Business Process Modeling from the Control Perspective: The AI Planning Approach

Joseph Natovich[1]* and Miklos A. Vasarhelyi[2]
[1]State University of New York at Albany, USA,
[2]Rutgers University, USA

Abstract **This paper proposes a new approach for modeling business processes using the AI-Planning paradigm. Based on the concepts of agents, actions, constraints and goals, the AI-Planning approach allows explicit representation of various types of internal controls. Different threats to business processes, such as fraud, can be modeled as planning tasks—that is, finding a sequence of actions that intend to achieve a defined set of goals. Applying modeled threats to models of business processes, a planning reasoner can generate hypothetical scenarios of exposures. Because of its ability to explicitly represent threats, controls and exposures, we argue that the AI-Planning approach is useful for business process modeling from the control perspective. © 1997 by John Wiley & Sons, Ltd.**

Intell. Sys. Acc. Fin. Mgmt. **6**: 121–139, 1997.

No. of Figures: 9. No. of Tables: 0. No. of References: 37.

INTRODUCTION

Business systems are not failure-proof machines. They are transaction processing systems comprising agents, each of whom is responsible for only a few actions in the process. Working separately and asynchronously, the agents cannot observe directly the entire process and the work done by other agents. Instead, they communicate by sending and receiving signals, usually in the form of documents and records. To coordinate their operations, agents are provided with role descriptions, formal or informal, that instruct them on how to act upon observing different signals or receiving various documents.

Ideally, agents act according to their assigned role, and thus the transactions are processed correctly. Unfortunately, human agents are driven by personal goals that might interfere with their role in the process. In addition, they are subject to various errors causing the process to deviate from its normal sequence of activities and incurring losses to the company. To prevent or detect errors and frauds and to assure the correct processing of transactions, the business system contains various constraints termed internal controls.

This paper proposes a new approach to modeling business processes within the AI-Planning paradigm. Based on concepts of agents, actions, constraints and goals, the AI-Planning approach allows explicit representation of various types of internal controls.

* Correspondence to: Joseph Natovich, Department of Accounting and Law, School of Business Administration, SUNY Albany, 1400 Washington Av., Albany, NY 12222, USA.

CCC 1055-615X/97/020121-19$17.50

Received June 1995
Revised February 1997

Threats to the business process, such as fraud, can be represented as a knowledge-based model comprising additional set of actions, defined goals and planning strategies. Applying the modeled threats to the model of the business process, a planning reasoner can generate hypothetical scenarios of exposure. Because of its ability to explicitly represent threats, controls and exposures, the AI-Planning approach should be useful for business process modeling from the control perspective. To demonstrate and evaluate this approach, we have implemented it in a working prototype of a modeling and analysis tool.

The following section discusses the motivation for our work. The third section presents the planning paradigm and the STRIPS representation for planning problems, which are the foundation of the proposed modeling approach. The fourth section presents the concepts of the business process modeling within the AI-Planning paradigm. The fifth section discusses the model strengths compared to current modeling approaches. The sixth section raises problems with the approach, which are to be addressed by future research.

BACKGROUND

Curtis *et al*. (1992) classify process modeling into four perspectives: functional, behavioral, organizational and informational. Auditors, however, have a different perspective—the control perspective. While the functional perspective focuses on the prescribed business processes, the control perspective focuses on the potential **deviations from the prescribed process** (termed also as **exposures**). For example, in a vendor disbursement process, the functional perspective deals with questions such as who issues the payment and how long the payment process takes. The control perspective, however, deals with questions such as (1) whether a payment to a vendor can be issued with no appropriate approval, and (2) whether a payment issued for the vendor can be diverted to an ineligible payee. Such deviations from the prescribed process occur when **threats** such as errors and frauds are not addressed by the internal controls.

Hypothesizing about exposures is a task widely performed by different professionals including internal auditors, EDP auditors, fraud examiners and external auditors. When assessing the likelihood of fraud, for example, one of the first things they do is to 'think like a crook', generating hypothetical fraud scenarios that might occur. This task is performed whether evaluating internal controls in existing systems or designing controls for new systems. This task may be difficult for auditors because fraud occurs relatively infrequently in the life of auditors (Loebbecke *et al*., 1989), and generating hypotheses about infrequent threats is difficult (Bonner and Pennington, 1991).

Unfortunately, current models including analytical models, (e.g. Yu and Neter, 1973; Cushing, 1974), simulation models, (e.g. Burns and Loebbecke, 1975; Knechel, 1985) and knowledge-based models (e.g. Bailey *et al*., 1985; Meservy *et al*., 1986; Hamscher, 1992; Gadh *et al*., 1993) do not fully support the task of generating hypothetical exposure.

The REAL model (Hollander *et al*., 1996) provides a theory based upon agents, events, resources and locations in business process models. In this work, we use similar concepts within knowledge-based modeling approach that has some unique features. It (1) allows modeling threats to the business process including fraud; (2) requires as input only a description of controls' mechanism without preliminary assumptions about their effectiveness; and (3) can generate exposure hypotheses as detailed scenarios of deviation from the prescribed process. Hence, we expect this modeling approach to be more useful than other approaches for control professionals.

THE AI PLANNING PARADIGM

The foundation of the modeling technique we use is based on the planning paradigm that has emerged from artificial intelligence research. Planning involves the construction of some plan of **actions** for one or more **agents** to achieve some specified **goal** or goals, given the **constraints** of the world in which these agents are operating (Georgeff, 1987). A basic representation technique is STRIPS (Fikes and Nilsson, 1971). Although a relatively old system, STRIPS (an acronym for Stanford Research Institute Problem Solver) has been most influential in the systems built to date. While much has been done to extend STRIPS in ways that are useful for particular applications, these extensions generally remain *ad hoc* (Allen, 1990). Hence, in this paper we primarily use the classic STRIPS. Nevertheless, we will point out future work which exploit some extensions to STRIPS.

STRIPS Representation of Planning Problem

Originally, STRIPS was used in the context of robot movement planning. All possible actions of the robot were modeled, including their preconditions and their effects on the state of the robot's world. By using this knowledge, the computational planner could generate a plan—i.e. construct a sequence of actions that achieves a defined set of goals, starting from a given initial state.

A STRIPS system is defined by an **initial** world model, which describes the initial state of the world, and by a set of **operators**, which correspond to actions changing the current state. The description of each operation consists of its **preconditions** (applicability conditions, expressed by a first-order formula), its **add-list** (a list of formulas that must be added to the current world model), and its **delete-list** (the list of formulas that may no longer be true and therefore must be deleted) (Lifshitz, 1990). STRIPS is the most commonly used domain representation for planning problems in different domains. It has been used in situations ranging from single-agent static domains to multi-agent, dynamic domains (Pednault, 1989).

A frequently discussed planning domain is the blocks world where the goal assigned to the robot is to stack blocks in a defined order. Figure 1 shows the STRIPS model of this domain. Starting from the initial state described at the left-hand side, the problem is to find a sequence of actions that achieves the defined goal at the right-hand side. The action 'move' is defined by STRIPS in three parts:

(1) Preconditions: defines when an action is feasible. For example, move(a,b,c) (i.e. move block a from block b to block c) is feasible only when both block a, the block moved, and block c, the target place, are clear and if block a is currently on block b.
(2) Add list: defines the effect of the action on the state in terms of the new elements added. For example, the effect of the action move(a,b,c) is that block a is on block c and the block b becomes clear.
(3) Delete list: defines the elements in the state that are deleted by the action. For example the action move(a,b,c) deletes the facts that a was on b and c was clear.

An important feature of STRIPS is that actions can be parameterized (e.g. as defined in the blocks world, the single 'move' action with the variables *B*, *F* and *T* as parameters can represent all possible block moves). This helps reduce the amount of modeling efforts.

