Communications of the Association for Information Systems

Volume 31 Article 3

7-2012

A Model for Investigating Internal Control Weaknesses

Jian Guan
Department of Computer Information Systems, College of Business, University of Louisville, j0guan01@louisville.edu

Alan S. Levitan
School of Accountancy, College of Business, University of Louisville

Recommended Citation
Guan, Jian and Levitan, Alan S. (2012) "A Model for Investigating Internal Control Weaknesses," *Communications of the Association for Information Systems*: Vol. 31 , Article 3.
DOI: 10.17705/1CAIS.03103
Available at: https://aisel.aisnet.org/cais/vol31/iss1/3

Communications of the Association for Information Systems

CAIS

A Model for Investigating Internal Control Weaknesses

Jian Guan

Department of Computer Information Systems, College of Business, University of Louisville

j0guan01@louisville.edu

Alan S. Levitan

School of Accountancy, College of Business, University of Louisville

Abstract:

Scandals in corporate finance in the early 2000s and subsequent policy changes led corporate executives to adopt a more risk-based approach in corporate governance. Therefore, identification and assessment of risks became extremely important. Risk assessment poses a particular challenge for auditors due to the highly complex structure and processes of internal control systems. Extant research in this area mostly focused on probabilistic models and expert systems that capture and model heuristic knowledge. However, evidence suggests that knowledge of the structure of the internal control system is also essential. There is relatively little research that focuses on the modeling of the structural aspects of financial processes and their internal control systems as a means of helping corporate executives and auditors perform their respective tasks of risk management and assessment. This article proposes an approach to risk management and assessment in internal control systems that models the structure and financial processes of an internal control system. The model uses a directed graph to represent the various elements in an internal control system, such as financial statement assertions, control activities, financial processes, and the causal relationships that exist among these elements. The article demonstrates the usefulness of the model by presenting and discussing algorithms based on this model to help corporate executives manage risk and to help internal and external auditors assess risk, for designing substantive testing and for tracing sources of errors.

Keywords: auditing, internal controls, directed-graph, risk management, substantive testing

Volume 31, Article 3, pp. 61-84, July 2012

The manuscript was received 4/13/2010 and was with the authors 12 months for 3 revisions.

Communications of the Association for Information Systems

I. INTRODUCTION

Scandals in corporate finance in the early 2000s and the subsequent policy changes required managers to adopt a more risk-based approach in corporate governance. Therefore, risk management and assessment of risks for the audit function became extremely important in this context. Risk management tasks pose a particular challenge for managers, as does risk assessment for auditors, because of the highly complex structure and processes of internal control systems [Davis, Massey, and Lovell, 1997; Felix and Niles, 1988; Krishnan, Peters, Padman, and Kaplan, 2005; Wand and Weber, 1989]. Pressures are also mounting for both internal and external auditors to perform their tasks more effectively and efficiently [Curtis and Payne, 2008]. The combination of increasing complexity of the information processing environment and the mounting pressure for efficiency and effectiveness provided the impetus and the advice for managers and auditors to switch from a process-oriented approach to a risk-based approach [McNamee and Selim, 1999; Weidenmier and Ramamoorti, 2006]. The importance of risk management is also evidenced by the establishment of policies and frameworks by professional organizations, such as the Committee of Sponsoring Organizations of the National Commission on Fraudulent Financial Reporting (COSO), to ensure a more formal approach to risk management. For example, COSO released the Enterprise Risk Management (ERM) framework to serve as guidelines for reliable reporting and regulatory compliance.

To effectively and efficiently perform risk management and risk assessment, tools and techniques could help navigate the complex internal control environment and help make decisions regarding risk levels [Denna, Hansen, and Meservy, 1991]. Most of the existing research in this area has focused on probabilistic models and expert systems [Bodnar, 1975; Kelly, 1985; Lenard, Alam, Booth, and Madeyet, 2001; Looi, Tan, Teow, and Chan, 1989; Meservy, Bailey, and Johnson, 1986; O'Donnell, Arnold, and Sutton, 2000; Srivastava and Shafer, 1992]. Though probabilistic and heuristic knowledge is important, evidence suggests that knowledge of the structure of the internal control system is also important [Frederick, 1991]. However, relatively little research focuses on the modeling of the structural aspects of internal control systems as a means to assist in managing and assessing risk. This article proposes an approach to modeling internal control systems by focusing on the representation of the structural aspects of internal control systems. The proposed model consists of two parts. The first part is the modeling of the financial processing structure of an internal control system as a directed graph, and the second part defines the reasoning processes that can be used to manage and assess risk. The directed graph-based model represents the various elements in a financial processing system, such as financial statements assertions, control activities, financial operations, and causal relationships that exist among the elements. Algorithms that help external auditors in diagnosing weaknesses to plan substantive testing and help management in evaluation of the system demonstrate the utility of the proposed model. When a financial operation fails or misstatement occurs, it may not manifest itself immediately. The problem may be detected downstream in the transaction flow. During tests of controls, corporate personnel and external auditors may follow the path forward to find vulnerable financial assertions. And, during substantive testing, an auditor may need to backtrack the complex internal control system to locate the source of the problem. The first algorithm helps managers and auditors assess the potential impact of a weak control or financial operation. The second algorithm helps auditors navigate the complex structure of an internal control system by reducing the search space during substantive testing.

The purpose of this article is to introduce a methodology to improve auditors' ability to provide assurance to the public of the reliability of a company's financial statements as representing the company's financial position and results of operations. Auditors are currently required to assess a company's internal control system before conducting the substantive examination of the company's records. The more that auditors may rely on their clients' controls, the more efficient the audit, since the nature and extent of the substantive examination may be reduced. Clients want a more efficient (less intrusive and costly) audit, while the public increasingly demands a more thorough and effective audit.

Our article thus presents a new modeling approach for internal control systems auditors may utilize without greatly increasing their time allocation to control assessment, but which will more expeditiously connect weaknesses to vulnerabilities in financial statements. Implemented, this model will improve auditors' ability to budget valuable auditing time to areas that are more vulnerable to misstatements, while also speeding the auditors' ability to trace errors back to weak controls, especially with large and complex companies.

The rest of the article is organized as follows. The next section provides a brief description of the internal control system environment, i.e., its risks, its management, and its assessment. A review of relevant literature follows in

Section III. Section IV introduces the proposed model and algorithms for risk management in internal control systems. Finally, Section V provides conclusions and discusses future research directions.

II. ASSESSMENT OF RISK IN INTERNAL CONTROL SYSTEMS

When accounting scandals (and so-called "audit failures") led to the 2002 Justice Department prosecution of corporate executives, the destruction of the previously venerable Arthur Andersen accounting firm, and the enactment of the Sarbanes–Oxley Act (SOX), attention to internal controls reached a new peak, while confidence in published financial statements plummeted. Among its provisions, SOX Section 302 required the corporate CEO and CFO in public filings to acknowledge their responsibility for establishing and maintaining internal controls and to report on the current operational effectiveness of the corporation's internal control system. Any material control weaknesses had to be publicly disclosed. Section 404 then required auditors to perform and report their own evaluation of the system of controls.

An audit is essentially an engagement in which the auditor examines and evaluates evidence about a set of management assertions (i.e., the components of the financial statements) and then issues a report attesting to the degree of correspondence between those assertions and established criteria (usually, generally accepted accounting principles, or GAAP). Each of those assertions by management contends that the amount is neither materially overstated nor understated, and that it is valued fairly. The strengths of the controls in the accounting information system (AIS) support these assertions.

An audit requires substantive testing of the account balances and the transactions which created those balances. Usually, statistical sampling is applied to select transactions for examination, resulting in reasonable (rather than absolute, prohibitively expensive) assurance. The degree and amount of substantive testing can be reduced if the internal control system can be relied upon. Thus, it is efficient to perform tests of controls before substantive testing in order to identify areas containing control weaknesses. Tests of the operation of controls may involve the entry of dummy transactions with deliberate errors in order to ascertain the operational effectiveness of the control procedures in transaction processing operations. The financial assertions affected by operations found to be poorly controlled would be more vulnerable to misstatement and, therefore, should be subjected to more extensive, time-consuming, substantive testing by auditors.

There are standard many-to-many relationships between financial statement assertions and transaction processing operations, and there are standard many-to-many relationships between transaction processing operations and control procedures. But these standard relationships, linking a control procedure through a complex network of operations to an assertion, may vary from one company to the next. The greater the comprehension of these linkages, the more effective and efficient will be the evaluation of controls.

With little specific guidance in SOX or from the Securities and Exchange Commission (SEC) and an atmosphere of excessive caution within the audit community, managers and auditors preferred to err on the side of effectiveness in their examination of internal controls rather than efficiency, identifying and testing nearly every control. As a result, some large corporations endured a tripling of audit fees on top of large new internal costs.

To constrain excessive management and audit costs, with their disproportional impact on smaller companies (scalability), while preserving SOX's fundamental objectives, the SEC and its Public Company Accounting Oversight Board (PCAOB) in 2005 launched a process to concentrate audits on areas of greatest risk, those areas most likely to result in a material misstatement in the financial assertions. This process resulted in the 2007 publication of Auditing Standard No. 5, "An Audit of Internal Control Over Financial Reporting That Is Integrated with an Audit of Financial Statements."

Thus Auditing Standard No. 5 put more responsibility on the auditors' judgment, directing auditors, as well as management, to use their professional judgment to determine the key controls—controls that mitigate potential misstatements. These are the controls that should be tested. As soon as auditors or managers are satisfied that a significant assertion has one (or more) well-designed and functioning control(s) in the network of financial operations leading to it so that its risk of material misstatement is reduced to a sufficiently low level, the controls of any other operations in that segment of the network do not need to be tested with respect to that assertion. The intention is to make audits more efficient (less costly) without reducing their effectiveness.

III. RELEVANT LITERATURE

From the description in the previous section it is apparent that auditors, and now enterprise executives, face a daunting task in balancing effectiveness and efficiency in managing and assessing risks. An early approach to model the audit knowledge and auditing processes uses probability and/or system reliability theory to model internal

control systems [Ahituv, Halpem, and Will, 1985: Bodnar, 1975; Cooley and Cooley, 1982; Cushing, 1974; Stratton, 1981; Yu and Neter, 1973]. These models regard an internal control system as consisting of a set of controls, control procedures that monitor the controls, and error correction procedures. They describe the sequential ordering of control procedures and use probability estimates for procedure errors. The distinct features of this approach include systematic error description and use of explicit numeric output to assist auditors in decision making. Though this approach can provide a mathematically sound description of an internal control system, it failed to gain acceptance because of the high cost of collecting the needed probabilities, the need for reassessment of these probabilities as internal control systems change, and the simplifying assumptions [Felix and Niles, 1988]. For example, Bodnar [1975] notes that, although the simplifying assumption of statistical independence among controls may be necessary, research supports interdependence of behavioral variables (e.g., dominance by one employee in a group, predisposition of an employee toward collusion).

Expert systems have also been proposed to assist auditors in planning and evaluating audit systems. Auditing often involves complex decision-making processes. The ability to make the right decision often requires years of experience and deep knowledge. Expert systems are a natural solution to codify human auditors' expertise and experience. Examples include identification of risks, appraisal of controls, and planning/assessment of control and substantive tests [Hunton and Rose, 2010]. Audit research has a long and rich history of using artificial intelligence (AI)-based techniques [Bailey, Duke, Gerlach, Ko, Meservy, and Whinston, 1985; Baldwin, Brown, and Trinkle, 2006; Changchit, 2003; Davis, Massey, and Lovell, 1997; Denna, Hansen, Meservy, and Wood, 1992; Gadh, Krishnan, and Peters, 1993; Kelly, 1985; Looi, Tan, Teow, and Chan, 1989]. A common weakness of expert systems is their lack of completeness in terms of covering the various aspects of an internal control system [Krishnan, Peters, Padman, and Kaplan, 2005]. In addition expert systems rely on the availability of expert knowledge and knowledge acquisition, both of which can be a hindrance to implementing expert systems. These systems have not lived up to their expectations [O'Leary, 2003]. However, the biggest weakness of expert systems lies in the fact that they tend to utilize only one of two important types of knowledge that auditors find useful when performing their tasks. In addition to traditional expert systems, artificial neural networks have also been used in audit-related tasks [Calderon and Cheh, 2002; Koskivaara, 2004]. Koskivaara criticizes the use of neural networks in auditor decision aids by stating one of their major weaknesses, i.e., the inability of such systems to trace the process by which the output is reached [Koskivaara, 2004]. Therefore, the results lack explanatory capabilities.

