BONKERS
Collaborative Review System

Edward Dean Willis
John J. Leggett

Hypermedia Research Lab

DECEMBER 1993
TAMU-HRL 93-012

Table of Contents

ABSTRACT

This paper discusses the design and implementation of a hypertext system designed to support the collaborative tasks confronting scientists and engineers during the review of experimental data. This system is called *BONKERS* (Browsable On-line NotebooK for Engineers and Research Scientists). BONKERS was designed to meet the needs of engineers reviewing measured building energy data but is actually a general-purpose hypermedia tool. BONKERS works under the Microsoft Windows environment on any standard personal computer network, and combines hypertext with multimedia through the use of Window's Object Linking and Embedding (*OLE*) facilities.

1. INTRODUCTION

The design for BONKERS is based on the notebooks used for the review of measured building energy data within the *Texas LoanSTAR (Loans to Save Taxes and Resources) Monitoring and Analysis Program*, hereafter referred to as LoanSTAR [Verdict 90]. These are conventional loose-leaf notebooks containing pages of text and graphs that act as the centerpiece of the collaborative review task discussed later in this document. The idea behind BONKERS is to make a "better" notebook -- to combine the comfortable spatial features of a paper notebook with the rapid distribution and mutual accessibility of a computer network. BONKERS achieves this goal by 1) providing virtual notebook pages upon which the engineers can freely place information, and 2) providing a navigational *hypertext* facility for moving between related pieces of information [Nelson 65].

2. BACKGROUND

The creation of BONKERS was inspired by observation of failings in the quality-control procedures used in LoanSTAR (see below). These procedures rely on conventional circulation of printed reports, and are excessively time-consuming. Since LoanSTAR is widely considered a pilot project for larger-scale implementation, an understanding of the limitations of procedures developed in LoanSTAR is critical to meeting national energy-conservation goals.

2.1. The LoanSTAR Program

The LoanSTAR Monitoring and Analysis Project (MAP) is administered by the Texas Energy Office (TEO). It is a statewide project to implement energy conservation measures in public-sector buildings. As the monitoring contractor for this project, Texas A&M University's Energy Systems Lab (ESL) carries out a number of subtasks. One of these tasks is the measurement of energy usage in many of the participating buildings. The purpose of such measurement is to provide data for the analysis of the effectiveness of building system retrofits and operational measures intended to conserve energy.

To achieve this purpose, the buildings are instrumented with automated data recorders. Time series data are collected from these instruments via telephone lines and used for analytical purposes. This collection process is relatively error-prone, as both the instrumentation and its operation are complex. To maintain a high standard of accuracy, the ESL carries out a review of acquired data to detect errors and provide for their timely correction.

Although data acquisition is principally an automatic process, review is not. Review is currently carried out in a labor-intensive manner by a group using minimal automated support. As described by Bottger and Yetton, the subject-matter (building systems) experts within the group dominate the process, and their availability is a bottleneck to system performance [Bottger 88].

As the number of buildings monitored by LoanSTAR increases, so does the amount of effort required to perform reviews of the acquired data. At some indefinite point, the time required to conduct even the simplest reviews will exceed the time available. As Bush [Bush 45] and Englebart [Englebart 63] pointed out, such an information explosion is occurring in all fields of study. Expansion of LoanSTAR methodology requires resolving this issue, and techniques for doing so may also be applicable to other areas.

2.2. Example Task: Inspection Plot Notebooks

LoanSTAR produces several major reports that are used in the data review process. The primary working tool is the *Inspection Plot Notebook*, or IPN. A subset of an IPN is attached as Appendix A. The ESL data processing staff prepares the IPN weekly. It contains several pages of time series and scatter plots for each building. Each page contains several plots, and may be unique to a given building. In practice, the IPN is divided into several sections, each of which is circulated separately. Multiple copies of each section may also be circulated. After preparation, the inspection plot notebook is circulated to each of the reviewers. Reviewers make comments on the report by typing their comments into a file on a diskette circulated with the report. Comments by the reviewers are collected by the Site Description and Agency Contacts group (SDAC), and problems detected at this stage are communicated to responsible parties for correction. Output is passed on into the Monthly Energy Consumption Report (MECR). Figure 1 is a diagram of this process.

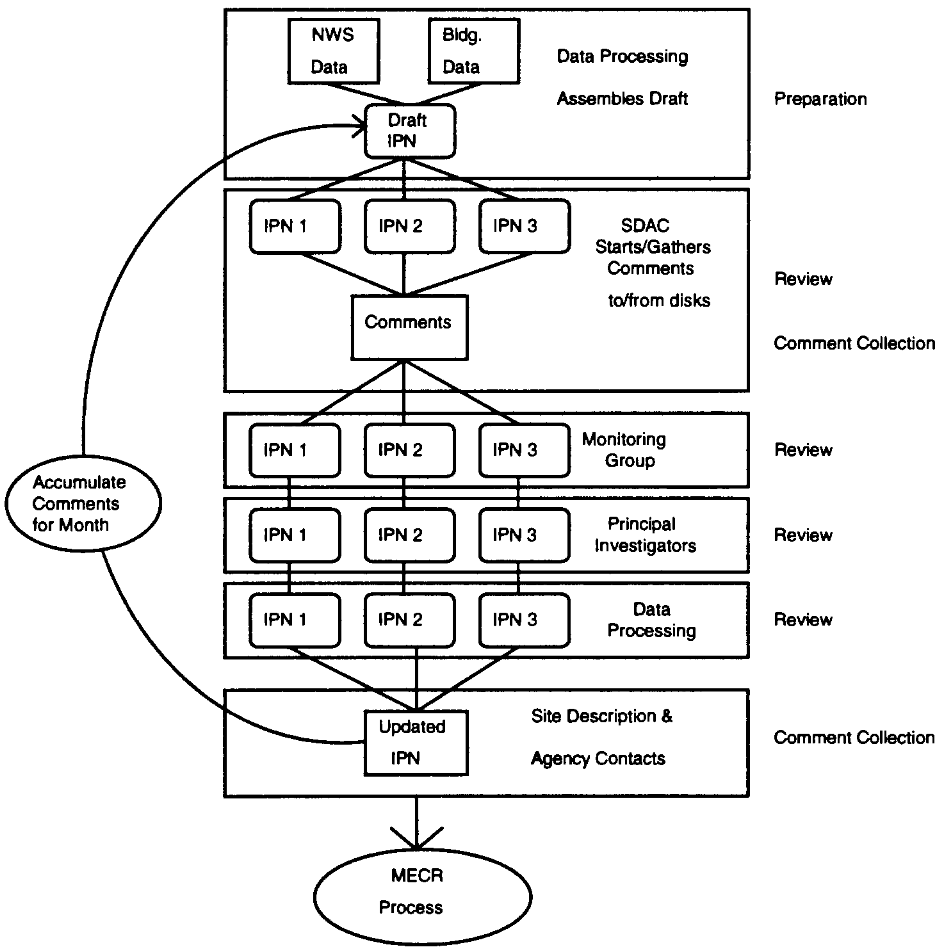


Figure 1: Current IPN Process

2.3 Limitations of Manual System

The existing manual system poses difficulties in terms of distribution, reduces effective collaboration, and does not adequately preserve the context of decision making.

2.3.1. Distribution

As can be seen from Figure 1, the IPN process involves distributing multiple copies of the notebook. This procedure was implemented after practice demonstrated that a single copy could not be serially passed through the entire set of reviewers in a week's time. Even with multiple copies, scheduling problems may develop which delay production or require skipping some reviewers to meet the production schedule. This most commonly occurs when one of the reviewers is busy or away, and the IPN copy routed to this reviewer is left idle. The numerous copies are also large consumers of physical resources -- paper, toner, and printer time and wear.