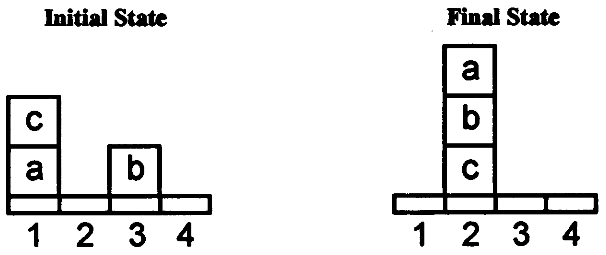
Reasoning about a Plan

An action in STRIPS representation serves as an operator that transfers the blocks from one state to the subsequent state. Based on this representation, a computational planner can generate a plan, which is a sequence of actions that transfers the blocks to a state that contains the desired goals.

The basic planning process is 'forward-chaining' (Figure 2). When given an initial state, the planner finds all actions for which their preconditions are met in the initial state. The planner selects the first of these actions, then calculates the state of the system as a result of that action, which is the previous state modified by the effect of the action. When the new state is known, the planner reasons about the next action and the state that follows and so on. If the planner reaches a state that meets the goals, the planning process stops and returns the sequence of selected actions as the solution. If the planner identifies a 'dead-end' state, where no action can be selected, then by backtracking the planner selects a different path of actions and continues with the forward-chaining process. After a solution is found, the forward-

a) A problem in the blocks world:

Find a sequence of actions that achieves the goals: a on b and b on c . These actions transform the initial state (on the left) to the final state (on the right).



b) STRIPS representation of the problem

- **A blocks world**

  blocks and places are objects

  a, b and c are blocks

  1, 2, 3, 4 are places
- **Initial state**

  2 is clear, 4 is clear, b is clear, c is clear, a is on 1, b is on 3, c is on a
- **Goals**

  a is on b, b is on c
- **Actions**

  move B from F to T :

  Preconditions to the action move B from F to T :

  B is a block, F and T are objects, B is clear, T is clear , B is on F, B is not F , F is not T , B is not T

  State transformation as a result of the action move B from F to T :

  Add elements to state: B is on T, F is clear.

  Delete elements from state: B is on F , T is clear

Figure 1 The blocks world—the classic planning problem example (Bratco, 1990)

chaining can continue with the systematic search and generate alternative solutions, if they exist.

The planning literature suggests a variety of planning algorithms that are more efficient and sophisticated than forward-chaining. Because we focus primarily on problem representation, however, a review of those planning algorithms is beyond the scope of this paper.

THE BUSINESS PROCESS WITHIN THE PLANNING PARADIGM

The plan-based modeling approach can be

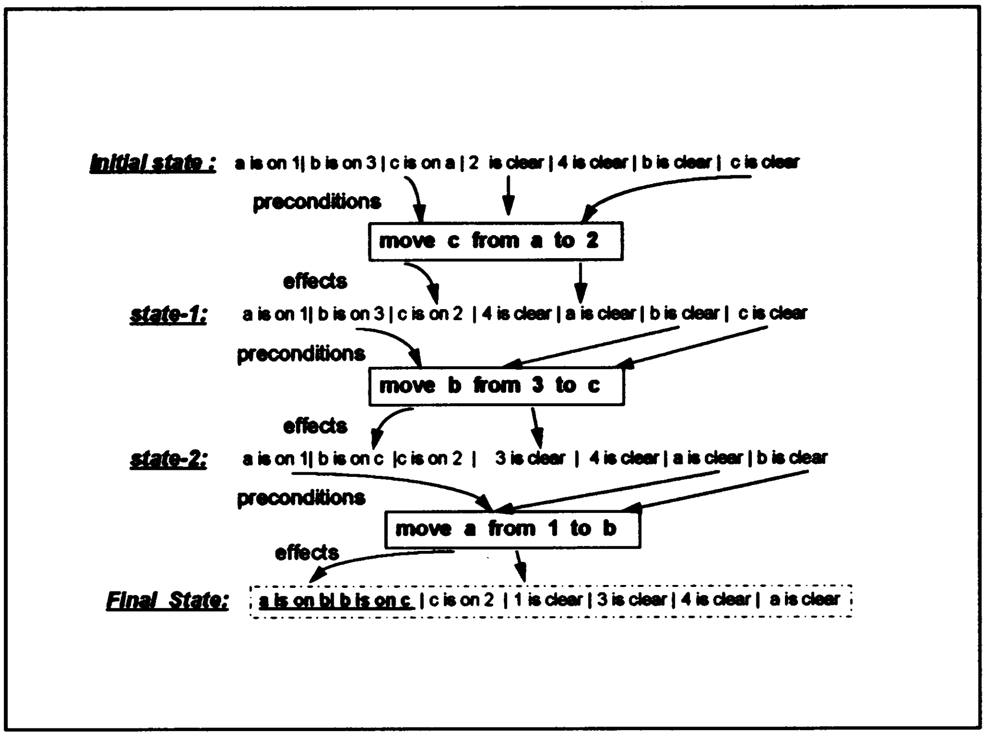

Initial state : a is on 1| b is on 3 | c is on a | 2 is clear | 4 is clear | b is clear | c is clear
preconditions
move c from a to 2
effects
state-1: a is on 1| b is on 3 | c is on 2 | 4 is clear | a is clear | b is clear | c is clear
preconditions
move b from 3 to c
effects
state-2: a is on 1| b is on c |c is on 2 | 3 is clear | 4 is clear | a is clear | b is clear
preconditions
move a from 1 to b
effects
Final State: a is on b| b is on c | c is on 2 | 1 is clear | 3 is clear | 4 is clear | a is clear

Figure 2 The forward-chaining process in the blocks world

applied in the business process domain. In this setting, instead of a single robot, there are multiple human agents: employees, contractors and other business associates. Even though human agents clearly do not behave like robots in a precondition-effect fashion and their set of possible actions is richer, it is still useful to model the role of agents in the business process using actions defined as operators. Plan-based business process modeling has been recently proposed for some functional tasks, including managing the system development process (Huff and Lesser, 1989) and business process reengineering (Ku *et al.*, 1996; Yu and Mylopoulos, 1996).

Within the plan-based model, each agent in the business process has a role, which in many cases is a routine performed for each transaction. Roles can be described in terms of actions, preconditions and effects. In some businesses there is a manual that formally describes roles and responsibilities. Even if such formal role descriptions do not exist, agents are familiar with their roles and the actions they need to perform in various situations. Role definitions instruct agents on the actions they should take, based on the observed state of the system, releasing them from the need to consider the previous actions of other agents.

Using job descriptions, a forward-chaining reasoning can produce the path of action. Good role design should allow only one or a small number of possible actions in a given state. Thus, the number of possible paths in the prescribed process is limited. Hence, assuming that agents perform their role only as prescribed, reasoning about the path of actions in the process is straightforward, and there is little need for planning techniques. As this assumption is relaxed and agent error and fraud is considered, however, in the absence of appropriate controls, additional (incorrect) actions may occur at each state. This increases possible paths including ones that deviate from the pre-

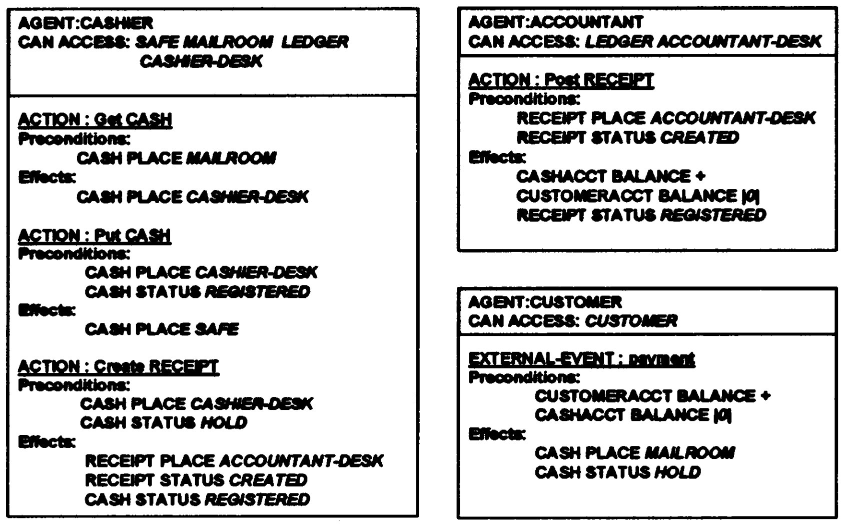


Figure 3 ROLEMAN'S role representation—cash receipts process

scribed process. Here, the strength of AI-Planning may be fully exploited to reason about possible frauds and errors and their effect on the process.