A study by Frederick shows that auditors, in particular experienced auditors, used more knowledge of the structure of an internal control system, or knowledge of the various financial operations and their interconnectedness or relationships in a financial processing system [Frederick, 1991]. Transaction flow or the flow of accounting data is an important source of information for auditors [Frederick, 1991; Whittington and Pany, 2001]. Gadh, Krishnan, and Peters [1993] point to the need to combine structured knowledge and heuristic knowledge in modeling internal controls. As further evidence of the importance of structured knowledge use in auditing, empirical work indicates that flowcharts are part of auditors' internal representation of internal control systems [Frederick, 1991; Kelly, 1985; Meservy, Bailey, and Johnson, 1986]. However, few papers focused on modeling this very important aspect of auditors' extant knowledge of internal control systems [Bailey, Duke, Gerlach, Ko, Meservy, and Whinston, 1985; Krishnan, Peters, Padman, and Kaplan, 2005; Wand and Weber, 1989]. The structure of an internal control system can be defined by the financial operations or their information system implementations, control activities, financial statement assertions supported by control activities, and the information flow among the first three elements. Bailey, Duke, Gerlach, Ko, Meservy, and Whinston [1985] use biologic-directed graph to capture both control and data flows in their TICOM model. The TICOM model represents accounting operations and control activities as nodes in the graph and uses constraints sets to represent relationships among the nodes. Procedures based on TICOM allow auditors to perform more efficient evaluation of an internal control system for a client. Gadh, Krishnan, and Peters [1993] extend Bailey's work by proposing a model that combines structural knowledge of internal controls with more heuristic knowledge of rules [Bailey, Duke, Gerlach, Ko, Meservy, and Whinston, 1985]. Wand and Weber propose an ontological model for accounting information systems to allow more efficient identification of controls affected by changes in the accounting information system [Wand and Weber, 1989]. Each component in the accounting information system is modeled as an entity (or "thing") with properties. The relationships among the entities are defined by the accounting events through which the related entities affect each other. A hierarchy of subsystems is defined through these entities, the accounting events, and the relationships. Procedures are proposed and discussed, procedures that allow the search for the affected controls to be localized to subsystems affected by the changes in the accounting information systems. Krishnan, Peters, Padman, and Kaplan [2005] present a process-oriented model for accounting information systems to help auditors make an effective and efficient selection of key controls for reliability assessment. The key components of the model include economic events, information transformation processes, control activities, and target error classes. Target error classes and the related controls and information transformation processes are modeled as a relation called *covers*. Procedures based on set-covering algorithms are proposed for both key control selection and evaluation.

Common to all these models is the representation of the structural aspects of an internal control system. Because of the nature of economic transactions in an AIS, how information flows and how such flows are monitored through controls are very important to the assurance that the representations of the economic transactions are not materially misstated. The information flow in an internal control system, particularly the propagative nature of such flows, has been well recognized by researchers in the field [Ahituv, Halpern, and Will, 1985; Bailey, Duke, Gerlach, Ko, Meservy, and Whinston, 1985; Bodnar, 1975; Cushing, 1974; Krishnan, Peters, Padman, and Kaplan, 2005; Wand and Weber, 1989]. However, relatively little research focuses on this important aspect of modeling in internal control systems. This article proposes a model for the structural aspects of an internal control system using a directed graph to represent its financial operations. An important feature of the model is its ability to model the propagative flow of economic information through capturing the causal relationships among the elements of an internal control system. Directed graph-based modeling techniques in engineering contexts proved to be an effective tool for knowledge representation of a financial reporting system [Guan and Graham, 1994, 1996; Narayanan and Viswanhadam, 1987]. The article also demonstrates how the model may be used to help managers and auditors in their new responsibility of selecting which controls to test and, moreover, to help auditors diagnose errors during substantive testing of an internal control system using an algorithm adapted from Guan and Graham [1994].

IV. A DIRECTED GRAPH REPRESENTATION OF A FINANCIAL REPORTING SYSTEM

This section describes a proposed model for internal control systems and presents algorithms for assisting in risk management and assessment. This approach to modeling the internal control system is motivated by the evidence that experienced auditors rely on knowledge of the structural aspects of the system and transaction flow within the system [Frederick, 1991] and the relative lack of research on models that assist auditors in utilizing this important knowledge component.

Auditors are required to document a client's control structure as proof for federal regulators and for use by the audit team. Economic events are captured by the AIS and then follow paths through a network of transaction processing operations, terminating with their effect on one or more financial statement assertions. Each operation, if not properly controlled, could introduce an error as it processes the data. Only through understanding these relationships can a testing plan be designed that is both effective enough to reduce the risk of material misstatement to an acceptable level, and efficient enough to keep the testing costs within reason.

The higher the materiality and the inherent risk in an assertion, the greater the need for strong controls. Thus, the paths backward, from assertion to operations, is likewise invaluable for planning effective and efficient tests of the functioning of the controls surrounding those operations.

The currently required documentation explains transactions flow from initiation, authorization, recording, processing, and reporting, along with the control procedures applied [Ramos, 2006]. "Walk-throughs" are recommended as the best way to confirm understanding of control design and operation [Ramos, 2006]. In a walk-through, the documenter follows a single transaction through its detailed procedures, making inquiries and gathering evidence. Thus, auditors already routinely gather and evaluate the inputs needed for the proposed model. The proposed model provides a formal representation to help auditors navigate the complex web of transaction flows to identify risks and help auditors locate weak controls that lead to misstatements.

The proposed model formally represents the complex web of financial operations and their relations to assertions using a directed graph. This model allows an auditor to navigate both forward from operations to assertions and backwards from assertions to operations. In an internal control system, the controls are associated with operations. Thus the ability of an auditor to navigate easily an internal control system provides a tremendous advantage for both effectiveness and efficiency purposes. Though the controls are part of the proposed model, the main objective of the model is to assist the auditor to locate the financial operations whose controls are to be assessed. The interactions and interdependence of the financial processes are represented in the model from which, in conjunction with a commonly available list of expected controls for given operations such as shown in Table 1 for the example financial reporting system in Figure 1, the key controls and their interrelationships may be inferred.

Modeling of Internal Control System

The financial processing system, *S*, is defined as

$$S = \{A, C, O, G\}$$

where

A = set of financial statement assertions

C = set of controls/control activities

O = set of financial operations such as recording a disbursement.

Communications of the Association for Information Systems

Table 1: Financial Operations and Their Controls	
Financial Operations	Associated Controls
Receive customer Purchase Order and prepare job requirements for Operations & Trucking	Order processed by authorized employee, properly trained, with password compatibility check
	Validity check on customer to approved customer list
	Credit check on customer's current status and credit limit before accepting order. Any exceptions must be approved by Treasury function.
	Validity check on inventory SKUs
	Range/reasonableness check on inventory quantities
	Acknowledgement of order sent to customer
	Orders sequentially numbered to highlight any missing/unbilled orders
Weigh trucks when empty	Trucks weighed by authorized employee, with password compatibility check
	Recorded weight subject to range check for reasonableness, and preferably entered automatically from scale
	Accuracy of scale checked daily against standard weights
Send to the bank any customer remittance payments erroneously sent directly to the company	Incoming mail opened by two employees, who are bonded with theft insurance, working together in a room with a video camera
	All checks received immediately endorsed, "For deposit only into account 9999"
	Remittance logged for subsequent comparison to bank's list of customer remittance payments received
	Monthly statements sent to customers who can notify the company if a payment they sent is not shown on the statement
If this is a custom job, create an internal manual Packing List	List prepared by authorized employee, properly trained, with password compatibility check
	Validity check on inventory SKUs
	Range/reasonableness check on inventory quantities
Weigh trucks when full	Trucks weighed by authorized employee, with password compatibility check
	Recorded weight subject to range check for reasonableness, and preferably entered automatically from scale
	Automatic check of difference between full weight and empty weight for reasonableness
	Accuracy of scale checked daily against standard weights
IF THIS IS A CUSTOM JOB, CREATE MORE DETAILED PRICED PACKING LIST	List prepared by authorized employee, properly trained, with password compatibility check
	Validity check on inventory SKUs
	Range/reasonableness check on inventory quantities
	Range/reasonableness check on inventory prices
Access secure bank website for list of customer remittance payments received. Record in Qb to reduce bank loan and Customer Ar. Report any non-customer remittance to bank, to be treated as cash, and report customer payment discrepancies to manager	Website accessed by authorized employee, properly trained, with password compatibility check
	Date-of-last-logon shown and consistent with authorized last logon
	List checked against log of remittance payments originally erroneously sent directly to company and forwarded to bank, for inclusion in list
	Amount check of payment against balance owed for reasonableness
	Date of payment (current date) automatically entered into QB by system
	AR total in QB before and after update confirmed to be reduced by total of remittance payments
	Monthly statements sent to customers as additional confirmation that payments have been properly credited
Create Invoice in Qb. "Big Co." is invoiced at only 90% of contract	Invoice created by authorized employee, properly trained, with password compatibility check who has no access to inventory, cash, or credit approvals
	Standard prices retrieved automatically from price file rather than keyed in
	Discount for "Big Co." calculated automatically
	Invoice products, quantities, and customer number compared for agreement against customer's original purchase order
	Range/reasonableness check on total dollar amount of invoice
	QB updated automatically, based on invoices, for sales, AR, inventory, and cost of sales, with date of invoice (current date) automatically entered into QB by system

Table 1: Financial Operations and Their Controls – Continued	
Send weights to clerk	Weights sent automatically, by authorized employee, properly trained, with password compatibility check
	Acknowledgement by clerk of weights received
Receive Monthly Bank Statement, reconcile and record adjustments in Qb	Bank statement received and reconciled by authorized employee, properly trained, and independent of previous processes involving cash
	Checks shown as outstanding for more than 30 days, and deposits shown as in transit for more than 2 days, are investigated
	Any bank charges are proved to be in accordance with contracts with bank
	Resulting reconciliation is reviewed by upper management
Receive Final Output Melt Weight from "Big Co." and record adjustments in Qb	Range/reasonableness check on adjustment compared to weight originally invoiced
	Adjustments recorded by authorized employee, properly trained, with password compatibility check
Create Bill of Lading for all jobs	Bill of lading created by authorized employee, properly trained, with password compatibility check
	Bill of lading compared against customer's original purchase order for products and amounts, and against weights previously calculated
	Report produced for management of time between receipt of customer order and shipment to highlight unusual delays
Record manager's decisions in Qb	Adjustments to AR and Sales Adjustment recorded by authorized employee, properly trained, with password compatibility check
	List of transactions affecting AR and Sales Adjustments automatically prepared at end of day for manager's review
Decide whether to accept/reject customer payment discrepancy	Manager compares amount of adjustment to total amount billed
	Manager reviews customer's previous discrepancies for habitual behavior

Each financial operation may be associated with a set of control activities. Let *f* be a function mapping each financial operation to a set of controls in *C*, then we have for each financial operation O_i:

$$f(O_i) = \{c_j\}, j = 0, n \text{ and } c_j \in C$$

G = a directed graph representing *S* such that

$$G = \{V, E\}$$

where *V* is the vertex set representing assertions and/or financial operations and *E* is the edge set representing the causal relationships among the vertices.

