

*The CPAS / CCM experiences:
prospectives for AI/ES research in Accounting*

Miklos A. Vasarhelyi
Rutgers University,
KPMG Peat Marwick Professor of Accounting
and
Technical Manager Advanced Computing Group,
AT&T Bell Laboratories.

To be presented at the Information Systems Audit, Control and Security
at
Hotel Atrium Hyatt Budapest
on
September 4-7 1996

This paper is preliminary in nature and should not be quoted without explicit consent of the author. I am very appreciative for the contributions of my co-authors in many parts of this effort in particular F. B. Halper, J. Snively, and K. Ezawa. The paper draws heavily in prior published works of this effort.

¹ CPAS stands for Continuous Process Auditing, CCM stands for Continuous Control Monitoring

Introduction

This paper discusses the learnings from the Continuous Process Audit System² (CPAS) and Continuous Control Monitoring (CCM) efforts and extrapolates into areas of potential research in AI & ES in Accounting. First we describe the motivation and key factors in the CPAS efforts, then we show how an effort a la CCM³ can be applied onto the basic framework, in the next section we discuss some technological features and problems, next the prospective approaches for the problems found are discussed, finally the conclusions summarize the discussion and propose some additional routes for future research.

Elements of the CPAS effort

The CPAS project was motivated by a survey of the Internal Audit organization of AT&T that identified large corporate systems as the potentially a very large exposure for the corporation. The CPAS methodology was developed to measure and detect any major problems that may be occurring during the day-to-day operation of large corporate computer systems. The methodology initially focused on very large mainframed corporate legacy systems where more than one copies of a system ran in multiple datacenters around the country. Later developments allowed for the conceptualization of the process in distributed and client-server environments.

Basic 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. Interim measures should be implemented to prepare for online monitoring.

The CPAS effort consisted of a data provisioning system and an advanced decision support system. Data can be gathered from tailored reports (files) from the application, reports from the application, and direct monitoring data. The approach used in CPAS is dual, evolving from a **measurement phase** without intrusion and minor system overhead, to a **monitoring phase** where intrusion is necessary⁴ but audit capability is substantially expanded.

Measurement

Copies of key management reports are issued and transported through a data network to an independent audit workstation at a central location. 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

² Vasarhelyi, M. A. & Halper, F. B., "The Continuous Audit on Online Systems," **Auditing: A Journal of Practice & Theory**, Vol. 10, No. 1, Spring 1991, pp. 11-125.

³ Vasarhelyi, M. A. & Halper, F. B., "**Continous Control Monitoring**," a monograph submitted to the Institute of Internal Auditors, forthcoming, 1996.

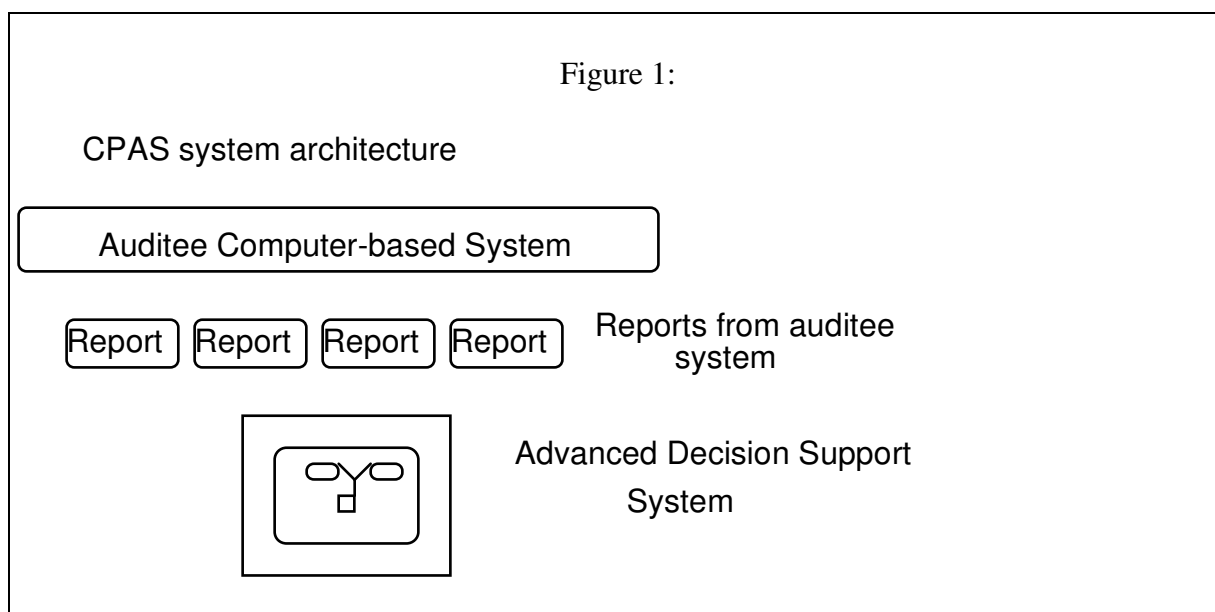
⁴ Intrusion and system overhead may be limited by utilizing database backup and recovery traces as the main source of transaction data, dumping a copy of these traces onto a local workstation, loading the workstation with some expert software and having it as a local interchange device.

with a symbolic algebraic representation of the system that is used to define the analysis. The database is placed on a workstation and analysis is performed at the workstation using the information obtained from the database.

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 provide sufficient control and monitoring points for management retracing of transactions. The level of aggregation and difficulties of balance and transaction tracing that are prevalent in current systems will decrease in the future as processing economies that dictated the limited traceability of transactions will not be needed as systems become more powerful.

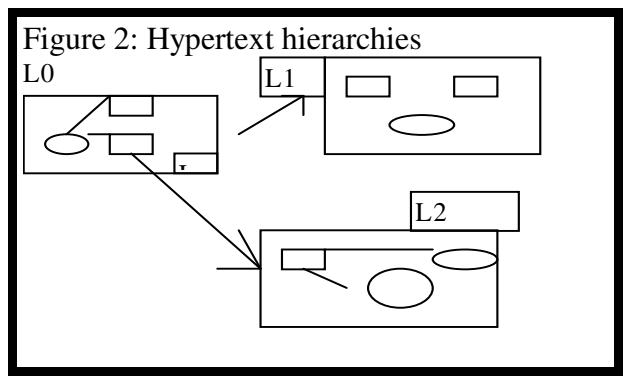
The Continuous Process Audit System (CPAS) used the "measurement" strategy of data procurement. This is illustrated in Figure 1. The auditor logs into CPAS and selects the system to be audited. The front end of CPAS allows the auditor to look at copies of actual reports used as the source of data for 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 (called data flow 0- DF level 0 in Figure 2) is linked hierarchically to other flowcharts representing more detail about the system modules being audited. This tree oriented view-of-the-world which allows the user to drill down into the details of a graphical representation is conceptually similar to the **Hypertext** approach [Gessner, 1990]⁵ ⁶The analysis is structured along these flowcharts leading the auditor to think hierarchically.