2.3.2. Collaboration

Each reviewer sees only comments made in one copy of the IPN, and only those comments which were made by reviewers earlier in the sequence of that copy. As a result, reviewers often remain unaware of critical observations made by others. There is little or no opportunity for the academic discussion which the reviewers feel is critical to improvements in quality.

2.3.3. Preservation of Context

Reviewers do see the main points of comments made by others, but generally only after those comments have been processed and summarized by the SDAC group. There is no direct accountability. Furthermore, the entire decision-making process becomes invisible, and it is difficult to impart reviewing skills and methodology to new staff members. Retaining this context could lead to an increase in the quality of organizational memory [Conklin 92].

3. REQUIREMENTS FOR AN AUTOMATION TOOL

In order to provide an effective solution, a collaboration support tool must address the task itself, be compatible with the existing systems infrastructure, and be both cost effective and within the financial scope of the operation.

3.1. Task

The collaborative review task as implemented in LoanSTAR can be broken down into a set of components which must be addressed by any system intended to assist in the performance of that task. These components include:

- Distribution of the materials to the participants
- Commenting by the participants
- Compilation of the comments into a report
- Coordination of the review activity
- Collaboration between the participants

3.1.1. Distribution

The distribution task component is the process of getting the materials to the reviewers. In the current manual process, the distribution component is labor intensive, involving:

1) Preparing multiple copies
2) Attaching routing slips to each copy
3) Hand-carrying each copy to its first reviewer
4) Hand-carrying between reviewers in each series
5) Hand-carrying back to SDAC staff for compilation

An optimal solution to the distribution task is one which gets the materials where they are needed, when they are needed, with the minimum amount of overhead, and the least susceptibility to bottlenecks. In this respect, the distribution component is "easy" -- optimality cannot be well-described for the other, more subjective components.

3.1.2. Commenting

The commenting task component is the recording of observations made by reviewers during the examination of the materials. In the current manual system, the preferred course of action is for each reviewer to write comments into a word-processing file on a diskette contained in each inspection plot notebook (figure 2). In practice, comments may also be written directly on the notebook page (or attached as PostIt notes) for later transcription. Also, detected problems may be handled directly, without record of the issue or the action taken to correct it.

Inspection Plot Notebook Comments
Volume 1: Sites 001 - 124
Data Beginning **11-16-93 & 11-23-93 to 11-29-93**

Note: Please initial your comments!

Weather Crossplots

AA:	Specific Humidity is getting bad at SanAntonio
AA:	Starting to get RH at Houston from 10/27/93
001	Zachry Engineering Center
AA:	Hot water may be bad. Ron could you also plot HW consumption from the venturi meter.
AA:	Please check the wind speed channel.
100:	UT/Education Building
AA:	Economizer cycle not working this year.
101	UT/UTC
AA:	Few hour of data lost on 11/21/93 due to power outage.
102	UT/PCL
118	UT/Garrison
119	UT/Gearing
AA:	Chw pump P1 is not running. Checked by Task A.

Figure 2: Examples of Comments from Current System

3.1.3. Compilation of Comments

Compilation of comments occurs after the review phase. This task involves transcribing and rearranging all comments into a consistent and orderly report. In the current manual process, this operation involves retrieving comments from files on the distributed diskettes and using of a word processing package to cut and paste the comments into a new document.

3.1.4. Coordination

The coordination task includes notifying the reviewers to begin, keeping the review process running (preventing bottlenecks), and exchanging of problem reports and resolutions between reviewers and field personnel. The current system does not provide a "standard" means of coordination, and the staff relies on a mix of memos, phone calls, electronic mail, and informal communications to coordinate their activities.

3.1.5. Collaboration

Baecker defines collaboration as "communication and problem solving carried out by a group" [Baecker 93]. In the context of the LoanSTAR project, this implies communication between producers, reviewers, task coordinators, field personnel, and trainees in any of these areas. The task coordinators (SDAC) work with the production team (Data Processing), then the reviewers, to start the process. The reviewers then work together to perform the review, bringing in field and production personnel as needed to resolve problems, while being monitored for timeliness by the task coordinators. Trainees may observe at any or all stages, and may pose questions to any of the participants at any time. This interaction is depicted in Figure 3.

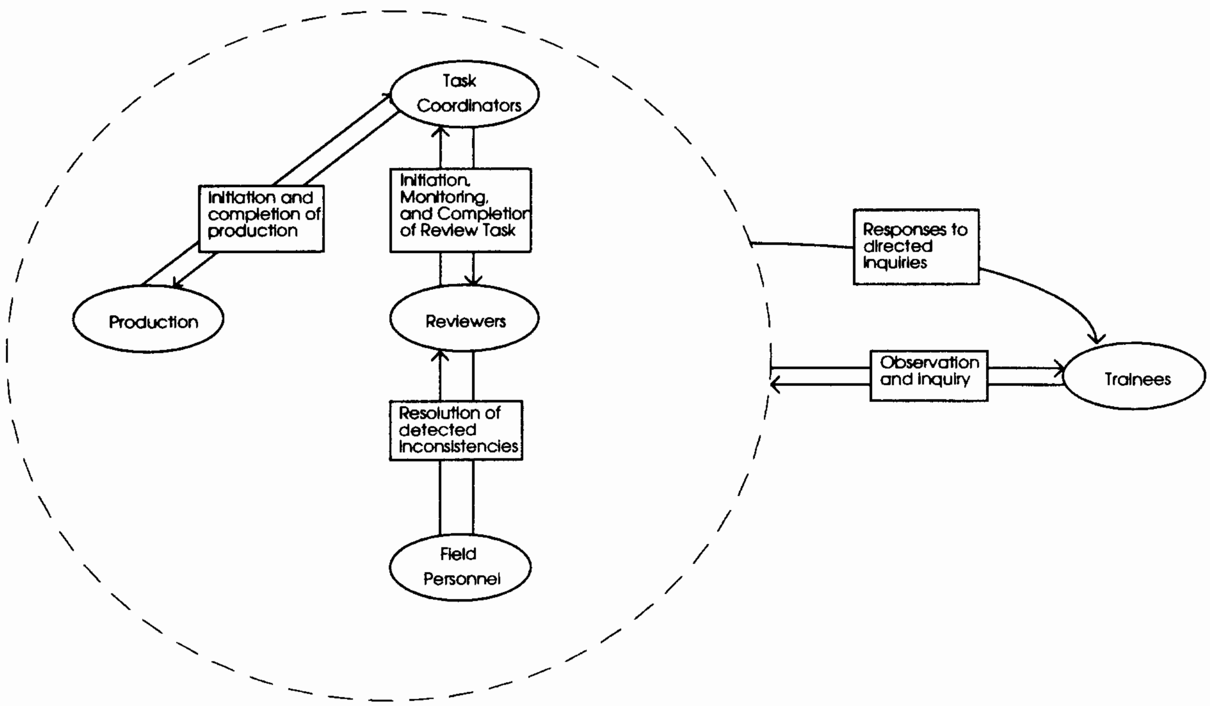


Figure 3: Collaboration in IPN Process

3.2. Systems

To be useful within the LoanSTAR project, any software must be usable with available computing facilities -- or be important enough to justify replacing those facilities. The existing facilities were put into operation by Dean Willis during the period of January 1990 through December 1993, and a substantial investment in software and training made during that period. All LoanSTAR personnel, including the IPN review team, are equipped with industry-standard personal computers using the Microsoft Windows operating system. These personal computers are configured for file and application sharing from a central server running the Novell NetWare network operating system over the existing campus Ethernet network. These computers can also be used to access the project's electronic mail server.

