



Continuous Auditing

The Continuous Audit of Online Systems¹

Miklos A. Vasarhelyi, Fern B. Halper,

Article information:

To cite this document: Miklos A. Vasarhelyi, Fern B. Halper, "The Continuous Audit of Online Systems¹" *In* Continuous Auditing. Published online: 12 Mar 2018; 87-104.

Permanent link to this document:

<https://doi.org/10.1108/978-1-78743-413-420181004>

Downloaded on: 13 March 2018, At: 02:28 (PT)

References: this document contains references to 0 other documents.

To copy this document: permissions@emeraldinsight.com

Access to this document was granted through an Emerald subscription provided by emerald-srm:178665 []

For Authors

If you would like to write for this, or any other Emerald publication, then please use our Emerald for Authors service information about how to choose which publication to write for and submission guidelines are available for all. Please visit www.emeraldinsight.com/authors for more information.

About Emerald www.emeraldinsight.com

Emerald is a global publisher linking research and practice to the benefit of society. The company manages a portfolio of more than 290 journals and over 2,350 books and book series volumes, as well as providing an extensive range of online products and additional customer resources and services.

Emerald is both COUNTER 4 and TRANSFER compliant. The organization is a partner of the Committee on Publication Ethics (COPE) and also works with Portico and the LOCKSS initiative for digital archive preservation.

*Related content and download information correct at time of download.

The Continuous Audit of Online Systems¹

Miklos A. Vasarhelyi and Fern B. Halper

Miklos A. Vasarhelyi is at AT&T Bell Laboratories, Murray Hill, N.J. 07974 and Rutgers University, Newark, N.J. 07102, and Fern B. Halper is at AT&T Bell Laboratories, Murray Hill, N.J. 07974.

Abstract

The evolution of MIS technology has affected traditional auditing and created a new set of audit issues. This paper describes the Continuous Process Auditing System (CPAS) developed at AT&T Bell Laboratories for the internal audit organization that is designed to deal with the problems of auditing large paperless real-time systems. The paper discusses why the methodology is important and contrasts it with the traditional audit approach. CPAS is designed to measure and monitor large systems, drawing key metrics and analytics into a workstation environment. The data are displayed in an interactive mode, providing auditors with a work platform to examine extracted data and prepare auditing reports. CPAS monitors key operational analytics, compares these with standards, and calls the auditor's attention to any problems that may exist. Ultimately, this technology will utilize system probes that will monitor the auditee system and intervene when needed.

The authors thank the two anonymous reviewers for their constructive comments, the editor for his thorough review of the manuscript, and Stanley Halper (Audit Committee Support Network) for his insightful comments. We also thank the participants of research seminars at Columbia University, Rutgers University, the University of Kansas, the University of Nebraska, and Boston University and the attendees of the EDPAA, IIA, and AICPA professional meetings for their comments and suggestions.

¹From *AUDITING: A Journal of Practice and Theory* 10(1), 110–125. Reprinted by permission of American Accounting Association.

We are particularly indebted to Sam Parker. Chris Calabrese. Tsyh-Wen Pao. John Snively. Andrew Sherman. Kazuo Ezawa. Ed Raefski, David Pollak, Ron Fritz, Joe Riley, and David Lillian for their work on the prototype system.

Introduction

This paper develops the concept and explores key issues in an alternate audit approach called the Continuous Process Audit Methodology (CPAM). The paper focuses on an implementation of this methodology, the Continuous Process Audit System (CPAS), developed at AT&T Bell Laboratories for the AT&T internal audit organization. Features of the methodology may also be applicable to external audit functions.

The paper is divided into three sections. In the remainder of the Introduction, changes in management information systems (MIS) that affect traditional auditing are discussed. In the second section, CPAM and CPAS are described and contrasted with the traditional audit approach. The audit implications related to the introduction of a CPAS-like technology also are examined. The last section discusses some of the knowledge issues involved in the implementation of a CPAS application and suggests paths for future work.

Technology and the Auditor

Traditional auditing (both internal and external) has changed considerably in recent years, primarily as a result of changes in the data processing environment [Roussey, 1986; Elliott, 1986; Vasarhelyi and Lin, 1988; Bailey et al., 1989]. These changes have created major challenges in performing the verification and attestation functions². These changes and the resulting technical challenges created for auditors are summarized in Table 1.

For example, the introduction of technology precluded auditors from directly reading data from its source (magnetic tape) and, unlike paper and indelible ink, this source could be modified without leaving a trace (phases 1 and 2 in Table 1); the advent of time sharing and data communications has allowed continuous access to data from many locations (phase 3) creating access exposures; database systems have added more complexity to auditing due to the lack of obvious mapping between the physical and logical organization of data (phase 4).

Auditors dealt with these changes, for example, by (1) tailoring computer programs to do traditional audit functions such as footing, cross-tabulations, and

²Both external and internal audits have attestation objectives. The external auditor must attest to the validity of financial statements. The internal auditor provides the primary appraisal function of company operations, including the controls over the company EDP systems. After testing and evaluation, the internal auditor can attest to the system of internal accounting controls [Halper et al., 1985].

Table 1: The Evolution of Auditing from a Data Processing Perspective.

Phase	Period	Data Processing of Functions	Applications	Audit Problem
1	1945-55	Input (I) Output (O) Processing (P)	Scientific & Military applications	Data transcription Repetitive processing
2	1955-65	I, O, P Storage (S)	Magnetic tapes Natural applications	Data not visually readable Data that may be changed without traces
3	1965-75	I, O, P, S Communication (C)	Time-sharing systems Disk storage Expanded Operations support	Access to data without physical access
4	1975-85	I, O, P, S, C Databases (D)	Integrated databases Decision Support Systems (decision aides) Across-area applications	Different physical and logical data layouts New- complexity layer (DBMS) Decisions impounded into software
5	1986-91	I, O, P, S, C, D Workstations (W)	Networks Decision support systems (non-expert) Mass optical storage	Data distributed among sites Large quantities of data Distributed processing entities Paperless data sources Interconnected systems
6	1991-on	I, O, P, S, C, D, W Decisions (De)	Decision support systems (expert)	Stochastic decisions impounded into MIS

confirmations; (2) developing generalized audit software to access information on data files; (3) requiring many security steps to limit logical access in multi-location data processing environments; and (4) developing specialized audit computers and/or front-end software to face the challenge of database-oriented systems.