⁵ Gessner, R., "Building A Hypertext System," *Dr. Dobb's Journal*, (June 1990), pp.22-33.

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



An integrated view of the system is available at DF level 0. This logical view of the system can be associated to diagnostic analytics that count the number of exceptions and/or alarms current in the system. Detailed information about each main module is available at the lower levels. This type of thinking is similar to "hypertext" conceptualization where symbolic and relational links can be specified across levels.

This information is presented primarily as **metrics** and **analytics**.

Metrics

Metrics are defined as direct measurements of the system, drawn from reports, in the measurement stage. These metrics are compared against system standards. If a standard is exceeded, an alarm appears on the screen. For example, in the auditing of a billing system, the number of bills to be invoiced is extracted from a user report. The number of bills not issued due to a high severity error in the data 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 and Alarms

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:

- its algebraic structure,
- the relationships and contingencies that determine its numeric value at different times and situations and
- rules-of-thumb or optimal rules on the magnitude and nature of variance that may be deemed as "real variance" to the extreme of alarms.

For example, a billing analytic would state that dollars billed should be equal to invoices received, minus values of failed edits plus (or minus) the change of the number of dollars in retained invoices. 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.

Actual experience with these issues indicates that several levels of alarms are desirable:

1. minor alarms dealing with the functioning of the auditing system,

2. low level operational alarms to call to the attention of operating management,
3. higher level alarms to call the attention of the auditor and trigger "exception audits" and
4. high level alarms to warn auditing and top management of serious crisis.

Establishing these alarm thresholds is a second harmonic development. 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 will aid in their development.

In Continuous Process Auditing, data flowing through the system are monitored and analyzed continuously (i.e., daily) using a set of auditor defined rules. System alarms and reports call the auditor's attention to any deterioration or anomalies in the system. Continuous Process Auditing then, is really an analytical review technique since constantly analyzing a system allows the auditor to improve the focus and scope of the audit.

Furthermore, it is also often related to controls as it can be considered as a meta form of control (audit by exception) and can also be used in monitoring control (compliance) either directly, by looking for electronic signatures, or indirectly by scanning for the occurrence of certain events. The accounting literature has suggested other forms of supplementing traditional control techniques by creating a formal methodology of internal control representation and analysis [Bailey et al., 1985⁷; Bailey et al., 1986⁸] or by using the entity-relationship approach [McCarthy 1979⁹, 1982¹⁰] The technology used in the CPAS effort is described by Vasarhelyi et al.¹¹

Auditor and knowledge issues

The set of analytics and heuristics used in CPAS included a wide variety of algorithms ranging from flow-based rules to expert algorithms drawn using techniques in knowledge engineering. These algorithms will be used both in the auditor platform, as analytical supplements, as well as impounded into software probes in the monitoring stage.

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

⁷ 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.

⁸ Bailey, A. D and R. D. Meservy, P. E. Johnson, "Internal Control Evaluation: A Computational Model of the Review Process," *Auditing: A Journal of Practice and Theory*, (Autumn 1986), pp. 44-74.

⁹ McCarthy, W. E., "An Entity-relationship View of Accounting Models," *The Accounting Review*, (October 1979), pp. 667-86.

¹⁰ 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

¹¹ Vasarhelyi M. A., F. B. Halper, and K. E. Ezawa, "The Continuous Process Audit System: A UNIX -Based Auditing Tool". *The EDP Auditor Journal*, 1991, vol. 3, PP 85-91.

Messier, 1987¹²; Vasarhelyi, 1988¹³]. 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, 1984¹⁴] necessary for programming the auditing probes.

The CPAS prototype was tested on two very large financial systems. 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. The audit knowledge elicitation process focused in three areas: archival recording, heuristic discovery, and methodological development.

Archival Recording:

Interviews with auditors and examination of working papers and audit reports for identification of current audit steps, items of data being examined, specific rules concerning required audit evidence; and any actual procedures of data gathering, search and analysis. This process is analogous to the work that tries to establish descriptive models of auditor behavior. For example "think aloud" techniques [Biggs and Mock, 1983¹⁵] provide some insight on the auditor's thought processes.

Heuristic Discovery:

Application of knowledge engineering techniques to identify non-formulated rules, desired tooling, types of inference, methods of fuzzy set resolution, etc. (Shimura and George, 1973¹⁶; Shank and Abelson, 1977¹⁷; Hayes-Roth, 1978¹⁸)

Methodological Development:

Working with auditors to further develop the "Continuous Process Audit" methodology, monitoring the usage of the auditor workstation in the measurement phase, and impounding more audit expertise into the audited system. [Shaw and Simon, 1958¹⁹; Simon 1973²⁰, 1979²¹)

¹² Hansen, J. V. and W. F. Messier Jr., "Expert Systems in Auditing," **Auditing: A Journal of Theory and Practice**, (Autumn 1987), pp. 94-105.

¹³ Vasarhelyi, M. A., "Expert Systems in Accounting and Auditing," **Artificial Intelligence in Accounting and Auditing**, (Markus Wiener Publishing Company, 1988), Vols. 2 & 3, 1994..

¹⁴ Buchanan, B. G. and E. H. Shortliffe, **Rule-Based Expert Systems**, (Addison-Wesley Publishing Company, 1984).

¹⁵ 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.

¹⁶ Shimura, M and F. H. George, "Rule-Oriented Methods in Problem Solving", **Artificial Intelligence**, (Vol.4 1973) pp.203-223.

¹⁷ Shank, R. G. & Abelson, R. P., **Scripts Plans & Understanding**, (Lawrence Erlbaum Associates Publishers, 1977).

¹⁸ Hayes-Roth, B., "Implications of Human Pattern Processing for the Design of Artificial Knowledge Systems", (Academic Press, Inc., 1978).

¹⁹ 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.

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 knowledge was ultimately captured from a much wider set of experts than originally conceived. 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 that an auditor has to perform. The level of auditor comprehension of the system tends to be deeper under this approach if the auditor (not a system analyst) is to perform knowledge capture.²²

Consequently, the CPAS approach probably requires a higher audit startup cost than the traditional audit but the level of audit examination is also consequently 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. 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 of the auditors in the engagement. to represent accounting events.

Ultimately, if a system is monitored over time using a set of auditor heuristics, the audit can rely purely on exception reporting and the auditor is called in only when exceptions arise. Impounding auditor knowledge into the system means that tests that would normally be performed once a year are repeated daily. This methodology will change the nature of evidence, timing, procedures and 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 related activities (e.g. preparedness audits). Audit work would be focused **on audit by exception** with the system gathering knowledge exceptions on a continuous basis.