A directed graph-based approach captures both the static structure of the system and the propagative relationships among the financial operations of the system [Guan and Graham, 1994, 1996; Narayanan and Viswanhadam, 1987; Warfield, 1974]. Let $X = \{x_1, ..., x_n | x_i \in A \cup O\}$ be a set of assertions and financial operations (hereafter referred to as *nodes*) in an internal control system. Then a propagation relationship Ψ can be defined on *X* such that $x_i \Psi x_j$ means that nodes x_i and x_j are related or an error in x_i can propagate to x_j. *A* propagation digraph *G* is then used to represent this relation as

$$G = \{V, E\}$$

where $V = \{x_i | x_i \in X\}$ is the vertex set;

and $E = \{(x_i, x_j) | x_i \text{ and } x_j \text{ are related and } i \neq j\}$ is the edge set.

Figure 1 is a simple example of an internal control system, representing the revenue cycle of an actual business in a Midwest city in U.S. Each node in the figure represents either a financial statement assertion or a financial operation of the system and is labeled by a number for ease of discussion. Given the definitions of the structure of the system above we have:

A = {Sales, Cost of Sales, Inventory, Accounts Receivable, Cash, Loan Payable, Sales Adjustment, Bank Service Charge}

C = Internal Controls (See Table 1 for the list of controls for each of the financial operations in the internal control system.)

O = Financial Operations (See Table 1 for a list of the financial operations in the internal control system.)

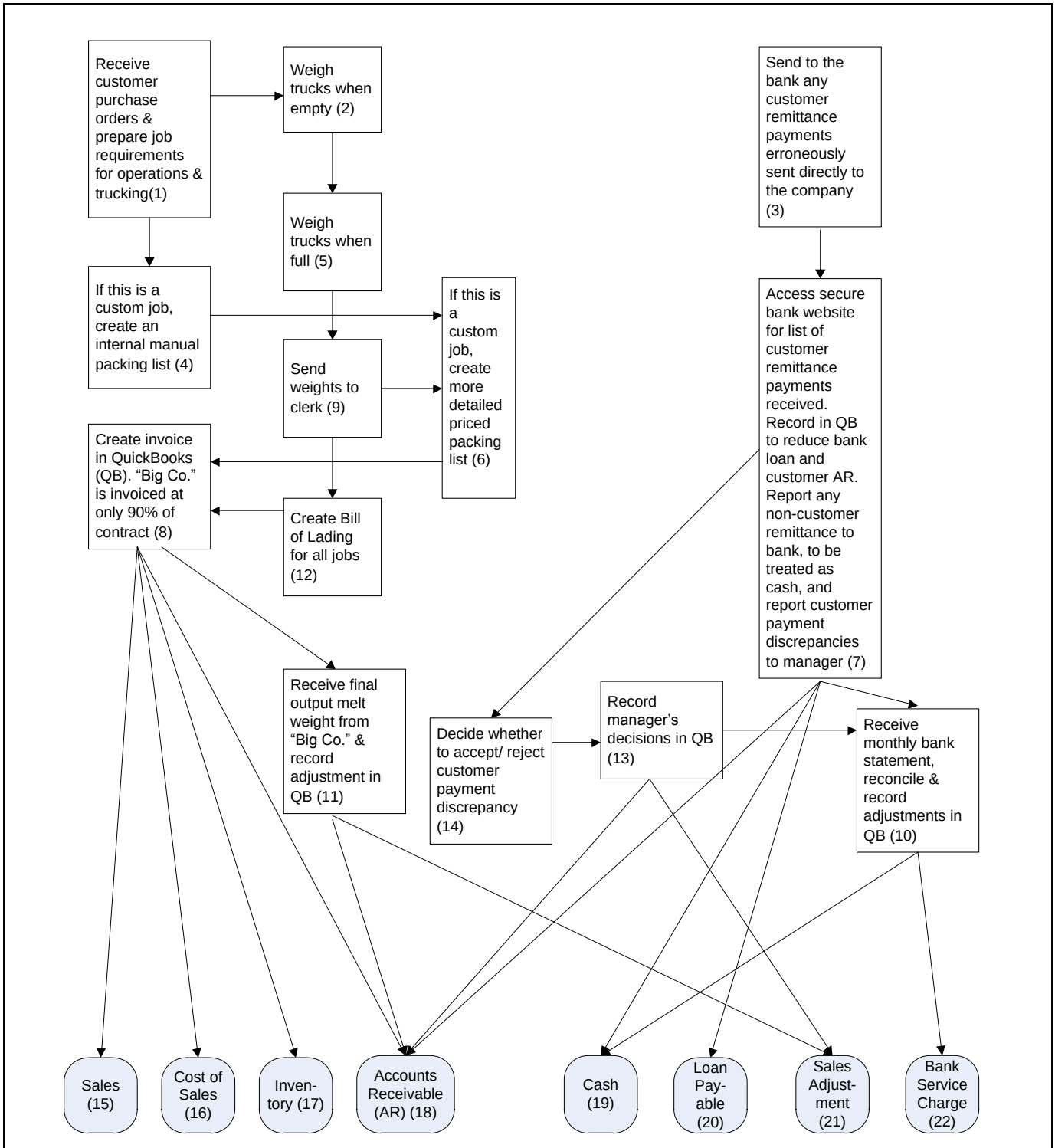

Figure 1. A Simple Financial Reporting System

Let $\mathbf{M} = [m_{ij}]$ be the adjacency matrix representing the internal control system digraph such that $m_{ij} = 1 \; if \; \{x_i, x_j\} \in \Psi$ and $m_{ij} = 0 \; if \; \{x_i, x_j\} \notin \Psi$. If there is a path in *G* from x_i to x_j, x_j is said to be reachable from x_i. In the context of an internal control system, this means that an error/risk occurring at x_i may propagate to x_j. For completeness, every vertex in *G* is defined to be reachable from itself by a path of 0. As defined above reachability is transitive.

The proposed model captures the key elements (financial statement assertions and financial operations) and their relationships in an internal control system. The model also allows heuristic knowledge, since corporate personnel and external auditors often use past experience in forming judgments and opinions about the tasks they are performing. Errors/risks in internal control systems can propagate through the system if not detected. The challenge is finding out from which one of the many sources the errors/risks may have propagated. For any financial operation *x,* the error may have propagated from any of the nodes upstream. These nodes are called the *ancestors* of *x*. In some cases there may be several immediate ancestors of *x* and the error could have propagated from any of these ancestors or any of the error/risk propagation paths headed by these ancestors. Obviously an error may be more likely to propagate from some of the nodes than others. This information can be captured through a propagation probability. Associated with each edge then is p_{ij}, the probability that an error/risk will propagate from x_i to x_j, $0 < p_{ij} \leq 1$. It is assumed that this information is available from auditors experienced with the internal control system and can be refined by the auditors' tests of controls. This assumption is based on the requirement that auditors assess all components of the risk that a material misstatement in assertions occur and not be prevented or detected by the company's procedures. "These components of audit risk may be assessed in quantitative terms, such as percentages, or in nonquantitative terms, such as high, medium, or low risk. The way the auditor should consider these component risks and combines them involves professional judgment and depends on the auditor's approach or methodology" (Aud Sec 312, AICPA Codification of Auditing Standards, 2010). The requirements further state that the auditor must have sufficient evidence of his/her evaluation of these risks through understanding and testing the controls. Thus obtaining the probabilities for this model does not require significant additional work beyond existing auditing regulations.

In case such information is not available, equal probability can be assigned to each p_{ij} and p_{ij} can be gradually refined through working with the internal control system. Also for each node x_j, p_{ij} is defined such that

$$\sum_{i=1}^{k} p_{ij} = 1$$

where *k* is the indegree of the vertex representing the node x_j. For example, Figure 2 (from an expenditure cycle) shows propagation probabilities for propagation paths into the node *Match 3 Documents & Record Payable(4)*. In this case if node 4 is found to contain an error, then the error most likely has propagated from node 3, i.e., *Receive & Record Invoice*. When the model is first implemented, there may not be knowledge about the propagation probabilities. In that case, equal probability is assigned to each p_{ij}. Guan and Graham [1996] have shown an approach to updating the probabilities p_{ij} based on results of testing using stochastic approximation-based methods [Saridis, 1977].

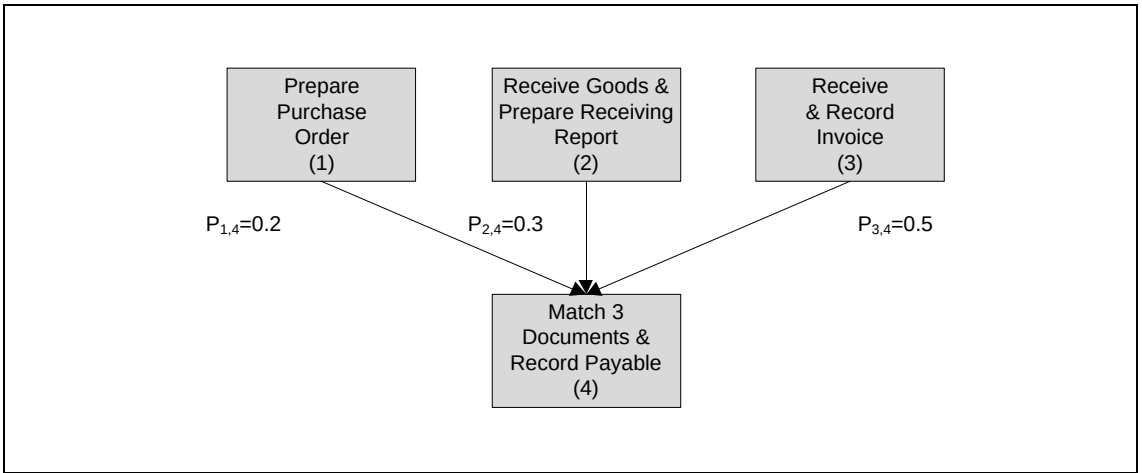

Figure 2. Error Propagation Probability

The adjacency matrix for the example system is given as follows:

$$\mathbf{M} = \begin{bmatrix}
1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&0&0&1&1&1&0&0\\
0&0&0&0&0&0&0&1&0&0&1&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&1\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&1&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{bmatrix}$$

The reachability matrix ***R*** is defined as follows:

$$\mathbf{R} = (\mathbf{M}+\mathbf{I})^r = (\mathbf{M}+\mathbf{I})^{r-1} \neq (\mathbf{M}+\mathbf{I})^{r-2}$$

where ***I*** is the identity matrix, *r* is an integer, ***(M+I)***^{r} is the r^{th} power of ***(M+I)***, and the operations are boolean. The reachability matrix ***R*** for the digraph in Figure 1 is calculated as follows:

$$\mathbf{R} = \begin{bmatrix}
1&1&0&1&1&1&0&1&1&0&1&1&0&0&1&1&1&1&0&0&1&0\\
0&1&0&0&1&1&0&1&1&0&1&1&0&0&1&1&1&1&0&0&1&0\\
0&0&1&0&0&0&1&0&0&1&0&0&1&1&0&0&0&1&1&1&1&1\\
0&0&0&1&0&1&0&1&0&0&1&0&0&0&1&1&1&1&0&0&1&0\\
0&0&0&0&1&1&0&1&1&0&1&0&0&0&1&1&1&1&0&0&1&0\\
0&0&0&0&0&1&0&1&0&0&1&0&0&0&1&1&1&1&0&0&1&0\\
0&0&0&0&0&0&1&0&0&1&0&0&1&1&0&0&0&1&1&1&1&1\\
0&0&0&0&0&0&0&1&0&0&1&0&0&0&1&1&1&1&0&0&1&0\\
0&0&0&0&0&1&0&1&1&0&1&1&0&0&1&1&1&1&0&0&1&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&1\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&1&1&0&0&1&1&1&1&0&0&1&0\\
0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&1&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0&0&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{bmatrix}$$

The reachability matrix as defined above can be processed to extract important properties of the matrix [Narayanan and Viswanhadam, 1987; Warfield, 1974]. Two partitions or operations defined on the reachability matrix ***R*** are the level partition and the separate parts partition. Let $RS(x_i)$, the reachability set, be the set of vertices reachable from x_i and $AS(x_i)$, the antecedent set, be the set of vertices which reach x_i. In other words $RS(x_i)$ refers to the set

of financial operations and/or assertions that node x_i can affect and $AS(x_i)$ refers to the set of financial operations that can affect x_i. Then the level partition $L(\mathbf{R})$ is defined as

$$L(\mathbf{R}) = [L_1; L_2; ...; L_l]$$

where l is the number of levels. If the 0^{th} level is defined as the empty set, $L_0 = \varnothing$, the level partition of $\boldsymbol{R}$ can be found iteratively as follows

$$L_j = \{x_j \in \mathbf{R} - L_0 - L_1 ... - L_{j-1} \,|\, RS_j(x_i) = RS_j(x_i) \cap AS_j(x_i)\},\ j = 1, l \text{ and } i = 1, n$$

The levels so obtained have the following properties:

a) $\cup L_i = V,\ for\ i = 1, l$

b) $L_i \cap L_j = \phi\ for\ i \neq j$

c) Edges leaving vertices in level L_i can go only to vertices in levels L_j such that $i \leq j$. In other words, an error in nodes in one level can impact only nodes in the same or lower levels. Please note lower-numbered levels appear in the lower parts of the level partition drawing (Figure 3).