3.3. Financial

At the time of this writing, the LoanSTAR project is in a mature phase. The bulk of systems capital has been expended, and is now being budgeted on a "maintenance only" basis. This means that essentially no funds are available for investment in new commercial software, which effectively restricts consideration to public-domain software and in-house developments. It is noteworthy that, under the existing contract, any software developed by LoanSTAR must be entered into the public domain.

4. INADEQUACIES OF CONVENTIONAL SOFTWARE

Given the wide variety of software tools available, it is not unreasonable to consider whether the LoanSTAR requirements might be met by some sort of package which is currently available, or might be available at a low cost. Discussions among the process development staff concluded that there are potential means of using existing tools, but that the training overhead and maintenance issues outweigh the received benefits. Discussion centered on:

- Conventional word processing
- Workflow routing tools
- NCSA Mosaic
- NCSA Collage

4.1. Conventional Word Processing

LoanSTAR personal computers are equipped with Microsoft's Word for Windows, arguably one of the most complete word processing packages available. The current manual process uses Word for commenting -- each reviewer uses Word to write comments into a file on a diskette distributed with the notebook.

One possibility for automating with Word is to distribute the entire IPN as a single document file on the NetWare server, then have each reviewer directly update this file. This approach fails in several ways. The locking granularity is too large, and no two people can edit the file at any one time. Also, making changes directly in the file means that the history of changes made is lost, and that little or no discussion is actually occurring.

Another technique is to distribute the IPN as a single document file, and to have the reviewers comment into new files, using some sort of naming protocol. Since the IPN can be opened in a read-only mode, this eliminates the large granularity of locking, and preserves at least as much context as the current system. However, it is still cumbersome for a reviewer to examine the comments made by others -- Word's collateral display capabilities are severely limited by the video display hardware, and navigating through large serial documents can be awkward. What is needed is a way to directly associate comments with the relevant IPN page -- perhaps even to include some visual referent on that page.

It is possible to develop such an application using only Word, as it does have an elaborate macro language (BASIC), supports textual *buttons* which can actuate a macro, and has the idea of *bookmarks* which can be used as link destinations [Microsoft 93a]. This approach seems feasible, but complex. It is difficult to structure the interplay of documents such that concurrent access can be controlled. Furthermore, it would also be easy for a participant to accidentally "break" the system in such a way that recovery would be difficult. Editing or deleting macros or deleting bookmarks could shut down the entire operation, and access control for these resources is minimal. Certainly these difficulties could be overcome by rigid social protocols, intensive training, and careful coding -- but the result would still be cumbersome to use.

4.2. Workflow-Routing Tools

One approach to a team task is to route each item of work through the sequence of steps needed to accomplish the desired result. This can be based either on a flow-model, such as that seen earlier in Figure 3, or on a formalized sequence of actions such as requests and promises [Flores 88].

The underlying concept of workflow routing is that there is a clearly defined sequence of transitions (a plan) such as: A gets the problem, routes to B, who processes, request details from C, receives the details, and routes to D who finishes and returns to A. Unfortunately, real tasks, especially those under discussion, are not so straightforward, and tend to be situationally based [Suchman 87]. In the case of IPN review, the scenario might be better described as: A produces report and passes to B. B makes it public. Several of C's edit, revise, and comment on the report, conferring with more D's and with A as needed, then return it to B. The D's may also confer with A's, and perhaps with others as well.

Workflow routing approaches are not well suited to this seeming chaos -- somewhere in the A to B to C transition the spontaneity, creativity, and downright necessity of informal communications are missed, only to result in the formation of contacts outside of the system and the eventual atrophy of the whole automated process.

4.3. NCSA Mosaic

NCSA *Mosaic* is a hyperdocument system based on the work done at CERN and revised by the National Center for Supercomputing Activities [NCSA 93]. A version is now available which is compatible with the ISA-PC and Microsoft Windows environment used in the LoanSTAR program. Although a limited "forms" response system is now available, Mosaic is primarily a delivery tool -- that is, it does not support modification of the hypertext from user application. Although Mosaic could be an effective distribution tool, it does not support in-place commenting. Mosaic could be combined with an external editor for commenting, but this would suffer from one of the weaknesses of the current system -- the association between the comment and the material to which it refers is not explicit. Furthermore, there would be no enhancement to collaboration or coordination over the existing manual system.

4.4. NCSA Collage

Like Mosaic, *Collage* is also a development of the National Center for Supercomputing Activities [Wilson 93]. It is derived from their earlier *Image* and *Audible Image* systems, and serves as the basis for the *Envision* visualization system. Collage seems to be primarily a synchronous collaboration tool, where multiple users share the same virtual screen. Collage seems to have potential for use in the given review task. It provides for the distribution, coordination, financial, and potentially the collaboration requirements.

A beta version of Collage is available for personal computers running the Windows environment. However, this version of Collage requires a specialized NCSA TCP/IP transport layer which is not compatible with current standards or applications, and in this light does not meet the systems requirement. All of the reported experience with Collage was acquired using a version running under UNIX on a RISC workstation -- unfortunately the port to this system was extremely bad, and the resulting software almost completely dysfunctional.

In short, Collage and its follow-ons have substantial potentials for collaborative review However, they are currently inadequate for even a proof-of-concept due to the severe systems problems encountered.

5. THE BONKERS APPLICATION

BONKERS is a *frame-based* [Robertson 81] hypertext system which is readily adaptable to supporting collaborative work. Used for collaboration support, BONKERS provides a *shared task space* in which the participants interact [Buxton 92]. This interaction occurs in an asynchronous manner -- multiple participants do not work simultaneously on the same material [Baecker 93]. BONKERS provides several of the "essential elements" described by Englebart -- mixed object documents and explicitl addressing [Englebart 90]. It does not provide object-level access control or addressing, as in the component-object model, these concepts become essentially meaningless. On the other hand, through use of OLE (Object Linking and Embedding) [Microsoft 93b], BONKERS capably handles the task of providing a hypertext front-end to non-hypertext applications [Bieber 91].

5.1. User Interface Issues

The BONKERS user interface is principally derived from the standards of the Microsoft Windows environment [Microsoft 91]. Adherence to this standard is important because the primary intended users are familiar with Windows conventions. As pointed out by Meister, close matching between system design and user expectations reduces users need to adjust the cognitive models by which they operate the system, thereby increasing acceptance and reducing errors [Meister 85].

Components of the Windows standard used in BONKERS include multiple overlapping windows, pull-down menus, use of a pointing device, cut-and-paste, and the context-sensitive help system. Objects are positioned and sized on-screen through *direct manipulation* -- the pointing device is used to drag the object around for positioning, or to drag *resizing handles* (little black rectangles at the corner of the object) to change the size of the object.

Additionally, implementation of the OLE specification in BONKERS allows any OLE server-compliant application to be used from within BONKERS. This eliminates the difficulty reported with the PREP system, where incompatibilities with other tools were considered a "major stumbling block" to users of the system [Neuwirth 90].

5.2. Document Structure

The BONKERS metaphor is that of a notebook, filled with pages, each of which has on it objects such as chunks of text or graphic images. This approach to information storage, widely seen in the *Notecards* system [Halasz 87] is ubiquitous to hypertext applications, leading Ted Nelson [Nelson 87] to observe:

> *"Most text systems today are built around the VPOP (Virtual Piece of Paper) and shower you with COMPLOP (Big piles of printout)."*

5.2.1. Pages in Notebook

Each page in a BONKERS notebook is actually a separate document, and is stored in a unique file in the underlying filesystem. Within the application, the page is represented as an object of class *CBonkDoc*, which serves as a collection class for the content objects displayed on that page. In Microsoft terminology, this is referred to as a *component-object model*, where the page functions as the container for the components of that page [Brockschmidt 93]. Figure 4 is a sample BONKERS page, which contains several chunks of text, an embedded freehand drawing, and a linked *Excel* (Microsoft's spreadsheet program) graph. OLE objects may be either *embedded* -- contained within the BONKERS document, or *linked* -- stored in an external application. Figure 5 depicts this relationship.