Elements of the CCM effort

Levels of Monitoring

While auditing is a form of ex-post-facto monitoring, it does not satisfy the three basic axioms of monitoring:

- a. that a process is constantly measured
- b. that standards exist of system functioning

²⁰ Simon, H., "The Structure of III Structured Problems", **Artificial Intelligence**, North-Holland Publishing Company, (Vol.4 1973) pp.181-201.

²¹ Simon, H, "Information Processing Models of Cognition", **Annual Review Psychology**, (Vol.30 1979), pp.363-396.

²² In 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, if not directly used, to the platform.

c. that variances are observed and management is given opportunity for prompt and close-to-the-event intervention.

It was desirable to differentiate between measurement and monitoring: **Measurement** entailed drawing metrics and actuals using the actual systems-cycle related data²³ to gather measurement. **Real-time-monitoring** implies status-checks through the process with the ability to interrupt or alter the process during its execution. These are actual extremes in the range of monitoring that must be explored as alternatives to the design on monitoring systems.

Definitions

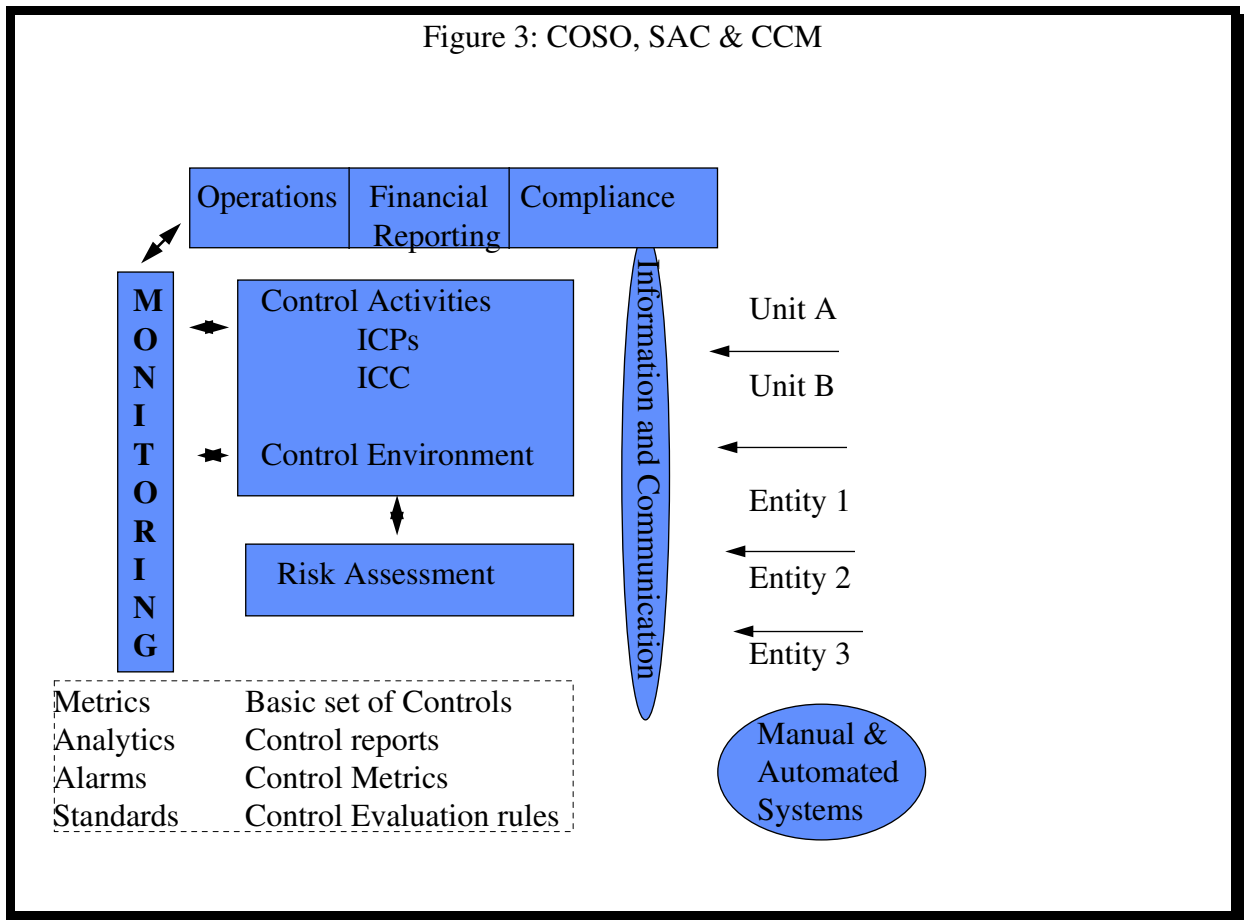
Both COSO²⁴ and SAC²⁵ present a comprehensive view of a framework for the study, understanding and review of internal controls. On the other hand substantial degree of operationalization is necessary for their use in practice. Consequently, most large audit firms and internal audit departments have developed operational manuals for internal control work. Auditing textbook²⁶ tend to organize these procedures at a higher level with emphasis on qualitative assessment.

²³ For instance, drawing copies of regular operational daily reports, at the end of each day, to measure what is happening to the system.

²⁴ Internal Control- An Integrated Framework. Committee of Sponsoring Organizations of the Treadway Commission, 1992, pp. 13-15.

²⁵ Institute of Internal Auditors, **Systems Auditability and Control: module 2, audit and control environment**, Altamonte Springs, Florida, 1991. (Page 2-1)

²⁶ See Wallace, W. A., **Auditing**, PWS-Kent , Boston, 1991, chapter 9 or Arens, A. A., & Loebbecke, J. K., **Auditing: An Integrated Approach**, Prentice Hall, Englewood Cliffs, N. J., 1991, Chapter 9.