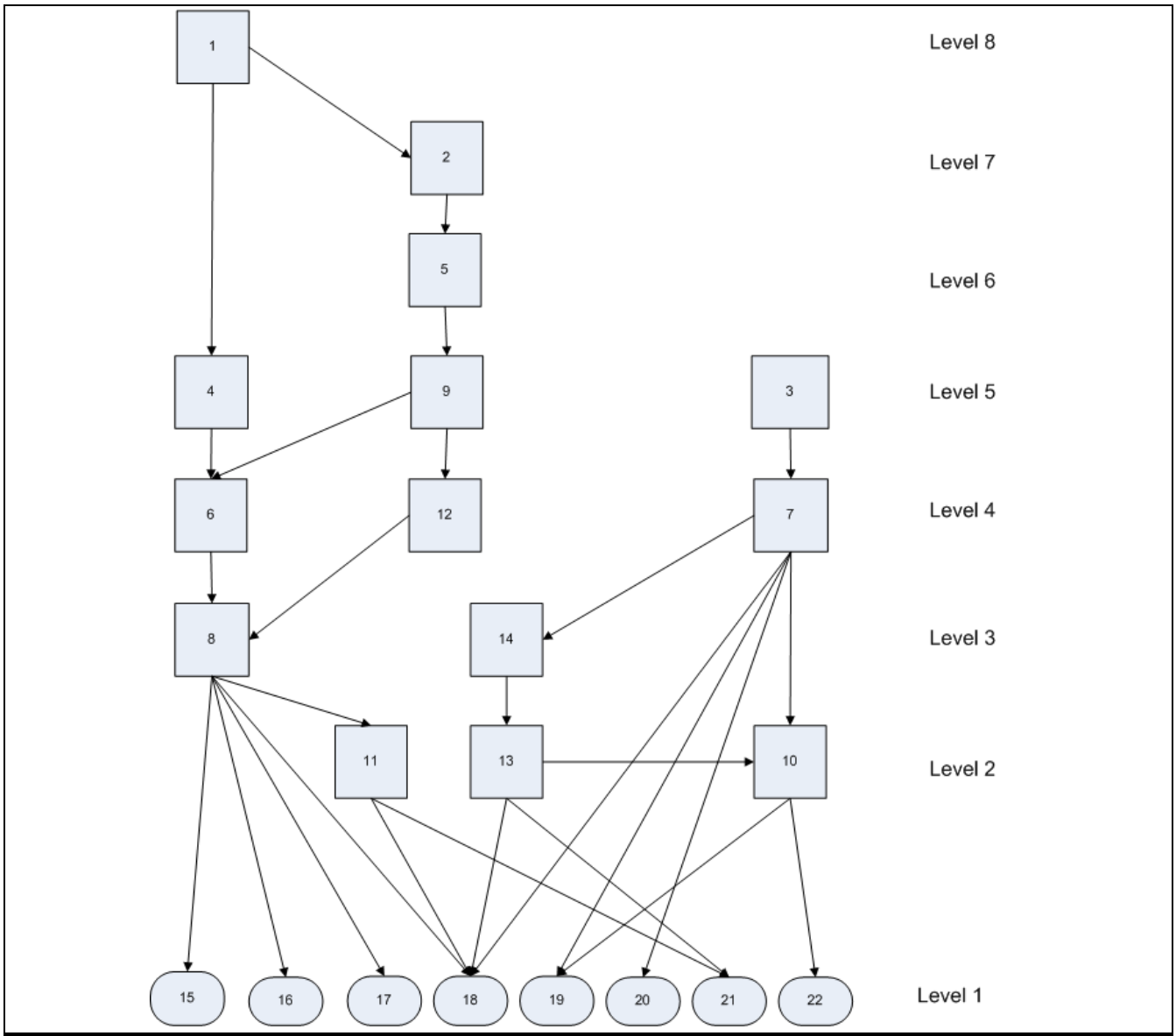

Figure 3. Level Partitioned Digraph of the Financial Reporting System

Level partitioning can be performed through tabulation. See [Rao, 1984] for an efficient algorithm of level partitioning. Tables 2–9 show the process of deriving the eight level partitions for the example in Figure 1. Therefore, we have the following level partitions:

$$L = \{(15,16,17,18,19,20,21,22),(10,11,13),(8,14),(6,7,12),(3,4,9),(5),(2),(1)\}$$

and

$$L_1 = \{15,16,17,18,19,20,21,22\}$$

$$L_2 = \{10,11,13\}$$

$$L_3 = \{8,14\}$$

$$L_4 = \{6,7,12\}$$

$$L_5 = \{3,4,9\}$$

$$L_6 = \{5\}$$

$$L_7 = \{2\}$$

$$L_8 = \{1\}$$

For example, in Table 3 the level partition L_2 consists of the nodes *10*, *11*, and *13,* as those nodes meet the condition that their reachability set is equal to the intersection of the reachability set and the antecedent set. The level partitioned digraph is given in Figure 3.

Table 2: Level Partition 1

x_i	$RS(x_i)$	$AS(x_i)$	$RS(x_i) \cap AS(x_i)$
1	1,2,4,5,6,8,9,11,12,15,16,17,18,21	1	1
2	2,5,6,8,9,11,12,15,16,17,18,21	1,2	2
3	3,7,10,13,14,18,19,20,21,22	3	3
4	4,6,8,11,15,16,17,18,21	1,4	4
5	5,6,8,9,11,15,16,17,18,21	1,2,5	5
6	6,8,11,15,16,17,18,21	1,2,4,5,6,9	6
7	7,10,13,14,18,19,20,21,22	3,7	7
8	8,11,15,16,17,18,21	1,2,4,5,6,8	8
9	9,6,8,11,12,15,16,17,18,21	1,2,5,9	9
10	10,19,22	3,7,10	10
11	11,18,21	1,2,4,5,6,8,9,11	11
12	12,8,11,15,16,17,18,21	1,2,9,12	12
13	13,18,21	3,9,13	13
14	14,10,13,18,21,22	3,7,14	14
15	***15***	***1,2,4,5,6,8,9,12,15***	***15***
16	***16***	***1,2,4,5,6,8,9,12,16***	***16***
17	***17***	***1,2,4,5,6,8,9,12,17***	***17***
18	***18***	***1,2,3,4,5,6,7,8,9,11,12,13,14,18***	***18***
19	***19***	***3,9,10,19***	***19***
20	***20***	***3,7,20***	***20***
21	***21***	***1,2,3,4,5,6,7,8,9,11,12,13,14,21***	***21***
22	***22***	***3,7,10,14,22***	***22***

Table 3: Level Partition 2			
x_i	$RS(x_i)$	$AS(x_i)$	$RS(x_i) \cap AS(x_i)$
1	1,2,4,5,6,8,9,11,12	1	1
2	2,5,6,8,9,11,12	1,2	2
3	3,7,10,13,14	3	3
4	4,6,8,11	1,4	4
5	5,6,8,9,11	1,2,5	5
6	6,8,11	1,2,4,5,6,9	6
7	7,10,13,14	3,7	7
8	8,11	1,2,4,5,6,8	8
9	9,6,8,11,12	1,2,5,9	9
10	***10***	***3,7,10***	***10***
11	***11***	***1,2,4,5,6,8,9,11***	***11***
12	12,8,11	1,2,9,12	12
13	***13***	***3,9,13***	***13***
14	14,10,13	3,7,14	14

Table 4: Level Partition 3			
x_i	$RS(x_i)$	$AS(x_i)$	$RS(x_i) \cap AS(x_i)$
1	1,2,4,5,6,8,9,12	1	1
2	2,5,6,8,9,12	1,2	2
3	3,7,14	3	3
4	4,6,8	1,4	4
5	5,6,8,9	1,2,5	5
6	6,8	1,2,4,5,6,9	6
7	7,14	3,7	7
8	***8***	***1,2,4,5,6,8***	***8***
9	9,6,8,12	1,2,5,9	9
12	12,8	1,2,9,12	12
14	***14***	***3,7,14***	***14***

Table 5: Level Partition 4			
x_i	$RS(x_i)$	$AS(x_i)$	$RS(x_i) \cap AS(x_i)$
1	1,2,4,5,6,9,12	1	1
2	2,5,6,9,12	1,2	2
3	3,7	3	3
4	4,6	1,4	4
5	5,6,9	1,2,5	5
6	***6***	***1,2,4,5,6,9***	***6***
7	***7***	***3,7***	***7***
9	9,6,12	1,2,5,9	9
12	***12***	***1,2,9,12***	***12***

Table 6: Level Partition 5			
x_i	$RS(x_i)$	$AS(x_i)$	$RS(x_i) \cap AS(x_i)$
1	1,2,4,5,9	1	1
2	2,5,9	1,2	2
3	***3***	***3***	***3***
4	***4***	***1,4***	***4***
5	5,9	1,2,5	5
9	***9***	***1,2,5,9***	***9***

Table 7: Level Partition 6			
x_i	$RS(x_i)$	$AS(x_i)$	$RS(x_i) \cap AS(x_i)$
1	1,2,5	1	1
2	2,5	1,2	2
5	***5***	***1,2,5***	***5***

Table 8: Level Partition 7			
x_i	$RS(x_i)$	$AS(x_i)$	$RS(x_i) \cap AS(x_i)$
1	1,2	1	1
2	***2***	***1,2***	***2***

Table 9: Level Partition 8			
x_i	$RS(x_i)$	$AS(x_i)$	$RS(x_i) \cap AS(x_i)$
1	***1***	***1***	***1***

Another partition is defined for the digraph that can help with the error tracing process, i.e., the separate parts partition [Warfield, 1974]. It is possible that some of the nodes of *G* (the digraph representing the internal control system) constitute a smaller digraph which is disjoint from the remainder of the digraph. The separate parts partition is used to identify these disjoint parts of the internal control system. To define the separate parts partition, a set of bottom-level nodes must be defined. *B* is a set of bottom-level nodes, if and only if, for any $x_i \in B$

$$AS(x_i) = RS(x_i) \cap AS(x_i)$$

Given the reachability matrix ***R*** as defined above, a separate parts partition *SP(**R**)* can be defined as

$$SP(\mathbf{R}) = [D_1; D_2; ...; D_m]$$

where *m* is the number of disjoint digraphs which constitute the digraph ***M*** represents. To find *SP(**R**)*, the set of bottom-level nodes *B* must be found, where

$$B = \{x_i \in \mathbf{R} | AS(x_i) = RS(x_i \cap AS(x_i)\}$$

Then, any two nodes $x_i, x_j \in B$ are placed in the same block if and only if

$[RS(x_i) \cap RS(x_j)] \neq [\varnothing]$ Once the nodes of *B* have been assigned to blocks, the remaining nodes of the reachability sets for each block are appended to the block.