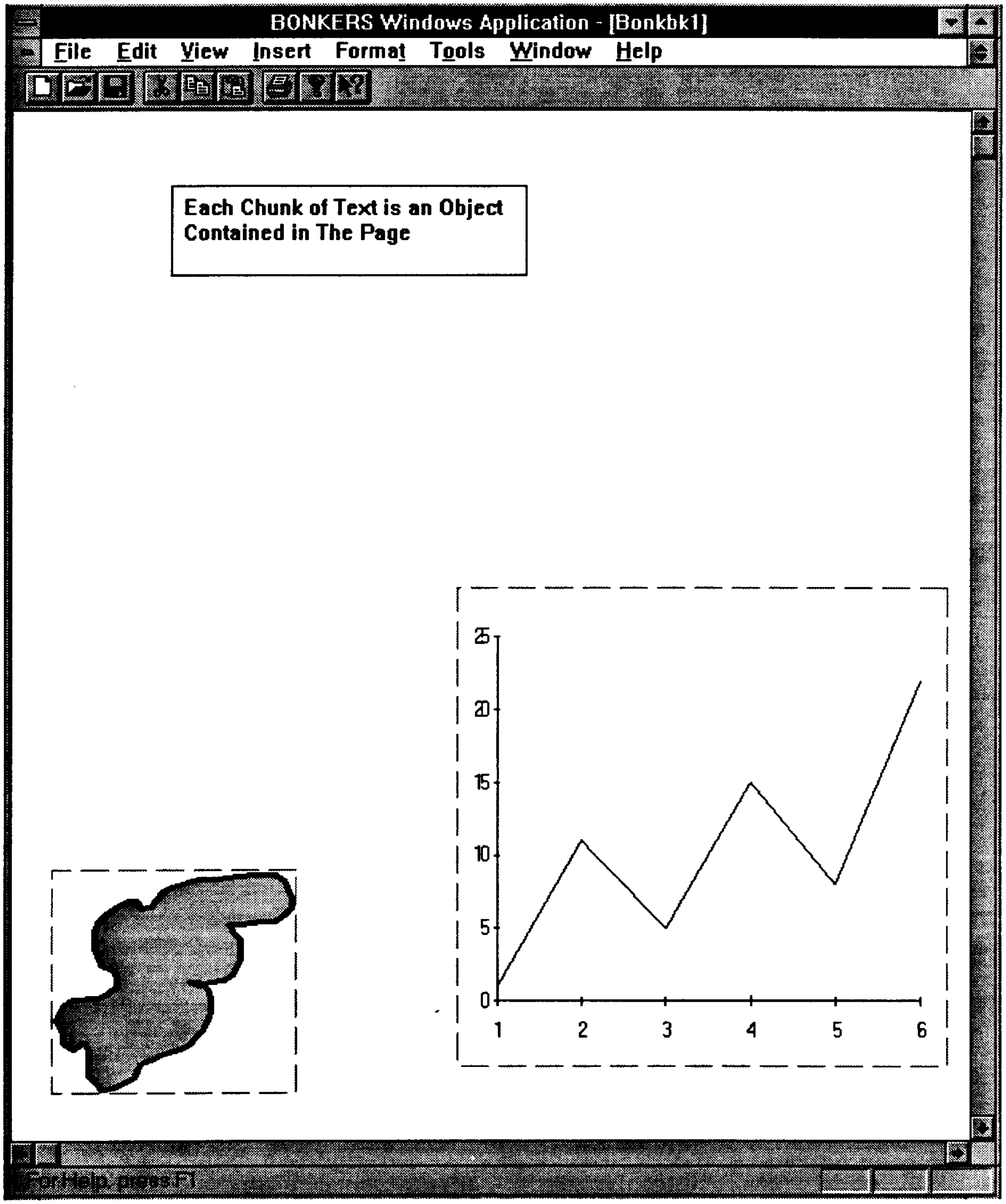


Figure 4: Sample BONKERS Page in Editor

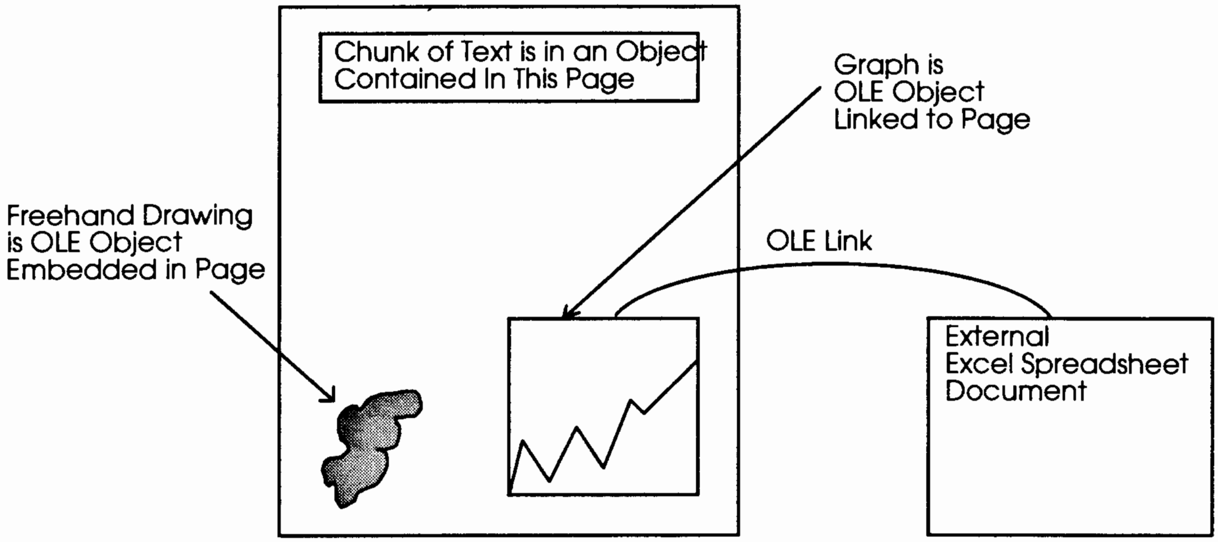


Figure 5: Component-Object Model Structure

5.2.2. Objects on Page

Each BONKERS page contains objects which are either chunks of simple text, or representations of OLE objects. Within the application, each displayed object is represented as an object of a subclass of *CBonkRgn*, the "region" class. Each displayed object is bounded by a rectangle, and may have a "border" property, which causes the object to be displayed or printed with the bounding rectangle drawn in a visible manner.

5.2.3. Object-Property Association

An arbitrary property-value list for each content object, like that of Common LISP, is implemented in BONKERS. [Koschmann 90]. Each object may have an arbitrary number of properties, each of which is identified by a unique string. Each property has a value, also a string. This technique allows the addition of new functionality without changing the existing class definitions or declaring new subclasses.

5.3. Hypertext Linking Structure

The hypertext linking mechanism in BONKERS principally supports construction of hierarchical trees of associated pages. However, there is no mechanism for preventing cycles in the current implementation. Dropped subtrees -- pages to which all links have been deleted -- can be accessed using the explicit address (DOS path and filename) of the page at the root of each tree.

5.3.1. Links and Anchors

BONKERS links are of the node-element to node variety. Each link is anchored (originates at) an object on a page, such as a text box. The destination of a link is another page, not an object on that page. Each anchor supports only one outgoing link, and there is no restriction on the number of links to any given page. Linking is implemented using the arbitrary property-value component of content objects and the explicit address of destination pages -- the explicit address of the destination is stored in the "link" property of the object.