We have built **ROLEMAN**, a tool for modeling the business process within the AI-Planning paradigm. ROLEMAN is built on Allegro CL (of Franz Inc.) platform and Josie (Nado *et al.*, 1991), a frame system that is supported by a graphical user interface. ROLEMAN follows the ontology of actions developed by Bailey *et al.* (1985) and Hamscher (1992). We have also built **CROOK**, a prolog program that analyzes fraud opportunities in ROLEMAN and generates possible frauds scenarios.

In the remainder of this section, we discuss how processes, controls and threats are represented within the AI-Planning paradigm. We support our discussion using an example of ROLEMAN's model of a cash receipts process, which is shown in Figure 3. There are three agents—a customer, a cashier and an accountant. The customer triggers the process by **Payment**—making a payment and mailing it, assuming that her account balance shows debit. As a result, the cash is received at the mailroom. The cashier's role includes three actions: (a) **Get CASH**—getting the cash when it is in the mailroom and putting it on her desk; (b) **Create RECEIPT**—creating a receipt when the cash is in her desk unregistered (the result is a receipt sent to the accountant, and a change in the status of the cash to 'registered'); and (c) **Put CASH**—depositing the cash in the safe when it is registered (as a result, the cash is placed in the safe). The accountant's role is **Post RECEIPT**—posting the receipt to the ledger when she receives it unposted. As a result, the balances in the cash account and the customer account are changed, and the receipt is stamped as 'posted'.

The Process

Current models, including knowledge-based models such as TICOM (Bailey *et al.*, 1985) and Savile (Hamscher, 1992), lack the rationale underlying the actions in the process. They view the business process as a **procedure**, defining the order in which actions are executed, assuming that each action is triggered upon the completion of the previous one, and ignoring contingencies that should be considered such as availability of resources and supporting documents. The AI-Planning representation can more effectively handle these contingencies. The business process can be simulated as forward-chaining of the defined

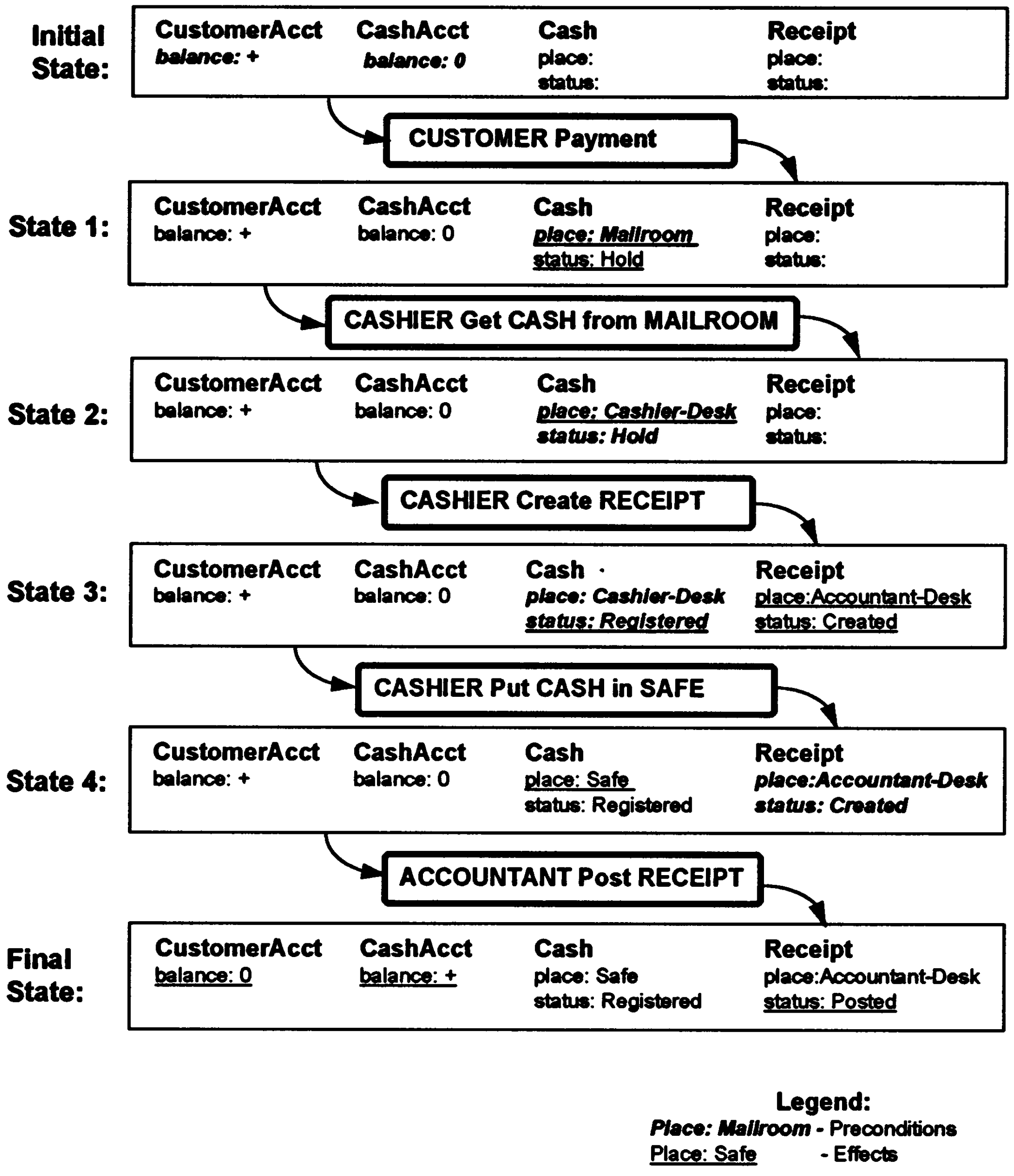


Figure 4 Forward chaining—cash receipts process

roles. In this approach, agents continually observe the state of the system. When they observe that, according to their role definition, the preconditions to an action are met, they will perform the action, resulting in a change in the state. The new state triggers a different action, which again results in a new state. This process continues until it reaches a final state where no more actions can be triggered.

Figure 4 illustrates ROLEMAN's simulation of the cash receipts process represented in Figure 3. In this chart, the properties in bold type in the states are the preconditions that trigger the next action. The underlined properties are the effects of the previous action. The CUSTOMER Payment is committed first, placing the cash in the mailroom (state 1). The existence of cash in the mailroom triggers the cashier to get the cash (state 2). This new state fulfils the preconditions for creating a receipt. The 'Create RECEIPT' action changes the status of the cash to 'registered' and initiates the object receipt (state 3). The new status of the cash prompts the agent to deposit the cash in the safe (state 4). Finally, the accountant observes the receipt she has received, and posts an entry to credit the customer account and debit the cash account, changing their balances to zero and debit respectively and stamping the receipt's status as 'posted' (final state). Since at this point no additional actions are possible, the process stops.

In our opinion, forward-chaining seems to much more closely reflect the behavior of business processes than procedural models because (1) actions are usually triggered by the system's state rather than by the previous action, and (2) actions affect the system's state rather than the next action.