Assessing Impact of Risk

Once corporate personnel or external auditors have modeled the internal control system as described above, they can establish procedures to assist them in evaluating the internal control system and/or performing auditing tasks. This section presents and discusses an algorithm to support the requirement to identify the principal exposures and the controls designed to reduce those exposures. The elements of the internal control system model follow from the risk-management decisions, which are the sole responsibility of corporate management, after a cost-benefit analysis of the likelihood and materiality of individual risks. Auditors, in turn, can use this model in their responsibility to assess risks and to plan the audit accordingly. The inputs to the algorithm are the adjacency matrix ***M*** representing the digraph *G* and the set of nodes *F* that are assumed/known to be risky. The output of the algorithm is a set of nodes, $\Omega = \{x_1, x_2, x_3, \ldots, x_l\}$, that are likely to be affected by the identified risks, where *l* is the number of level partitions.

1. Compute the reachability matrix ***M*** of *G*.
2. Compute the level-partition of *G*.
3. Compute the separate-parts partition of *G*.
4. For each separate part
   a. find the impact set $Q = \{\cup RS(x_i) | x_i \subset F\}$

b. find the leveled impact set $LQ = \{LQ_i | \cup LQ_i = Q\}$ for *i = 1, l,* where *l* is the number of level partitions.

The final output Ω is the collection of all the leveled impact sets in all the partitions. Steps 1 through 3 preprocess the digraph representing the internal control system by finding the reachability matrix, level partitions, and separate parts partitions. Step 4.a finds the impact set in each partition. An impact set is defined as the nodes that may be impacted by the risky nodes(s) identified in each separate part partition. It follows that these are the nodes in the reachability set(s) of the risky nodes. Step 4.b divides each impact set into levels using level partitions. This helps distinguish nodes that are more likely to be impacted from those that are less likely to be impacted. Nodes in a higher level partition are more likely to be affected or more immediately affected by the risky nodes because of their closer proximity to the risky nodes.

For example, assume that management (or its auditor) wants to assess the impact of an error in operation 3 (Send to the bank any customer remittance payments erroneously sent directly to the company). Maybe the clerk at the company did not realize that this was a customer remittance payment, and thought it was a payment for something else. The next step, operation 7, will be missing that customer payment from the bank's list of customer remittance payments received. This will cause misstatements in the assertions of Accounts Receivable (AR), Cash, and Loan Payable. This algorithm will demonstrate the full impact of the risk in operation 3. In addition, it will show the auditor which operation(s) will be most immediately impacted, i.e., operation 7 or Record in QuickBooks the effect on AR, Cash, and Loan Payable based on the bank's remittance list.

Risk Identification by Company Personnel, and by Auditors During Audit Planning and Substantive Testing

This section presents and discusses algorithms for error diagnosis. Here error diagnosis refers to the process of locating the source(s) of observed misstatements in assertions. The purpose of *internal* auditing includes the improvement of the organization's operations, especially by evaluating the effectiveness of risk management and control. Thus internal auditors could use this tool to trace and repair the control problem(s) that triggered the misstatement. *External* auditors, upon finding an error during substantive testing, might merely propose an adjustment to the financial statements. But if they had a tool like this available, they, too, may trace the error to a significant control deficiency which would require prompt reporting to the Board of Directors. Error diagnosis is a fairly well-established field in applied artificial intelligence [Guan and Graham, 1994, 1996; Narayanan and Viswanhadam, 1987; Pauker, Clancey, and Shortliffe, 1976; Reggia, Nau, and Wang, 1983; Reiter, 1987]. However, researchers have produced relatively little in the domain of internal controls to help locate sources of errors. The algorithms in this section have been adapted from Guan and Graham [1994].

The algorithms help auditors navigate the internal control system to locate those financial operations that likely caused the observed errors/misstatements. The results will provide necessary evidence to support management and the external auditor's assessments of the corporation's attainment of risk management. The inputs to the algorithms are the adjacency matrix ***M*** representing the internal control system digraph *G* and the set of assertion nodes *F* that the auditor assumes/knows to be erroneous. The output of the algorithm is a set of financial operations, Ω, that are the likely root cause(s) of the observed errors. Since more than one financial operation in the internal control system may have caused the observed errors, this article will consider both the single error case and the multiple error case. In the single error case, one financial operation (unreliable ancestor) has likely led to the observed misstatements. In the multi-error case, two or more financial operations (independent unreliable ancestors) have led to the observed misstatements. In either case the following calculations are performed. Please note that these are one-time operations that need be repeated only if there were changes in the underlying financial operations.

Compute the reachability matrix ***R*** of *G*.

Compute the level-partition of *G*.

Compute the separate-parts partition of *G*.

The inputs to the single error algorithm are as follows:

- The adjacency matrix ***M*** representing the digraph *G*.
- The reachability matrix ***R*** of *G*.
- The level-partition *L* of *G*.
- The separate-parts partition of G.
- The set of observed errors, such as misstated assertions, *F,* in the internal control system.

The output of the algorithm is a set of nodes, Ω, that are likely causes of the observed misstatements. The diagnostic process starts when misstatements are observed. If there is one or more common ancestor(s) to the misstatements, the single error algorithm is invoked. If there is no common ancestor to the misstatements, the multi-error algorithm is invoked. If the number of separate parts is greater than 1, then ancestors of the misstatements are computed for each separate part and the single error or multi-error algorithm is invoked according to whether there is at least one common ancestor or not. The single error algorithm is as follows:

1. $\sum = \cap AS(x_i)$ *for* $x_i \in F$
2. $\Omega = \varnothing$
3. *While* $\sum <> \varnothing$
4. $T = \{x \mid x \in \sum$ *and level(x) = min{level(*x_i*) for all* $x_i \in \sum\}\}$[1]
5. *TESTCONTROLS(T)*
6. *Return* Ω *as the error source*

where the TESTCONTROLS algorithm is as follows:

TESTCONTROLS(T)

1. *If* $|T| = 1$
2. $v = T$
3. *If* $|T| > 1$
4. $v \in T$ *such that* p_{vz} = *max{*p_{xz} *for all* $x \in T$ *and z is the common descendent of all* $x \in T$*}*
5. *If v's controls have not been tested*
6. $\sum = \sum - \{v\}$
7. *Test the controls associated with node v and mark v as tested*
8. *If the tests reveal control risk to be high*
9. $\Omega = \Omega \cup \{v\}$[2]
10. *If Ancestor(v) <>* $\varnothing$
11. *T = Immediate Ancestor(v)*
12. *If* $|T| <> \varnothing$
    TESTCONTROLS(T)

The logic of this algorithm will help the auditor navigate the complex internal control system to search for a possible error source when the misstatements have common ancestors. The assumption is that weak controls of one financial operation (node) resulted in the misstatements (they are represented as assertion nodes in *F*, which is part of the input to the algorithm). In auditing the auditor's job is to identify the weak controls of a financial system through testing controls of operations. The algorithms in this section are designed to backtrack through the complex internal control system of financial operations so that the auditor can more efficiently utilize this important source of knowledge, i.e., the structure of an internal control system [Frederick, 1991]. The diagnosis algorithms guide the auditor in choosing a node to test so that the result can reduce the search space maximally and lead the search in the most likely error propagation path.

The algorithm starts by first finding the common ancestors of the misstatements (Step 1). This set of common ancestors, ∑, the set of candidate error sources, is defined as the intersection of the ancestors of the misstatement nodes in *F*. The algorithm will then guide the auditor to test the controls of these nodes and/or their ancestors to identify error source. The "while" loop (Steps 3–5) examines each of the potential error sources in ∑ starting with the one at the lowest level of the level-partitioned digraph. Once the node on the lowest level is identified (Step 4), a modified depth-first search algorithm (*TESTCONTROLS*) guides the auditor by backtracking along all possible error propagation paths. In *TESTCONTROLS, T* contains the set of nodes to examine next. If *T* contains only one node, the controls of this node will be tested (Step 7).[3] Once the controls of a node are tested and found to be error-free, the node is removed from the set of candidate error sources and marked as tested (Steps 6 and 7). If the controls test error-free, the error is likely to have originated from nodes further upstream, and the ancestors of the tested node will be examined next (Steps 11 and 12). Otherwise the node whose controls have been found to be weak in the testing is added to the set of error sources Ω (Step 9). At this point the auditor has the option of stopping the diagnostic process or continuing until all the potential error sources are tested. If the search is to continue with the parent(s) of the current node, there are two possible cases. In the first case, there is only a single parent, and the auditor will test the controls associated with this parent (condition in Step 1). In the second case, there is more than

[1] If there is more than one node at the lowest level in ∑,, one of them is chosen randomly to start the testing process.
[2] The auditor can stop here once a node has been identified as the likely error source or continue further upstream.
[3] Please note that when TESTCONTROLS is first invoked from the single error source algorithm, *T* contains only one node.

one parent, and the parent from which the error is most likely to have propagated can be determined according to the propagation probabilities. This is done so that the search will stay as closely as possible on the most probable path to the source (Step 4). The idea is that a node is chosen for testing so that the test result can prune away at least one path of nodes. If the node whose controls have just been tested does not have any parent, then the recursive algorithm will allow the testing of the remaining possible error paths [Bang-Jensen and Gutin, 2000]. After the examination of all possible paths originating from the chosen node in the main algorithm, the node at the next lowest level will be used to guide the next round of testing (Step 4 in the main algorithm). This process continues until all potential error sources in ∑ are exhausted or, as noted above, when the auditor determines a node with its identified weak controls is sufficient to explain the observed misstatements.

As an example of a single error source causing the misstatements, let us assume there is an error in weighing the trucks when full for a non-custom job (node 5), propagating through nodes 9, 12, and 8, resulting in incorrect calculation of the cost of the job, causing errors in assertions 16 (cost of sales) and 17 (inventory) (see Figure 4). As a result, misstatements occur in assertions 16 and 17. The candidate error sources ∑, the intersection of the ancestors of the misstatements *{16, 17}*, will be equal to *{8,6,12,4,9,5,2,1}*. The algorithm would first identify node 8 as the common ancestor at the lowest level of the partitioned digraph (Step 2). Assume the controls of node 8 prove to be strong, resulting in error-free transactions; the error may have propagated from a node further upstream. Tracing back from node (operation) 8, auditors would presume, ex ante, that the probability that the error arose in node (operation) 12, which applies to all jobs, is higher than the probability that the error arose in node (operation) 6, which applies only to custom jobs, given that the auditors knew that the error was related to a non-custom job. If, however, the auditors had learned that the clerk who performed operation 6 was inexperienced and likely to create prices for all jobs without scrutinizing them for customization, or if the auditors did not have enough evidence to determine whether the error source was from a custom job or not, then $p_{6\text{-}8}$ would be greater and $p_{12\text{-}8}$ would be less. The auditors' experience and prior knowledge would determine the probability of each path, as is common in practice. Assume that the auditor-assigned probabilities would direct the search along the 8–12 path. First the auditor would test to assess whether operation 12 (Create Bill of Lading for all jobs) contained the necessary controls as listed in Table 1, whether they were designed properly and operating as designed. Perhaps examples of carelessness or unfamiliarity, or even fraud, would occur in that operation. But if Bills of Lading were indeed being prepared correctly based on weights received from operation 9, then attention would be directed to operation 9, and so on. Eventually, the auditor would find the erroneously processed transaction which caused the misstatement. The auditor would then recommend specific preventive controls which should be added to that process. It is quite possible, in addition, that this search would have revealed downstream nodes where additional controls could have detected the upstream error.