5.3.2. Template-Based Instantiation

New BONKERS pages may be created as *referenced* or *unreferenced.* An unreferenced page, one which is the destination of no links at the time of its creation, is created from the "File" menu. A referenced page is created by right double-clicking a content object which will be linked to the new page. One feature believed unique to BONKERS is that new referenced pages may be easily created from user-configurable templates. Right double-clicking a content object which has no link property results in the display of a dialog box. This dialog box contains a simple combo-box (a standard dual-mode list pane) which lists all BONKERS files in the template directory. A user may select from one of the templates by clicking the template in the combo-box, can enter the explicit path to another frame to be used as a template, or can select a new blank page. Figure 6 displays this dialog box.

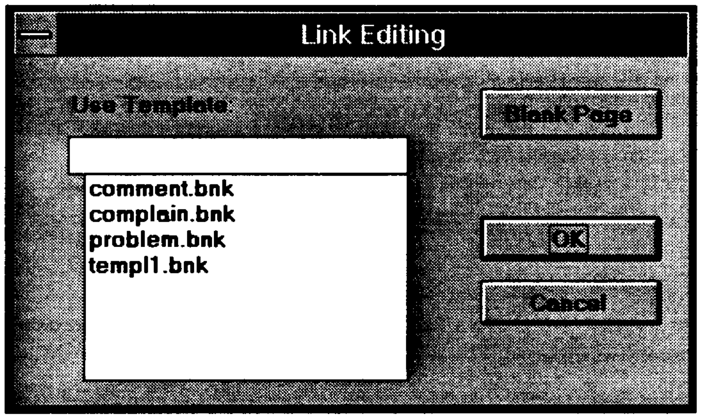


Figure 6: Dialog Box for Creating New Referenced Page from Template

5.4. Concurrency Control

In this context, concurrency control refers to the mechanism for handling multiple simultaneous accesses to components of the system. Multiple users may attempt to update the same piece of information, or one user may be viewing a piece while another updates it. There may also need to be an exchange of information between users about planned accesses. These issues can be classified as access contention, monitoring, and communication [Trigg 86]. This version of BONKERS provides absolute minimal concurrency control, probably far below the level required to effectively support collaboration in an intensive production environment [Wiil 93].

5.4.1. Page Level

The current version of BONKERS provides only minimal non-locking operating-system level concurrency control on page files. Multiple users may open a document in edit mode, without notification of other editors. The application does not notify users when changes are made by others. If multiple edits are made, the last one saved is the one retained. Implementation of an edit-locking facility is intended in the very near term. The operating system does provide exclusive locking on output, such that files cannot be accidentally interleaved in the case of simultaneous write attempts.

5.4.2. OLE Object Level

OLE objects may either be embedded (stored within the page) or linked (stored outside the page). No concurrency control other than that provided at the page level is provided for embedded objects. However, the system framework provides some notification control for linked objects, and may provide access contention if the server application for the link provides access contention control. Linked object notification is implemented via dispatch of an *OnChanged* event by the server application to all registered OLE clients.

5.4.3. Control File Level

BONKERS currently uses two control files in Windows INI format -- the BONKERS.INI file on the client machine, and the BONKNAME.INI file in the working directory specified within the first file. The Windows environment provides access contention control for these files.

5.5. Naming Strategies

Within BONKERS, some components must be identified by unique (or effectively unique) names. The current system provides for the naming of document sets, individual pages, and the OLE objects within a page.

5.5.1. Document Sets

A document set is actually the file-system directory containing a name initialization file, and generally most or all of the page files associated with that set. These directory names are user-specified, and are not created arbitrarily by the system. BONKERS determines the current document set directory by examining the "WorkPath" setting in the BONKERS.INI file. Pages may be written or read from any available directory, but new referenced frames are created in the document set directory.

5.5.2. Page Names

The names of the files in which page contents are stored may be either arbitrary (user-supplied) or system created. All file names for new referenced pages are system created, using sequential numbering based on the "LastFileIndex" setting of the document set directory's BONKNAME.INI file. Page file names must be unique within the directory containing the file. The system detects attempts to overwrite existing files, and will warn the user.

5.5.3. OLE Objects

Each OLE object must be uniquely named within the page, while that page is active in BONKERS. However, there is no requirement for this name to be preserved from session to session. BONKERS addresses this by sequentially naming each object as it is created during page serialization.

5.6. Access Control

In this context access control refers to permitting or denying access to system components dependent on who is attempting the access. The current implementation of BONKERS relegates this task to the network operating system, which provides all common access controls based on the user-ID provided at login. Since the system controls access on a file level, the resultant access control appears in BONKERS at the page level -- a user with a given level of access to any page has that same level of access to all components of that page.

5.7. Implementation Methodology

The idea behind BONKERS is to produce a simple, low cost tool for collaboration support which meets the minimal requirements for usability in the LoanSTAR project. Towards this end, the target environment and development tools were selected to provide the maximal use of existing resources and minimize the amount of new code required, as well as the level of maintenance effort needed over a longer term.

5.7.1. Target Environment

The target environment, simply put, is the computer systems operated by the Energy Systems Lab in support of the LoanSTAR project. In practical terms, this is:

- Industry standard (Intel CPU) personal computers
- Microsoft Windows operating system
- Color displays with 1024x768 resolution displays
- Adequate disk and memory resources for common applications
- A central file server using Novell NetWare
- Installed base of applications including Word and Excel

5.7.2. Development Tools

Initial prototypes, such as that documented in [Willis 92] were developed using the Microsoft Visual BASIC environment. As the complexity of the design increased, it became apparent to the authors that the framework imposed by Visual BASIC was not adequately flexible, and that this lack was detrimental to application design. The authors then adopted the AT&T C++ language, which allows direct access to underlying structures. Initially, the Borland C++ compiler (version 3.1) was tried, but was found to contain many errors. The authors then adopted Microsoft Visual C++ (version 1.0), which is the compiler for the current version of BONKERS. The Microsoft Foundation Class library provided a code base of over one million lines of code for reuse. The object orientation of C++ helped make this large library both manageable and useful, resulting in a final code size on the order of 5,000 lines.

6. HOW BONKERS MEETS REQUIREMENTS

BONKERS is specifically designed to meet the task, systems, and financial requirements of a collaboration support system for LoanSTAR data review.

6.1. Task

The task requirements, as discussed earlier, include distribution, commenting, compilation of comments, coordination, and collaboration.

6.1.1. Distribution

BONKERS essentially eliminates the distribution task. Since it allows shared access to the set of pages making up a notebook, there is no requirement to route copies of the notebook to the participants. Once the BONKERS notebook has been created, the task coordinator simply advises the reviewers (via an email distribution list) that they may begin work.

6.1.2. Commenting

BONKERS allows reviewers to make comments directly on the relevant page, or to link to a new page or pages as needed. Reviewers can make new pages using user-defined templates which simplify the commenting process.

6.1.3. Compilation of Comments

The SDAC group can peruse the BONKERS workspace to extract comments, much as they would peruse an ordinary notebook -- except that BONKERS comments can be copied and pasted directly into other reports. It may be possible to adapt through software modification and/or social protocols to provide for more intelligent, even automated, compilation. The authors expect that after some usage of BONKERS additional measures will become apparent.

6.1.4. Coordination

Review activity takes place around a single shared artifact -- the BONKERS notebook. This artifact then serves as the centerpiece of coordination. Questions, answers, and problem reports can all be exchanged and dealt with within the system. The task coordinator can look into the notebook to see the status of the review process rather than polling the participants.