However, MIS continue to advance in design and technology. Corporate MIS, and particularly financial systems, are evolving towards decentralization, distribution, online posting, continuous (or at least daily) closing of the books, and paperlessness [Vasarhelyi and Yang, 1988]. These changes are causing additional challenges for auditors and provide opportunities for further evolution in audit tooling and methodology. The current systems environment and new audit challenges in this environment are described in the next section.

Current Environment for Large Applications

Many large applications today typically use one type of Database Management System (DBMS) (e.g., IBM's IMS) spread among several databases that relate to different modules of a system. Data may be kept in several copies of the database with identical logical structure and may be processed at the same location and/or in many different locations. These systems typically can support both online and batch data processing and are linked to a large set of related feeders acting in asynchronous pat-terms feeding transactions and receiving adjustments and responses from the main system. Additionally, the main system can be the information base for downstream systems supporting management decisions and operations.

This system may store a related family of databases including the master database, a transaction database, a pending transaction database, a control database, and an administrative database. The DBMS typically will have its own software for resource accounting and restart and recovery facilities, a query language, a communication interface, a data dictionary, and a large number of utility packages. In many corporations, system software consists of different systems with a large majority of the systems still operating in mainframe computers, programmed in traditional programming languages, and interfacing primarily with mainframe-based databases.

System hardware is a mix of different technologies with bridges among different standard environments, including microcomputers acting as feeders and analysis stations, large mainframes, a large number of telecommunication interfaces, medium-size system buffers, and large data storage devices.

The corporate system is generally developed application by application, often at different sites. Copies of system modules may be distributed to different data processing sites, and version control plays a very important role in the consistent processing of an application. Application data typically come from both the operating entities (branches) and from headquarters. Data can be transmitted in a burst mode (accumulated by or for batch processing) or in an intensive flow (where data are entered when a transaction is measured and not accumulated for transmission) for online or close-to-online processing [Fox and Zappert, 1985]. Perhaps most importantly, many of these systems are real-time systems, meaning that they receive and process transactions continuously.

Auditing these systems requires both the audit of the system itself and the examination and reconciliation of the interfaces between systems. These interfaces and the error correction and overhead allocation loops pose additional problems to system audits. Table 2 displays some of the characteristics of database systems and two audit techniques (labeled level 1 and level 2) that can be used to evaluate and measure these systems.

Audit work on these systems originally was constrained by strong dependence on auditee system staff (for the extraction of data from databases) and typically entailed reviewing the manual processes around the large application system. In traditional system audits, these procedures were labeled as "auditing around the computer." These procedures are labeled as "level 1" in Table 2 and are characterized

Table 2: Database Systems and their Audit.

System Characteristic	Audit (level 1)	Audit (level 2)
Database	Documentation	Data dictionary query
Database size	User query	Auditor query
Transaction flows	Examine levels	Capture sample transactions
Duplicates	Sorting and listing	Logical analysis and indexes
Field analysis	Paper oriented	Software based
Security issues	Physical	Access hierarchies
Restart & Recovery	Plan analysis	Direct access
Database interfaces	Reconciliation	Reconciliation and transaction follow-through

by examination of documentation, requests for user-query of the database, examination of application summary data, sorting and listing of records by the user (not the auditor), a strong emphasis on paper, physical evaluation of security issues, plan analysis for the evaluation of restart and recovery, and manual reconciliation of data to evaluate application interfaces. Level 2 tasks, listed in Table 2, would use the computer to perform database audits as well as eliminate the intermediation by the user or systems people (auditees) in the audit of database systems¹. This hands-on approach utilizes queries to the data dictionary, involves direct use of the system by the auditor, and relies on transaction evidence gathered by the auditor using the same database technology. The level 2 approach reduces the risk of fraudulent (selective) data extraction by the auditee and allows the audit to be conducted more efficiently if the auditor is well versed in database management. Furthermore, audit effectiveness is increased because the auditor has greater flexibility in the search for evidence and it is not obvious to the auditee what data are being queried by the auditor (resulting in improved deterrence of fraud). Differences in desired audit approach and the resultant technological tooling necessary for performing level 2 tasks led to the development of some of the concepts used for continuous process auditing.

Continuous Process Auditing

There are some key problems in auditing large database systems that traditional auditing (level 1) cannot fully solve. For example, given that traditional audits are performed only once a year, audit data may be gathered long after economic events

¹At level 1, the auditor (internal or external) must obtain data through the services of a database user who has access to the DBMS. Such intermediation has been proven to be dangerous in several cases (e.g., equity funding).

are recorded. This often is too late to prevent economic loss by the audited entity. Traditionally, the attestation function has not been relevant in the prevention/detection of loss on a timely basis. However, internal auditors have increasingly been asked to assume a much more proactive role in loss prevention. Another problem is that auditors typically receive only a “snapshot” of a system via several days of data supplied by the auditee. Unless these data coincide with some sort of problem in the system, the data may not be a good indication of system integrity. Evaluating the controls over real-time systems requires evaluating the controls at many points in time, which is virtually impossible after the fact, even if a detailed paper transaction trail exists. Surprise audits seldom are effective in this kind of environment and compliance is difficult to measure because major and obtrusive preparation is necessary in the “around-the-computer” audit of systems.