Continuity equations

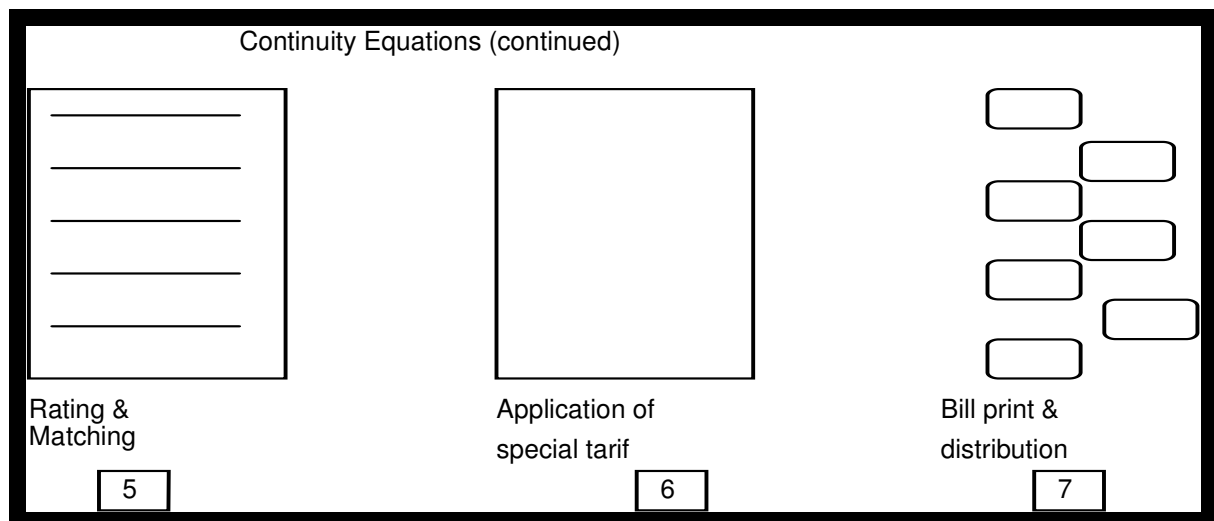
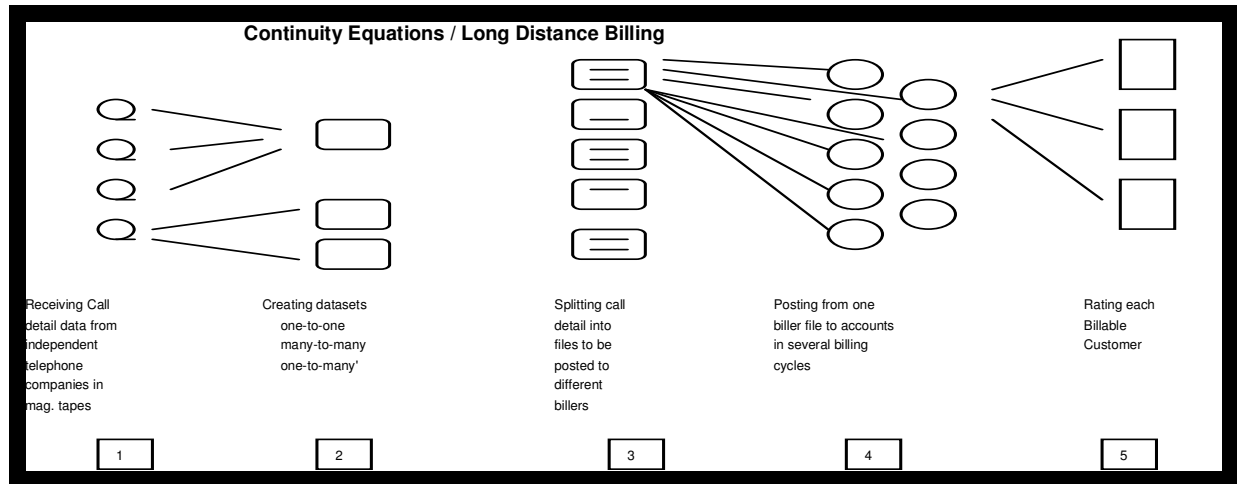
Technology is substantively changing the architecture of data processing systems. They are evolving from large batch oriented mainframes to a more distributed, relational-database environment. Eventually they will evolve towards hardware using massive parallelism on a distributed basis, with intelligent sharing of data and processes across a flexible network.

Nevertheless, the intrinsic nature of the business transaction will continue to rely on two main elements; (1) a "batch" transaction²⁷ representing an economic activity, and (2) the updating of a "master status" file. Furthermore, it will be essential to keep records of the individual transactions on a one-by-one basis (for accountability), and be able to trace the path of a transaction through the business processing entities.

On the other hand, financial system architectures are progressively forcing upward and downward integration among systems through its main elements. Data transfers are the hinges among many modules. The modules are part of a diverse set of business functions that may or may not belong to the same division, department, organization, or company. Figure 4 illustrates a fictitious system for billing long distance telephone calls and their intrinsic elements.

²⁷ An online posting can be seen as a nearly immediate "batch posting" of one (or more) records.

Figure 4: Continuity Equations



The point being made in this rather detailed description, is the need for control **along multiple variables**, and the progressive change of **which control variable** on which to focus. This example will be continued in terms of the development of the concepts of CPAS and CCM. Table 1 continues this discussion linking steps and control variables.

Table 1
Steps, Process, and Control variables

S t e p #	Process	Control variables
1	tape management	number of tapes
2	dataset management	number of datasets
3	call detail separation	number of datasets
		number of messages
		call-minutes
4	account posting	number of messages
		call-minutes
		number of accounts
5	rating & matching	call-minutes
		dollars
		number of accounts
6	special tariff application	modified dollar unit
		dollars
		minutes / messages
7	bill print & distributions	number of bills printed
		number of customers
		messages / dollars/ mod. dollars

CCM definitions and processes

Key control variables, Transition ratios, and Stability

Once processes are defined in terms of **key control variables**, **transition ratios** and their **stability** must be specified. For example, step # 1 measures manually the number of tapes received each day. Tapes are manually labeled, read for heading labels and placed in holding racks. The location of these racks is encoded into the system, or a process code-based location used.

Key control variables are the ones deemed by the auditor/analysis as key to the measurement process. Transition ratios are the standards of conversion between the key control variables between stages (vertically and horizontally). Stability ratios are the variance standards expected between stages.

The ensuing step # 2 encompasses reading the tapes into the system and creating "datasets." The **key control variable** in this stage is datasets, and the transition ratio may be estimated to be 2.3, where on average for every 2.3 tapes one dataset it received. This **transition ratio** of

2.3 creates a more specific and usable standard than something like average number of datasets for the 7th day of the month. If the transition ratios vary between 1.8 and 2.8 the process is considered more **stable** than a process whose ratio (for one standard deviation as an example) varies between 1.6 and 3.0. While, for example, generic transition ratios may be adequate, further information about the process may divide suppliers of data into IOCs (independent operating companies) where the transition ratio is **.7** and BOCs (Bell Operating Companies) where the transition ratios tend to be **2.7**.

The same reasoning may follow between steps #2 and #3 where it may be formulated that each dataset has in average 130,000 messages. Consequently for every tape in step 1 one should expect $130,000/2.3$ messages. The chained extension of the argument continues. If each message is rated in average at \$.71 and volume discount plans draw 25% discount to 47% of the population, each tape may entail billing of $130,000/2.3 \times .71 - 130,000/2.3 \times .71 \times .25 \times .47$.

Integrating manual and Automatic controls

While the evolution of information processing technology has clearly affected operations, management and auditing, little attention has been given to the conceptual evolution of automation integration.

For example, in the front-end processing of receivables, several controls may take place:

1. sequential numbering of sale invoices;
2. batch headers in their transmission;
3. recalculation by a clerk;
4. supervision by a supervisor of the clerk with sign-off.

