and collaboration of other users. Shorter messages could be transmitted directly to a user's Initial File (the file seen on entering the system, similar to the desktop on current systems). Citations to larger documents would be delivered.

On seeing one of those citations, which included links to the document's location in the Journal, a user could jump to that document. The documents in the journal were permanent, read-only records of the dialogue within the community. Links to these documents were created, and evolving commentary on the design and implementation issues were always available. These facilities are similar to those currently advocated as "hypertext publishing systems."

NLS also had tools for interactive real-time collaboration. For example, users could link their terminals together and share screens; this made it possible for them to view the same material and collaboratively edit it.

As the ARPANET became available, we were among its first users. We found it necessary to tune the network to the then unique characteristics of our highly interactive system. It was also useful to separate the architecture of the system into a front end (which handled the user-interface interactions) and a back end (which handled the execution of commands).

The front end could operate on a separate machine and communicate with back-end resources through a network. Commonly used resources could be resident on the front-end machine; resources that were most usefully shared would reside on the back end.

We also created the Network Information Center (NIC) at the Stanford Research Institute to serve as an information resource for the emerging ARPANET. We used our tools to create the ARPANET Resource Directory, which was made available in both on-line and hard-copy form.

NLS included facilities for document development, production (including early computer phototypesetting facilities), and control. These facilities incorporated tools for successive refinement and editing by teams of writers, editors, and reviewers and were built on other parts of the core workshop, such as the editor, Journal, and programming tools.

any of the capabilities of the workshop. Such capabilities assume a high degree of responsiveness and bandwidth in the communication channel in order to support the high degree of interactivity in the system. (Our developments in this area required extensive tuning of the original ARPANET algorithms.)

• *Document development, production, and control.* This system capability includes tools for composing, studying, and modifying document drafts and for high-quality photocomposition. In addition to the page-layout tools that have become widely available, Augment offered tools for collaboration between several authors and editors in the process of evolving a final draft. These included tools for controlling changes, new version distribution, and automatic index generation for complex documents or sets of documents.

Page-layout programs such as PageMaker have entered widespread use in recent years. However, the tools for collaborative control of other aspects of a document's evolution are equally important. Augment permitted establishing superdocuments that were hypertextually linked combinations of the whole or parts of many pieces of information. This linking implies and reflects underlying meaning in ways that mere typesetting, which deals primarily with layout, cannot. While the typeset, WYSIWYG view should be available, it should not be the only way to view a document in its larger sense.

We also assume the need for tools to authenticate submissions and comments, provide administrative support to editors, offer sequential delivery and tracking for approval chains, and show automatic "ticklers" to those who do not respond to requests for comments, modifications, and approvals.

A backlinking facility within the recorded dialogue system is also necessary to handle superseding of old documents by new. Recent versions of the Augment Journal provide such a capability, permitting users to request current or older versions of an evolving document.

• *Research intelligence.* The tools within the Collaborative Dialogue Support System for cataloging and indexing internally generated items should also support managing *externally* generated items—bibliography, contact reports, clippings, notes, and so forth.

With centrally supplied (and hence uniformly available) services such as these, a community can maintain a dynamic and highly useful "intelligence" database to help it stay up-to-date on external happenings that affect it. Citations of external items from within the internally generated dialogue base, in the form of annotations, commentary, or supportive references, offer computer-sensible interlinking of the external information with the internal information and facilitate browsing, retrieval, searching, back-citation, and so on.

• *Community handbook development.* This includes extending this research service toward the coordinated handling of a very large and complex body of documentation and its associated external references. This material, when integrated into a monolithic whole, may be considered a "superdocument." Tools for the responsive development and evolution of such a superdocument by many (distributed) individuals within a discipline- or project-oriented community could lead to the maintenance of a "community handbook," a uniform, complete, consistent, up-to-date integration of the special knowledge representing the current status of the community.

The handbook would include principles, working hypotheses, practices, glossaries of special terms, standards, goals, goal status, supportive arguments, techniques, observations, how-to-do-it items, and so forth. An active community would be constantly involved in dialogue concerning the contents of its handbook. Constant updating would provide a "certified community position structure" about which the real evolutionary work would swarm; flexible tools for on-line navigation and view generation would be very important, as would the facility for generating hard-copy equivalents.

The "handbook cycle" includes the incorporation of ongoing dialogue and intelligence mediated by professional facilitation to create evolved versions of the community handbook.

• *Computer-based instruction.* We assume that the special training needs of a community of collaborating knowledge workers will be supported by computer-based instructional tools. These would make use of the other knowledge workshop services described, especially dynamic filtering of the community handbook.

A "shared screen" facility is useful for instruction so novices can get access to expert users or coaches in parts of the system for which other instructional tools are inadequate and for which local teachers are unavailable. Having an expert take you along for a ride is an extremely effective learning technique.

*continued*

• *Meetings and conferences.* At ARC, we made extensive use of augmentation tools in our local and distributed meetings. Projected display images, video overlays, and split-screen image superimposition were first used to great effect by Engelbart in the 1968 IFIP Fall Joint Computer Conference in San Francisco.

Dynamic control of the agenda and the collaborative creation of position papers are some typical uses of these services.

• *Community management and organization.* Conventional project-management operations can be augmented through the use of computer-based project-management tools with the enriching services of dialogue support, document development, and the handbook, which would include plans, commitments, schedules, and specifications.

• *Special knowledge work by individuals and teams.* The tools supporting a collaborating community should be available to the team members in their roles as individuals and members of other teams. A user-programming facility in Augment made it possible for individual users to customize parts of the system according to their needs and abilities. Some of these specialized extensions became part of the more widely available tools for the entire workshop community.

### A Formula for Success

As Augment evolved, we realized some assumptions that we think are applicable to any successful CSCW system:

• *Coordinated set of user-interface principles.* There should be a common set of principles over the many application areas. This does not mean that the user interface itself is necessarily the same across all domains. It does mean that a common underlying style of communication is present. While each domain within the core workshop area or specialized application system may have a vocabulary unique to its area, this vocabulary should be used within language and control structures common throughout the tool environment. Users learn new functions by increasing vocabularies, not by learning separate "foreign" languages. When in trouble, they will invoke help or tutorial functions in a standard way.

This point has become apparent in the Apple Macintosh environment. Users of different applications have a common method of interacting with each application. This makes it easier to learn new applications and to move between systems.

A single interface metaphor is neither required nor ideal. Interaction styles suitable for a particular application domain and user group may differ from those for other domains and users. Apple's HyperCard provides an example of an environment that offers interaction metaphors different from the original Apple Desktop with minimal confusion to users.

• *Grades of user proficiency.* Users who are not experienced in using the system are part of the community; they will want to be able to get at least a few straightforward things done with a minimum of learning. Even an expert user in certain domains of the collaborative workshop environment will be a novice in less frequently used domains. Attention to novice-oriented "easy to use" features is required.

However, users should be rewarded for their increasing proficiency with a rich tool environment that offers advanced vocabularies and the opportunity

*continued*

for individual customization in every specialized domain.

• *Ease of communication among, and addition of, workshop domains.* We think that there will be many different parts of an augmented-knowledge workshop, each with its own tools. You should never be bound to isolated areas of the workshop. It should be possible to move and communicate information between domains easily. It should also be possible to install new tools as needed.

• *User-programming capability.* Users must be able, with various levels of ease, to add or interface new tools and extend the language to meet their needs. They should be able to do this in a variety of programming languages in which they may have training, or in the basic user-level language of the workshop itself (e.g., through a macro facility.)

*We can't ignore the social implications of our technical progress.*

• *People-support services.* The computer-based tools will be insufficient by themselves. The CSCW technologies will create opportunities and needs for highly specialized professional services, such as database design and administration, training, cataloging, and retrieval formulation.

• *Recognition of standards for information interchange and ranges of hardware.* We should not have to assume the presence of a particular type of machine in a user's work environment. It should be possible to exchange information and get a reasonable representation of the information shared across system environments.

• *Careful development of methodologies.* The elements involved in augmenting communities of knowledge workers include the development of both "tool systems" and "human systems" (the set of skills, methods, languages, customs, procedures, training, and organization structures needed for effective use of tools). New technologies, even those such as CSCW that aim at improving group interaction, contribute directly only to the tool system. The cultural evolution that led to the current state of the human system took place with a very primitive tool system.

As much care and attention needs to be paid to developing the procedures and methodologies associated with the people-support services and the organizational and societal effects of introducing new technologies as is spent on developing the technologies themselves.

• *Co-evolution of roles and organizational structures and technologies.* The widespread availability of successful CSCW services will create the need for new organizational structures and roles. These structures and roles need to co-evolve with the technologies. For example, we found there was a need for what we called knowledge-workshop architects who served as "change agents" in introducing new technologies into their organizations.

To take advantage of the radical, emerging tool-system inventions associated with CSCW, it is inevitable that the evolution of the human system will begin to accelerate. The optimum design for either a tool system or a human system is dependent on the match it must make with the other. The high degree of mutual dependence implies that a balanced co-evolution of both is necessary. The bind we are in is that our society encourages and rewards progress in the technological and material sense and often ignores the human and social implications of that progress. ■

FURTHER READING

Ambron, Sueann, and Kristina Hooper. *Interactive Multimedia.* Redmond, WA: Microsoft Press, 1988.

Greif, Irene. *Computer-Supported Cooperative Work: A Book of Readings.* San Mateo, CA: Morgan Kaufman Publishers, 1988.

Johansen, Robert. *Groupware: Computer Support for Business Teams.* New York: Free Press, 1988.

Grimsdale, R. L., and F.F. Kuo, eds. *Computer Communication Networks.* Leydon: Noordhof, 1975.

Goldberg, Adele, ed. *A History of Personal Workstations.* New York: ACM Press, 1988.

Rheingold, Howard. *Tools for Thought.* New York: Simon and Schuster, 1987.