6.1.5. Collaboration

Using BONKERS for review allows communication to be carried in the context of the issue -- each discussion leaves a visible artifact, and every observation made is available to the entire team. Students and trainees can see the processes by which decisions are made, follow the trails of discussions, and learn. As Vannevar Bush wrote [Bush 45]:

> *"The inheritance of the master becomes not only his additions to the world's record, but the entire scaffolding by which they were erected."*

6.2. Systems

BONKERS was designed from the start for the systems environment of the LoanSTAR program. Furthermore, it was developed and tested in that same environment, and as such is ideally suited to handle the requirements of this environment. In particular, BONKERS is a native Windows application, supports the standard PC network in use, and interacts transparently via OLE and cut-and-paste with the other software in use in this program.

6.3. Financial

As a student project developed within the Texas A&M University System, BONKERS is available at no charge to the LoanSTAR Program. Furthermore, primary BONKERS development was carried out outside of funded time using private computers. Final BONKERS testing and pilot project development were performed with University computers. So, for all practical purposes, BONKERS is free -- a strong financial argument, and certainly one meeting the financial requirements described earlier.

7. FUTURE WORK

Although BONKERS represents a substantial step towards a complete environment for collaborative review in an environment like that of the LoanSTAR program, it is by no means a complete solution. Each advance suggests new alternatives, and only by living with the system will new advances become apparent. Usage during development and proof-of-concept testing has demonstrated the need for advances in several areas:

- Concurrency Control
- Richer Formatting
- Property Editing and Utilization
- OLE 2.0 and In-Place Activation
- Batch Notebook Construction
- Detached Work Sets

7.1. Concurrency Control

As noted earlier, the current version of BONKERS provides only the minimum of concurrency control needed to prevent abnormal termination, and does not guarantee consistency or preservation of changes. The highest priority for development of future revisions is more complete concurrency control, at least to the full locking model described in the EHTS system [Wiil 92].

7.2. Richer Formatting

Text regions provided in BONKERS provide most of the content. However, formatting options for these regions are currently minimal -- one size and face of text, with all paragraphs left justified. Additional formatting could be added by utilization of *face*, *size*, and *justification* properties (using the existing property-value schema), coupled with a revised *Draw* method for the text region class and a few new menu items and dialog boxes. This formatting would apply on a "region-wide" level -- a selection of font, face, or justification would apply to all characters in the region. Mixed typefaces and more elaborate features might be implemented using OLE 2.0, as discussed below.

7.3. Property Editing and Utilization

The property-value schema for each display object provides rich possibilities. The first step in further exploiting them is to provide a property editor, invocable by a menu operation when an object is selected. This editing facility could be used to view or change the properties of a selected object. It is also desirable to create a record of the creator of each object, as well as the creation and modification dates -- object properties easily developed using the property-value schema and modifications to the base-class constructor and edit methods.

7.4. OLE 2.0 and In-Place Activation

Microsoft has announced new extensions to Object Linking and Embedding. OLE version 2 will provide greater application transparency, including a new feature called "in-place activation." An OLE object contained in a page can be edited with all the function of the server program, while still in the context of the client program. As an example: Once OLE version 2 is implemented in BONKERS, double-clicking an embedded Excel spreadsheet will, rather than popping up an Excel window, activate the Excel functionality directly on the BONKERS page. The part of the screen occupied by the spreadsheet object will actually become a "live" spreadsheet, with the relevant Excel menus replacing BONKERS menus in the menu bar.

7.5. Batch Notebook Construction

For use in a production basis, it must be at least as easy to create an Inspection Plot Notebook (or other large report) in BONKERS as it is with the current manual system. This means that there must be some mechanism for producing an entire BONKERS document from the existing UNIX-hosted relational database. The proposed mechanism is to define an intermediate file format which can be produced by a straightforward modification of the existing production routines, (running on the UNIX server against the RDBMS). This intermediate file format could then be translated in by a modified version of BONKERS, which would result in the creation of a new BONKERS notebook made up of many individual page files.

7.6. Detached Work Sets

LoanSTAR reviewers spend a substantial proportion of time traveling and/or working at home, using portable computers. Several have already requested the ability to split off a chunk of BONKERS data, load it onto the portable, work on it, and merge it back into the main system. While this is possible using manual manipulations and some discipline, the merging phase is tedious and error-prone. Current thinking has produced no reasonable solution to this problem.

8. CONCLUSION

BONKERS meets the task requirements discussed in this document. As such, it has the potential to be a "better notebook," and to substantially improve the collaborative task of data review in the LoanSTAR program. It provides for improvements in distribution, coordination, and collaboration over the existing system, and preserves a far larger proportion of the context in which a decision is made, as well as the rationale of the decision-making. It provides this functionality at a minimal cost, and is compatible with the systems requirements and user expectations pertaining to the available computer systems. For BONKERS to completely replace the existing paper system it needs substantial improvements in the areas of concurrency control and automated production, and could certainly benefit from added functionality in formatting and tracking of component histories. The BONKERS system is general-purpose in nature, with substantial multimedia capabilities, and may prove useful outside of its original scope. In summation, BONKERS appears to be a successful and reasonably useful system, and its descendants are likely to radically change the nature of the collaborative review of measured building energy data as performed in the LoanSTAR program.

REFERENCES

[Baecker 93] Baecker, R., *Readings in Groupware and Computer-Supported Cooperative Work.* Morgan Kaufman Publishers, San Mateo CA, 1993.

[Bieber 91] Bieber, M., "Issues in Modeling a Dynamic Hypertext Interface for Non-Hypertext Systems", *Hypertext '91 Proceedings*, San Antonio, TX, December 15-18, 1991, pp. 203-218.

[Bottger 88] Bottger, P. and P. Yetton, "An Integration of Process and Decision Scheme Explanations of Group Problem Solving Performance", *Organizational Behaviour and Human Decision Processes,* vol. 42, 1988, pp. 234-249.

[Brockschmidt 93] Brockschmidt, K., "Introducing OLE 2.0, Part I: Windows Objects and the Component Object Model", Microsoft Systems Journal, August 1993, pp. 15-23.

[Bush 45] Bush, V., "As We May Think", *The Atlantic Monthly,* 176:1, Jul. 1945, pp. 101-108.

[Buxton 92] Buxton, W. "Telepresence: Integrating Shared Task and Person Spaces", *Proceedings of Graphic Interface '92*, Morgan Kaufman, pp. 123-129.

[Conklin 92] Conklin, E. J., "Capturing Organizational Memory", *Proceedings of Groupware '92*, Morgan Kaufman, pp. 133-137

[Englebart 63] Englebart, D., "A Conceptual Framework for the Augmentation of Man's Intellect", *Vistas in Information Handling,* Vol. 1, Spartan Books, Washington, DC, 1963, pp. 1-29.

[Englebart 90] Englebart, D. "Knowledge-Domain Interoperability and an Open Hyperdocument System", *Proceedings of the Conference on Computer Supported Collaborative Work*, Oct. 7-10, Los Angeles, pp. 143-156.

[Flores 88] Flores, F., and M. Graves, B. Hartfield, T. Winograd, "Computer Systems and the Design of Organizational Interaction", *ACM Transactions on Office Information Systems*, Vol. 6, No. 2, April 1988, pp. 153-172.

[Haberl 92] Haberl, J. and V. Jagannathan, R. Lopez, E. Sparks, K. Kissock, D. Willis, and D. Claridge, "Exploring an Integrated Database Structure for Building Energy Monitoring Data", *International Building Performance Simulation Association Proceedings*, Nice, France, August, 1992.

[Koschmann 90] Koschmann, Y. *The Common LISP Companion*. John Wiley and Sons, New York, 1990.

[Meister 85] Meister, D., *Behavioral Analysis and Measurement Methods*, John Wiley and Sons, New York, 1985.