Actions are Triggered by the System's State

Consider the disbursement process, where the vendor invoice goes through verifications and authorizations by different functions and is then transferred to the payment clerk. Before the clerk issues a check to the vendor, she must verify that the invoice was approved for payment. Because the clerk works separately from other agents, it is impossible for her to observe directly the previous actions they performed. Thus, how can she be sure that the invoice properly passed the authorization and verification activities? The answer is that she does not have to observe directly the authorization action. Instead, she observes some signals that inform her about these authorizations. Such signals may include the fact that the invoice was sent for payment, various signatures exist on the invoice, the signed payment order is attached to the invoice, or the credit balance of the vendor's account. The clerk's role description tells her what signals she must observe as a precondition for making the payment. If a signal is missing (for example, one of the authorized signatures) the clerk is not allowed to process the payment. Note that the clerk relies solely on the signals with no regard to the previous actions. Indeed, signals such as signatures can be falsified, and the invoice might actually receive no approval. That supports the AI-Planning modeling approach: Agents perform an action because of some signals they observe in the current state and not because of the completion of some actions.

Figure 5 shows a payment action modeled by ROLEMAN. In this example, the preconditions for the action are (1) the vendor invoice is at the clerk's desk, and (2) the invoice is stamped as 'approved' for payment.

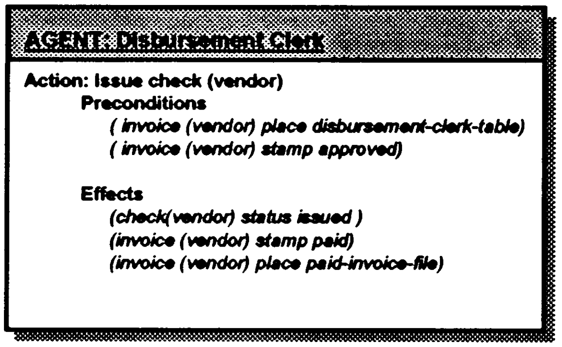
Actions affect the system's state

Continuing the example of the disbursement clerk, when the check is finally issued, the state of the system is changed. First, the new check is added. Second, and even more important, the clerk makes sure that the invoice won't be paid twice. In other words, after the payment, the system's state should be changed such that the payment preconditions will no longer be met. This change can take place in various ways. The clerk, for example, can move the invoice to a different place (e.g. file the invoice in the 'paid invoices' file). This action, however, is risky because the invoice might be removed and routed back to the clerk for payment. A better way is to attach a copy of the check to the invoice, or, even better, to stamp the invoice as 'paid'. All these actions are different ways of affecting the state of the system. Again, the role description of the agent defines how the action she performs affects the state.

Figure 5 shows ROLEMAN's representation of the action 'issue check'. The effects of the actions are (1) a new check is added bearing the initial status 'issue', (2) the vendor invoice is stamped as 'paid', and (3) the invoice is filed in a special 'paid invoices' file.

Internal Controls

To ensure the correct processing of transactions, various types of internal controls are implemented, including preventive controls, detective controls and management controls.


AGENT: Disbursement Clerk

Action: Issue check (vendor)
Preconditions
(invoice (vendor) place disbursement-clerk-table)
(invoice (vendor) stamp approved)

Effects
(check(vendor) status issued)
(invoice (vendor) stamp paid)
(invoice (vendor) place paid-invoice-file)

Figure 5 Disbursement clerk's role

Within the AI-Planning paradigm, internal controls are a set of constraints over the process that limits the possible paths of action. The role definitions do not determine a specific path of action in which the process is executed. Therefore, without appropriate constraints, and especially when threats are introduced, the process may be executed in multiple ways. The internal control constraints are implemented to rule out incorrect paths.

The AI-Planning paradigm can represent the various types of control as different types of constraints. Preventive controls are represented as constraints over actions. Detective controls are represented as constraints over the system's final state. Management controls are represented as constraints over agents and the repositories they may access.

Preventive Controls are Constraints over Actions

Preventive controls are checkpoints within the business process where some factors are examined and searched for any indication of a problem. The process is blocked until the problem is corrected. Preventive controls are modeled within the AI-Planning paradigm as constraints over actions. They require specific conditions for actions to be executed. Preventive controls are an integral part of the role of agents. Although they differ from other actions and preconditions in their purpose, they are represented similarly. They can take the form of preconditions to actions. Also, some actions can be considered as preventive controls if their whole purpose is to put constraints over subsequent actions.

For example, in the disbursement process, the action 'approve vendor invoice' (for payment), has two preconditions, which are actually preventive controls: (1) the invoice must match the purchase order, and (2) the invoice must match the receiving report. Moreover, this whole action is considered a preventive control because its purpose is to restrict payments in the subsequent actions.

Detective Controls are Constraints over the Final State of a Process

Detective controls are usually detached from the business process, and they examine the already-processed transactions to identify and report the presence of problems. Within the planning paradigm, detective controls are regarded as performed at the final state of the process (i.e. after its completion).

Detective controls check either agreement of one element to a defined standard, or agreement between two elements in the state. Figure 6 illustrates the representation of these two types in ROLEMAN. Aging, on the right-hand side, is an example of the former. This is a control over the accounts receivable process that ensures the customer account balance at the final state is zero. Aging control triggers an examination of any customer account with a non-zero balance. A debit balance might indicate that either the customer has not paid or the payment has not yet been recorded, while a credit balance might indicate a payment duplication. Cash count control (on the left-hand side of the exhibit) is an example of the latter. It compares the cash in the safe to the balance of the cash account. The cash in the safe should match the cash account balance, that is when cash exists the balance of the cash account must be debit or when cash does not exist the balance of the cash account must not be debit. In case of mismatch between the cash and the cash account, the final state is not accepted. The final state of the process shown in Exhibit 3 passes both controls because the customer account balance is 0, the cash account balance is debit and the cash is in the safe.

Management Controls are Constraints over Agents and Objects

Management controls include a broad set of measures such as recruiting and screening procedures, rotation policy, segregation of duties, and access restrictions. (Although some may argue that access controls are preventive controls, we prefer to classify them here as management controls because they are global controls that are not related to a particular process.) In this paper, we refer only to those controls that directly affect the role of agents, namely segregation of duties and access controls. Within the AI-Planning, access controls restrict the agents' ability to use objects in repositories they are unauthorized to access. Segregation of duties restricts the type of actions that the agent can perform.

In ROLEMAN, access controls are implemented for each agent by the 'can-access' slot that lists the repositories to which the agent has access. In the example of the cash receipts system (Figure 3), the cashier may access the safe, the mailroom, the ledger and her own work area (cashier-desk) while the accountant may access only the ledger and her own work area (accountant-desk). When running a simulation, agents cannot perform actions on objects that are not placed in the assigned repositories. Segregation of duties are represented by assigning actions to their authorized agents. Agents are restricted to perform only actions to which they are assigned. In the example (Figure 3), only the accountant is assigned to post the receipt.

Threats

If agents performed only the assigned actions as defined, then unless their design was incorrect, processes would always be executed correctly. In practice, however, various threats including error and fraud may affect agents' behavior, thereby causing them to deviate from their defined role. In the absence of appropriate controls, the process may be executed incorrectly and the business may suffer from losses.