This article considers two cases for multi-errors. The first case occurs when separate parts partition yields more than one disjoint digraph. In this case each of the disjoint digraphs, along with their associated misstatements, can be diagnosed with the single error algorithm described above. This type of multi-error scenario can occur when the digraph represents multi-cycles in an organization and an auditor ruled out certain propagation paths, resulting in disjoint digraphs or disjoint cycles. The second multi-error case occurs when more than one financial operation could have propagated errors to all the misstatements in the assertions. The following algorithm addresses this case. As in the previous algorithm *F* represents the set of misstatements, financial assertions, in the internal control system suspected/observed to be abnormal and Ω represents the set of financial operations that are likely causes of the suspected/observed errors. The input to the algorithm is the same as that for the single error source algorithm. The following is the multi-error algorithm.

1. $\sum = \cup AS(x_i)$ for $x_i \in F$
2. $S = \varnothing$
3. While $\cup_{x \in S} RS(x) \subset F$
4. $T = \{x \mid x \in \sum$ and $level(x) = min\{level(x_i)$ for all $x_i \in \sum\}\}$
5. v = a node in T such that $|RS(v) \cap F| = max_{x \in T}(|RS(x) \cap F|)$
6. If $(RS(v) \cap F) - (\cup_{t \in S} RS(t))) <> \varnothing$
7. $S = S \cup \{v\}$
8. $\sum = \sum - \{v\}$
9. $\Omega = \varnothing$
10. While $S <> \varnothing$
11. v = the node such that $|RS(v) \cap F| = max(|RS(x_i) \cap F|$ for all $x_i \in S)$ and $level(v) = min(level(x_i)$ for all $x_i \in S)$
12. If $v \in S$, $S = S - \{v\}$
13. $T = \{v\}$
14. TESTCONTROLS(T)

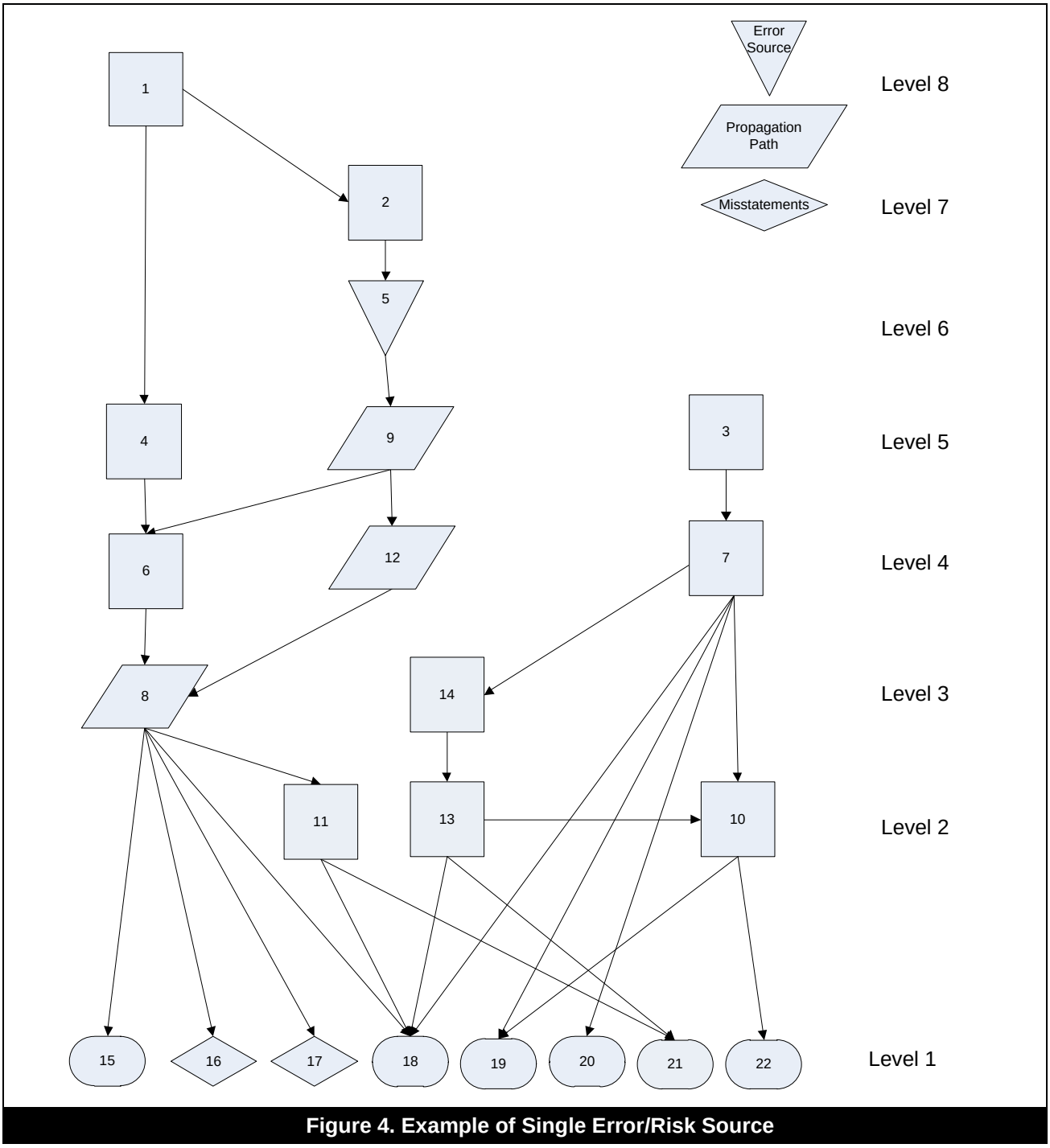

Figure 4. Example of Single Error/Risk Source

The objective of the algorithm is to find Ω, a set of financial operations whose collective failure would propagate errors to all the known misstatements, *F*. The algorithm starts by finding ∑, the set of nodes in the digraph whose failure can cause all the misstatements/assertions in *F* (Step 1). Since there is no common ancestor to the misstatements, ∑ will be the union of the ancestors of the misstatements in *F*. Next the algorithm finds *S*, the smallest set of nodes in ∑ whose weak controls could explain all the observed misstatements (Steps 3–8). The "while" loop in Steps 3–8 builds the set *S*. The loop stops when the nodes in *S* cover all the misstatements in *F* (Step 3). The objective of this step is to reduce the number of nodes to examine for possible error sources. In this loop the algorithm adds a node from ∑ to *S* if this node is at the lowest level and the failure of the node will propagate error to most nodes in *F* that are not already covered by the current nodes in *S* (Steps 4–5). Once the algorithm has added a node is to S in Step 7, it removes it from ∑ (Step 8). This process continues until *S* contains nodes whose collective failure will propagate error to all the assertion nodes in *F* (condition in line 3).

The next set of steps directs the auditor to test controls of nodes in *S* and their ancestors to localize error sources. The next node to test is the lowest level node in *S* that will propagate error to most misstatements in *F* (Step 11).

The same *TESTCONTROLS* algorithm will help the auditor navigate the part of the internal control system affecting this node (Step 14). Upon the examination of a node and its propagation paths, the algorithm removes it from further consideration (Step 12). This process is repeated until all the nodes (and their propagation paths) in *S* are tested (condition in Step 10). The set, Ω, returned by the algorithm will contain nodes whose collective failure will have propagated the error to all the misstatements in *F*.

As an example let us consider an error scenario where multiple error sources exist (see Figure 5). An error in calculating the price of a custom job (6) propagated through 8, causing errors in assertions 15 (Sales) and 18 (AR). Also, an error occurred in sending to the bank some customer remittance payments erroneously sent directly to the company (3) because a newly hired clerk did not know what to do with such checks. Instead, he put them in a drawer, intending to ask his supervisor about them when time would permit. This caused an error to propagate through operation 7 (the list of customer remittance payments on the bank's website), misstating assertion 20 (Loan Payable) and aggravating the misstatement in assertion 18 (AR). In this scenario the set of misstatements *F* comprises {15, 18, 20}. Because there is no common ancestor to all the nodes in *F,* the multi-error source algorithm will be initiated. The set of potential error sources ∑ will contain all the non-assertion nodes in the digraph except for node 10 (Step 1). The next move is to find those nodes (assigned to the set *S*) in ∑ in the lowest level (Steps 3–7), such that the nodes in S can collectively cause the observed misstatements. In the current example, the set *S* = {7, 8}. The algorithm next tests the controls of nodes in S. The testing for node 7 will lead the auditor to node 3. The testing for node 8 will lead the auditor to node 6 or node 12, depending on the probabilities. In this example, the search for poor controls stops at node 3 and node 6, but if node 6 turns out to be error-free, the search will continue upstream to the highest level of the digraph.