If this process gets automated, with a hand-held sale receipt device collecting the data, basically all these four functions will be replaced by computer processing.

1. Transactions may be automatically numbered (does not perform a control function but may help in tracking).
2. Batch headers may disappear if online processing is being performed.
3. Recalculations are not necessary, but their function is replaced both by reviews in system approval and system audits of general controls.
4. Supervision disappears, and with it the "sniff test" ability, which entail the detection of the very unusual²⁸ transactions or activities.

The above example illustrates the problems, changes and opportunities that arise therefore with automation and integration of corporate recording activities. While the development of

²⁸ Advanced techniques of pattern recognition, may in the future create software, with the ability of performing "sniff tests" or to replace the human ability of recognizing the unusual without anticipation by the programmer.

integrated transaction processing systems and its simultaneous reengineering²⁹ are highly desirable events from the control standpoint, the most common occurrence is partial automation where a particular process, say creation of sale invoices, is automated.

Spot automation leaves us typically, with inadequate controls as controls are considered and created for the new automated process, but its complementary elements are not reviewed.

Continuous Control Monitoring defined

Continuous Control Monitoring is a management methodology aimed at facilitating corporate operations, supervision, and meta-supervision, through the constant measurement of corporate activity, its comparison against standards, and the reporting of discrepancies, leading to corrective management action. Continuous Control Monitoring puts the emphasis on controls and formalizes the control monitoring process. CCM adds focus to the role of controls in CPAS by viewing monitoring data from a control framework.

Continuous Control Monitoring entails audit involvement from cradle-to-grave in the design, operation and modification of corporate controls. Its main steps entail:

1. involvement in the design of corporate/system controls both of manual and computerized nature;
2. involvement in the setting of standards for control and operations;
3. development of a system of monitoring the operation of controls;
4. development of a system of reporting control functioning;
5. interfaces and support of the internal/external audit activity.

For control monitoring an expanded set of systems features is desirable. The basic concepts necessary to create this expanded features are:

basic-set of controls: the knowledge engineering process defines key-controls to be monitored. While the total set of controls in a system is a fuzzy set, as the human component allows for open-concept, open-ended observation and scrutiny, experience tends to dictate the key controls to be defined and monitored.

For control monitoring, an expanded set of systems features is desirable. These include a basic-set of controls to be monitored, a series of **control reports**, a macro-schema of **control indices aggregation rules**, a schema of **control relationships** and ultimately a set of **control metrics** that aggregate control performance, leading to a final **control evaluation measure**. The idea of a meta control number is similar to various quality measures that certain world-class companies use. For example, Federal Express uses a Service Quality Indicator (SQI) to

²⁹ Davenport & Short, " The New Industrial Engineering: Information Technology and Business Process Redesign," Sloan Management Review, Summer 1990.

manage the quality of its delivery process. The measure consists of twelve components that relate to customer dissatisfaction which are weighted based on customer feedback.³⁰

- **control reports**, specific reports that contain information on particular control(s)
- **control indices aggregation rules**, particular rules on how to mix controls
- **control relationships**: formal specification of the relationship among controls; for example if a control is **redundant** if another control is in place, or it is **complementary** to another control.
- **control metrics**: particular measures relative to a control, typically contained in a control report
- **control evaluation measure**: the result of the aggregation of existing controls upon a process

While CCM can be accomplished in purely manual systems with actual manual CCM procedures, the cost/benefits of the approach indicate the need of at least a CCM system even on mainly manual customer systems.

The Continuous Control Monitoring System (CCMS)

The CPAS³¹ system allowed for continuous process audit at AT&T financial systems. A overview of this system was described in Figure 1 above. This figure, describing the CPAS system, shows the auditee system, electronic copies of operational reports being extracted for audit purposes, and a hierarchical analysis and reporting system, for audit and review. The knowledge engineering process, identifies system features, flowcharts system flows, defines metrics and analytics, defines reports and alarms and requires the information provisioning. Auditor reports and reconciliations are "wired-in" for **audit by exception**.

As stated above, for control monitoring, an expanded set of systems features is desirable. These include a **basic-set of controls** to be monitored, a series of **control reports**, a macro-schema of **control indices aggregation rules**, a schema of **control relationships** and ultimately a set of **control metrics** that aggregate control performance, leading to a final **control evaluation measure**.

³⁰Batting 1000, AT&T Quality Publication, 1992.

³¹ Vasarhelyi, M. A. , Halper, F. B. & Ezawa, K. J., "The Continuous Process Audit Systems: A UNIX-based Auditing Tool," *The EDP Auditor Journal*, Vol. 3, pp. 85-91, 1991.

A system for such an approach could be designed with a modified version of the CPAS system.. Figure 5 below summarizes such a system:

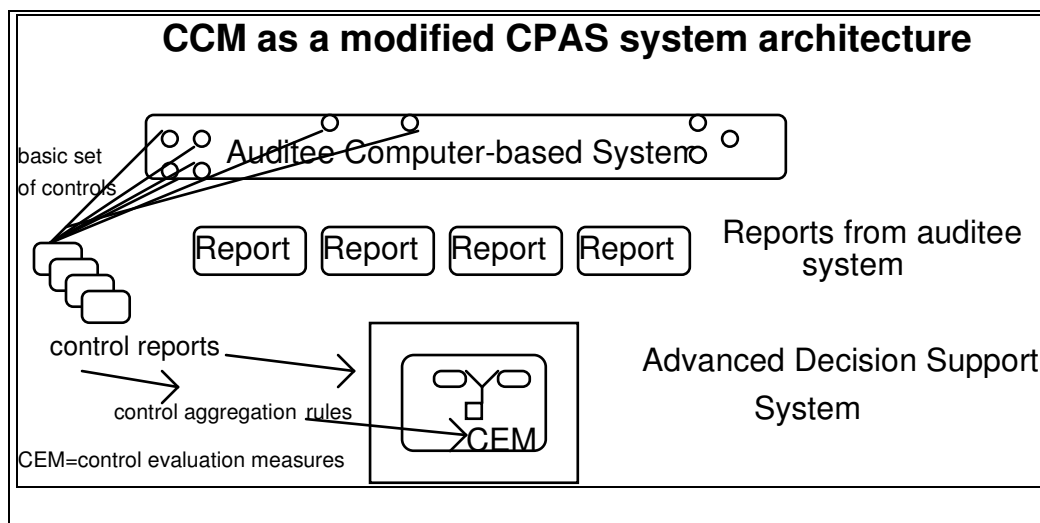


Figure 5

The schema for controls works similarly to the basic CPAS concepts. Control metrics are drawn from manual or automated systems. While in traditional systems reports focus on transactions, in a CCM application, obedience to controls and control performance are metered, reports issued, an aggregation schema concocted and control evaluation measures issued for different cycles and for the system / organization in its entirety.