*Douglas Engelbart, a senior scientist at McDonnell Douglas, recently created the Bootstrap Institute to further CSCW research. Harvey Lehtman is manager of the New Media Group at Apple Computer. They can be reached on BIX c/o "editors."*

Reprinted from *GroupWare '92, the* proceedings of the GroupWare '92 Conference , San Jose, CA, Aug 3-5 1992. Morgan Kaufmann Publishers, Inc. pp.77-100. (AUGMENT,132810,)

# TOWARD HIGH-PERFORMANCE ORGANIZATIONS: A STRATEGIC ROLE FOR GROUPWARE

**Douglas C. Engelbart**
Bootstrap Institute
6505 Kaiser Drive
Fremont, CA 94555
Document #: (AUGMENT,132810,)

## ABSTRACT

Achieving tomorrow's high-performance organizations will involve massive changes throughout their capability infrastructures. The complexity of implementing these changes will be daunting, and deserves a strategic approach. Groupware will support important, special new knowledge capabilities in these infrastructures, and also can play a key role in an evolutionary strategy.

## 1 INTRODUCTION

### 1.1 Shared Visions and the "Groupware Community"

Groupware to me, personally, is a strategic means to an important end: creating truly high-performance human organizations. My pursuit began in the '50s, aiming to make our organizations and institutions better able to handle complexity and urgency. By 1962 I had evolved a basic conceptual framework for pursuing that goal (Ref-1 and Ref-2). I have essentially lived and worked within that framework ever since, steadily evolving and enriching it via many relevant experiences.

It is becoming relatively common of late, in the increasing flow of literature about organizational improvement, to highlight the need for the members of an organization to have a shared vision of where and how the organization is moving, in its marketplace and in its internal evolution. I assume that the same principle should be applicable to a looser organizational unit, in this case, to the community consisting of organizations and researchers interested in the overlapping domains of organizational improvement and "groupware," and including the information-system marketplace whose business is providing products and services to end-user organizations.

From my experience, the nature of this shared vision will be the single most important factor in how directly and how well the digital-technology marketplace will indeed support significantly higher organizational capability — which I assume is our basic objective in the evolution of groupware.

My own vision about pursuing high-performance organizations has matured over the years into a quite comprehensive, multi-faceted, strategic framework. It may seem a bit radical in nature, but my continuing hope is that it will be merged into such a shared community vision.

The full purpose of our Bootstrap Institute is to promote constructive dialog with critical stakeholders in the community about this "bootstrap strategy," to facilitate its trial adoption, and to further the strategy's own "continuous improvement."

In this paper I summarize the key elements of this strategic framework and highlight the role that would be played by the "groupware community." In Ref-3 is an explicit historical treatment that provides a good deal of background on framework development up to 1986. Also, Ref-4 gives a relatively balanced description of our associated groupware and application developments with an underlying framework treatment.

### 1.2 Capability Infrastructure and its Augmentation System

Any high-level capability needed by an organization rests atop a broad and deep *capability infrastructure*, comprised of many layers of composite capabilities, each depending upon the integration of lower-level capabilities. At the lower levels lie two categories of capabilities: Human-Based and Tool-Based. The functional capabilities of groupware fit into the latter category, along with a wide variety of facilities, artifacts, and other tools.

In pursuit of higher organizational performance, this infrastructure is the obvious focus of attention. Then it is a matter of establishing system and goal perspectives to determine how much of this infrastructure to include as serious candidates for change, and how radical a change to contemplate. I arrived at a singularly global perspective from the following considerations.

Figure 1 shows the result of a great deal of thought about how over the centuries our cultures have evolved rich systems of things that, when humans are conditioned and trained to employ them, will *augment* their basic, genetically endowed capabilities so that they, and their organizations, can exercise capabilities of much higher nature than would otherwise be possible. For lack of a ready-made term, I named this our *Augmentation System*, and found it valuable to partition it into the two parts as shown — a Human System and a Tool System. I have developed many things from this model that have proved useful and valid over the years — including essentially everything I've developed in the groupware arena (tools, concepts, strategies).

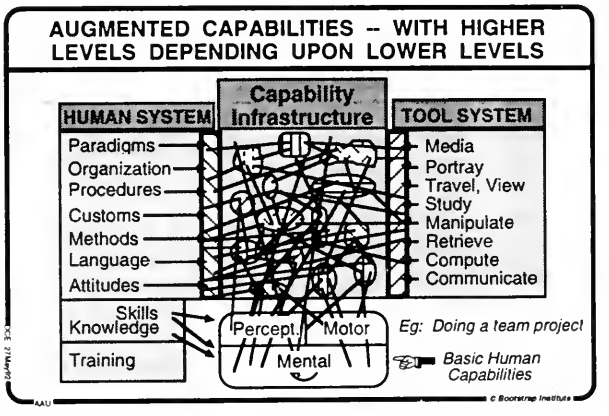



Figure 1

A bit of thinking about this model brought me the realization that we are far short of being able to do a one-pass re-design of any major portion of this capability infrastructure — if only because of their pervasive, underlying dependence upon human processes.

And as we pursue significant capability improvement, we need to appreciate that we will be trying to affect the evolution of a very large and complex system that has a life and evolutionary dynamic of its own. Concurrent evolution of many parts of the system will be going on anyway (as it has for centuries). We will have to go along with that situation, and pursue our improvement objectives via facilitation and guidance of these evolutionary processes. Therefore, we should become especially oriented to pursuing improvement as a multi-element, *co-evolution* process. In particular, we need to give explicit attention to the co-evolution of the Tool System and the Human System.

And, along with these foregoing perceptions, another factor popped into the scene to create a very significant effect on my emergent framework.

### 1.3 The Relevant Implications of Radical Scale Change

Some years earlier, I had studied the issues and prospects associated with extreme miniaturization of functional devices, towards assessing the likelihood of digital equipment becoming extremely small, fast and cheap. I was personally motivated because I would have to be relatively confident of very significant progress in that regard in order to commit a career towards facilitating widespread computer augmentation.

I learned enough to convince myself that, with the expected high industrial and military demand toward digital technology, the achievable limits on micro scalability were far beyond what would be enough to warrant my particular pursuits. And in the process, looking into references dealing with dimensional scale in living things, I became aware of a very important general principle: if the scale is changed for critical parameters within a complex system, the effects will at first appear as quantitative changes in general appearance, but after a certain point, further scale change in these parameters will yield ever-more striking *qualitative* changes in the system.

For example: The appropriate design for a five-foot creature is not that much different from that for a six-foot creature. But the design for either of these would be totally inappropriate for a one-inch creature, or for a thirty-foot creature. A mosquito as big as a human couldn't stand, fly or breathe. A human the size of a mosquito would be badly equipped for basic mobility, and for instance would not be able to drink from a puddle without struggling to break the surface tension, and then if his face were wetted, would very likely get pulled under and be unable to escape drowning.

The lesson: Expect surprising qualitative changes in structural assemblage and functional performance when a complex system adapts effectively to drastic changes in critical parameters.

I could only assume that the same is very likely to be true for the complex Augmentation System that supports an organization's capability infrastructure. Here, the radical change in the scale of Tool System capability — in speed, function, capacity, presentation quality, transmission, etc. of emergent digital technology — greatly transcends any other perturbation in system parameters that our organizations have ever needed to adapt to in so short a time as a few decades.

Much more could be said about the scaling issue that is relevant to the general theme of organizational change. Sufficient here to say that these thoughts drove me definitely to

view as global and massive both the opportunity and the challenge that we humans were facing with respect to increasing the performance level of the organizations and institutions upon which mankind's continuing existence depends.

### 1.4 The Underlying Importance of Paradigms

In the ensuing thirty years since the model of Figure 1 first evolved, I have become ever more convinced that human organizations can be transformed into much higher levels of capability. These digital technologies, which we have barely learned to harness, represent a totally new type of nervous system around which there can evolve new, higher forms of social organisms.

In the face of mounting evidence that our organizations and institutions can not cope adequately with the increasing complexity and urgency of our society's problems, it seems highly motivating to explore every avenue that offers reasonable probability of improving their capability to cope.

Those were my thoughts thirty years ago; they seem even more germane today. The technologies have been demonstrated, and our organizations are aligning toward internal improvement. What seems still to be lacking is an appropriate general perception that:

(a) huge changes are likely, and really significant improvements are possible;

(b) surprising qualitative changes may be involved in acquiring higher performance; and

(c) there might actually be an effective, pragmatic strategy for pursuing those improvements.

In developing a basic, scalable strategy, the above issues of perception are important enough to warrant being explicitly factored into it. In other words, the strategy should provide for the need of significant shifts in our perception of our likely and possible futures.

Perceptions, shared visions, paradigms — their evolution is *critical*, yet they receive little or no direct developmental attention. The slow, un-shepherded paradigm drifting of the past isn't an adequate process for times when deeper global changes are occurring than ever-before accommodated by such massive social bodies. And the rates of such change are more likely to increase than to diminish.

I interject such thoughts here because I actually believe that what can be produced by the groupware community can make a very large difference (in a proper strategic framework) to our capability for coping with large, complex problems. The ability to acquire this new capability is heavily dependent upon evolving an appropriate paradigm, which result in itself represents the type of complex challenge that our institutions need to become more capable of handling.

This leads to an assumption that an important factor to hope for, in an early stage of the future paradigms possessed by key players in this transformation of our organizations, is the perception of importance and a can-do attitude about consciously cultivating appropriate evolutionary trends and change rates in our future paradigms. Shifting our paradigm about paradigms.

What role will *you* play?

## 2 IMPROVING THE IMPROVEMENT PROCESS

The next step in developing an explicit strategic framework was generated from the conceptual content of Figure 1 by asking what sort of investment principles would make sense. I hoped to solicit R&D money and wondered how we might get the best return on those funds in facing this very large, unstructured problem. I also was prepared to invest essentially the rest of my professional career: how should I invest that time to get best net progress? And what basic guidelines should be adopted for launching (bare handed, so to speak) such a program?

The only serious approach that I could imagine, towards really significant improvement, would be a long-term, pragmatically guided, whole-system evolution. I was addressing a very complex system, and the challenge would be further complicated by the fact that the subject organizations would have to keep functioning at better than survival level while undergoing large, systemic changes.

So the image depicted in Figure 2 emerged from realizing that the capability of an organization to improve itself would have to become much more prominent and effective. It then seemed natural to consider a strategy wherein the earliest improvement efforts might be concentrated upon improving this capability (i.e., to improve the organization's improvement capability).