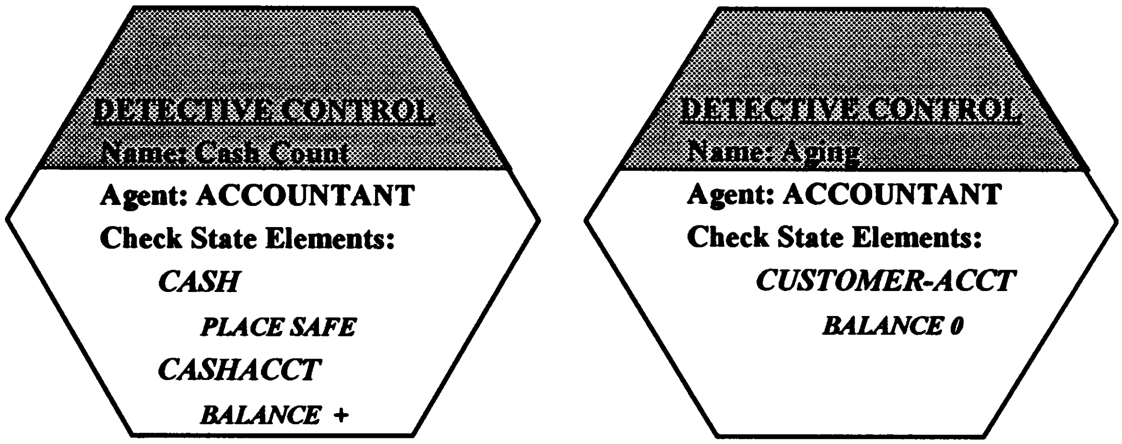


Figure 6 Detective controls representation

Fraud occurs when an agent, disregarding her assigned role, acts in a manner driven by personal goals to take possession of an asset or funds. Error occurs when an agent unintentionally performs an action not in accordance with her assigned role.

The AI-Planning paradigm allows modeling of various threats. These models can be applied to the execution of the business process, revealing their effect on agents' behavior and examining the effectiveness of the controls. These models of threats are independent of a specific business process. They can be applied to any process and they do not have to be specifically built in accordance with each.

Fraud Occurs when Agents are Motivated by Some Personal Goals

Human agents have personal goals that may interfere with those of the business. Motivated by their own goals, agents might commit fraud and cause losses to the business. A common goal of fraud is to obtain an asset from the assets used in the process. As a result, in the final state of the process, a business asset is retained by the perpetrator. However, the success of the fraud requires an additional goal—namely, the final state of the fraud plan must comply with the defined detective controls so that the fraud will not be discovered.

To achieve these goals, the perpetrator must plan the actions to be taken. In such a plan, the perpetrator behaves differently from the role she is assigned: (1) the perpetrator follows strategies to achieve her goals instead of following her role; (2) the perpetrator uses, in addition to the action defined in her role, a set of fraudulent actions such as forging a document and stealing an asset; (3) the perpetrator might override some of the preventive controls, for example issuing a check in absence of an appropriate payment approval.

The AI-Planning paradigm accommodates such computational fraud planning. Basically, planning algorithms can be classified as 'generative' planners and 'case-based' planners. While generative planners (such as forward-chaining) start their search from scratch, case-based planners use a library of previously synthesized plans or plan fragments (Weld, 1994). A simple generative planner can plan frauds (Vasarhelyi and Natovich, 1993). Johnson *et al*. (1993) have identified several cognitive tactics for constructing deception that may be useful for such planners. We have chosen to use a case-based fraud planner, however, because it is more efficient (Hammond, 1989). Furthermore, West (1988, p. 23) argues that 'very few frauds are original in concept, but there are always people who think up new variations on an old theme'. Case-based planning seems to much more closely reflect this human behavior.

CROOK is a planner based on CHEF (Hammond, 1989) and MEDIC (Turner, 1994) that generates fraud scenarios within the business process modeled by ROLEMAN. It contains domain knowledge including scripts of generic fraud schemes, a set of fraudulent actions and general strategies for plan adaptation. CROOK's algorithm is shown in Figure 7. The **Selector** searches for a promising generic fraud scheme for the specific process. The **Initiator** instantiates the variables of the generic scheme and turns it into an initial plan. Because this initial plan does not consider internal controls, it is likely to fail. The **Simulator** runs the plan, and in case of failure, the **Analyzer** finds out which controls cause it to fail and how they work. Then the **Repairer** applies a strategy to circumvent these controls and repair the problems found. The repaired plan is passed again to the **Simulator**. Because new problems can occur in the repaired plan, the process of repair, analysis and simulation is iterative. Success is not guaranteed in this process, but CROOK will not give up without a 'fight'.

Figure 8 shows an example of a fraud plan generated by CROOK. The gray boxes represent the fraudulent actions taken by CROOK. In this example, the cashier embezzles the cash and, to conceal the theft, posts a journal entry to credit the cash account against an expense account. As shown, neither aging control nor cash count controls detect the anomaly in the final state because the customer account has been credited by the accountant, leaving a balance of zero, and the cash account balance has been tampered to match the lack of cash in the safe. This fraud could have been detected if a control to examine the expense account existed.

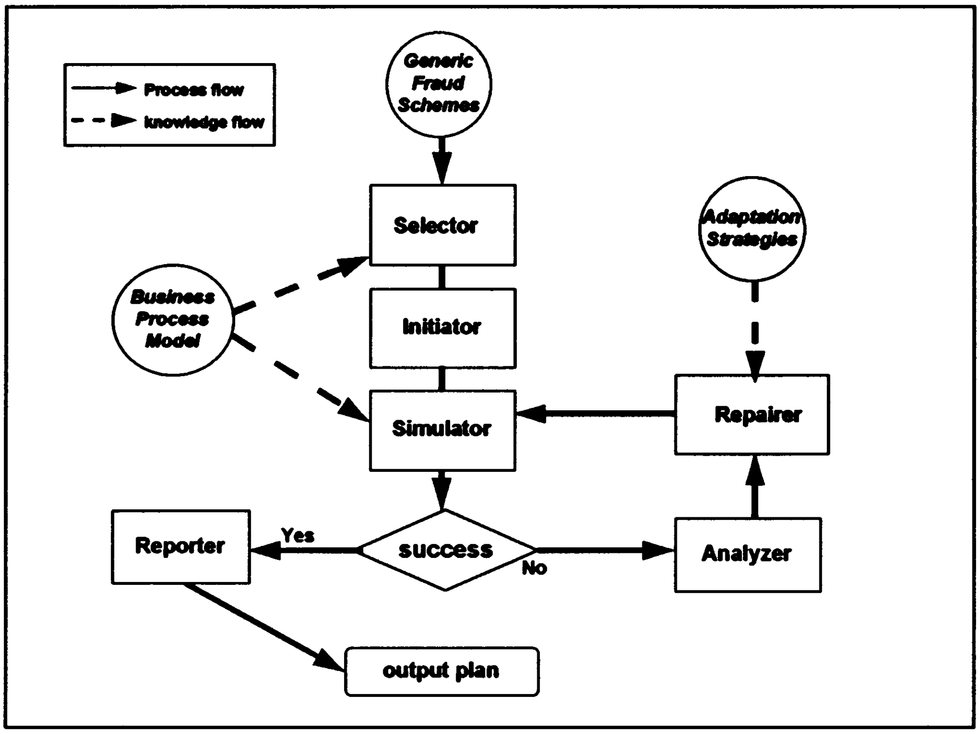


Figure 7 The fraud planning process

How is the plan generated? CROOK starts by identifying an opportunity for the cashier to embezzle an asset (cash). CROOK generates the initial plan in which the cashier steals the cash as soon as she gets it. In the simulation, however, the initial plan fails because the cash count control detects that the debit balance of the cash account disagrees with the lack of cash in the safe. CROOK searches for an action that can repair this disagreement and turn the cash account balance to zero. It finds such an action—posting a false entry to credit cash account against expenses. CROOK inserts this action into the plan and the repaired plan runs successfully. Detailed discussion about CROOK can be found in Natovich (1996).