[Microsoft 91] *Microsoft Windows Style Guide*, Microsoft Press, Seattle, WA, 1993.

[Microsoft 93a] *Word for Windows User Manual*, Microsoft Corporation, Seattle WA, 1993.

[Microsoft 93b] *User's Guide to Viusal C++*, Microsoft Corporation, Seattle WA, 1993.

[NCSA 93] Software Development Group of the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign, *NCSA Mosaic*, NCSA Mosaic Home Page, World Wide Web.

[Nelson 65] Nelson. T., "A File Structure for The Complex, The Changing, and the Indeterminate," *ACM Proceedings of the 20th National Conference*, 1965.

[Nelson 87] Nelson, T., *Literary Machines/Dream Machines*, Project Xanadu, San Antonio, TX, 1987.

[Neuwirth 90] Neuwirth, C., and D. Kaufer, R. Chandhok, J. Morris, "Issues in the Design of Computer Support for Co-Authoring and Commenting", *Proceedings of the Conference on Computer Supported Collaborative Work*, Oct. 7-10, Los Angeles, pp. 183-193.

[Robertson 81] Robertson, G., D. McCracken, and A. Newell, "The ZOG Approach to Man-Machine Communication," *Int. Journal of Man-Machine Studies*, Vol. 14, 1981, pp. 461-488.

[Shneiderman 83] Shneiderman, B., "Direct Manipulation: A Step Beyond Programming Languages," *IEEE Computer*, 16(8), 1983, pp. 57-69.

[Trigg 86] Trigg, R., and L. Suchman, F. Halasz., "Supporting Collaboration in NoteCards", *Proceedings of the CSCW '86 Conference*, Austin TX, December 1986, pp. 153-162.

[Verdict 90] Verdict, M., and J. Haberl, D. Claridge, D. O'Neal, W. Heffington, D. Turner, 1990. "The Texas LoanSTAR Monitoring and Analysis Program: Overview", *Proceedings of the 1990 ACEEE Summer Study*, August 1990, pp. 7.261-7.271.

[Wiil 92] Wiil, U.K., "Issues in the design of EHTS: A Multiuser Hypertext System for Collaboration", *Proceedings of the 25th Hawaii International Conference on Systems Sciences*, B. Schriver, ed., Kauai, HA, January 1992, pp. 629-639.

[Wiil 93] Will, U. K. and J. Leggett, "Concurrency Control in Collaborative Hypertext Systems", *Proceedings of the Hypertext '93 Conference*, Seattle, WA, November 14-18, 1993, pp. 14-24.

[Willis 92] Willis, E. D. and J. Haberl, "A Collaborative Support System for the Review of Building Energy Data in the LoanSTAR Monitoring and Analysis Project", *Proceedings of the Eighth Symposium on Improving Building Systems in Hot and Humid Climates*, May 13-14, 1992, Dallas TX, pp. 213-221.

[Wilson 93] Wilson, C., *Release Notes for NCSA Audible Collage for Microsoft Windows Version 1.05 BETA 1*, distributed with executables via Internet.

APPENDIX A: Sample Inspection Plot Notebook Subset

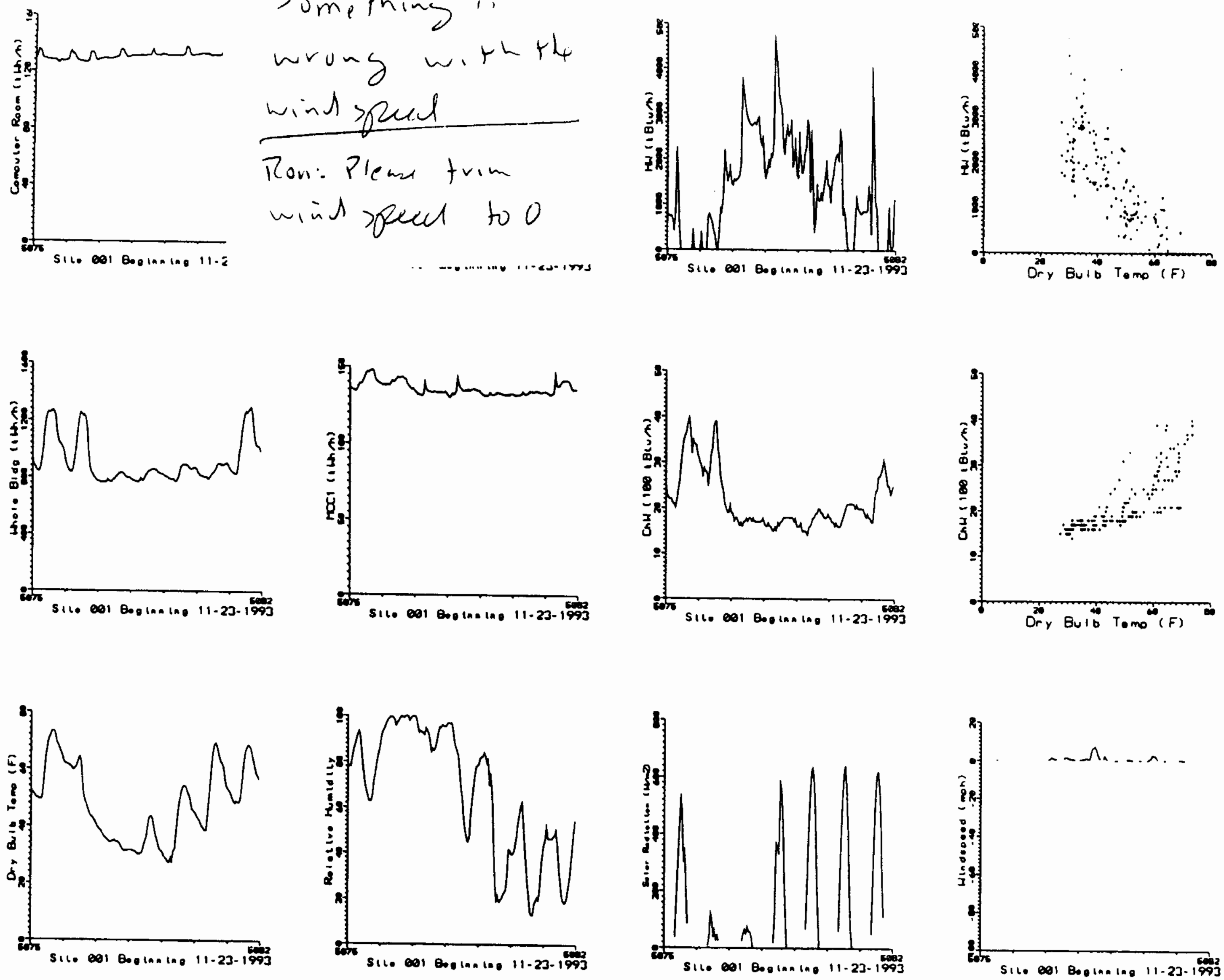
The following pages were photoreduced to 78% of the original size in order to capture comments written on Post-It notes pasted in a "tab" manner along the outside edge of the pages.

The material presented is the Inspection Plot Notebook for LoanSTAR Site #001, Zachry Engineering Center, College Station Campus, for the week of 11/23/93.

The complete IPN of site #001 for the week of 11/23/93 contains 25 pages, including a number of "background" pages which document the metering installation and vary little on a weekly basis. In the interest of brevity, only a subset of pages which demonstrate commenting activity are presented in this appendix. The overall review task is quite large -- currently 84 sites are being reviewed weekly.