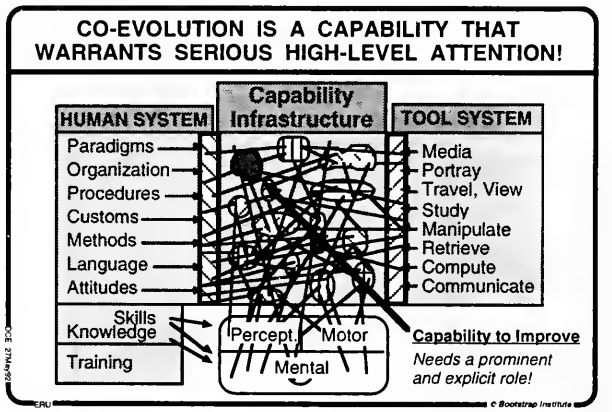



Figure 2

## 3 THE ABC MODEL OF ORGANIZATIONAL IMPROVEMENT

In doing some further thinking about improvement activities and the capabilities that support them, I found it useful to extract from Figure 2 a simpler abstraction dealing with organizational improvement, as in Figure 3. Here we separate the two types of activities, *A* and *B*, and show that the capability for each type of work is supported by its respective Augmentation System (comprised of Human and Tool systems).

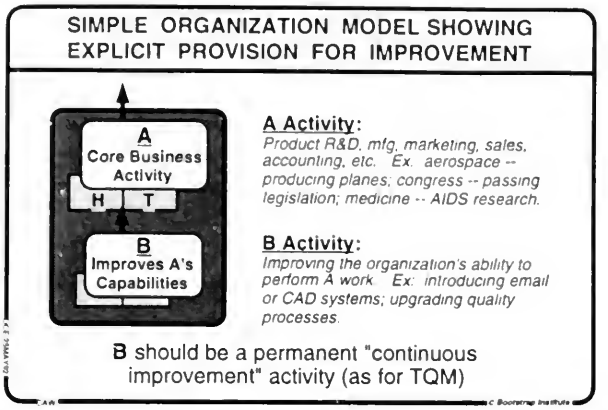



Figure 3

Given this model, we can now consider the prospects of improving the organization's improvement capability, as discussed earlier in Figure 2, as *improving the capability of the B Activity*. And for such a critical pursuit to be effective requires yet another explicit organizational activity, depicted in Figure 4 as the organization's C Activity. Executive efforts to assess and improve B-Activity funding, staffing, and high-level approach would qualify as a C Activity. C Activities would also include introducing new knowledge and skills into the B Activity, providing better means for participatory interaction with its A-Activity clients, or improving how pilot operations are managed.

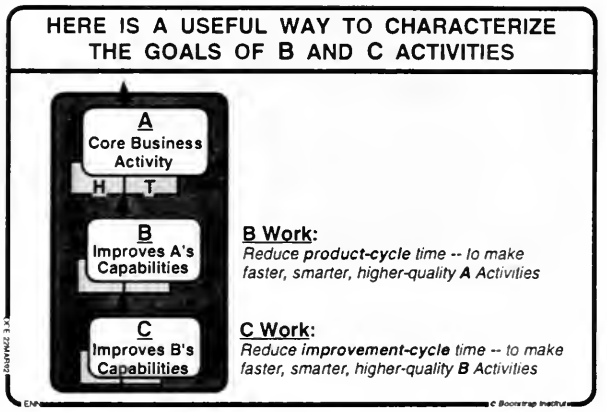



Figure 4

## 4 LOOKING FOR A MULTI-PAYOFF CAPABILITY CLUSTER

In considering the infrastructure elements that support this higher-level, self-improvement B Capability, I realized that many of its important subordinate capabilities are also actively employed by many of the higher-level A Capabilities that are important to the basic operations of the organization. For example, identifying needs and opportunities, designing and deploying solutions, and integrating lessons learned. This led to the following rhetorical question:

> *Is there a set of basic capabilities whose improvement would significantly enhance both the higher-level operational A Capabilities and this self-improvement B Capability?*

The answer was a clear "Yes!" A core set of knowledge-related capabilities rapidly emerged as the prime candidate.

An investment that boosts the A Capability provides a one-shot boost. An investment that boosts the B Capability boosts the subsequent rate by which the A Capability increases. And an investment that boosts the C Capability boosts the rate at which the rate of improvement can increase. (To be slightly mathematical, investing in B and C boosts respectively the first and second derivative of the improvement curve — single and double compounding, if you wish.)

We are assuming here that selected products of the two capability-improvement activities (B and C) can be utilized not only to boost the capabilities of their client activities, but can also to a significant extent be harnessed within their own activities to boost their subsequent capability. This is depicted in Figure 5 by the "feedback" paths.

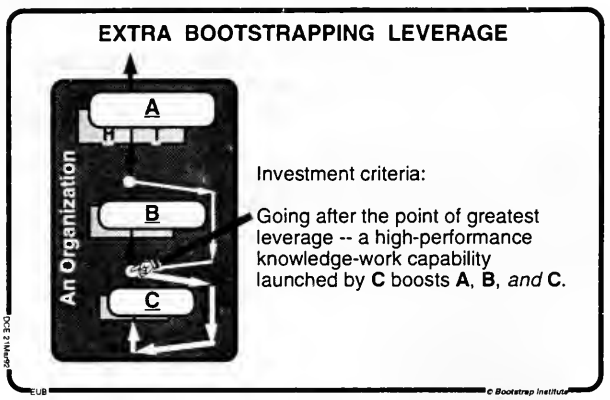



Figure 5

This was where the term *bootstrapping* became welded into my continuing professional framework. It turns out that there are many choices that we will face where balanced consideration of the bootstrapping possibilities can make a difference. I place much confidence in the potential payoff from thoughtful application of the principles that have evolved from such thinking.

## 5 THE CODIAK PROCESS CLUSTER: BEST STRATEGIC APPLICATION CANDIDATE

Over the years I have tried various ways to label and characterize the above-mentioned key knowledge capabilities. For lack of an established term, I have settled on an acronym that embraces the main concepts of this cluster of high-leverage capabilities — *CODIAK*:

*The concurrent development, integration and application of knowledge.*

As complexity and urgency increase, the need for highly effective CODIAK capabilities will become increasingly urgent. Increased pressure for reduced product cycle time, and for more and more work to be done concurrently, is forcing unprecedented coordination across project functions and organizational boundaries. Yet most organizations do not have a comprehensive picture of what knowledge work is, and of which aspects would be most profitable to improve.

The CODIAK capability is not only the basic machinery that propels our organizations, it also provides the key capabilities for their steering, navigating and self repair. And the body

of applicable knowledge developed represents a critically valuable *asset*. The CODIAK capability is crucial in most A Activities across the organization, whether in strategic planning, marketing, R&D, production, customer support, or operations. It is also crucial in the B and C Activities, whether identifying needs and opportunities, designing and deploying solutions, or incorporating lessons learned — which of course is also used in key A-Activity work. As such, the CODIAK capability should be considered a core business competency in the organization's capability infrastructure, and is an ideal candidate for early improvement to achieve the extra bootstrapping leverage discussed above in Figure 5.

For best exposure to full CODIAK issues, it helps to consider heavy knowledge-intensive activities such as a large, complex project. Figure 6 represents the high-level core of such a CODIAK process. In the center is a basic organizational unit, representing the interactive knowledge domains of a single individual, or of individuals or groups within a project team, department, functional unit, division, task force, committee, whole organization, community, or association (any of which might be inter- or intra- organizational).

Each organizational unit is continuously analyzing, digesting, integrating, collaborating, developing, applying, and re-using its knowledge, much of which is ingested from its external environment (which could be outside of, or within, the same organization).

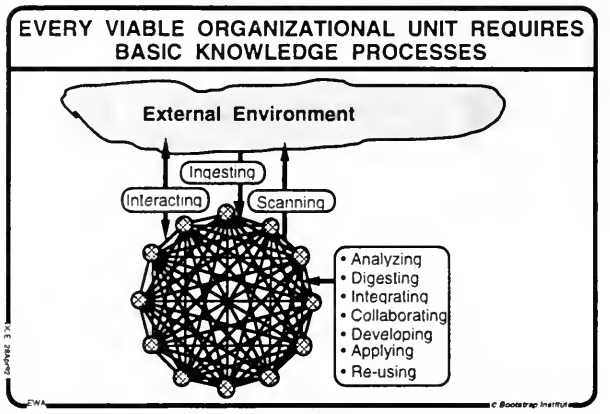



Figure 6 *CODIAK: The COncurrent Development, Integration, & Application of Knowledge*

A result of this continuous knowledge process is a dynamically evolving knowledge base as shown in Figure 7 below, consisting of three primary knowledge domains: intelligence, dialog records, and knowledge products (in this example, the design and support documents for a complex product).

***Intelligence Collection:*** An alert project group, whether classified as an A, B, or C Activity, always keeps a watchful eye on its external environment, actively surveying, ingesting, and interacting with it. The resulting *intelligence* is integrated with other project knowledge on an ongoing basis to identify problems, needs, and opportunities which might require attention or action.

***Dialog Records:*** Responding effectively to needs and opportunities involves a high degree of coordination and *dialog* within and across project groups. This dialog, along with resulting decisions, is integrated with other project knowledge on a continuing basis.

***Knowledge Product:*** The resulting plans provide a comprehensive picture of the project at hand, including proposals, specifications, descriptions, work breakdown structures, mile-

stones, time lines, staffing, facility requirements, budgets, and so on. These documents, which are iteratively and collaboratively developed, represent the *knowledge products* of the project team, and constitute both the current project status and a roadmap for implementation and deployment. The CODIAK process is rarely a one-shot effort. Lessons learned, as well as intelligence and dialog, must be constantly analyzed, digested, and integrated into the knowledge products throughout the life cycle of the project.

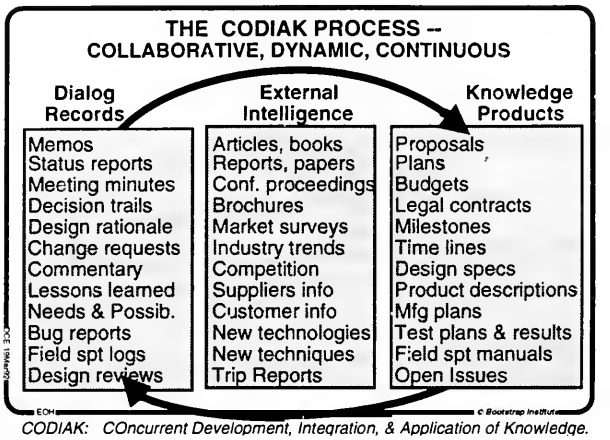



Figure 7 *CODIAK: COncurrent Development, Integration, & Application of Knowledge.*

With minor adjustments in the boxed lists in Figure 7, this basic generic CODIAK model seems to apply equally well to academic scholarship, heavy industry, government, medical research, social institutions, consumer product businesses, consulting firms, trade associations, small non-profits, and so on.