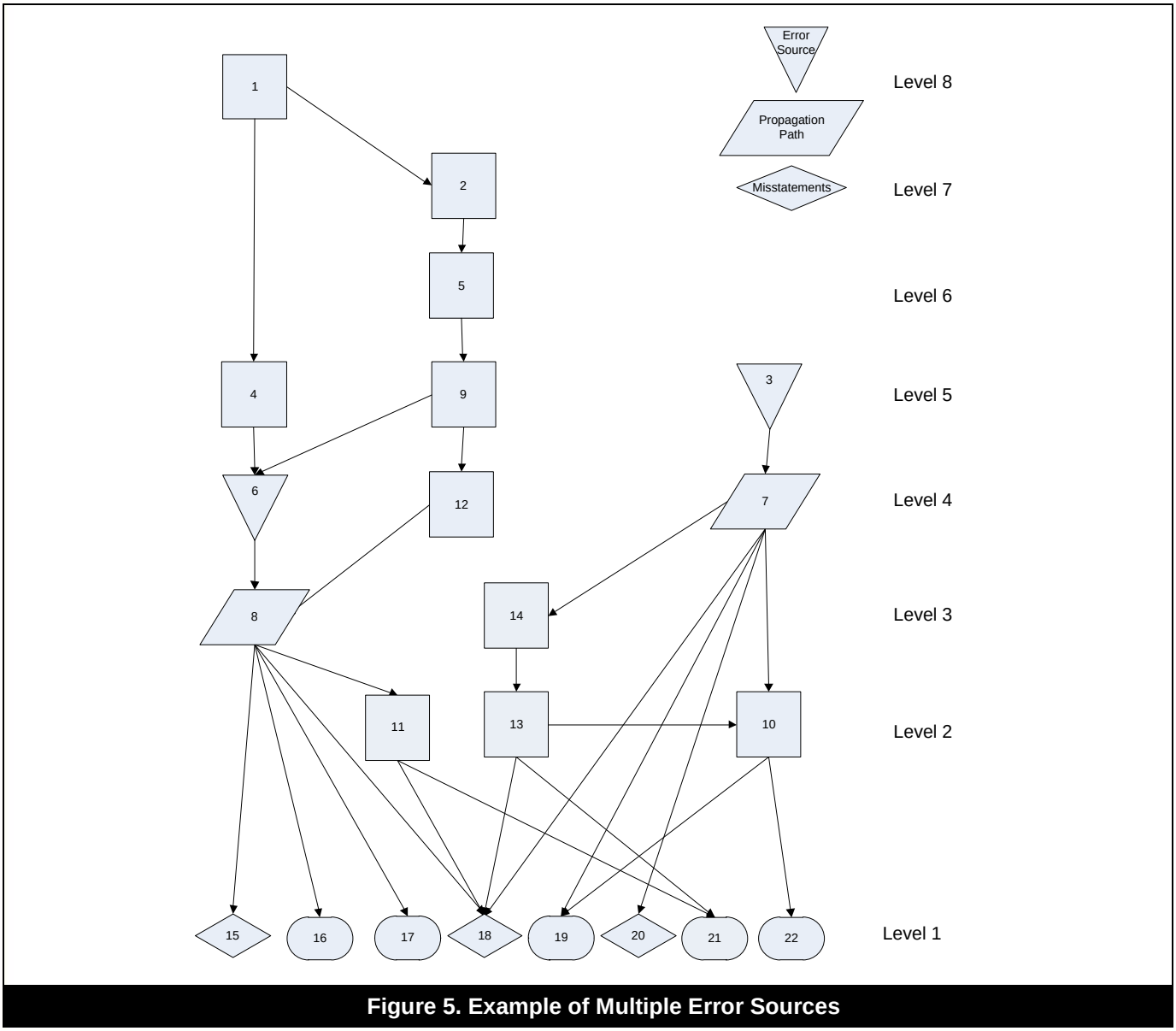

Figure 5. Example of Multiple Error Sources

Analysis of the Diagnosis Algorithms

Design science artifacts need to be validated [Hevner, March, Park, and Ram, 2004]. In addition to validation through audit scenarios in an internal control system of a real business described in the last section, this section will provide an analytical discussion (such as complexity analysis) of the algorithms [Hevner, March, Park, and Ram, 2004]. The analysis will be conducted in two parts. The first part will consider the preprocessing steps and the second part will examine the algorithms. The construction of the reachability matrix, the separate parts partition, and level partition is referred to as preprocessing steps, in that these steps need be performed once in most cases unless the underlying financial operations change and/or their propagation relationships change. The complexity of reachability matrix construction is generally $O(n^3)$ where n is the number of nodes in the digraph representing the internal control system [Warshall, 1962]. Level-structuring also has a complexity of $O(n^3)$ [Rao and Viswanadham, 1987]. For separate parts partition, the determination of the bottom level nodes can be performed in $O(n^2)$ steps, as this step consists of finding the intersection of the reachability set and ancestor set of each node and comparing the result with the ancestor set for each node. The assignment of nodes to each block is $O(n^3)$, since each block (separate part) is determined by finding the intersection of each node with all the other nodes. Therefore, the preprocessing steps are $O(n^3)$.

The risk impact algorithm has a complexity of $O(n^2)$. Given a set of financial operations the output of the algorithm is a set of other financial operations and/or assertions that may be impacted if the first set of financial operations has weak controls. After the one-time preprocessing steps described above, Step 4 finds the set of financial operations and assertions to be impacted and sorts them by levels so that auditors can see which operations are more immediately affected because of their proximity to the operations with weak controls. Step 4.a finds all the operations impacted and Step 4.b sorts them. If k is the number of risky operations/nodes, then Step 4.a loops through all the k nodes and adds their reachability sets to the result. Since each reachability set has a maximum of $(n-1)$ number of nodes (minus one because the risky node itself is excluded), Step 4.a is $O(kn)$. Step 4.b sorts by levels the impacted nodes by examining each node in the impact set Q (a $O(n-k)$ operation) and placing it in the appropriate leveled impact set. Therefore, the entire Step 4.b is $O(ln)$ where l is the number of levels in the level partition. Hence, the risk impact algorithm is $O(max(kn, ln))$ where $k \leq n$ and $l \leq n$.

The error source diagnosis algorithms are more involved in terms of complexity analysis. In the single error algorithm the first step finds the set of potential error sources or $\sum$. Again, if the number of misstatements/observed errors is k, then this step is $O(kn)$. Step 3 loops through all the nodes in $\sum$ so that their controls and their ancestors' controls may be tested to locate weak controls, resulting in the observed errors. Therefore, this step will repeat a maximum of (n-k) times, i.e., the step is $O(n)$ as the number of potential error sources cannot exceed the number of nodes. Step 4 finds the node at the lowest level in $\sum$ and, therefore, is $O(n-k)$. Step 5 is a call to the *TESTCONTROLS* algorithm, which is a variation of the well-established depth-first search algorithm. Depth-first search is $O(n+e)$, where n is the number of nodes and e is the number of edges [Bang–Jensen and Gutin, 2000]. Hence, this algorithm is $O(n(n+e))$ or $O(n^2+ne)$ in the worst case.

The multi-error algorithm is more complex than the single error algorithm. As in the single-error algorithm, the first step in the multi-error algorithm is finding the potential error sources, a $O(kn)$ operation. Steps 3–8 find the smallest set of nodes (S) whose failure or whose ancestors' failure would explain the observed misstatements. Step 3 checks to see if the union of all the reachability sets of the nodes in S covers all the misstatements in F, a $O(n)$ operation. Step 4 finds the nodes in $\sum$ at the lowest level and saves the result in T. This is $O(n)$, as $\sum$ has a maximum of $n-k$ number of nodes. If T contains more than one node, Step 5 selects the node that reaches the most nodes in F. Step 5 contains two intersections, each a $O(n)$ operation. Since T is usually a small set, we can ignore the number of times these intersections are performed in our time complexity analysis. Thus Step 5 is $O(n)$. A similar analysis shows that Step 6 is also a $O(n)$ operation and Steps 7 and 8 can be performed in constant time. In the next "while" loop Step 10 is $O(n)$, as the number of nodes in S will not exceed $(n-k)$. However, in most cases the number of nodes in S is smaller than $(n-k)$, as it represents the smallest number of nodes in the lowest levels that cover the error nodes in F. Step 11 finds the lowest level node in S that affects the most nodes in F. The intersection of the reachability set of each node in S with those nodes in F is $O(n^2)$, as the number of operations in this step is at most $(n-k)*(n-k)$. The other steps, with the exception of Step 14, can be performed in constant time. Since the loop is $O(n)$, Step 11 is $O(n^2)$, and *TESTCONTROLS* is $O(n+e)$, the multi-error algorithm is also $O(nn^2+n(n+e))$ or $O(n^3+n^2+ne)$.

As the above analysis shows, the algorithms are either $O(n^2)$ and $O(n^3+n^2+ne)$. In other words, these algorithms are tractable in terms of their running time. It is important to point out that, although the analysis in this section provides a better understanding of the algorithms, auditing is inherently a very interactive process. The main contribution of the model is its ability to guide auditors in a very interactive process (auditing) by helping auditors utilize an important knowledge source, i.e., the structural attributes of the internal control system, as opposed to just relying on experience [Frederick, 1991].

While this model's advantages may not be obvious for a small system, where an auditor needs to remember only a few processes, its value increases with the size and complexity of the company. Figure 6 illustrates how, in a moderate or large company, not only are there many data transformation processes within a cycle, but there are key inter-cycle information flows as well to processes within the other cycles before being ultimately transformed into assertions about the activities and status of the company. With all the interrelationships among the five common cycles as depicted in Figure 6, a digraph model incorporating the entire company offers the auditor a much more reliable methodology for planning which processes to test and for searching backward for error sources.

Figure 6 may also serve as an example of when separate parts partition may be used to improve the efficiency of the audit process. We might have a case in which an audit team is performing substantive tests on assertions affected by the Revenue and the Payroll/HR Cycles, therefore using a digraph with nodes from both cycles. A misstatement in the salaries paid to a salesperson might originate from a weakly controlled node in the Payroll/HR Cycle, which calculates pay for salespersons based, in part, on salespersons' commission data received from the Revenue Cycle. Alternatively, the misstatement may arise from a weakly controlled node in the Revenue Cycle which calculates and sends commissions data based on sales made.

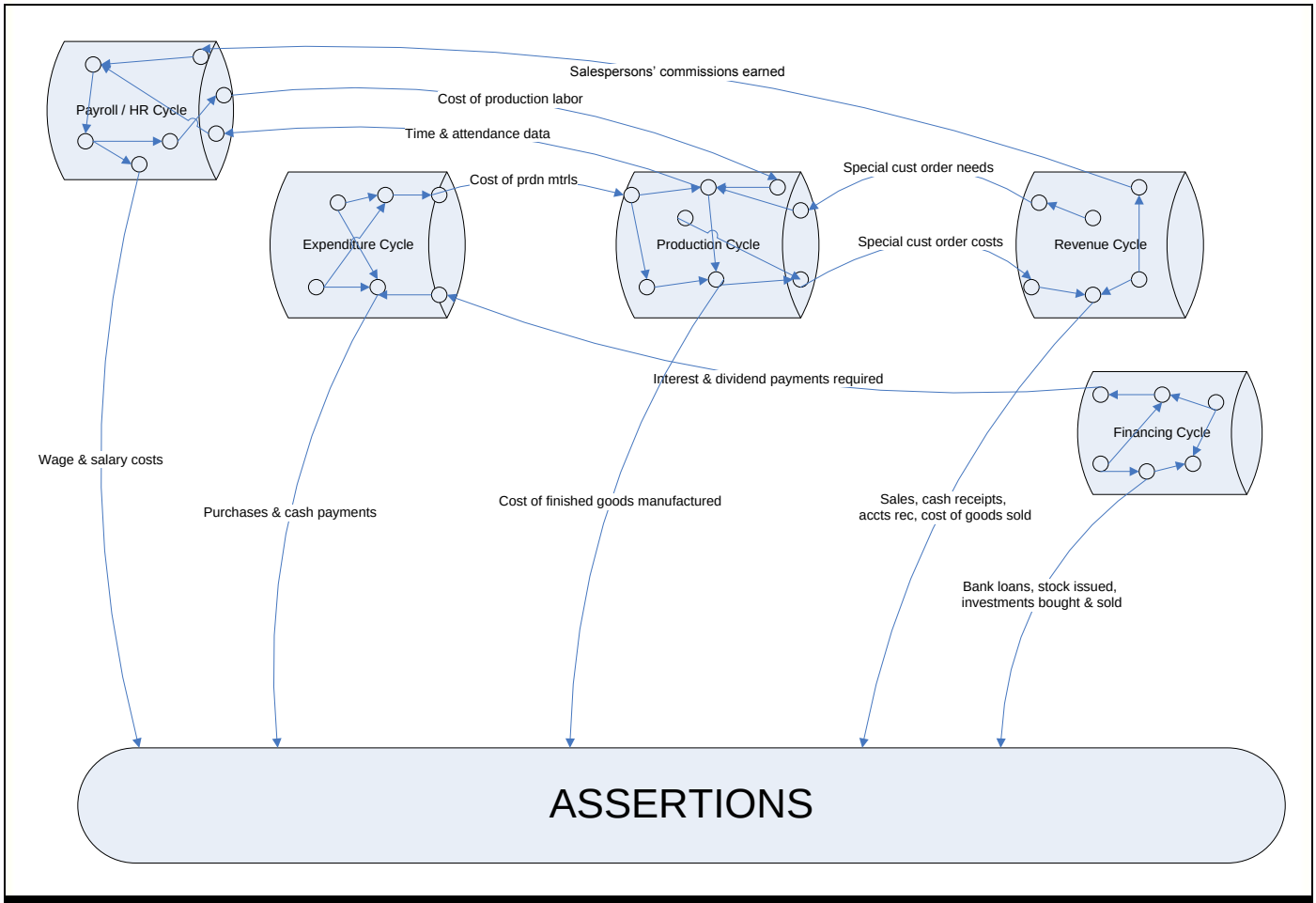

Figure 6. Information Flow Within and Among a Company's Processing Cycles

But if the misstatement is in salaries paid to salaried administrative managers rather than salespersons, the Revenue Cycle could not possibly be involved. The path from the Revenue Cycle to the Payroll/HR Cycle may be ruled out, and the Payroll/HR Cycle can be examined as a disjoint digraph with no physical connections to other cycles. This is where separate parts partition can be used to break the digraph into two separate disjoint digraphs so auditors have to navigate only a much smaller set of nodes.