In continuous process auditing, data flowing through the system are monitored and analyzed continuously (e.g., daily) using a set of auditor defined rules. Exceptions to these rules will trigger alarms which are intended to call the auditor’s attention to any deterioration or anomalies in the system. Continuous process auditing amounts to an analytical review technique since constantly analyzing a system allows the auditor to improve the focus and scope of the audit. Furthermore, continuous process auditing can be considered as a meta form of control and can be used in monitoring control (compliance) either directly, by looking for electronic signatures, or indirectly by scanning for the occurrence of certain patterns or specific events².

Ultimately, if a system is monitored over time using a set of auditor heuristics, the audit can rely mainly on exception reporting and the auditor is called in only when exceptions arise³.

Impounding the auditor function into the system means that tests which would normally be performed once a year are repeated daily.

This methodology (CPAM) will change the nature of evidence, the timing and types of procedures, and the allocation of effort involved in audit work. The auditor will place an increased level of reliance on the evaluation of flow data (while accounting operations are being performed) instead of evidence from level data (e.g., level of inventory, receivables) and from related activities (e.g., internal audit’s

²The accounting literature has suggested other forms for supplementing traditional control techniques by creating a formal methodology of internal control representation and analysis [Bailey et al., 1985; Bailey et al., 1986] or by using the entity-relationship approach [McCarthy 1979, 1982] to represent accounting events.

³Many different types of auditor heuristics can be “wired” into the CPAS system. The examples below are representative, but generic in nature.

- => If there is an increase in telephone traffic of 20 percent over a period, then expect an increase of similar size in billing 20 days from that day. Investigate if there is more than five percent variance.
- => If total number of errors dropped by edits is larger than two percent of the transactions processed, then investigate error type breakdown.
- => If there is a sudden large drop (in a day) of the size of the retained error file, then an error correction audit is necessary.

preparedness reviews). Audit work will be focused on audit by exception with the system gathering exceptions on a continuous basis.

The continuous process audit scenario entails major changes in software, hardware, the control environment, management behavior, and auditor behavior, and its implementation requires a careful and progressive approach. The next subsection discusses some of the key concepts in the actual implementation of the approach, using a prototype software system.

Key Concepts

The placement of software probes into large operational systems for monitoring purposes may imply an obtrusive intrusion on the system and can result in performance deterioration. The installation of these monitoring devices must be planned to coincide with natural life-cycle changes of major software systems. Some interim measures should be implemented to prepare for full-fledged online monitoring. The approach adopted at AT&T, with the current CPAS prototype, consists of a data provisioning module that feeds an independent and external advanced decision support system.

Data provisioning can be accomplished by three different, though not necessarily mutually exclusive, methods: (1) data extraction from "standard" (existing) application reports, using pattern matching techniques; (2) data extraction from the file that feeds the application report; and (3) recording of direct monitoring data. The approach actually used in CPAS entails first a measurement phase without intrusion and minor system overhead, evolving in the future to a monitoring phase where intrusion is necessary but the audit capability is substantially expanded⁴.

Measurement. Copies of key management reports are issued and transported through a data network to an independent audit work-station at a central location⁵. These reports are stored in raw form and data are extracted from these reports and placed in a database. The fields in the database map with a symbolic algebraic representation of the system that is used to define the analysis. The database is tied to a workstation, and analysis is performed at the workstation using the information obtained from the database. The basic elements of this analysis process are described later in the paper.

Monitoring. In the monitoring phase, audit modules will be impounded into the auditee system. This will allow the auditor to continuously monitor the system and

⁴It is envisaged that, in the monitoring phase, intrusion and system overhead may be limited by utilizing database backup and recovery traces as the main source of transaction data, dumping a copy of these traces onto a local workstation, loading the workstation with some expert software, and having it as a local interchange device.

⁵Large application systems have many user reports generated for the operational purposes of different organizational entities. For example, marketing has sales reports that report total sales activity and also break down activity by region; production has inventory reports to gauge production needs; finance has payments and receivables reports for their cash flow administration purposes. Most likely, these existing reports will contain all information necessary for continuous auditing purposes.

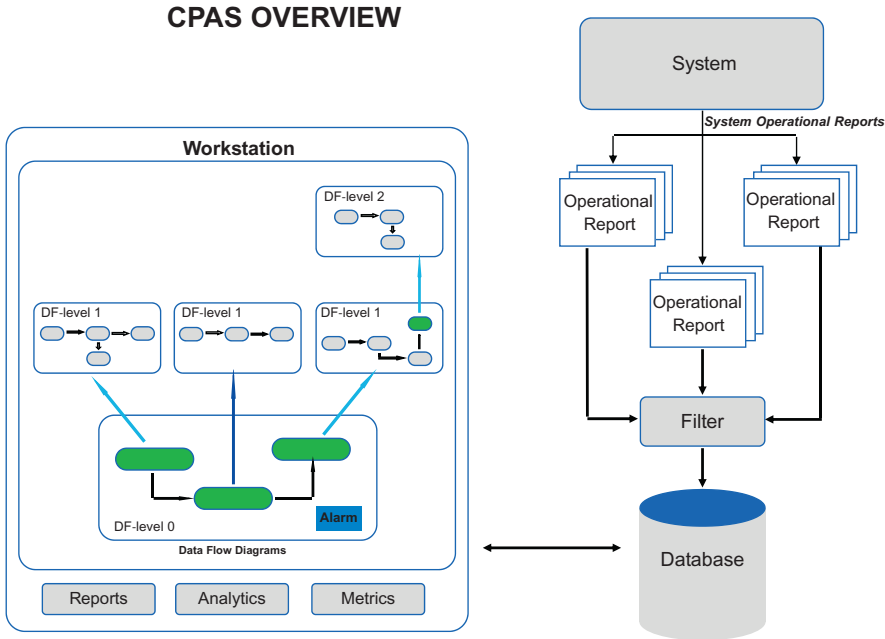


Figure 1: CPAS Overview.

provide sufficient control and monitoring points for management to be able to retrace any transactions. In current systems, individual transactions are aggregated into account balances and complemented by successive allocations of overhead. These processes create difficulties in balancing and tracing transactions.