We need to note here that basic CODIAK processes have practically forever been a part of society's activity. Whether the knowledge components are carried in peoples' heads, marked on clay tablets, or held in computers, the basic CODIAK process has always been important.

What is new is a focus toward harnessing technology to achieve truly high-performance CODIAK capability. As we concurrently evolve our human-system elements and the emergent groupware technology, we will see the content and dynamics represented in Figure 7 undergo very significant changes.

More and more intelligence and dialog records will end up usefully recorded and integrated; participants will steadily develop skills and adopt practices that increase the utility they derive from the increased content, while at the same time making their contributions more complete and valuable.

Generally, I expect people to be surprised by how much value will be derived from the use of these future tools, by the ways the value is derived, and by how "natural and easy to use" the practices and tools will seem after they have become well established (even though they may initially be viewed as unnatural and hard to learn).

Inevitably, the groupware tools which support the CODIAK processes within and across our organizations will need to be fully integrated and fully interoperable. Consider the larger

organization depicted in Figure 8 in which our representative complex project may be embedded (for example, in the Engineering Department of a manufacturing organization).

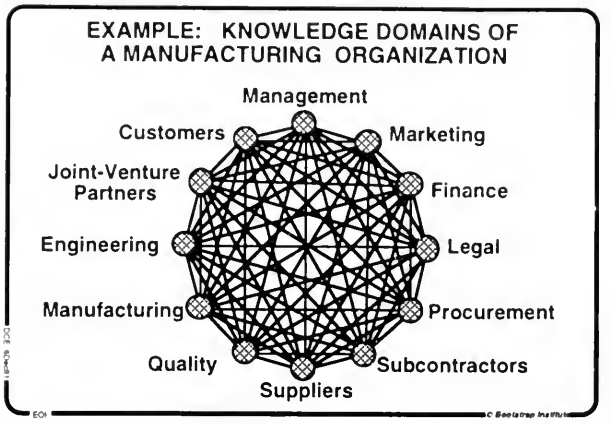



Figure 8 *Enterprise integration: Interoperability within & across knowledge domains*

Each of the enterprise's functional units studded around the circle represents an activity domain that houses at least one CODIAK process. Then, because of their mutual involvement with the operations of the whole enterprise, the CODIAK processes within each of these enterprise sub-domains would with strong likelihood benefit from being interoperable with those of the other sub-domains.

As operations between enterprises steadily become more closely knit, the interaction processes with customers, subcontractors and suppliers also want to become increasingly effective — and therefore the issue of knowledge-domain interoperability becomes ever more global.

As developed in the sections that follow, our framework assumes that all of the knowledge media and operations indicated in Figure 7 will one day be embedded within an Open Hyperdocument System (OHS). Every participant will work through the windows of his or her workstation into his or her group's "knowledge workshop."

With this in mind, consider the way in which the project group's CODIAK domain, with all of its internal concurrent activity, will be operating within the larger enterprise group depicted in Figure 8.

And consider that the whole enterprise, acting as a coherent organizational unit, must also have a workable CODIAK capability and possess its own evolving, applicable CODIAK knowledge base.

Here an important appreciation may be gained for the "concurrency" part of the CODIAK definition. CODIAK was introduced above with the sense that all of the development, integration and application activities within a given organizational unit were going on concurrently. *This establishes a very important requirement for the groupware support.*

In Figure 9 we get the sense of the multi-level "nesting" of concurrent CODIAK processes within the larger enterprise. Each of the multiply-nested organizational units needs its own coherent CODIAK process and knowledge base; and each unit is running its CODIAK processes concurrently, not only with all of its sibling and cousin units -- but also with larger units in which it is embedded, and with smaller units that are part of its own makeup.

Furthermore, there are many valuable organizational units that cut across the organizational structure — such as a corporate-wide task force — and each of these units also needs a coherent CODIAK process and knowledge base. And beyond that, significant working relationships will be going on with external organizational units, such as trade associations, professional societies, consultants, contractors, suppliers, special alliance partners, customers, regulatory agencies, and standards groups. Each such "external" unit needs to have a coherent CODIAK knowledge domain; all such domains will have some knowledge elements and evolutionary dynamics that are mutual with those of many other units in the enterprise's total CODIAK environment.

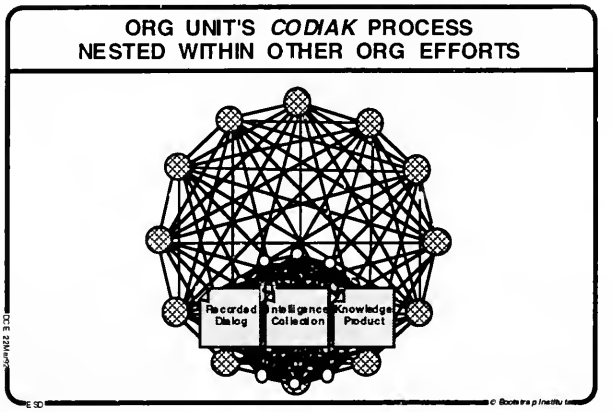



Figure 9

So, consider the much extended sense of concurrency and inter-dependency arising from the above picture: the CODIAK processes of all of the inter-dependent organizational units within the larger enterprise are going on concurrently; and further, among these concurrently active processes there is a great deal of mutual involvement with parts of the whole knowledge base.

It is easy to realize that significant parts of what the smaller group works with, as being in its "external environment" intelligence collection, will actually be shared-access knowledge from other domains within the enterprise — from other's dialog, from their external intelligence, or from their finished or evolving knowledge products.

Then the entire enterprise has a collective CODIAK domain, with knowledge elements that to some extent will be actually in a "whole-enterprise" domain, but where much of what lies in the collective enterprise domain is an active part of the CODIAK domains of subordinate organizational units within the enterprise.

And further, consider that as the availability of highly effective online CODIAK support becomes widespread, suppliers, contractors and customers will engage in a non-trivial degree of CODIAK-domain sharing with the enterprise. One needs only a brief glance at the supplier network of Figure 10 to realize the magnitude of critical, interoperable CODIAK processes and shared CODIAK knowledge domains that will prevail when (or if) suitable groupware becomes widely available.

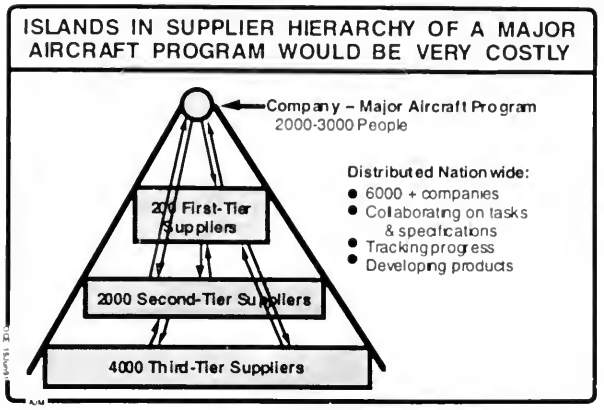



Figure 10

This is representative of the scale of global challenge that I think faces the groupware marketplace.

The foregoing dictates some very significant requirements for any groupware system that attempts to support the CODIAK processes of our future, high-performance organizations. Immediately apparent is the need for very flexible, wide-area sharing of pieces of the knowledge base. What has only recently begun to be generally apparent is the associated need for a new way of thinking about the nature of the knowledge packages we have called "documents." This above requirement for flexibly arranged sharing of essentially arbitrary knowledge chunks provides a very strong argument for documents becoming built from modular-concept nodes with arbitrary inter-node linking — *hypertext*.

So, how (and when) will the marketplace learn enough and be cooperative enough to develop truly effective OHS standards? The prospects for achieving truly high levels of performance in larger organizations and institutions pretty much await that day.

> This question is a significant part of what an effective bootstrapping strategy needs to address.

## 6 OPEN HYPERDOCUMENT SYSTEM (OHS): FOR GENERIC CODIAK SUPPORT

My early assumption, amply borne out by subsequent experience, is that the basic supporting technology for future high-performance knowledge work will be an integrated system based upon multi-media hyperdocuments.

Furthermore, there will be critical issues of interoperability within and between our organizations and their knowledge domains. The ever-greater value derived from online, interactive work within a hyperdocument environment will require a significantly higher degree of standardization in document architecture and usage conventions than heretofore contemplated.

It is inevitable that this service be provided by an "open system" of hyperdocuments and associated network and server architectures. The basic arguments for this Open Hyperdocument System (OHS) are presented in Ref-5; and the hyperdocument system features described below are assumed by me to be strong candidates for requirements for the eventual OHS whose evolution will be so critical to the productivity of industries and nations.

Following is a brief general description of the system design that has evolved from the conceptual orientation described in this paper, through the experience of many years and trial events. Please note that the term "system" is very important here.

***Shared Files/Documents*** — the most fundamental requirement. Generalized file sharing is to be available across the entire global domain in which any online collaborative working relationship is established (e.g., world-wide).

***Mixed-Object Documents*** — to provide for an arbitrary mix of text, diagrams, equations, tables, raster-scan images (single frames or live video), spread sheets, recorded sound, etc. — all bundled within a common "envelope" to be stored, transmitted, read (played) and printed as a coherent entity called a "document."

***Explicitly Structured Documents*** — where the objects comprising a document are arranged in an explicit hierarchical structure, and compound-object substructures may be explicitly addressed for access or to manipulate the structural relationships.

***Global, Human-Understandable, Object Addresses*** — in principle, every object that someone might validly want/need to cite should have an unambiguous address, capable of being portrayed in a manner as to be human readable and interpretable. (E.g., not acceptable to be unable to link to an object within a "frame" or "card.")

***View Control of Objects' Form, Sequence and Content*** — where a structured, mixed-object document may be displayed in a window according to a flexible choice of viewing options — especially by selective level clipping (outline for viewing), but also by filtering on content, by truncation or some algorithmic view that provides a more useful portrayal of structure and/or object content (including new sequences or groupings of objects that actually reside in other documents). Editing on structure or object content directly from such special views would be allowed whenever appropriate.