Error Occurs when Agents Unintentionally Act not in Accordance with their Role

Because actions are performed by human agents, errors can occur. Norman (1981) has classified unintentional errors in action execution (termed 'slips' as opposed to 'mistakes', which are errors in intentions) to a small set of classes, based on the stage in the human cognitive process in which they occur. Many of these error classes can be represented within the AI-Planning paradigm as erroneous perception of states, erroneous selection of action and erroneous effects. For example

- **Mode errors**: Failure to perceive the state of the process. The error causes the agent to execute an action although the preconditions are not met. For example if a disbursement clerk fails to recognize that the status of the vendor invoice is 'paid' she might issue another check to the vendor for that invoice.
- **Description errors**: Failure to select the right object for the action. It causes the agent to perform the right action but with the wrong object. For example the disbursement clerk, upon an approved invoice, correctly issues a check but erroneously writes it to a different vendor.

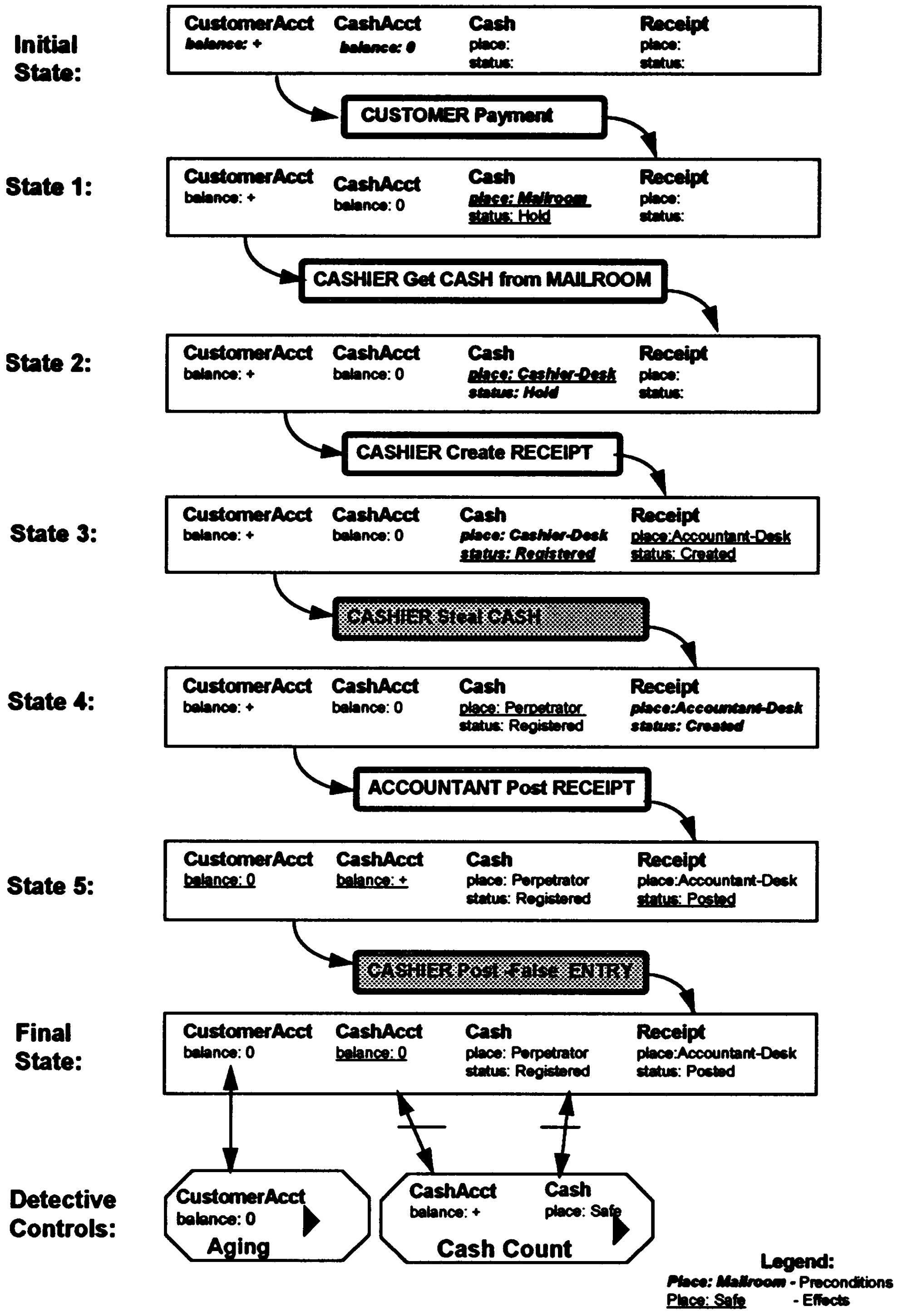


Figure 8 Cash receipt fraud scenario

- **Activation errors**: Failure to carry out an action when its preconditions are met by the state. Usually this lapse is a result of memory failure. For example, in the cash receipt process the cashier may fail to issue a receipt when the cash has been accepted.
- **Defective processing**: Erroneous effects of an action. For example, in posting a journal entry of the receipt, the accountant erroneously does not stamp the receipt as "posted".

The AI-Planning paradigm allows a computational error generator that injects errors during a process simulation by mutating elements in the states, action preconditions and action effects. To reduce the complexity, because probability is low that two independent errors occur within the same transaction, only one error is injected at a time. After an error has been injected, the process simulation continues to examine the effectiveness of the controls to detect and correct the error.

One possible result is that the error is detected at a later stage by one of the controls. For example, in the cash receipt process, if the cashier fails to get the customer's payment from the mailroom, the aging control will indicate the unrecorded payment and invoke investigation to find its cause. As the process continues, another possibility is that more errors happen as a result of the initial error. For example, if instead of issuing a debit note, a sales clerk erroneously issues a credit note, more errors can occur along the process and that might end up in issuance of a check to the client.

SLOPPY is a model of error that injects errors into ROLEMAN and evaluates the strength of the controls to handle these errors. Because our implementation efforts have been focused on CROOK, SLOPPY has yet to be implemented. At this point, our claim about the feasibility of error modeling is supported only by analogy from CROOK's prototype.

COMPARISON TO CURRENT MODELING METHODS

As opposed to current internal control models, AI-Planning modeling can be useful for generating and evaluating hypotheses of exposures. Reliability models (e.g. Yu and Neter, 1973; Cushing, 1974) and simulation models, (e.g. Burns and Loebbecke, 1975; Knechel, 1985; Wiggins and Smith, 1987) do not support this task. Instead, they use as input hypotheses generated by the user to quantify the overall exposure. Some knowledge-based models (e.g. Hamscher, 1992) support generation of hypotheses of errors only. Only the AI-Planning approach with its abilities to generate goal-driven plans can fully support fraud hypotheses as well.

In addition, the AI-Planning approach enhances the representation of some concepts that are relevant to the control perspective. Despite differences, models of the business process from the control perspective share a common format, in which **threats** and **controls** are applied to a **process** and result in an evaluation of the **exposure** (Figure 9). Throughout this section, we use this format as a framework for discussing and comparing AI-Planning representation to other approaches. We argue that the AI-Planning improves the current models by (1) supporting representation of the threat of fraud, (2) supporting reasoning about control effectiveness with respect to particular threats (i.e. control-threat matrix), thereby eliminating the need of users to input this assessment manually, and (3) representing the exposures found in the form of detailed scenarios that serve as proof of feasibility.

Modeling Threats

Many internal control models, such as reliability models, view the business process as a technical device comprising a set of procedures and controls (Srinidhi and Vasarhelyi, 1986). They ignore the fact that these procedures are executed by human agents. This might explain why those models concentrate on the threat of error only.