The AT&T CPAS prototype uses the “measurement” strategy of data provisioning. This is illustrated in Figure 1. The auditor logs into CPAS and selects the system to be audited. The front end of CPAS allows the auditor to look at copies of actual reports used as the source of data for the analysis. From here the auditor can move into the actual analysis portion of CPAS. In CPAS, the system being audited is represented as flowcharts on the workstation monitor. A high-level view of the system (labeled DF level 0 in Figure 1) is linked hierarchically to other flowcharts presenting more detail about the system modules being audited. This tree oriented view-of-the-world which allows the user to drill deeper into the details of a graphical representation is conceptually similar to the Hypertext approach [Gessner, 1990]⁶. The analysis is structured along these flowcharts leading the auditor to think hierarchically.

Analysis. The auditor’s work is broken down into two phases: fast, the startup stage where he/she works with developers, users, and others to create a view of the

⁶The Hypertext approach is not new, being traceable to the 1960s work of Ted Nelson. It currently is quite popular due to its implementation in personal computers, its affinity to object-oriented thinking, and many implementations in both commercial and public domains.

system, and second, the use stage when he/she actually uses the system for actual operational audit purposes. The auditor's (internal or external) role in this context is not very different from its traditional function.

At the setup stage, the auditor acts as an internal control identifier, representer, and evaluator using existing documentation and human knowledge to create the system screens (similar to flowcharts) and to provide feedback to the designers/management. Here, audit tests, such as files to be footed and extended or reconciliations to be performed, as well as processes to be verified, are identified. Unlike the traditional audit process, the CPAS approach here requires the "soft-coding" of these processes for continuous repetition. Furthermore, at this stage, the CPAS database is designed and, unlike in the traditional process, standards are specified and alarm conditions designed.

In the use stage, the system is monitored for alarm conditions and these alarm conditions are investigated when they arise and the symptoms and diagnostics identified and impounded into the CPAS knowledge base. The current baseline version of CPAS provides auditors with some alarms for imbalance conditions, the ability to record and display time-series data on key variables, and a series of graphs that present event decomposition.

This logical view of the system can be associated with diagnostic analytics that count the number of exceptions and/or alarms current in the system. Detailed information about each main module is available at lower levels through a drill-down procedure. This information is presented primarily as metrics, analytics, and alarms.

Metrics. Metrics are direct measurements of the system, drawn directly from reports. These metrics are compared to system standards. If a standard is exceeded, an alarm appears on the screen. For example, in the audit of a billing system, the number of bills to be issued is extracted from a user report. The number of bills not issued due to a high severity error in the data, detected by the normal data processing edits, is captured, as well as the total dollar amount of bills issued. These three numbers are metrics that relate to the overall billing process.

Analytics. Analytics are defined as functional (natural flow), logical (key interaction), and empirical (e.g., it has been observed that ...) relationships among metrics. Specific analytics related to a particular system module can be derived from the auditor, management, user experience, or historical data from the system. Each analytic may have a minimum of three dimensions: (1) its algebraic structure; (2) the relationships and contingencies that determine its numeric value at different times and situations; and (3) rules-of-thumb or optimal rules on the magnitude and nature of variance that may be deemed as "real variance," potentially triggering an alarm. For example, a billing analytic would state that dollars billed should be equal to invoice amounts received, minus values of failed edits, plus (or minus) the dollars billed that were not issued due to quality control factors. The threshold number of expected invoices for that particular day or week (allowing for seasonality) must be established to determine whether an alarm should be fired.

Alarms. An alarm is an attention-directing action triggered, for example, when the value of a metric exceeds a standard. Actual experience with these issues indicates

that several levels of alarms are desirable: (1) minor (type 1) alarms dealing with the functioning of the auditing system; (2) low-level operational (type 2) alarms to call exceptions to the attention of operating management; (3) higher-level (type 3) alarms to call exceptions to the attention of the auditor and trigger “exception audits;” and (4) high-level (type 4) alarms to warn auditors and top management of serious crisis.

For example, a type 1 alarm may be triggered if two sets of data are produced by the audited system, for the same module, for the same day, and it is unclear from the information given which data to load into the database. Of course, cycle and re-run information should be clearly passed along with the data, but sometimes this will not be as clean as expected. A type 1 alarm might also be triggered if the reports change format and data extraction procedures need to be modified. These type 1 alarms will need to be acted upon immediately, usually with a call to the system administrator or system management organization.

A type 2 alarm might be triggered if data pertaining to the same process are inconsistent. For example, data from many different reports might be used to perform an intra-module reconciliation. The data must come from different jobs in order for the reconciliation to be meaningful. A well designed CPAS application will try to gather data from different jobs and compute the same reconciliation in more than one way. If the value for the same variable (for the same run, etc.) is inconsistent between reports, this indicates a problem either with the system or the system reports and should be investigated immediately to determine how severe it is.

A type 3 alarm might be triggered if an error or suspense file is getting too large, or if some other threshold is exceeded. These exceptions are cause for concern and should be investigated because they may pose a danger to the company if not corrected.

A type 4 alarm is the most severe. For example, if, at the time of billing, many customers can not be accounted for, or if all customers were billed the same amount, regardless of how much they used a particular service, or if it appears that duplicate paychecks were sent to employees, the system should be shut down and promptly corrected.

The data and experience needed to understand the phenomena being measured to the level of specification of alarm standards are probably not available in most organizations. Experience with a CPAS-like system aids in their development.

Software Implementation

Figure 2 was prepared using CPAS and has the look-and-feel of any CPAS application⁷. It shows a high-level view of a hypothetical billing system. This billing system processes transactions from multiple locations, bills customers around the country,

⁷The CPAS software was implemented under a NeWS windowing system and on a SUN workstation. The entire software was constructed using standard UNIX tools with a minimum of low-level programming. A commercially available relational database was used in the delivery device. The concept, however, can be extended and can be implemented piece by piece using standard PC tools. Conceivably, the methodology can be implemented in many different ways, from a pure PC implementation to a full-fledged distributed computing solution with the “audit computer” as the self-contained destination of monitoring/measurement data.