***The Basic "Hyper" Characteristics*** — where embedded objects called *links* can point to any arbitrary object within the document, or within another document in a specified domain of documents — and the link can be actuated by a user or an automatic process to "go see what is at the other end," or "bring the other-end object to this location," or "execute the process identified at the other end." (These executable processes may control peripheral devices such as CD ROM, video-disk players, etc.)

***Hyperdocument "Back-Link" Capability*** — when reading a hyperdocument online, a worker can utilize information about links from other objects within this or other hyperdocuments that point to this hyperdocument — or to designated objects or passages of interest in this hyperdocument.

***Link Addresses That Are Readable and Interpretable by Humans*** — one of the "viewing options" for displaying/printing a link object should provide a human-readable description of the "address path" leading to the cited object; AND, the human must be able to read the path description, interpret it, and follow it (find the destination "by hand" so to speak).

***Personal Signature Encryption*** — where a user can affix his personal signature to a document, or a specified segment within the document, using a private signature key. Users can verify that the signature is authentic and that no bit of the signed document or document segment has been altered since it was signed. Signed document segments can be copied or moved in full without interfering with later signature verification.

***Hard-Copy Print Options to Show Addresses of Objects and Address Specification of Links*** — so that, besides online workers being able to follow a link-citation path (manually, or via an

automatic link jump), people working with associated hard copy can read and interpret the link-citation, and follow the indicated path to the cited object in the designated hard-copy document.

> Also, suppose that a hard-copy worker wants to have a link to a given object established in the online file. By visual inspection of the hard copy, he should be able to determine a valid address path to that object and for instance hand-write an appropriate link specification for later online entry, or dictate it over a phone to a colleague.

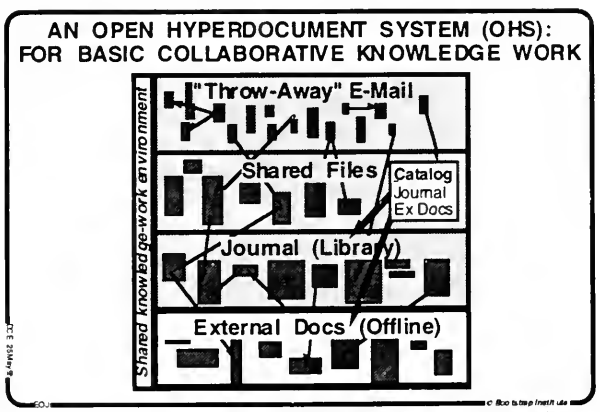
***Hyperdocument Mail*** — where an integrated, general-purpose mail service enables a hyperdocument of any size to be mailed. Any embedded links are also faithfully transmitted — and any recipient can then follow those links to their designated targets that may be in other mail items, in common-access files, or in "library" items.

***The Hyperdocument "Journal System"*** — an integrated library-like system where a hyperdocument message or document can be submitted using a submittal form (technically an email message form), and an automated "clerk" assigns a catalog number, stores the item, notifies recipients with a link for easy retrieval, notifies of supercessions, catalogs it for future searching, and manages document collections. Access is guaranteed when referenced by its catalog number, or "jumped to" with an appropriate link. Links within newly submitted hyperdocuments can cite any passages within any of the prior documents, and the back-link service lets the online reader of a document detect and "go examine" any passage of a subsequent document that has a link citing that passage.

***Access Control*** — Hyperdocuments in personal, group, and library files can have access restrictions down to the object level.

***External Document Control (XDoc)*** — (Not exactly a "hyperdocument" issue, but an important system issue here.) Documents not integrated into the above online and interactive environment (e.g. hard-copy documents and other records otherwise external to the OHS) can very effectively be managed by employing the same "catalog system" as for hyperdocument libraries — with back-link service to indicate citations to these "offline" records from hyperdocument (and other) data bases. OHS users can find out what is being said about these "XDoc" records in the hyperdocument world.

The overview portrayal in Figure 11 shows the working relationships between the major system elements described above. Note the shared catalog service that supports use of the Journal and External Document services.




"Hyperdoc" provides flexible linkages to any object in any multi-media file;
"Open" provides vendor-independent access within and across work groups.

Figure 11

Details of features and designs for well-developed prototypes of some of the above may be found in Ref-6, Ref-7 and Ref-8.

## 7 FOUR GENERAL GROUPWARE ARCHITECTURAL REQUIREMENTS

Besides the aforementioned Hyperdocument Mail and Hyperdocument Library features that depend upon special larger-scale architectural features, there are at least four other important tool-system capabilities that are very important to wide-area groupware services such as being considered here:

***Global and Individual Vocabulary Control*** — somewhat new in the history of computer services are issues regarding the evolution and use of a common "workshop vocabulary" among all the users of the forthcoming "global knowledge workshop." Common data dictionaries have been at issue, of course, but for a much more limited range of users, and for a more limited and stable vocabulary than we will face in the exploding groupware world.

Our own architectural approach (see Ref-6, Ref-9 and Ref-10) has been to introduce into every user-interface environment a common Command-Language Interpreter (CLI) module that derives the user's available operations (verbs) as applied to the available classes of objects (nouns) from a grammar file (individualized if desired with respect to the size and nature of the verbs and nouns utilized from the common vocabulary). The CLI interprets user actions, based upon the contents of the currently attached grammar file, and executes appropriate actions via remote procedure calls to a common application program interface of the "open system environment."

Each of us knowledge workers will become involved in an ever richer online environment, collaborating more and more closely within an ever more global "knowledge workshop," with multi-organizational users of widely divergent skills and application orientations who are using hardware and software from a wide mix of vendors.

Without some global architectural capability such as suggested above, I can't see a practical way to support and control the evolving global "workshop vocabulary" in a manner necessary for effectively integrating wide-area groupware services.

***Multiplicity of Look-and-Feel Interface Choices*** — Based upon the same Command-Language Interpreter (CLI) architecture as above, a "look-and-feel interface" software module would be located between the CLI and the window system. Providing optional modules for selected look-and-feel interface characteristics would serve an important practical as well as evolutionary need.

There would be a basic constraint necessary here. When working interactively, no matter what particular look-and-feel style is being used, a user has a particular mental model in mind for the significance of every menu item, icon, typed command, or "hot, command-key combination" employed.

The necessary constraint needed here is that the resulting action, via the interface module that is being employed for this user, must be produced through the underlying execution of processes provided by the Command Language Interpreter module as derived from use of common-vocabulary terms. And the users should learn about their tools and materials, and do their discussing with others about their work, using the underlying common-vocabulary terms no matter what form of user interface they employ.

Besides relaxing the troublesome need to make people conform to a standard look and feel, this approach has a very positive potential outcome. So far, the evolution of popular graphical user interfaces has been heavily affected by the "easy to use" dictum. This has served well to facilitate wide acceptance, but it is quite unlikely that the road to truly high performance can effectively be traveled by people who are stuck with vehicular controls designed to be easy to use by a past generation.

As important classes of users develop larger and larger workshop vocabularies, and exercise greater process skill in employing them, they will undoubtedly begin to benefit from significant changes in look and feel. The above approach will provide open opportunity for that important aspect of our evolution toward truly high performance.

***Shared-Window Teleconferencing*** — where remote distributed workers can each execute a related support service that provides the "viewing" workers with a complete dynamic image of the "showing" worker's window(s). Used in conjunction with a phone call (or conference call), the parties can work as if they are sitting side-by-side, to review, draft, or modify a document, provide coaching or consulting, support meetings, and so on. Control of the application program (residing in the "showing" worker's environment) can be passed around freely among the participants. Generic provision of this service is discussed in Ref-6.

***Inter-Linkage Between Hyperdocuments and Other Data Systems*** — for instance, a CAD system's data base can have links from annotations/comments associated with a design object that point to relevant specifications, requirements, arguments, etc. of relevance in a hyperdocument data base — and the back-link service would show hyperdocument readers which passages were cited from the CAD data base (or specified parts thereof).

Similarly, links in the hyperdocuments may point to objects within the CAD bases. And, during later study of some object within the CAD model, the back-link service could inform the CAD worker as to which hyperdocument passages cited that object.

## 8 THE CODIAK PROCESS SUPPORTED BY AN OHS

With the above tool capabilities, together with well-developed methods and other human-system elements as discussed in section 1.2, the organization's capability infrastructure could support the following types of online CODIAK scenarios.

Note that the following online interactions are designed to work even if the users are in different organizational units, in different organizations, using different application packages on different workstations (assuming access to the data is not barred by the stringent privacy features, naturally). The real test of an OHS is when you can click on a link you received via email from someone in a different organization, jumping directly to the passages cited, and then comfortably maneuver through the "foreign" knowledge domain, possibly jumping up a level with an outline view to see the context of the given passage, following other links you find there, and so on, *without having to fumble through unfamiliar processes.*

***Intelligence Collection:*** Now an alert project group, whether classified as an A, B, or C Activity, can keep a much enhanced watchful eye on its external environment, actively surveying, ingesting, and interacting with it mostly online. Much of the external intelligence is now available in hyperdocument, multimedia form, having been captured in an OHS Journal facility. When I send you an email to let you know about an upcoming conference, I can cite the sessions I think you'd be interested in, and you can click on the enclosed citation links to quickly access the cited passages (taking advantage of hypertext links and object addressability). When I do a search through the Journal catalogs to research

a question for the proposal I am writing, I can see who has cited the material and what they had to say about it. If the material is offline (i.e. in XDoc), I can quickly discover where it is stored and how to obtain a copy, probably requesting it via email. If the material is online, I can access it instantly, usually starting with a top-level outline view of the document's titles (taking advantage of the OHS document structure and custom viewing features), possibly setting a simple filter to narrow the field, then quickly zooming in on the specific information I require. I can quickly build an annotated index to the intelligence documents, or objects within those documents, that I want to keep track of. I can share with you a macro I wrote to trap certain incoming intelligence items and reformat them in a certain way, and you could fire this up in your own environment to work off your pet keywords (taking advantage of the common-vocabulary architectural feature). All the intelligence collected is easily integrated with other project knowledge.