Measuring Manual Control

The term "manual controls" is a misnomer. It describes processes like authorizations, segregation of duties and supervision which do not imply actual "manual" activity, but the existence of a particular process, activity or organizational structure. A better nomenclature would divide controls into "automatic" and "non-automatic." We will use these terms interchangeably.

The ensuing control measurement procedures illustrate methods of capturing the use/effectiveness of a particular manual control to be entered into the CCM system.

- a. authorizations: a program of systematic review and count of authorization signatures and the preparation of a report into the system. Verification of electronic signatures or manual review of signatures.
- b. validity: systematic review of validation procedures into the Continuous Control Monitoring System (CCMS)
- c. population controls: counts entered and matching number of entries into the CCMS

- d. process controls: manual reconciliations recorded, their variances recorded and these results entered into the overall control scores.
- e. coverage: periodic review through observation and trial testing of issues such as segregation of duties, supervision, obedience to rules and procedures, and insurance. Coverage controls relate to human processes that potentially decrease the incidence of discrepancies. Issuing a rating to these measures and recording into the CCMS.
- f. access: periodic review, review drills, voluntary reporting by the controlee and recording into the CCMS. These relate to physical access to computer facilities, warehouses, etc.
- g. audit: not necessary unless the other measures are not performed or report problems.
- h. compliance with GAAP: independent peer reviews and or by accounting standards units in the organization. Use attested financial statements as references and look for changes to be reviewed.
- i. management controls: selection of key manual systems (or analyses) that management relies to run the company, review of these with an eye towards accuracy, regularity and what they are saying. Draw on key economic health indicators. use these as "going concern" warning lights.

Measuring in Automatic Controls

The naive view of data processing control assumes that once a system is thoroughly reviewed and tested, its computations are correct and the data flow reliable. An exception to this reasoning are the well publicized events of computer fraud with "Trojan horses," "backdoors" or "exemption blocks" created by fraudulent system developers, which are difficult to detect in the voluminous maze of large extant application systems. The reality however, is that as thoroughly as a system is tested, all combination of potential circumstances are nearly impossible to test in large systems. The combinatorics of the paths that data may travel or explode are numerous. All exceptions cannot be accounted for. Furthermore, even if a system is (1) thoroughly tested, (2) has not been changed and (3) has operated for many years weaknesses may exist. For example, upstream systems may have been changed, manual controls prior to the automated portion modified and/or data capture systems automated. These lead to the often observed strange data occurrences, freaky data variances and crashes that may wipe out systems.

The following control measurement procedures illustrate methods of capturing the use/effectiveness of a particular automated control to be routed into the CCM system. Procedures marked with a double asterisk (**) are reasonably similarly evaluated both in manual and non-manual systems.

- a. authorizations: are programs of systematic review and count of electronic authorization signatures. Spot-checks on these authorization and range tests on the allowability of particular authorization. Maintenance of statistics on the actual approval helps avoid

opportunistic approval circumventing corporate authorization policy³². High level statistics on overrides of standard system controls should also be maintained.

b. validity: is a systematic review of validation procedures entered on a random review basis into the Continuous Control Monitoring System (CCMS) (**)

c. population controls: are automatic population counts extended by time-series comparisons over time of population counts, values and other control variables that are part of the continuity equations.

d. process controls: are automatic reconciliations, and their variances recorded, and these results entered into the overall control scores.

e. coverage: is a periodic review thorough observation and trial testing of issues such as segregation of duties, supervision, obedience to rules and procedures, and insurance. Issuing a rating to these measures and recoding into the CCMS. (**) These tests and evaluations should have a different meaning and rating in automated systems.

f. access: is a periodic review, reviews drills, voluntary reporting by the controlee and recording into the CCMS. Automatic pattern evaluation of accesses and usage of facilities should be observed.

g. audit: is not necessary unless the other measures are not performed or report problems. (**)

h. compliance with GAAP: involves independent peer reviews by accounting standards units in the organization. Use attested financial statements as references and look for changes to be reviewed. (**)

i. management controls: involve selection of key manual systems (or analyses) that management relies to run the company, review of these with an eye towards accuracy, regularity and what they are saying. Draw on key economic health indicators; use these as "going concern" warning lights.

In the event of an EIS³³, automatic capture of key-indices is available. Otherwise, a more extensive review, analogous to the type needed for the development of an EIS, should be performed.

Knowledge & Computer Technology

Knowledge Acquisition

The process of knowledge acquisition presented the major challenge of the CPAS/CCM efforts as unlike most of the traditional ES efforts the sources of knowledge we multiple and

³² For example, if a manager's authorization level is \$1,000 and he/she repeatedly breaks purchases down into parts of lesser value but in total costing over the authorized level.

³³ Rockart & DeLong, op. cit., "Executive Support Systems", Dow Jones - Irwin, Homewood, IL. 1988.

disperse. While it was possible to rely on experts for focus and heuristic development much knowledge was impounded in the software itself, contained in systems documentation, and / or obtainable only by system observation. Table 2 describes the natural progression for knowledge acquisition.

Step	Purpose
unstructured interview	understand problem domain
structured open-ended interview	understand problem structure
task analysis	determine expert vs. novice
	use reports to map
	out basic snap-shot analysis
CPAS utilized	build expertise into system
	by using CPAS to map out more
	sophisticated analysis

Discussion

The knowledge base for a CPAS-like application needs to contain information about the system itself (functional and operational) as well as how to analyze the system. We found it convenient to think of the financial system in terms of an algebraic type model and to define parts of the analysis in terms of algebraic variables. Although the knowledge engineers did write up the specifications using algebraic type equations, there was definite reluctance on the part of some people to think in these terms. The greater the expertise the more comfortable the users were with the concept. In fact, these people did actually have a model of the system in their head and found the variable notation natural.

Novice auditors seemed to focus only on system flowcharts and metrics when asked how they analyzed the system. This may be because they didn't really understand the concept of a continuous audit. Typically auditors will work off only a few days of system data when doing an audit. The continuous process audit concept was new and not appealing to some. So our initial analysis focused on a snapshot analysis of the system (i.e. looking for completeness of input). However, once the auditor/manger used the system and saw the trending information available, they would start to ask for additional analytics. We therefore saw the process as iterative and the goal was to use CPAS itself, to help build audit and management expert knowledge.

We found knowledge to be dispersed and heterogeneous. Furthermore, its was very difficult to a priori estimate if experts actually existed in a particular area and if they would provide usable heuristic rules. Ultimately, knowledge came from manuals, system charts, system analysis, user management and auditors.

The automation of this knowledge, when we impounded diagnostics into the system, was questionable as there was very little validation of the rules being "wired in."

In the future a taxonomy of elements of system influence must be developed and measured over time. Human diagnostics of system problems must be coded along this taxonomy and system adjustments/corrections validated across history.