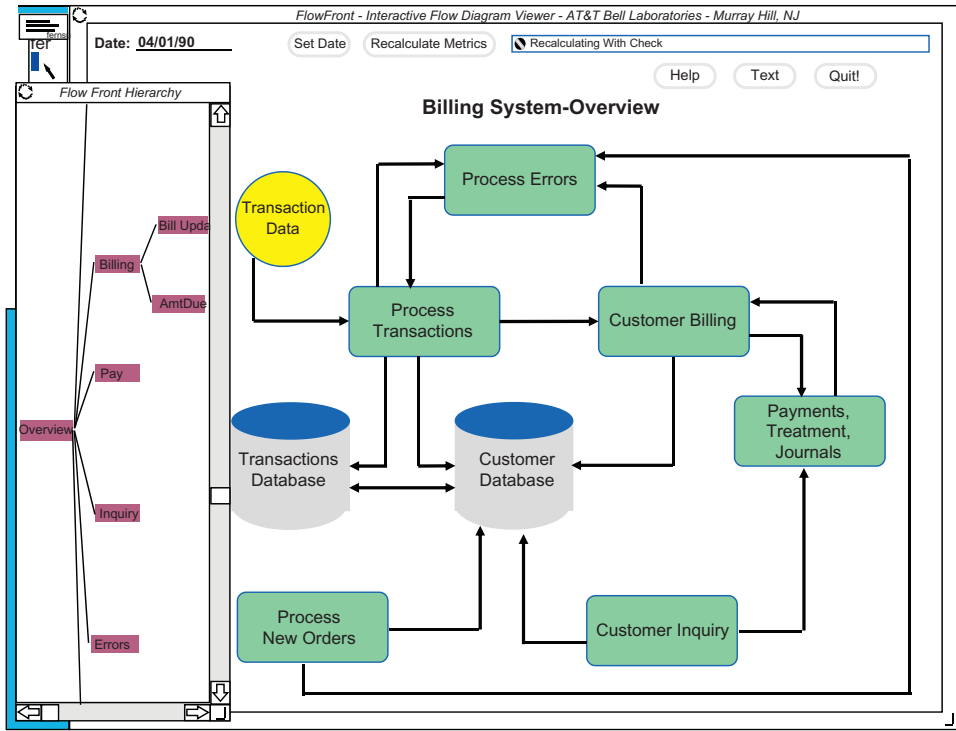


Figure 2: Billing System Overview.

and performs multiple bill pulls a month (i.e., not all customers are billed on the same day). The hierarchy window on the left in the figure indicates what part of the billing system is represented by the flowchart. In this example, the flowchart represents the base node of the billing system hierarchy, i.e., an overview of the system. The auditor can use the hierarchy window to move to any flowchart in CPAS by simply selecting the desired node.

As can be seen in Figure 2, the billing system consists of six major modules. Billing data first enters the Process Transaction module where high-level edits are performed. Any errors from this process are sent to the Error Processing module. Corrected errors are sent back through the front-end of the system. Transactions that successfully pass through the front-end are sent to the Billing module where customer accounts are extracted, amounts due are calculated, and the bills are produced. Errors from this process are sent to the Error Processing module. Billing information is sent to the Journals function where payment and treatment information is processed and the customer database is updated. The system also contains a module that deals with any questions a customer may have about his/ her account and a module that processes new orders for service. The date displayed in the

figure indicates the date that the analysis uses as the base date⁸. In the example presented here, the base date (4/1/90) is also assumed to be the current date.

The CPAS application may be testing that the following controls are in place: (1) completeness and accuracy of input; (2) completeness and accuracy of update; (3) timeliness of data arriving to the system and timeliness of system processing; (4) maintenance of data in the database; (5) accuracy of computer programs; and (6) reasonableness of the data. For example, the auditor might have defined tests (and had them built into the CPAS application) to answer the following questions:

- Were all transactions sent to the biller, received? Can all of the transactions be accounted for? Were all of the transactions loaded into the Process Transactions module? Were they loaded correctly?
- How many transactions were in error? Has the error threshold been exceeded? How long does it take errored transactions to re-enter the system?
- Were all transactions posted to the database correctly? Were all the transactions initiated, executed, and recorded only once? Can all of the transactions that entered the system be accounted for (i.e., either on the database or in an error file, or rejected back to the source)? How accurate are the data that were loaded to the database (i.e., does the sum of the dollars on the database match what was to be posted to the database)? Are all databases synchronized?
- Were the bills calculated properly? How reasonable are the amounts billed? Were all customers who were supposed to be billed actually billed?

The alarm report displayed in [Figure 3](#) states that there are three alarm conditions outstanding in the system on 4/1/90. Two of these are type 3 alarms, and one is a type 4 alarm. The report also shows the module where the error occurred, the value that caused the error, the standard that the value was compared against, and the average value of the error (computed for a 30-day period). The most severe alarm is, of course, the type 4 alarm. Here, ten accounts that should have been billed were not billed. This indicates a break-down in the system and should be dealt with immediately.

The two type 3 alarms indicate that a threshold was exceeded. In this case, 2,000 transactions out of 10,000 transactions processed on 4/1/90 were in error and sent to the Error Processing module. The alarm report indicates that the standard was 850 errors per processing day. The large amount of errors also resulted in triggering the second type 3 alarm, because the dollar value associated with these errors caused the dollar value of the error file to exceed the threshold (here \$200,000). The auditor should investigate this to find out the cause for the large number of errors. The auditor also should follow up to determine whether these errors are being corrected. If the errors are not being corrected in a timely manner, it may indicate that the

⁸In order to be able to focus on a discrete period and obtain the desired level of detail, the auditor chooses a base date to which all figures will relate. For example, if the auditor identifies a major drop in the outstanding errors file on March 18 (today being June 15), he/she will choose, say March 30, as a base date and perform weekly data analyses for that base date.