***Dialog Records:*** Responding effectively to needs and opportunities involves a high degree of coordination and *dialog* within and across project groups. In an OHS environment, most of the dialog will be conducted online via the Journal. Email would be used mostly for "throw-away" communiqués, such as meeting reminders. All memos, status reports, meeting minutes, design change requests, field support logs, bug reports, and so on, would be submitted to the Journal for distribution. Asynchronous online conferencing would be supported by the Journal, with each entry tagged and cataloged for easy future reference. Document exchange would be a matter of submitting the document to the Journal with a comment such as "Here's the latest version — please note especially the changes in Section G, differences are listed in File Y" including links to that section and that file for easy access. The reviewers would click on the links, and proceed to review the document. To make a comment, the reviewer would click on the object in question, and enter the comment, such as "Replace with 'Xyz'," or "Watch out for inconsistencies with Para G4!" with a link to the passage in G4. The author then gets back the indexed comments, and has many options for quickly reviewing and integrating them into the document. Such dialog support will obviate the need for many same-time meetings.

Same-time meetings, when needed, would be greatly enhanced by an OHS. The dialog motivating the meeting would already be in the Journal. Agenda items would be solicited, and the agenda distributed via the Journal. At the meeting, the agenda and real-time group notes can be projected on a large screen, as well as displayed on each participant's monitor (using the "shared screen" feature), and any participant can point to the displayed material (e.g. using a mouse). Controls can be passed to any participant to scribble, type, or draw on this virtual chalkboard. Any presentation materials and supporting documents can be instantly retrieved from the knowledge base for presentation. All resulting meeting documents, along with references to supporting documents cited, would subsequently be submitted to the Journal for immediate access by all authorized users.

In addition, tools will soon become generally available for flexibly contributing, integrating, and interlinking digitized speech into the OHS knowledge base. Early tools would be available for speaker recognition, for special-word recognition, and even for basic transcription to text — and for installing and following links between modules as small as a word embedded in a long speech string. This will greatly enhance the development, integration, and application of dialog records. More elegant tools will follow, and as human conventions and methods evolve to make effective use of the technology, the quantity and completeness of recorded dialog will become much more significant.

***Knowledge Product:*** Throughout the life cycle of the project, the online OHS knowledge product will provide a truly comprehensive picture of the project at hand. Intermediate project states, including supporting intelligence and dialog trails, can be bundled as document collections in the Journal for document version management. All knowledge

products will be developed, integrated, and applied within an OHS, with concurrent contributions from many diverse and widely distributed users. These users can also work as if sitting side by side, reviewing a design, marking up a document, finalizing the changes, etc. (using the shared screen feature). Finding what you need among the thousands of project documents will be a simple matter of clicking on a link (provided by the Journal catalogs, or by your project's indices), and zooming in and out of the detail, or by having someone else "take you there" (using the shared screen feature). Accountability is absolute— Journal submittals are guaranteed to be authentic, and each object can be tagged by the system with the date and time of the last write, plus the user who made the change. Documents can be signed with verifiable signatures.

Everyone is but one quick "link hop" away from any piece of knowledge representation anywhere in the whole knowledge collection. Smart retrieval tools can rapidly comb part or all of the collection to provide lists of "hit links" with rated relevance probabilities.

Conventions for structuring, categorizing, labeling and linking within their common knowledge domain will be well established and supportive of a high degree of mobility and navigational flexibility to experienced participants — much as residents get to know their way effectively around their city if they get much practice at it.

As a group adapts its ways of working to take better advantage of a tool system such as projected here, the classes of knowledge objects will grow, as will the functions available to operate upon them—and that growth will be paralleled by the concurrent evolution of an ever richer repertoire of the humans' "workshop knowledge, vocabulary, methodology and skills."

There is tremendous potential here, and many methods, procedures, conventions, organizational roles to be developed in close association with the tools. And, if the OHS is to be *open*, there is much deep exploration to be done into different application domains, such as Computer-Supported Cooperative Work (CSCW), organizational learning, Total Quality Management (TQM), Enterprise Integration (EI), program management, Computer-Aided Software Engineering (CASE), Computer-Aided Engineering (CAE), Concurrent Engineering (CE), organizational memory, online document delivery and CALS, and so on. This will require many advanced pilots, as will be discussed further on.

## 9 RECAP: THE FRAMEWORK TO THIS POINT

To this point in the paper, we have outlined steps in the development of a strategy to provide a high-leverage approach toward creating truly high-performance organizations.

We considered the concept of the organization's *capability infrastructure* upon which any of the organization's effectiveness must depend.

Further, what enables humans to exercise this infrastructure of capabilities is an *Augmentation System*, which is what provides the humans with all capabilities beyond their genetically endowed basic mental, motor and perceptual capabilities. It was useful to divide the Augmentation System into two sub-systems, the Human System and the Tool System. "Organic style co-evolution" among the elements of our Augmentation System has been the process by which it evolved to its current state.

New technologies are introducing an unprecedented scale of improvement in the Tool System part of the Augmentation System. This promises that subsequent co-evolution of our Augmentation Systems will likely produce radical qualitative changes in the form and functional effectiveness of our capability infrastructures, and hence of our organizations.

Very large and challenging problems are envisioned in pursuing potential benefits of such changes, towards truly high-performance organizations. A strategy is sought to provide an effective approach.

It would be profitable to consider early focus on improving the organizational improvement process so that further improvements can be done more effectively.

To help with this analysis, the *ABC* categorization of improvement-process was established. And the thesis was developed that the *CODIAK* set of knowledge capabilities — the concurrent development, integration, and application of knowledge — is important to all three types of activities. Therefore, if CODIAK improvement was concentrated upon early, the result could improve the first and second derivatives of the return on future improvement investments.

An Open Hyperdocument System (*OHS*) would be a key "Tool System" development towards improving general and widespread CODIAK capabilities within and between organizations. And creating a truly effective OHS would in itself be an extremely challenging and global problem for our groupware marketplace.

So, high-performance organizations: great opportunities, interesting concepts, tough challenges. What next regarding strategy?

## 10 C COMMUNITY: HIGH-PAYOFF BOOTSTRAPPING OPPORTUNITY

Returning to the basic ABC Model in Figure 4, we can make a few useful observations toward a next step in strategy development. This model will be useful even if the Bootstrapping approach is not followed; it is valuable to become explicit about differentiating responsibilities, functions and budgets between the two levels of improvement activity (B and C).

If explicit C roles are designated and assumed, basic issues will soon arise for which the C-Activity leaders find it valuable to compare experiences and basic approaches with their counterparts in other organizations. For instance, what budgeting guidelines and targets make sense for these improvement activities? How much can it help the B Activity to document the way things are done now? What role should pilot applications play? How large an improvement increment, for how big a group, does it make sense to try for a pilot? How much "instrumentation" of a pilot group — before, during, and after transition — to measure the value of the effort? These are all relevant to making the B Activity more effective.

So let us consider formalizing and extending the above type of cooperation among improvement activities, especially the C Activities. In the mid-60s I began to think about the nature and value of communities of common interest formed among different improvement activities. This led me very early to build explicit planning into the bootstrap strategy for forming improvement communities.

In Ref-11 (1972), I presented the concept of a "community knowledge workshop" -- outlining the tools we had developed for supporting it (including many of the hyperdocument system capabilities outline above), and described the three basic CODIAK sub-domains: recorded dialog, intelligence collection, and what I then called the "handbook" (or knowledge products).

After the ABC Model emerged in the framework, this evolved into a special emphasis on an important launching phase, for forming one or more special bootstrapping *C Communities* as shown in Figure 12.

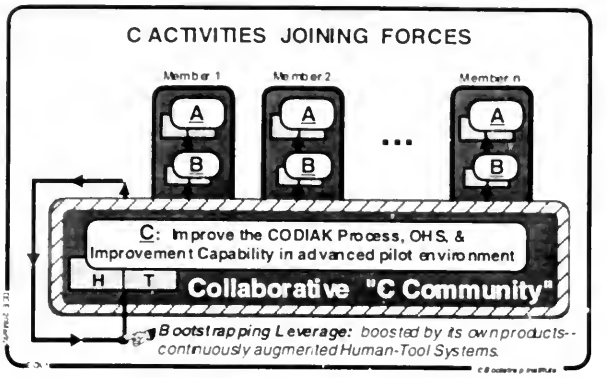



Figure 12

The value of such a cooperative activity can be very high — we'll unveil some of that later. First, there are some other questions that naturally arise which need to be addressed. An early and common pair of comments are: "I can't imagine sharing things with my competitors, there is so much about what we do that is proprietary;" and, "If they aren't in the same business, I don't see what useful things there would be that we could share."

About proprietary matters: The A Activity of each organization may be very competitive, with considerable proprietary content. The B Activity of each would tend to be less so — having quite a bit that is basic and generic. The C Activity of each would be much less involved in proprietary issues, and much more in basic, generic matters. So even competitors could consider cooperating, "out of their back doors" — "while competing like hell out of our front doors," as a trend that seems to be appearing among companies heavily into Total Quality Management and pursuit of the Malcolm Baldridge Award.

About being in very different business: Again, their B Activities will be much less different, and their C Activities surprisingly alike in important basic and generic issues.

Now, consider how a C Community could operate if it had the basic hyperdocument tools described above. For several decades, my colleagues and I have had such a system available, so all of our scenarios began there, using that system and calling it our "OHS, Model 1" — or "OHS-1."

And how would an ideal bootstrapping C Community operate? Its earliest focus would be on augmenting its own CODIAK capability. Using OHS-1 to do its work; making an important part of its work at first be to establish requirements, specifications and a procurement approach for getting a set of rapidly evolving prototype hyperdocument systems (e.g. OHS-2, -3, etc.), to provide ever better support for serious pilot applications among the C Community participants.

The Community's basic knowledge products could be viewed as dynamic electronic handbooks on "how to be better at your improvement tasks," with two customer groups: its B-Activity customers; and the C Community itself. Pooling resources from the member organizations enables a more advanced and rapidly evolving prototype CODIAK environment, which serves two very important purposes:

1. It provides for the Community getting better and better at its basic "C Activity;"

2. It provides advanced experience for its rotating staff of participants from the member organizations. They thus develop real understanding about the real issues

involved in boosting CODIAK capability — this understanding being absorbed by "living out there in a real, hard-working CODIAK frontier."

Note that it would be much more expensive for each member organization to provide equivalent experience by operating its own advanced pilot. Also the amount of substantive knowledge product developed this way would be very much more expensive if developed privately.