Knowledge Representation

In the initial implementation of CPAS, SQL was used to extract data from the database. For example, individual metrics were calculated in separate SQL queries. More complex analytics were calculated using SQL embedded in C. Since the queries did not resemble the algebraic specifications, it made the analysis more difficult to validate and maintain. We felt the process could be improved if the implementation language more closely resembled the algebraic language used by the knowledge engineers. Ideally, this language could be used directly by the knowledge engineers in place of the specification language as a knowledge acquisition tool. This would cut down on implementation time and make the analysis easier to maintain and validate.

For example, the user might be interested in electrical usage (in kwats and records) that is accepted to be billed from three different places in a northeast region (i.e. New York City, Queens, and Brooklyn). Data related to this analysis might be found on three different daily reports, one each for New York City, Queens, and Brooklyn. Included in the report is information about the date, region, location, usage errored out, and usage accepted to be billed. The data can be extracted from the reports and put in a database table. The database table adds structure to the analysis that the individual reports did not have.

Table 3 represents an output table from a relational database. This table was created using a simple SQL statement. The table shows a schematic view of usage (in records and kwats) from three different sources (New York City, Queens, and Brooklyn) for a hypothetical utility company. Certain parameters which were supplied to the query (date and region) do not appear in the table. The database contains information about errors (err1, err2) and the number of rec and kwats finally accepted to be billed. This data may be used to determine error rates in the northeast region for a particular day.

Table 3- Output of relational database

units	src	err1	err2	acpt
rec	NYC	0	2	300
rec	QUEENS	2	3	350
rec	BKLYN	3	0	350
kwats	NYC	0	50	1000
kwats	QUEENS	8	88	732
kwats	BKLYN	25	0	1500
date=4/1/89	region=NE			

In the table, units and src are indexing information. The other three column labels (err1, err2, and acpt) are values associated with these names. The software uses the label information in output tables to produce lists of name-value pairs. For example:

```
rec[NYC,err1] = 0
rec[NYC,err2] = 2
rec[NYC,acpt] = 298
.
.
kwats[BKLYN,acpt] = 1500
```

are name value pairs that represent the exact information found in Table 3. To use the representation, the user needs to know where the key columns stop and the data begins (err1,err2,acpt) and how the brackets are used. These name-value associations are produced automatically by the software, for later use in algebraic equations. Once the name-value pairs are generated the array language will operate on them. In our array language, we define "index variables" such as NYsrc and Errors (see below), whose values are implicitly iterated over when they are used in expressions. Also, the language uses a summation convention, whereby index variables appearing on the right side of an equation and not on the left are summed over before assignment takes place. Other features of the language include the ability to deal with missing data.

.P

For example, we may want to calculate the error rate for units= rec for error types 1 and 2 for the entire northeast region. We can sum all of the accepted records, calculate all errors, and the error rates using the following equations: together by using the equations:

Input calculation module

```
[NYsrc] = {NYC, QUEENS, BKLYN}
```

```
tot.acpt[NY] = rec[NYsrc, acpt] #implicit summation
tot.input[NY] = rec[NYsrc,err1] + rec[NYsrc,err2] + rec[NYsrc,acpt]
```

Error calculation module

```
[Errors] = {err1,err2}
tot.rec[Errors] = rec[NYsrc,Errors]
```

```
percent.error1 = (tot.rec[err1]/tot.input[NY])*100
percent.error2 = (tot.rec[err2]/tot.input[NY])*100\f1
```

The line "tot.acpt[NY] = rec[NYsrc,acpt]" is equivalent to

```
tot.acpt[NY] = rec[NYC,acpt] + rec[Queens,acpt] + rec[BKLYN,acpt]
```

and the line "tot.rec[Errors] = rec{NYsrc,Errors}" is equivalent to two assignments:

```
tot.rec[err1] = rec[NYC,err1] + rec[QUEENS,err1] + rec[BKLYN,err1]
```

$\text{tot.rec[err2]} = \text{rec[NYC,err2]} + \text{rec[QUEENS,err2]} + \text{rec[BKLYN,err2]}$

percent.error1 and percent error 2 can be compared against a standard. The output from this "model" can also be used as input to more expert rules.

Discussion

The knowledge base for a CPAS-like application needs to contain information about the system itself (functional and operational) as well as how to analyze the system. We found it convenient to think of the financial system in terms of an algebraic type model and to define parts of the analysis in terms of algebraic variables.

Although the knowledge engineers did write up the specifications using algebraic type equations, there was definite reluctance on the part of some people to think in these terms. The greater the expertise the more comfortable the users were with the concept. In fact, some of these people did actually have a model of the system in their head and found the variable notation natural.

Novice auditors seemed to focus only on system flowcharts and metrics when asked how they analyzed the system. This may be because they didn't really understand the concept of a continuous audit. Typically auditors will work off only a few days of system data when doing an audit. The continuous process audit concept was new and not appealing to some. So our initial analysis focused on a snapshot analysis of the system (i.e. looking for completeness of input). However, once the auditor/manger used the system and saw the trending information available, they would start to ask for additional analytics. We therefore saw the process as iterative and the goal was to use CPAS itself, to help build audit and management expert knowledge.

UNIX-based CPAS Implementation

The CPA concept required flexible-modular design and a high degree of flexibility with the purpose of concept-testing and prototyping. Mainframe-based development was deemed too intrusive and too costly. Consequently a workstation-based approach with UNIX-type transitivity and pixel-oriented graphics were chosen.

The CPAS software was implemented under a NeWS windowing system and a SUN workstation. The NeWS system, at that stage, possessed the best set of "widgets" and development tools. It used "postscript" as its imaging language and could use a screen, a file, as well as a laser printer as an output medium.

The entire software was constructed using standard UNIX tools with a minimum of low-level programming. The data was generated in the IBM mainframes in the form of standard user reports. These reports, created with the traditional system development process were analyzed by a "knowledge engineer" and specific fields chosen for collection. JCL specs were included in the application control procedures to specify that a particular report's copy needed to be sent to a particular report distribution node. This JCL specification was the only (and minimal) intrusion in the application.

Once the report was sent to the receiving destination it was placed in an electronic storage bin. A connected UNIX gateway would run periodic (say every 10 minutes) "DAEMONS" and capture (snurf) these reports, transform them into mail messages and **mail** (a standard UNIX function) them to the CPAS workstation. Under certain conditions **uucp** (UNIX to UNIX Communication Protocol) was used to transfer the report file to the CPAS workstation. These reports, upon arrival at the CPASW were identified and scanned for the desired data. For example a re[port named A121 would be identified and an A121.awk *XXXxxx program scanning routine³⁴ would be activated. The data extracted would be placed in a relational database. A commercially available relational database (INGRES) was used as a storage device separating the data gathering portion of the system from its data analysis and delivery device.