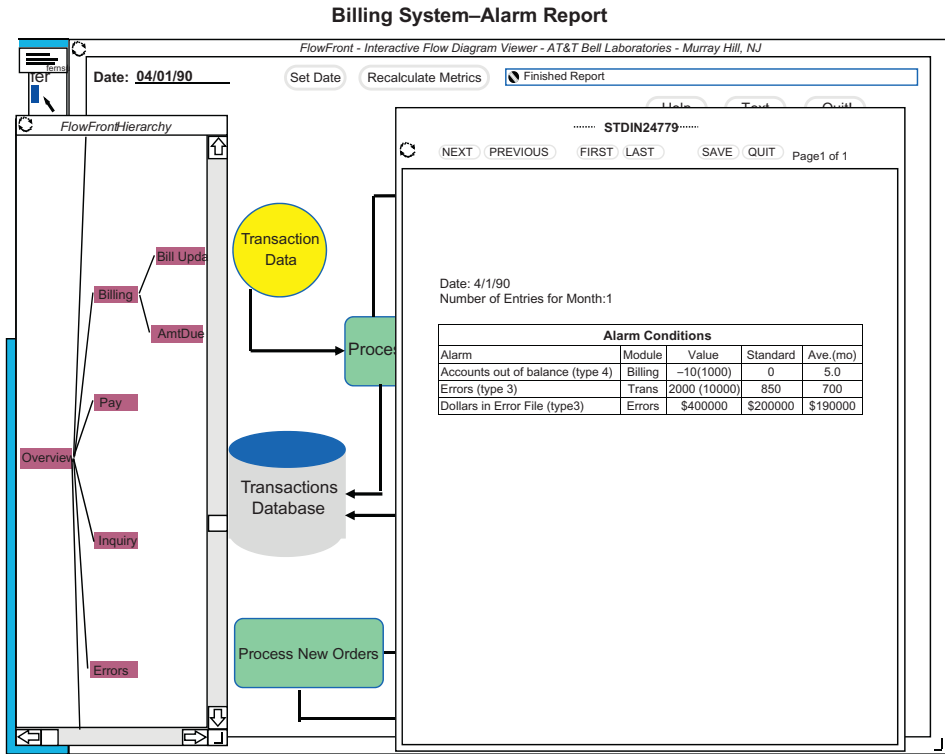


Figure 3: Billing System—Alarm Report.

system cannot deal with certain kinds of data or that there is a staffing problem at the error investigation unit. Additionally, the auditor (if he/she is not familiar with the history of the size of the error file) may want to change the base date to investigate whether this has been a problem in the past.

The auditor may wish to look at the Customer Billing module in more detail to gather more information about the out-of-balance condition before alerting management. The auditor would select the Billing node in the hierarchy window. A new flowchart representing the Customer Billing module would appear on the workstation monitor. This is illustrated in Figure 4. Here, the metrics, indicated as boxes next to the flow chart, show the flow of accounts through the Customer Billing module on 4/1/90⁹. The alarm (found on the lower left of the figure) indicates that

⁹At this, or any level of the system, the auditor can choose to look at alternate metric dimensions (e.g., transactions, records), if appropriate. Additionally, if multiple copies of the software exist in different locations, the auditor can choose the level of aggregation he or she is interested in (e.g., data from the specific data center, from a specific branch, or for the entire entity). These metrics are used to perform a reconciliation and different modules would have different metrics associated with them.

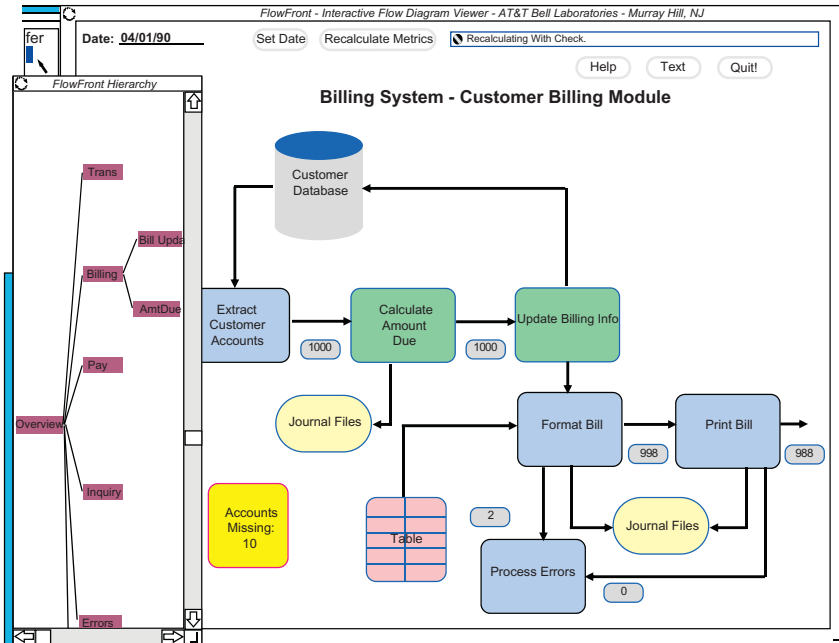


Figure 4: Billing System—Customer Billing Module.

there were ten accounts lost in the process, but more importantly, it illustrates that the loss occurred between the Format Bill module and the Print Bill module.

The auditor may wish to look at the history of the reconciliation. Figure 5 is a two-level time-series showing the number of accounts lost and the total number of accounts billed for a three-week period ending 4/1/90. The graph indicates that the out-of-balance condition occurred once at the beginning of the period and again on 4/1/90. The condition appeared to have been corrected at the beginning of the period, since the reconciliation did not fail again until the current day's processing. The auditor should reset the date to 3/13/90 and check the metrics to determine if the reconciliation failed for the same reason that it did on 4/1/90. This could indicate inadequacy of controls or poor compliance with internal controls. More detailed analytics and metrics relating to the actual billing process and the interface between this module and other modules in the system are found at different levels. This information, taken together, presents an integrated diagnostic view of the system being audited¹⁰.