An important feature: once the Community stabilizes with effective groupware tools, methods and operating skills, the participants from the respective member organizations can do most of their work from their home-organization sites. This provides for maintaining the organizational bonding which is very important in effective C and B activities.

This home-site residency also facilitates the all-important "technology transfer" from the C Community into its customer B Activities. And, while considering the issue of "technology transfer," note that a strong feature of an augmented CODIAK process is the two-way transfer of knowledge. Developing dialog with the B clients via joint use of the hyperdocument system not only facilitates directly this two-way knowledge transfer, but provides critically important experience for the B people in the close witnessing of how advanced CODIAK processes work.

To characterize the value of facilitating this two-way transfer, consider Figure 13, which highlights the basic importance of improved CODIAK processes in the organization's improvement activity. The "1, 2, 3" points all are basic to the CODIAK process. As augmented CODIAK capabilities make their way up from C to B and into A, the over-all improvement process can't help but improve. And also, note that when the A Activity for this organization, as well as those for its customers, become based on interoperable CODIAK processes, the dynamics of the whole business will begin to sparkle.

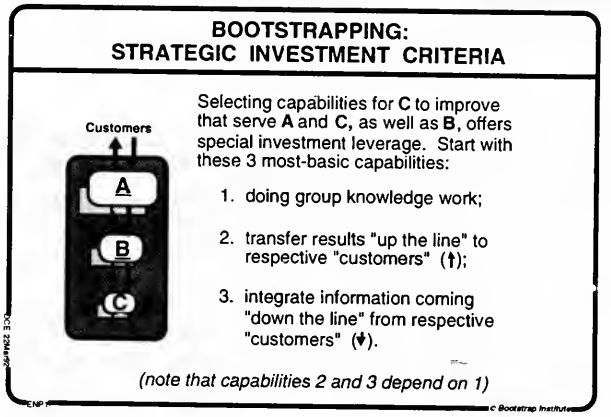



Figure 13

Now consider Figure 14, and note that the indicated types of knowledge flow are basic to the CODIAK processes, and that augmenting those processes for the C Community directly boosts one of its core capabilities. Conversely, Figure 15 emphasizes the previous basic point of the naturalness for enhanced CODIAK to improve this outflow, and highlights again the basic bootstrapping value that is obtained from early focus on these CODIAK processes.

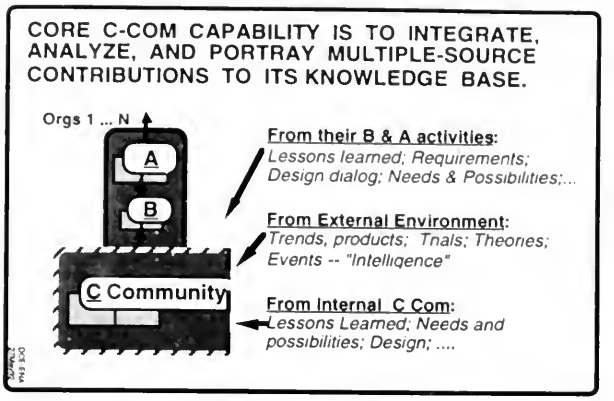



Figure 14

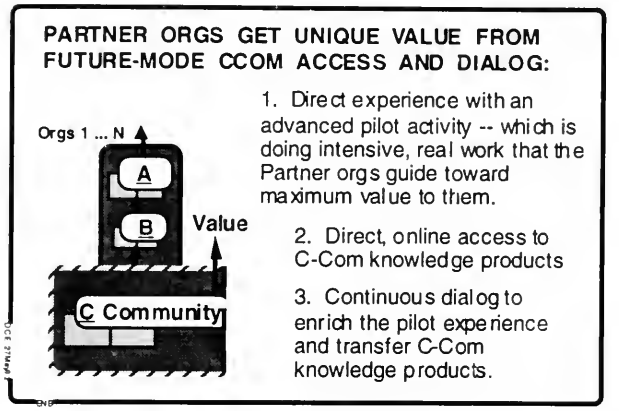



Figure 15

In the organizational improvement domain, there are several immediately apparent large and explicit issues for which a lone organization would need to consider a multi-party alliance. An immediate such issue, from the bootstrapping point of view, is to procure appropriate groupware systems that can support advanced pilot applications. Other large-sized issues have to do with "exploration and outpost settlements."

Relative to the options opening to our organizations for transforming into new states, there is a very large, unexplored, multi-dimensioned frontier out there. Both its dimensionality and its outer boundaries are expanding faster and faster. To really learn about that frontier, in order to decide where we would want to "settle our organizations," we must somehow do a great deal of basic exploration work. We also need to establish a significant number of outpost settlements in promising places so as to find out ahead of time what it would be like to really live and work there. (Translate "outposts" into "advanced pilot groups.")

Yet we are launching very few exploratory expeditions and developing very few significant outposts.

From the viewpoint that I have acquired, there is a great need for such explorations and trial settlements. Much of my motivation for advocating such as C Communities, bootstrapping, CODIAK and OHS pursuits, etc., is to find a strategy for exploring and settling that territory. It is almost like a military strategy: "first we get a firm settlement here in

CODIAK territory; then with that as a base, we encircle the OHS and C territories; when we get those under reasonable control, we will be in a most advantageous posture to pour through the rest of the B and C Improvement Territories to get the whole area under control and ..."

As the C Community and its working relationship with its "B customer" matures, there can be integrated into the substance of their joint efforts an ever larger sphere of involvement with the whole set of issues of organizational improvement.

Potential customers for augmented CODIAK capabilities can be seen everywhere in today's global society: e.g., all of the "Grand Challenges" earmarked in the U.S. for special support. Essentially every professional society will eventually operate this way; as will legislative bodies and government agencies, and university research programs.

In short, our solutions to every other challenging problem that is critical to our society will become significantly facilitated by high-performance CODIAK capabilities. Provides a stimulating challenge for the groupware community, doesn't it?

In closing, I would like to re-emphasize the comments in Section 1.4 about paradigms. I am convinced that cultivating the appropriate paradigm about how to view and approach the future will in the pursuit of high-performance organizations be the single most critical success factor of all.

*[Note: The Bootstrap Institute has developed basic plans for several scales of C-Community launching — a medium-sized consortium approach on the one hand, and a more conservative, organic evolution approach on the other hand. Interested inquiries are invited.]*

## REFERENCES

Ref-1: Engelbart, D.C. 1962. *Augmenting Human Intellect: A Conceptual Framework,* Summary Report, Stanford Research Institute, on Contract AF 49(63-8)-1024, October, 134 pp.

Ref-2: Engelbart, D.C. 1963. A Conceptual Framework for the Augmentation of Man's Intellect. *Vistas in Information Handling*, Howerton and Weeks (eds), Washington, D.C.: Spartan Books, pp. 1-29. Republished in Greif, I. (ed) 1988. *Computer Supported Cooperative Work: A Book of Readings,* San Mateo, CA: Morgan Kaufmann Publishers, Inc., pp. 35-65.

Ref-3: Engelbart, D.C. 1988. The Augmented Knowledge Workshop. Goldberg, A. [ed], 1988. *A History of Personal Workstations*, New York: ACM Press, pp. 185-236.

Ref-4: Engelbart, D.C. and Lehtman, H.G. 1988. Working Together, *BYTE Magazine,* December, pp. 245-252.

Ref-5: Engelbart, D.C. 1990. Knowledge Domain Interoperability and an Open Hyperdocument System. *Proceedings of the Conference on Computer-Supported Cooperative Work*, Los Angeles, CA, October 7-10, pp. 143-156. (AUGMENT,132082,). Republished in Berk, E. and Devlin, J. [eds] *1991. Hypertext / Hypermedia Handbook*, New York: Intertext Publications, McGraw-Hill, pp. 397-413.

Ref-6: Engelbart, D.C. 1982. Toward High Performance Knowledge Workers. *OAC'82 Digest*, Proceedings of the AFIPS Office Automation Conference, San Francisco, CA, April 5-7, pp. 279-290. (AUGMENT,81010,). Republished in Greif, I. (ed) 1988. *Computer Supported Cooperative Work: A Book of Readings*, San Mateo, CA: Morgan Kaufmann Publishers, Inc., pp. 67-78.

Ref-7: Engelbart, D.C. 1984. Collaboration Support Provisions in AUGMENT. *OAC '84 Digest*, Proceedings of the 1984 AFIPS Office Automation Conference, Los Angeles, CA, February 20-22, pp. 51-58. (OAD,2221,).

Ref-8: Engelbart, D.C. 1984. Authorship Provisions in AUGMENT. *COMPCON '84 Digest*, Proceedings of the COMPCON Conference, San Francisco, CA, February 27 - March 1, pp. 465-472. (OAD,2250,). Republished in Greif, I. (ed) 1988. *Computer Supported Cooperative Work: A Book of Readings*, San Mateo, CA: Morgan Kaufmann Publishers, Inc., pp. 107-126.

Ref-9: Irby, C.H. 1976. The Command Meta Language System. *AFIPS Conference Proceedings*, NCC Vol. 45, Montvale, NJ: AFIPS Press. (AUGMENT,27266,)

Ref-10: Watson, R.W. 1976. User Interface Design Issues for a Large Interactive System. *AFIPS Conference Proceedings*, Vol. 45, Montvale, NJ: AFIPS Press, pp. 357-364. (AUGMENT,27171,).

Ref-11: Engelbart, D.C. 1972. Coordinated Information Services for a Discipline- or Mission-Oriented Community. *Proceedings of the Second Annual Computer Communications Conference*, San Jose, CA, January 24,. Republished in Grimsdale, R.L. and Kuo, F.F. (eds) 1975. *Computer Communication Networks*, Leyden: Noordhoff. (AUGMENT,12445,).

Ref-12: Grenier, R., Metes, G. 1992. *Enterprise Networking: Working Together Apart*. Digital Press. (Very relevant general treatment; special emphasis given to "Capability-Based Environment" along the lines outlined in this paper.)

Ref-13: Parunak, H.V.D. 1991. Toward Industrial Strength Hypermedia, *Hypertext / Hypermedia Handbook*, Kerk, E. and Devlin, J. (eds), New York: McGraw Hill, pp. 381-395. (Provides very useful considerations relevant to requirements for the Open Hyperdocument System as discussed in this paper.)