The primary user interface (called Flow Front) displayed six main items: 1) a symbolic representation of the application system, 2) a hierarchy box representing the main levels of the system, 3) boxes displaying the values of specific metrics, 4) window with analytic-graphs, 5) window with representative tables, and 6) windows with text containing helps and system text. Furthermore the screen contains buttons, slide bars and touch-sensitive areas.

The graphic interface design device was called "**Flow-Edit**" and is not unlike many graphic design devices now available both in the UNIX (e.g. xxxx) and the DOS (e.g. Harvard Graphics) worlds.

The hierarchy box showed the hierarchical levels of the screens and allowed for moving rapidly among the levels.

Specific metrics boxes contained data represented to be moving along a flow or contained in a level. These metrics were the result of direct **sql** queries to the RDB.

The graphs contained in the windows representing analytics were drawn by a statistical package indigenous in the UNIX environment called **S**. This package developed for "exploratory data analysis contains graphical features of great value and was used in the generation of graphs.

Both the tables and text were generated using UNIX's text editing, formatting and WYSIWIG features enriched by the power of postscript display on to the screens

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.

Discussion

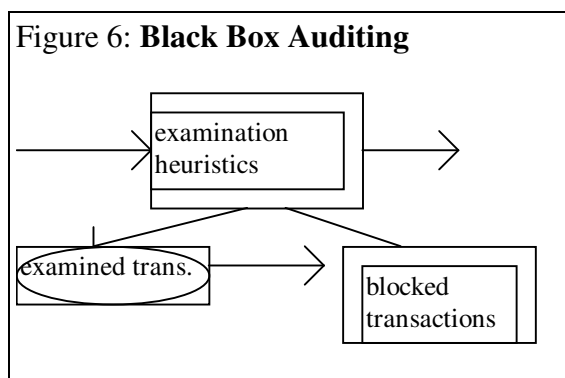
The first software steps in the CPAS effort were performed in 1987/88 when GUI, object-oriented, and workstation-based technologies were still at a rather primitive stage. Many of the envisaged tools are now reality even in the PC/DOS environment. Consequently the

³⁴ Both **sed** and **awk** are pattern scanning languages designed for the identification of specific sequences in text.

scope and domain of the effort could be considerably changed. Now it is quite feasible to develop an audit monitoring system under Windows 95 and run it on a Pentium system. Furthermore the methodology can substantially enrich particular transaction oriented software packages and it is conceivable that it could be impounded into its features.

Forthcoming Elements

Several of our papers have predicted the need for “**Black box auditing**” (**BBA**) where the transaction flow would pass through a black box containing auditor heuristics that identify fraudulent or erroneous transactions.



We have performed some experimentation in this area and several basic questions arose:

- do auditors have rules that can be extracted for emulation?
- how do you assess probabilities to flagged transactions?
- how do you decide on the cost x benefits of further examination of flagged transactions?
- what rules do you put in effect for process interruption on a particular (or particular set of) transaction(s)?
- are there different categories of transactions to be scrutinized (e.g. fraudulent vs erroneous vs irrelevant)?

Furthermore, it seems, from our preliminary experimentation that **BBA** as conceived originally is not sufficient. While individual transactions may be adequate patterns may develop that are of importance to system monitoring from unexpected types of fraud to recurring erroneous postings. Two technologies of great appeal, that can be used simultaneously and coordinately, as well as independently are emerging: Data mining³⁵ and intelligent agents³⁶.

Data Mining

Data mining lets the power of computers sift through large data stores. A wide variety of applications are emerging for the technology³⁷ among which patterns on sales data, semantic construct identification, telephone calling patterns, etc. Three major types of tools are available³⁸:

- query and reporting tools

³⁵ DeJesus, E. X., “Data Mining,” **BYTE**, October 1995, p.81

³⁶ Riecken, D. (Ed.) , “ Intelligent Agents,” **Communications of the ACM**, July 1994, Vol 37, No. 7, pp. 18-21.

³⁷ Hedberg, S. R., “The Data Gold Rush,” **BYTE**, October 1995, p.83-89.

³⁸ Watterson, K., “A Data Miner’s Tools,” **BYTE**, October 1995, p.91-96

- multidimensional analysis tools
- intelligent agents

Agents

Agents^{39 40 41} are natural elements for system monitoring and analysis. One can imagine an alternate paradigm whereby the transactions do not flow through an audit detection pipe as described in Figure 6 but agents roam through all parts of the system with the same objectives specified. While this approach would require wider and less identifiable algorithms, it has the appeal of providing a wider and more subtle monitoring, that cannot be bypassed easily by intelligent algorithms.

Conclusions

While the original work on the CCM / CPAS may have been somewhat premature great interest in this type of work is emerging. This is due to the fact that integrated online systems are not tractable by traditional auditing methods and losses can be too large prior to the realization, through traditional methods, that there is a control leak.

While, algorithmic and analytic solutions have been prevalent in the past, brute force solutions, very rich in computational requirements are now possible and likely of use. Solutions that subject each transaction to intensive analytic (computational) scrutiny as well as mining agents that frequently travel through large databases are not only possible but considering the economics of modern computing quite likely and desirable. The minimal cost per computer cycle makes these efforts clearly cost beneficial due to their deterrent and early detective action.

The agent terminology is recent but some of its main principles have been explored in the last decade. Interlocked networks and large databases give the opportunity for substantial benefits from software agents. System and data monitoring problems offer great potential areas for exploratory, pattern recognition and review agents.

Extant research has not yet defined the parameters of the utilization of agents in AI/ES in accounting research. The advent of the Internet as a pipeline for accounting systems and as a source for data gathering, as well as the increased use of EDI in corporate information systems makes this research even more important.

Research on the nature of agents is of essence. Regardless of auditor's desires modern online systems will be permeated by functional agents performing specific tasks. These agents will

³⁹ "Conversations with Marvin Minsky about agents", **Communications of the ACM**, July 1994, Vol. 37, No. 7, pp. 23-29.

⁴⁰ Maes, P., "Agents that Reduce Work and Information Overload," **Communications of the ACM**, July 1994, Vol. 37, No. 7, pp. 31-40.

⁴¹ Norman, D., "How Might People Interact with Agents," **Communications of the ACM**, July 1994, Vol. 37, No. 7, pp. 31-40.

be part of the environment but will also increase the risk of systems. It is not inconceivable that between friendly agents systems will also find viruses and intrusive objects. While computer science and architectures will work to provide a safe and productive environment for agents this same work is creating substantial business threats. It is up to AI/ES researchers in accounting to deal with these threats.

Concluding, it is important to mention what we view on of the major weaknesses in extant accounting research. We have been unable to well characterize, represent and deal with the “soft controls” of information systems. It is still very difficult to conceptualize and measure controls such as coverage and supervision as well as to integrate these measures with an overall view of system controls. It is highly desirable that AI/ES research in accounting focus on automatic methods of measurement of manual systems and their interaction with computer-based processes.