Auditors' Use of Tests of Controls to Plan Substantive Testing

The previous sections followed an error or vulnerability in an assertion back to its source in a weakness in a control over a financial operation. This is useful first in the planning phase of the audit, where the auditor's objective is to design an efficient and effective set of tests of controls to focus on those key controls most essential to mitigating a material misstatement in an assertion. This algorithm quickly guides the auditor from an assertion judged particularly

material or risky back through the chain of operations and controls affecting it for more thorough testing. Furthermore, it is useful for actual errors discovered much later in the audit during the detailed substantive testing of the account balances, to find the control flaw which allowed that error to occur and not be detected. The auditor has a responsibility to disclose control weaknesses that need fortification.

However, with external auditors now required to assess controls prior to substantive testing of the assertions in all audits, this model's greatest economic value may be in the other direction, with its explicit connection starting from the controls surrounding financial operations and progressing to the assertions affected by those operations. Between the planning phase and the substantive testing phase, auditors perform actual tests of controls. After evaluating the results of these tests, auditors assign control risk more precisely to specific financial operations. Using this algorithm, auditors can quickly trace any financial operation to the affected assertion(s) in order to refine the audit plan to make it even more efficient and effective. Auditors will reallocate budgeted time toward those assertions affected by weaker controls and away from those at the end of a path of the more reliable controls.

V. CONCLUSIONS

This article introduces a graph-based model of internal control systems that focuses on the structural aspects of the system rather than the probabilistic and heuristic aspects. The proposed model can be used to support risk management, risk assessment, and audit tasks such as error diagnosis. The proposed model captures the structure of an internal control system and the propagative relationships among its structural elements. The article presents and discusses two types of algorithms based on the model to help company personnel manage risk and auditors assess risk and trace sources of errors. The use of knowledge of the structure of an internal control system is important as managers and auditors rely not only on experience and heuristic knowledge, they rely also on more structured knowledge of the transaction processing systems. The model is based on a simple conceptualization of the internal control system. It is simpler to implement and easier to maintain than the existing models [Bailey, Duke, Gerlach, Ko, Meservy, and Whinston,1985; Krishnan, Peters, Padman, and Kaplan, 2005; Wand and Weber, 1989]. Though the model has been presented as a tool for managers and auditors in evaluating the internal control system or in substantive testing, it may also serve as a possible solution to automate all or parts of error detection and analysis in auditing, especially in light of the need for fast and constant monitoring of financial systems/control systems in continuous auditing [Chou, Du, and Lai, 2007].

The digraph model presented in this article offers a simple yet workable approach to help auditors perform their tasks in this difficult environment of increasing auditing requirements and limited resources. The natural next step in this research is to validate the model through its implementation in both internal and external audit engagements. Since each company's accounting information systems and business processes are different, validation of the model's usefulness will have to be established within a company and compared against a similar audit in the same company without the model. Another promising direction for further research is incorporating a model like this into continuous auditing. As companies start to publish financial information in almost real time, the need to assure the reliability of such information is becoming more pressing. Continuous auditing is the use of automated tools to monitor internal controls. The model presented in this article may be implemented relatively simply, as it is merely a different representation of documentation that auditors already routinely accumulate.

REFERENCES

Ahituv, N., J. Halpern, and H. Will (1985) "Audit Planning: An Algorithmic Approach," *Contemporary Accounting Research* (2)1, pp. 95–110.

Bailey Jr., A.D., G. L. Duke, J. Gerlach, C. Ko, R. D. Meservy, and A. B. Whinston (1985) "TICOM and the Analysis of Internal Controls," *The Accounting Review* (60)2, pp. 186–211.

Baldwin, A.A., C.E. Brown, and B.S. Trinkle (2006) "Opportunities for Artificial Intelligence Development in the Accounting Domain: The Case for Auditing," *Intelligent Systems in Accounting, Finance and Management* (14)3, pp. 77–86.

Bang–Jensen, J., and G. Gutin (2001) *Digraphs: Theory, Algorithms and Applications*, London, England: Springer.

Bodnar, G. (1975) "Reliability Modeling of Internal Control Systems," *The Accounting Review* (50)4, pp. 747–757.

Calderon, T., and J. Cheh (2002) "A Roadmap for Future Neural Networks Research in Auditing and Risk Assessment," *International Journal of Accounting Information Systems* (3)4, pp. 203–236.

Changchit, C. (2003) "The Construction of an Internet-based Intelligent System for Internal Control Evaluation," *Expert Systems with Applications* (25)3, pp. 449–460.

Chou, C.L., T. Du, and V.S. Lai (2007) "Continuous Auditing with a Multi-agent System," *Decision Support Systems* (42)4, pp. 2274–2292.

Cooley, J.W., and B.J. Cooley (1982) "Internal Accounting Control Systems: A Simulation Program for Assessing Their Reliabilities," *Simulation & Gaming* (13)2, p. 211.

Curtis, M.B., and E.A. Payne (2008) "An Examination of Contextual Factors and Individual Characteristics Affecting Technology Implementation Decisions in Auditing," *International Journal of Accounting Information Systems* (9)2, pp. 104-121.

Cushing, B.E. (1974) "A Mathematical Approach to the Analysis and Design of Internal Control Systems," *The Accounting Review* (49)1, pp. 24–41.

Davis, J.T., A.P. Massey, and R.E.R. Lovell (1997) "Supporting a Complex Audit Judgment Task: An Expert Network Approach," *European Journal of Operational Research* (103)2, pp. 350–372.

Denna, E.L., J.V. Hansen, and R.D. Meservy (1991) "Development and Application of Expert Systems in Audit Services," *IEEE Transactions on Knowledge and Data Engineering* (3)2, pp. 172–184.

Denna, E.L., J. V. Hansen, R. D. Meservy and L. E. Wood (1992) "Case-based Reasoning and Risk Assessment in Audit Judgment," *International Journal of Intelligent Systems in Accounting and Finance Management* (1)3, pp. 163–171.

Felix Jr., W.L., and M.S. Niles (1988) "Research in Internal Control Evaluation," *Auditing: A Journal of Practice & Theory* (7)2, pp. 43–60.

Frederick, D.M. (1991) "Auditors' Representation and Retrieval of Internal Control Knowledge," *The Accounting Review* (66)2, pp. 240–258.

Gadh, V.M., R. Krishnan, and J.M. Peters (1993) "Modeling Internal Control and Their Evaluation," *Auditing: A Journal of Practice & Theory* (12)4, pp. 113–129.

Guan, J., and J.H. Graham (1994) "Diagnostic Reasoning with Fault Propagation Digraph and Sequential Testing," *IEEE Transactions on Systems, Man, and Cybernetics* (24)10, pp. 1552–1558.

Guan, J., and J.H. Graham (1996) "An Integrated Approach for Fault Diagnosis with Learning," *Computers in Industry* (32)1, pp. 33–51.

Hevner, A.R., S. T. March, J. Park, and S. Ram (2004) "Design Science in Information Systems Research," *Management Information Systems Quarterly* (28)1, pp. 75–106.

Hunton, J. and J. Rose (2010) "21st Century Auditing: Advancing Decision Support Systems to Achieve Continuous Auditing," *Accounting Horizons* 24(2), pp. 297-312.

Kelly, K.P. (1985) "Expert Problem Solving Systems for the Audit Planning Process," unpublished Ph.D. dissertation, University of Pittsburgh, PA.

Koskivaara, E. (2004) "Artificial Neural Networks in Analytical Review Procedures," *Managerial Auditing Journal* (19)2, pp. 191–223.

Krishnan, R., J. Peters, R. Padman, and D. Kaplan (2005) "On Data Reliability Assessment in Accounting Information Systems," *Information Systems Research* (16)3, pp. 307–326.

Lenard, M.J., P. Alam, D. Booth, and G. Madey (2001) "Decision-making Capabilities of a Hybrid System Applied to the Auditor's Going-Concern," *Intelligent Systems in Accounting, Finance and Management* (10)1, pp. 1–24.

Looi, C.K., S. L. Tan, P. C. Teow, and H. S. Chan (1989) "A Knowledge-based Approach for Internal Control Evaluation," *Proceedings of the Second International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems—Volume 1,* pp. 254–261.

McNamee, D., and G. Selim (1999) "The Next Step in Risk Management," *Internal Auditor* (56)3, pp. 35–38.

Meservy, R.D., A. Bailey, and P. Johnson (1986) "Internal Control Evaluation: A Computational Model of the Review Process," *Auditing: A Journal of Practice and Theory* (6)1, pp. 44–74.

Narayanan, N H., and N. Viswanhadam (1987) "A Methodology for Knowledge Acquisition and Reasoning in Failure Analysis," *IEEE Transactions on Systems, Man, and Cybernetics* (17)2, pp. 274 - 288.

O'Donnell, E., V. Arnold, and S.G. Sutton (2000) "An Analysis of the Group Dynamics Surrounding Internal Control Assessment in Information Systems Audit and Assurance Domains," *Journal of Information Systems* (14)s-1, pp. 97–116.

O'Leary, D. (2003) "Auditor Environmental Assessments," *International Journal of Accounting Information Systems* (6)4, pp. 275–294.

Pauker, S G., W.J. Clancey, and E.H. Shortliffe (eds.) (1984) *Readings in Medical Artificial Intelligence: The First Decade*, Reading, MA: Addison-Wesley.

Ramos, M. (2006) *How to Comply with Sarbanes-Oxley Section 404: Assessing the Effectiveness of Internal Control*, Hoboken, NJ: Wiley.

Rao, S.V.N., and N. Viswanadham (1984) "Graph Algorithms for Fault Diagnosis in Large Scale Systems," *Technical Report, HIREL-SA-8, Indian Institute of Science, Bangalore, India*.

Reggia, J.A., D.S. Nau, and P.Y. Wang (1983) "Diagnostic Expert Systems Based on a Set Covering Model," *International Journal of Man-machine Studies* (19)5, pp. 437–460.

Reiter, R. (1987) "A Theory of Diagnosis from First Principles," *Artificial Intelligence* (32)1, pp. 57–95.

Srivastava, R.P., and G.R. Shafer (1992) "Belief-function Formulas for Audit Risk," *Accounting Review* (67)2, pp. 249–283.

Stratton, W.O. (1981) "Accounting Systems: The Reliability Approach to Internal Control," *Decision Sciences* (12)1, pp. 51–67.

Wand, Y., and R. Weber (1989) "A Model of Control and Audit Procedure Change in Evolving Data Processing Systems," *The Accounting Review* (64)1, pp. 87–107.

Warshall, S. (1962) "A Theorem on Boolean Matrices," *Journal of the ACM* (9)1, pp. 11-12.

Warfield, J.N. (1974) *Structuring Complex Systems*, Columbus, OH: Battelle Memorial Institute.

Weidenmier, M.L., and S. Ramamoorti (2006) "Research Opportunities in Information Technology and Internal Auditing," *Journal of Information Systems* (20)1, pp. 205–219.

Whittington, O.R., and K. Pany (2001) *Principles of Auditing and Other Assurance Services*, Boston, MA: Irwin/McGraw-Hill.

Yu, S., and J. Neter (1973) "A Stochastic Model of the Internal Control System," *Journal of Accounting Research,* pp. 273–295.

ABOUT THE AUTHORS

Jian Guan is an Associate Professor of Computer Information Systems at the College of Business, University of Louisville. His research interests include accounting information systems, data mining, and knowledge management. He has published in journals such as *Journal of Information Systems*, *Computers in Industry* and *IEEE Transactions on Systems, Man, and Cybernetics*.

Alan S. Levitan, DBA, CPA, is a Professor of Accountancy at the College of Business, University of Louisville. Prior to that, he worked in Chicago as a CPA, consultant, and corporate controller. His research interest is in accounting information systems, and he has published widely in accounting and in information systems.

Communications of the Association for Information Systems

ISSN: 1529-3181