In a few models of the business process, such as SeaDoc (Eliott, 1983) or TICOM (Bailey *et al.*, 1985), threats are not represented explicitly. Reliability models represent errors in terms of the probability that a process can be executed erroneously. Simulation models define dollar amount of errors in a single transaction as a

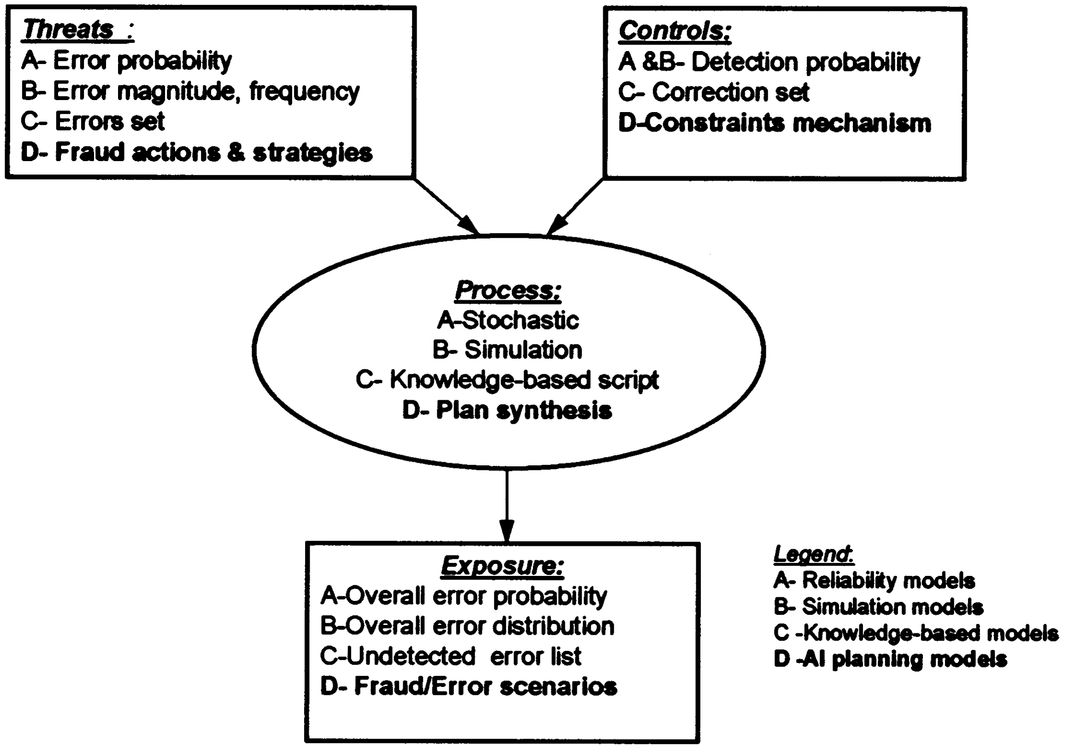


Figure 9 Comparison between representation methods of internal control models

random variable belonging to some distribution type such as normal or exponential. In contrast, Savile (Hamscher, 1992) models errors qualitatively, by defining for each type of action a list of potential errors that independently may occur.

Current models do not have the means to represent fraud. Some models regard fraud as a type of error, thereby representing both similarly. However, fraud differs substantially from error. In contrast to intentional errors, fraud may involve injection of multiple dependent errors into one transaction to circumvent internal controls. Also, since perpetrators try to circumvent internal controls, fraud is usually deterred only by multiple controls. Finally, hypothetical frauds are not easily identified by users, as required by quantitative models for errors.

A major advantage of the AI-Planning approach is the ability to represent threats including frauds. The model of fraud is represented by general definitions of fraudulent actions and adaptation strategies. Using this global definition, the AI-Planning approach automatically 'tailors' specific fraud scenarios to any given process, identifying the existence of such exposures.

Modeling Internal Controls

In current models, internal controls are represented in ways that express their effectiveness with respect to threats. In both reliability and simulation models, a control is modeled by its probability of detecting errors. Savile represents controls as a separate set of action types, each of which can detect and eliminate a defined set of errors. (Hamscher, 1992).

In contrast to current models, the AI-Planning approach does not represent controls by measures of effectiveness. Instead, the AI-Planning approach allows an explicit representation of the various controls as different types of constraints over agents and actions. The approach supports automated reasoning about the control effectiveness against threats. Thus, the user of the model is required to describe the control mechanism rather than to evaluate its effectiveness.

Evaluating Exposures

Internal control models use various methods to evaluate the threats that are not addressed by internal controls and to represent their effect on the business process. The reliability models evaluate deviations from the prescribed process in terms of the overall probability of the process failure. This value is calculated from the controls and the error probabilities in accordance with the structure of the process. Simulation models evaluate the deviations in terms of the expected dollar amount of error. This amount is calculated by seeding errors in transactions according to error distribution and applying controls to correct errors according to control effectiveness probability.

In Savile, the exposure is represented by a list of possible errors with respect to each account in the model in a simulation process (Hamscher, 1992). The simulation generates an error list for each action along the process flow. When the process flows through a control action, the errors that are effectively handled by that control are eliminated. At the end, only errors that were not addressed by the controls are left in the list.

The AI-Planning paradigm allows richer representation of the exposures than other internal control models. The reasoners of various exposures (i.e. CROOK or SLOPPY) search for all the plausible threats to the business process and generate scenarios, describing in detail how the errors or frauds may affect the prescribed process. These scenarios serve also as the proof of the feasibility of such exposures.

MODEL DEFICIENCIES AND FUTURE RESEARCH

Like all models, the AI-Planning approach suffers from some drawbacks. Compared to other modeling approaches, it (1) requires more input, thereby increasing the modeling cost, (2) relies on restrictive assumptions, and (3) is subject to combinatorial complexity that may restrict its ability to scale-up. These problems are partially addressed in our work, but they require additional research. Despite these problems, we believe that AI-Planning has potential to gain practical acceptance among control professionals.

Cost of Modeling

For every process modeled, the AI-Planning approach requires additional input including (1) providing the rationale of actions, (namely—preconditions to actions), and (2) providing the goals of actions, (namely—the effects). In addition, it requires an additional one-time effort in building models of the threats to the business process such as CROOK and SLOPPY.

This may increase the cost involved in its construction, including data collection and data input costs. Some measures can be taken to reduce this cost. With respect to data collection, although the amount of data needed is more extensive, AI-Planning uses only description of the process. The data can be obtained directly from observation, as opposed to some other models which require a preliminary threat analysis and/or evaluation of controls effectiveness. For example, in simulation models, the input includes possible error types at each step in the process and estimation of their probability. Obtaining such information requires an in-depth and time consuming analysis of threats and internal controls.

In addition, input cost can be controlled by using emerging information technology, such as high-level modeling languages, graphical user interface (GUI), CASE tools and reusability of models' objects. In our current implementation of ROLEMAN, we concentrated primarily on the functionality of model. More research is needed to address modeling productivity by employing the above techniques.

Model Assumptions

ROLEMAN, as a STRIPS-like model, is based on some general assumptions about the planning domain (Weld, 1994). In addition, some domain specific assumptions stem from this representation. These assumptions include:

- **Atomic time**: execution of actions is transformation from one world state to another. Each state in the sequence lasts a single time unit, and the actual time that passes is not considered. The model does not consider the state of the world while execution is proceeding. As a result, STRIPS does not allow simultaneous exectution of actions. Also, because time is represented only by the number of state transformations, some temporal elements such as end-of-month or end-of-year are ruled out.
- **Deterministic effects**: The effect of executing an action is a deterministic function of the action and the state of the world when the action is executed. Unless explicitly specified, arbitrary errors in actions are ruled out.
- **Omniscience**: The agents have complete knowledge of the initial state of the world and the nature of the possible actions.
- **Sole cause of change**: The only way the world changes is by the agents' own actions. There are no other agents to be represented in the model and, by default, the world is static.
- **Single transaction**: Although a business process is continuous and handles multiple transactions simultaneously, STRIPS, like other qualitative business process models do, focuses on the process with respect to one transaction only. As a result, some concepts that are associated with the relationships between transactions, such as aggregation and completeness, are not represented.
- **Qualitative properties**: As a qualitative model, quantitative properties concepts in STRIPS need to be converted to qualitative terms. The amount of a transaction can be either positive (+) or negative (–). Accordingly, an account balance in the model can be represented as Debit, Credit or 0. This qualitative representation ignores some quantitative concepts such as discrepancies between amounts of documents and account balances.