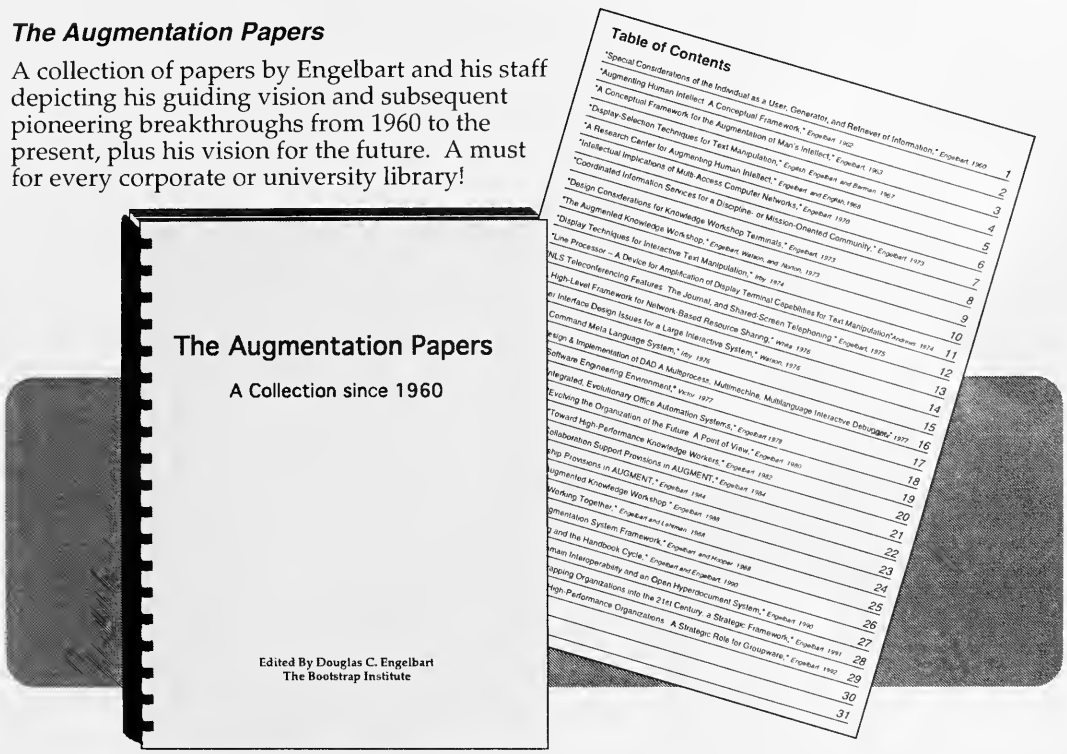
# BOOTSTRAP PUBLICATIONS

Is your organization poised and fit, ready to face the rapidly increasing complexity and urgency of the next decade and beyond? These Bootstrap publications offer a glimpse of the vision which has driven Doug Engelbart for over 30 years, and inspired his seminal work—from inventing the mouse, to designing high-performance organizations. Learn what Engelbart thinks it will take to make quantum leaps in productivity and competitiveness, and what role you and your organization should be playing.

***"Doug Engelbart has forever changed the way we do business in America"***
**—Coors American Ingenuity Award 1991.**

### *The Augmentation Papers*

A collection of papers by Engelbart and his staff depicting his guiding vision and subsequent pioneering breakthroughs from 1960 to the present, plus his vision for the future. A must for every corporate or university library!




## Bootstrap Order Form

| ✔ | Publication | Price | Ship Wt. |
|---|---|---|---|
| ☐ | *The Augmentation Papers* | $125 | 4 lbs |
| ☐ | *Bootstrap Seminar Binder* | $125 | 4 lbs |
| ☐ | *Augmented Knowledge Workshop (VHS)* | $125 | 1 lb |
| ☐ | *Together We Can Get There! (VHS)* | $125 | 1 lb |
| ☐ | Package Deal *(all of the above)* | $425 | 10 lbs |
| ☐ | Individual Papers *(please specify)* | $ 15 | 0.2 lbs |
| | **Subtotal** = | _____.___ | |
| | 20% Discount for Non-Profits - | _____.___ | |
| | Sales Tax *(CA residents only)* + | _____.___ | |
| | Shipping *(see below)* + | _____.___ | ___lbs |
| | **Total** = | _____.___ | |

***Shipping:*** For cont'l US UPS ground shipping add $5 per $125-item, $10 for Package Deal, and $1 per Individual Paper. For other destinations and shipping options, please call or refer to Bootstrap Institute brochure.

Please make check payable to ***Bootstrap Institute*** and mail with this form to: *Bootstrap Institute, 6505 Kaiser Drive, Fremont, CA 94555.*

*Name:* _____

*Title:* _____

*Org:* _____

*Addr:* _____

_____

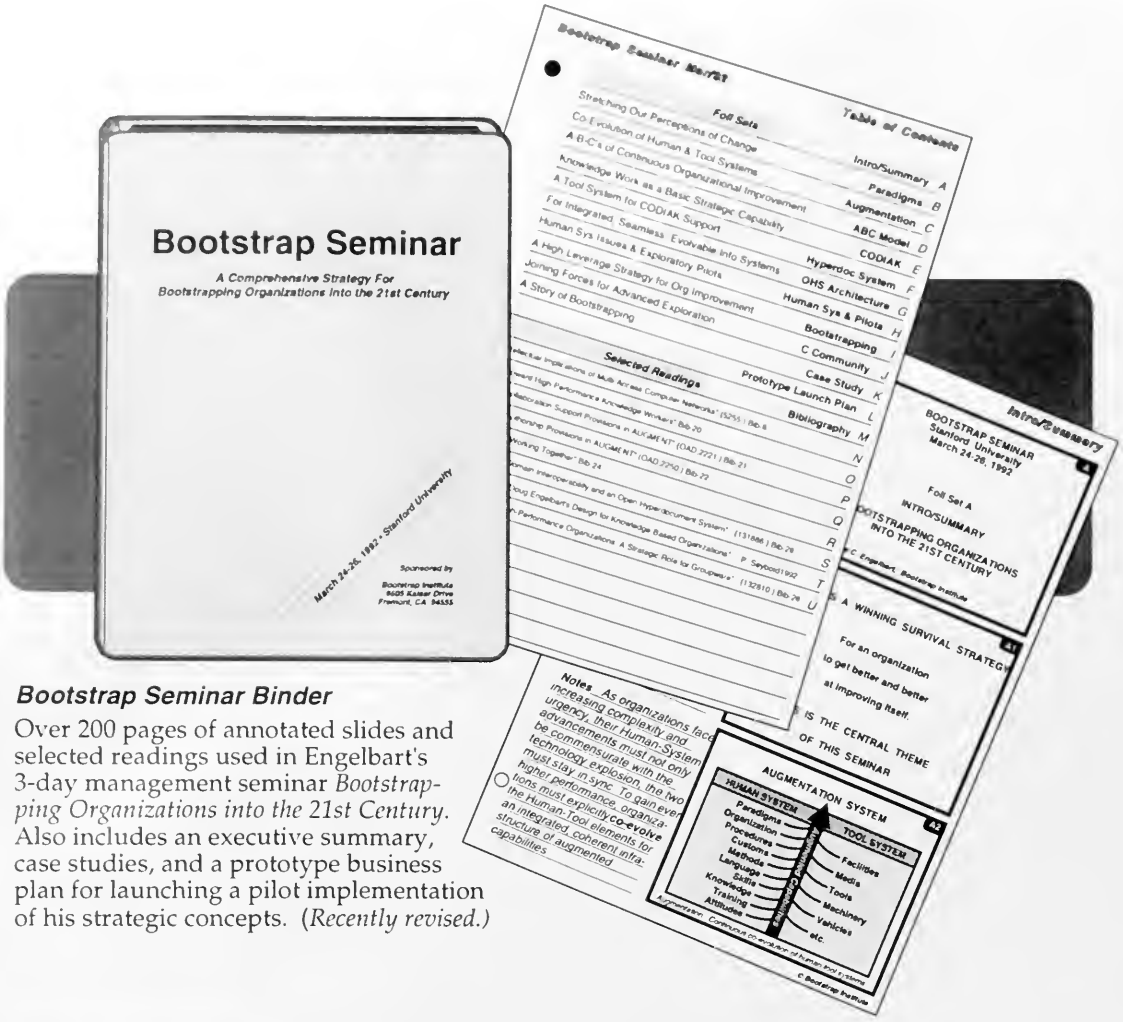
*Phone:* _____ *Fax:* _____

*EMail:* _____

*Interests (e.g.TQM, concurrent eng, groupware, hypermedia)* _____

_____

***For more information***
*Phone: (510)713-3550 • Fax: (510)792-3506*
*Email: info@bootstrap.org*




### *Bootstrap Seminar Binder*

Over 200 pages of annotated slides and selected readings used in Engelbart's 3-day management seminar *Bootstrapping Organizations into the 21st Century*. Also includes an executive summary, case studies, and a prototype business plan for launching a pilot implementation of his strategic concepts. (*Recently revised.*)

### *The Augmented Knowledge Workshop*


Engelbart presents the history of his award-winning NLS system at the 1986 ACM Conference on the History of the Personal Workstation. With rare photos from his legendary SRI lab, design rationale, and a sprinkling of candid anecdotes. Also a 20-minute clip from a demo given at the 1968 Fall Joint Computer Conference—the world debut of the mouse, windows, hypermedia, and shared-screen video tele-conferencing. Many of the groupware/hypertext features integrated into NLS are still on the wish lists of today's organizations. (©1986 82-min) Comes with 50-page companion paper.

[*Notes*: Selected footage from this video is on display at the Smithsonian Museum Exhibit on The Information Age. NLS was awarded the 1990 ACM Software System Award as "*a fitting recognition of the importance of this seminal work on interactive system design.*"]

### *Together We Can Get There!*


Patricia Seybold interviews Engelbart about his Bootstrap Strategy for high-performance organizations. This video, and companion two-part article by Seybold, covers the Bootstrap Seminar highlights in a dynamic question-and-answer format. What exactly is *group knowledge work*? What strategic role does it play? How would an *open hyperdocument system* support complex knowledge-intensive projects? What are the basic usage requirements? How can the computer industry deliver such a system? What role will end-user organizations play? How can they effectively harness these tools? At what level should they be cooperating on these issues? How does this fit with an organization's improvement strategy? What should organizations be doing now? (©1991 90-min)