00190001.cht   Fri May 7 10:01:42 1993   1

Date MM/DD/YY (YY DDD)	Time HH:mm	Raw-Data lin pos	Raw-Data coln pos	Arch coln pos	Name of Channel	Archive Units	Arch Format	Conv'n Code	Conv'n Constants	Error Code	Error Constants	Channel Description
44												
03/12/90	00:00	1	0	0	Begin	Zachry						Beginning date
03/12/90	00:00	1	1	1	Mon-Raw	MM	13	1		0		Month
03/12/90	00:00	1	2	2	Mon-Raw	DD	13	1		0		Day
03/12/90	00:00	1	3	3	Mon-Raw	YY	13	1		0		Year
03/12/90	00:00	1	3	4	Greg-Jul	MMDDYY	15	24	1 2	0		Gregorian Date to Julian
03/12/90	00:00	1	4	6	Time	HH mm	15	16	5	0		Time
03/12/90	00:00	1	3	5	Greg-Dec	DDD.frac	F10.4	28		0		Gregorian Date to Jul.Decimal
03/12/90	00:00	1	6	7	Main#1	F9.3	F9.3	1		1	0 600	Main Service Leg #1 (kWh/h)
03/12/90	00:00	1	7	8	Main#2	F9.3	F9.3	1		1	0 600	Main Service Leg #2 (kWh/h)
03/12/90	00:00	1	8	9	Main#3	F9.3	F9.3	1		1	0 600	Main Service Leg #3 (kWh/h)
03/12/90	00:00	1	9	10	MCC1#1	F9.3	F9.3	1		1	0 120	MCC #1 Leg #1 (kWh/h)
03/12/90	00:00	1	10	11	MCC1#2	F9.3	F9.3	1		1	0 120	MCC #1 Leg #2 (kWh/h)
03/12/90	00:00	1	11	12	MCC1#3	F9.3	F9.3	1		1	0 120	MCC #1 Leg #3 (kWh/h)
03/12/90	00:00	1	12	13	MCC2#1	F9.3	F9.3	1		1	0 80	MCC #2 Leg #1 (kWh/h)
03/12/90	00:00	1	13	14	MCC2#2	F9.3	F9.3	1		1	0 80	MCC #2 Leg #2 (kWh/h)
03/12/90	00:00	1	14	15	MCC2#3	F9.3	F9.3	1		1	0 80	MCC #2 Leg #3 (kWh/h)
03/12/90	00:00	1	15	16	Cry#1	F9.3	F9.3	1		1	0 200	Crystal Palace #1 (kWh/h)
03/12/90	00:00	1	16	17	Cry#2	F9.3	F9.3	1		1	0 200	Crystal Palace #2 (kWh/h)
03/12/90	00:00	1	17	18	Cry#3	F9.3	F9.3	1		1	0 200	Crystal Palace #3 (kWh/h)
03/12/90	00:00	1	18	19	Blank	F9.3	F9.3	1		0		Blank
03/12/90	00:00	1	19	20	AHU#1	F9.3	F9.3	1		1	0 50	AHU Fan Leg #1 (kWh/h)
03/12/90	00:00	1	20	21	AHU#2	F9.3	F9.3	1		1	0 50	AHU Fan Leg #2 (kwh/h)
03/12/90	00:00	1	21	22	AHU#3	F9.3	F9.3	1		1	0 5C	AHU Fan Leg #3 (kWh/h)
03/12/90	00:00	1	22	23	OA DB	F9.3	F9.3	1		1	-10 150	Rooftop Dry Bulb (F)
03/12/90	00:00	1	23	24	OA RH	F9.3	F9.3	1		1	0 110	Rooftop Relative Hum. (%)
03/12/90	00:00	1	24	25	Solar	F9.3	F9.3	1		1	-10 1100	Rooftop Global Solar (W/m2)
03/12/90	00:00	1	25	26	Wind	F9.3	F9.3	1		1	0 130	Rooftop Wind Speed (mph)
03/12/90	00:00	1	35	45	CWtot	F9.3	F9.3	1		1	0 100	Whole Bldg. CW (100kBtu)
03/12/90	00:00	1	36	43	HWtot	F9.3	F9.3	1		1	0 60	Whole Bldg. HW (100kBtu)
03/12/90	00:00	1	26	29	HDTMP	F9.3	F9.3	1		1	60 180	Hot Deck Temperature (F)
03/12/90	00:00	1	27	30	CDTMP	F9.3	F9.3	1		1	35 80	Cold Deck Temperature (F)
03/12/90	00:00	1	28	31	CDRH	F9.3	F9.3	1		1	0 105	Cold Deck Relative Humidity (%)
03/12/90	00:00	1	29	32	MATMP	F9.3	F9.3	1		1	50 100	Mixed Air Temperature (F)
03/12/90	00:00	1	30	33	MARH	F9.3	F9.3	1		1	0 100	Mixed Air Relative Humdity (%)
03/12/90	00:00	1	31	34	RATMP	F9.3	F9.3	1		1	60 90	Return Air Temperature (F)
03/12/90	00:00	1	32	35	RARH	F9.3	F9.3	1		1	0 100	Return Air Relative Humidity (%)
03/12/90	00:00	1	33	36	Flow	F9.3	F9.3	1		1	0 25000	Air Flow Rate (SCFM)
03/12/90	00:00	1	34	37	Diff P	F9.3	F9.3	1		1	-8 0	Pressure Drop Across the fan (in)
03/12/90	00:00	1	37	38	HPMP1	F9.3	F9.3	1		1	0 100	Percent ON time Hot Water Pump
03/12/90	00:00	1	38	39	HPMP2	F9.3	F9.3	1		1	0 100	Percent ON time Hot Water Pump
03/12/90	00:00	1	39	40	CPMP1	F9.3	F9.3	1		1	0 100	Percent ON time Cold Water Pump
03/12/90	00:00	1	40	41	CPMP2	F9.3	F9.3	1		1	0 100	Percent ON time Cold Water Pump
03/12/90	00:00	1	41	42	HWGHTUSA	F9.3	F9.3	1		1	0 99999	Whole Bldg. HW flow gallons
03/12/90	00:00	1	42	28	HWCHTUSA	F9.3	F9.3	1		1	0 99999	Whole Bldg. HW kBtu
03/12/90	00:00	1	44	27	CWCHTUSA	F9.3	F9.3	2	0.1 0	1	0 99999	Whole Bldg. CW consumption (10kBtu)
03/12/90	00:00	1	43	44	CWGHTUSA	F9.3	F9.3	1		1	0 99999	Whole Bldg. CW flow gallons
11/07/90	09:00	1	24	25	SolarOFF	F9.3	F9.3	0		1	-10 1100	Rooftop Global Solar (W/m2)
11/07/90	12:00	1	24	25	NewSolar	F9.3	F9.3	2	3.1525 0	1	-10 1100	Rooftop Global Solar (W/m2)
09/30/91	11:00	1	44	27	CWCHTUSA	F9.3	F9.3	1		1	0 99999	Whole Bldg. CW consumption (100kBtu
)												
04/28/92	00:00	1	45	46	AHU10HW	F9.3	F9.3	1		1	0 99999	AHU10 HW (kBtu)
04/28/92	00:00	1	46	47	AHU10HWF	F9.3	F9.3	1		1	0 99999	AHU10 HW Flow (gal)
04/28/92	00:00	1	47	48	AHU10CW	F9.3	F9.3	1		1	0 99999	AHU10 ChW (kBtu)
04/28/92	00:00	1	48	49	AHU10CWF	F9.3	F9.3	1		1	0 99999	AHU10 ChW (gal)
03/11/99	23:00	1	0	0	End	zachry						



Curtis:
Something is wrong with the wind speed

Ron: Please trim wind speed to 0

150,000 Btu/hr
÷ 60 min/hr
2500

Ron: Where are all the other channels?

Curtis: Please tell Tia 2 of the RH are down.