Complementing the actual hands-on audit work is an auditor platform, accessible at any level, which can include a series of different functions. This platform should

¹⁰“Text,” explaining the flowcharts and “Help,” explaining how to use the system, are available at each level. The auditor can print out screens, reports, or graphs at any time for writing his/her audit reports.

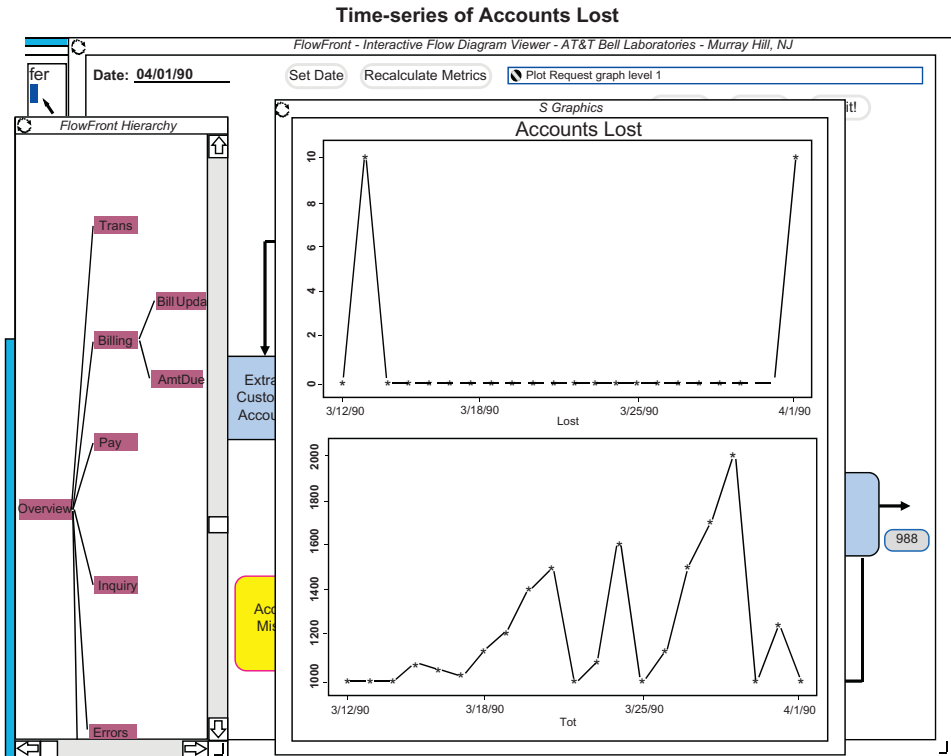


Figure 5: Time-Series of Accounts Lost.

ultimately contain at least a statistical package, a graphics package, a spreadsheet package (including a filter to the database), a report generator, and a text editor. These tools can be used for ad hoc analysis or be linked to the “wired-in” procedures in CPAS. An even richer technological environment may incorporate specific audit document preparation tools that use high technology hardware to read and interpret printed materials [Kahan et al., 1986], and large amounts of information can be stored and accessed directly using optical disk (WORM) technology¹¹.

Discussion

The set of analytics and heuristics used in CPAS will ultimately include a wide variety of algorithms ranging from flow-based rules to expert algorithms developed

¹¹Many firms (e.g., Imnet Corporation and Teletrak Advanced Technologies Systems Inc.) are developing document image technology to access large optical data storage devices.

using techniques in knowledge engineering¹². [13] These algorithms will be used both in the auditor platform, as analytical supplements, as well as impounded into software probes in the monitoring stage. Audit knowledge is needed to supplement the simple comprehension of the system being audited and to deal with the very complex stage of data gathering, analysis, and knowledge organization [Buchanan and Shortliffe, 1984] necessary for programming the auditing probes.

The CPAS prototype was tested on two very large financial systems and is currently being applied to a third. The first application of the CPAS technology was an evolving system whose features changed rapidly. The idea was to put a prototype in place that contained basic analytics and then work with the auditors, as they used CPAS, to build more expertise into the system. It was found that only a few heuristics really existed, perhaps because of the nature of tools available to the auditor or because of the lack of longevity of auditors on the job. With the use of CPAS, auditors started to suggest heuristics that previously required cumbersome or not economically feasible audit procedures (e.g., time series tracking of discrepancies in a particular reconciliation). Another explanation for the limited number of heuristics identified is that the problem domain in question tended to be one with “diffuse knowledge” [Halper et al., 1989], where a large set of sources of knowledge were necessary and where knowledge ultimately was captured from a much wider set of experts than originally conceived.

The two early experiences served to point out tools needed and auditors’ reactions. A long-term effort in conjunction with the system standards organization would be of great use in providing the base for establishing a company-wide continuous audit methodology. Substantive research is needed to determine the best approaches to operationalize and standardize the methodology in internal and external audit contexts.

The issue of startup cost to impound the system description into the CPAS platform and the maintenance of the knowledge base became very important. However, the process of knowledge acquisition and recording used under CPAS is not unlike the phases of internal control evaluation and documentation for workpapers. The level of auditor comprehension of the system tends to be deeper under CPAM than in the traditional audit if the auditor (not a system analyst) performs knowledge capture¹³.

In the two original applications, the CPAS approach required a higher audit startup cost than the traditional audit, but the level of audit examination was also deeper and more reliable. The CPAS approach is substantially different from the traditional one and requires balancing of audit evidence and timing of the audit process. Auditors currently are used to budgeting for a particular audit and perform it

¹²Expert systems techniques have been examined by several auditing researchers [see Kelly et al., 1988] as well as implemented on a limited basis dealing with certain tax (tax accruals) and financial accounting issues (e.g., bank loan portfolio estimation) in practice [Messier and Hansen, 1987; Vasarhelyi, 1988].

¹³In the long range, much of this work can be linked to the use of CASE-type tools where the knowledge is captured at design and could be easily transported to, if not directly used, the platform.

as an intense effort. CPAM requires long-term monitoring and reaction to emerging evidence, something that, with limited experience, is difficult to manage. Given this, the issue of resistance to change may arise. This can be handled by the issuance of an audit manual that describes how to audit with CPAS and extensive training and technical support for the auditors.

Ideally, management also has its own version of CPAS, so they are aware when major problems occur in their system. Auditors could browse their own version of CPAS (with independent analytics) on a periodic basis and follow up on any alarm conditions to see what management has done about them.

Future work on CPAS will focus on increasing the quality of auditor work by integrating the auditor platform with the auditor workstation, increasing the use of monitoring probes, improving the quality of the auditor heuristics, and impounding more expertise into the system.

The introduction of real-time systems will require that the auditor be able to attest to the system of internal accounting controls at different points in time. Continuous process auditing can effectively help the auditor to evaluate these controls, but will require substantive changes in the nature of evidence, the types of procedures, the timing, and the allocation of effort in audit work.

References

- Bailey, A. D., G. L. Duke, G. E. Gerlach, C. E. Ko, R. D. Meservy, and A. B. Whinston, "TICOM and the Analysis of Internal Controls," *The Accounting Review* (April 1985), pp. 186–201.
- Bailey, A. D., R. D. Meservy, and P. E. Johnson, "Internal Control Evaluation: A Computational Model of the Review Process," *Auditing: A Journal of Practice & Theory* (Autumn 1986), pp. 44–74.
- Bailey, A. D., K. Hackenbrack, P. De, and J. Dillard, "Artificial Intelligence, Cognitive Science and Computational Modeling in Auditing Research: A Research Approach," *Journal of Information Systems* (Spring 1987), pp. 20–40.
- Bailey, A. D., L. E. Graham, and J. V. Hansen, "Technological Development and EDP," in Abdel-Khalik, A. R. and I. Solomon, eds., *Research Opportunities in Auditing: The Second Decade* (American Accounting Association: Auditing Section, Sarasota, Florida, 1989).
- Biggs, S. F., and T. J. Mock, "An Investigation of Auditor Decision Processes in the Evaluation of Internal Controls and Audit Scope Decisions," *Journal of Accounting Research* (Spring 1983), pp. 234–255.
- Buchanan, B. G., and E. H. Shortliffe, *Rule-Based Expert Systems* (Addison-Wesley Publishing Company, 1984).
- Elliott, R. K., "Auditing in the 1990s: Implications for Education and Research," *California Management Review* (Summer 1986), pp. 89–97.
- Fox, C., and F. Zappert, "Information Systems Evolution in the Near Future" (AT&T Bell Laboratories, Private Communication, December 1985).
- Gessner, R., "Building A Hypertext System," *Dr. Dobb's Journal* (June 1990), pp. 22–33.
- Halper, F. B., J. P. Snively, and M. A. Vasarhelyi, "CPAS-Knowledge Engineering and Representation," Second International Symposium on Expert Systems in Business, Finance, and Accounting (Newport Beach, Ca., November 1989).

- Halper, S. D., G. C. Davis, P. J. O'Neil-Dunne, and P. R. Pfau, *Handbook of EDP Auditing* (Warren, Gorham and Lamont, 1985).
- Hayes-Roth, B., *Implications of Human Pattern Processing for the Design of Artificial Knowledge Systems* (Academic Press, Inc., 1978).
- Kahan, S., T. Pavlidis, and H. S. Baird, "On the Recognition of Printed Characters of any Font Size" (AT&T Bell Laboratories, Private Communication, January 1986).
- Kelly, K., G. Ribar, and J. Willingham, "Interim Report on the Development of an Expert System for the Auditor's Loan Loss Evaluation," in M. A. Vasarhelyi, ed., *Artificial Intelligence in Accounting and Auditing* (Markus Wiener Publishing Company, 1988).
- McCarthy, W. E., "An Entity-relationship View of Accounting Models," *The Accounting Review* (October 1979), pp. 667-686.
- McCarthy, W. E., "The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment," *The Accounting Review* (July 1982), pp. 554-578.
- Messier, W. F., Jr., and J. V. Hansen, "Expert Systems in Auditing," *Auditing: A Journal of Practice & Theory* (Autumn 1987), pp. 94-105.
- Rousseau, R., "The CPA in the Information Age: Today and Tomorrow," *Journal of Accountancy* (October 1986), pp. 94-107.
- Schank, R. G., and R. P. Abelson, *Scripts Plans and Understanding* (Lawrence Erlbaum Associates Publishers, 1977).
- Shaw, A. N., and H. A. Simon, "Elements of a Theory of Human Problem Solving," *Psychology Review* (Vol. 65, No. 3, 1958), pp. 151-166.
- Shimura, M., and F. H. George, "Rule-Oriented Methods in Problem Solving," *Artificial Intelligence* (Vol. 4, 1973), pp. 203-223.
- Simon, H., "The Structure of Ill Structured Problems," *Artificial Intelligence* (Vol. 4, 1973), pp. 181-201.
- Simon, H., "Information Processing Models of Cognition," *Annual Review Psychology* (Vol. 30, 1979), pp. 363-396.
- Vasarhelyi, M. A., "Expert Systems in Accounting and Auditing," in M. A. Vasarhelyi, ed., *Artificial Intelligence in Accounting and Auditing* (Markus Wiener Publishing Company, 1988).
- Vasarhelyi, M. A., and W. T. Lin, *Advanced Auditing* (Addison-Wesley Publishing Company, 1988).
- Vasarhelyi, M. A., and D. C. Yang, "Technological Change and Management Information Systems," Proceedings of the Twenty-First Annual Hawaii International Conference on System Sciences (Hawaii 1988), pp. 191-197.