Contemporary AI-Planning work proposes extensions and alternatives to STRIPS. Future research may focus on enhancing the process representation by incorporating these extensions. For example, using partial-order representation (Sacerdoti, 1975; Tate, 1977) allows for simultaneous actions in the process. Hierarchical planning models (Dean *et al.*, 1988) support representation of the business process in various levels of tasks and actions, as DFD and flowcharts, and they may be important in modeling large-scale processes. Uncertainty representation in planning (e.g. Feldman, and Sproull, 1977) allows for integrating probability into the model. It may support reasoning not only about the plausibility of exposures but also about their likelihood. *Ad-hoc* work is also needed to address representation of additional domain-specific concepts including various types of actions (e.g. automated actions), controls (e.g. aggregation controls) and threats (e.g. natural disasters, errors and omissions).

Model Scalability

Another issue to be examined is the ability of the approach to reduce the combinatorial complexity of planning tasks and to be effective in large and detailed processes. To address this problem, CROOK adopts recent case-based planning techniques (Hammond, 1989; Turner, 1994) that rely on experience to reduce search effort and claim to support solutions for real-world problems. Even the forward-chaining reasoning that is used to represent the prescribed process is expected to maintain low complexity because, as mentioned earlier, to eliminate ambiguity, roles define only one or a small number of alternative actions for a given situation. This gives us a reason to believe in the scalability of the proposed approach. More work is needed to examine this point, by implementing a representation of a real-world process larger than the process used for our current work.

SUMMARY

This work explores application of AI-Planning in the audit domain. This technology has rarely been applied in management and, to our best knowledge, has never been applied in auditing. The main attraction of AI-Planning is that, as opposed to expert systems, it does not require extensive expert knowledge. Rather, it employs a model of the problem domain and a search algorithm based on general heuristics.

In this paper, we have proposed an approach to modeling business processes within the AI-Planning paradigm. We have discussed its advantages in representing threats and controls, and have shown its potential to generate exposure hypotheses for internal control evaluation. Although further research is needed to extend the usefulness of this approach, the results shown here are encouraging.

Acknowledgments

Price Waterhouse Technology Centre provided software and hardware for this work. Also, the authors wish to thank Glenn Shafer, Walter Hamscher, Wenling Hsu, Alex Kogan, Hannah Rothstein, Debbie Nachshon, participants of the AAA workshop on AI/ES in Orlando 1995 and three anonymous reviewers for their useful comments.

References

Allen, J., 'Formal models of planning', in Allen, J., Hendler, J. and Tate, A. (eds.), *Readings in Planning*, Morgan Kaufman, San Mateo, CA, 1990.
Bailey, A.D., Jr *et al.*, 'TICOM and the analysis of internal control', *The Accounting Review*, **LX**: 2, April 1985, pp. 186–201.
Bonner, S. and Pennington, N., 'Cognitive processes and knowledge as determinants of auditor expertise', *Journal of Accounting Literature*, **10**, 1991, pp. 1–50.
Bratko, I., *PROLOG, Programming Language for Artificial Intelligence*, 2nd edition, Addison-Wesley, Reading, MA, 1990.
Burns, D.C. and Loebbecke, J.K., 'Internal control evaluation how the computer can help', *Journal of Accountancy*, August, 1975, pp. 60–70.
Curtis, B., Kellner, M.I. and Over, J., 'Process modeling', *Communications of the ACM*, **35**:9, September, 1992, pp. 75–90.
Cushing, B., 'A mathematical approach to the analysis and design of internal control system', *The Accounting Review*, January, 1974, pp. 335–365.
Dean, T., Firby, J. and Miller, D., 'Hierarchical planning involving deadline travel time and resources', *Computational Intelligence*, **4**:4, 1988, pp. 381–398.
Eliott, R.K., 'Unique audit methods: Peat Marwick International', *Auditing: A Journal of Practice and Theory*, **2**:2, Spring, 1983, pp. 1–12.
Feldman, J.A. and Sproull, R.F., 'Decision theory and AI II, the hungry monkey', *Cognitive Science*, **1**, 1977, pp. 158–192.
Fikes, R.E. and Nilsson, N.J., 'STRIPS: a new approach to the application of theorem proving to problem solving', *Artificial Intelligence*, **2**:3/4, 1971, pp. 189–208.
Gadh, V.M., Krishnan, R., Peters, J.M., Abdolmohammadi, M.J. and Houghton, C.W., 'Modeling internal controls and their evaluation', *Auditing: A Journal of Practice and Theory*, **12**, Supp., 1993, pp. 113–136.
Georgeff, M.P., 'Planning', *Annual Review in Computer Science*, **2**, 1987, pp. 359–400.
Hammond, K.J., *Case-based Planning: Viewing Planning as a Memory Task*, Academic Press, San Diego, CA, 1989.
Hamscher, W., 'Modeling accounting systems to support multiple tasks: a progress report', *Proc. 10th National Conf. on Artificial Intelligence*, San Jose, CA, July, 1992.
Hollander, A.S., Denna, E.L. and Cherrington, J.O. *Accounting, Information Technology and Business Solutions*, Irwin, Homewood, IL, 1996.
Huff, K.E. and Lesser, V.R., 'Plan-based intelligent assistant that supports the software development process', Proc. of the Third Software Engineering Symposium on Practical Software Development Environment, *Soft. Eng. Not.* **13**, 5, 1989, pp. 97–106.
Johnson, P., Grazioli, S. and Jamal, K., 'Fraud detection: intentionality and deception in cognition', *Accounting Organization and Society*, **18**:5, 1993, pp. 467–488.
Knechel, W.R., 'A simulation model for evaluating accounting system reliability', *Auditing: A Journal of Practice and Theory*, **4**:2, Spring, 1985, pp. 38–62.
Ku, S., Suh, Y. and Tecuci, G., 'Building an Intelligent business process reengineering: a case-based approach', *International Journal of Intelligent Systems in Accounting, Finance and Management*, **5**,1, March, 1996, pp. 25–40.
Lifschitz, V., 'On the semantics of STRIPS', in Allen, J., Hendler, J. and Tate, A. (eds), *Readings in Planning*, Morgan Kaufman, San Mateo, CA, 1990.
Menzefricke, U. and Smieliauskas, W. 'A survey of simulation studies in statistical auditing', *Journal of Accounting Literature*, **6**, 1987, pp. 26–54.
Meservy, R., Bailey, A.D. Jr and Johnson, P.E., 'Internal control evaluation: a computation model

of the review process', *Auditing: A Journal of Practice and Theory*, **6**:1, Fall, 1986, pp. 44–73.

Nado, B., Van Baalen, J. and Fikes, R., 'JOSIE: an integration of specialized representation and reasoning tools', *ACM SIGART Bulletin* special issue on implemented knowledge representation and reasoning systems, **2**:3, June, 1991, pp. 101–107.

Natovich, J., *Reasoning about internal controls and fraud: the AI-Planning approach to modeling the business process*, unpublished dissertation, Rutgers University, 1996.

Norman, D.A., 'Categorization of action slips', *Psychological Review*, **88**, 1981, pp. 1–15.

Pednault, E.P.D., 'Formulating multiagent, dynamic world problems in the classical planning framework', in Allen, J., Hendler, J., and Tate, A. (eds), *Readings in Planning*, Morgan Kaufman, San Mateo, CA, 1990.

Sacerdoti, E.D., 'The nonlinear nature of plans', *Proceedings IJCAI-75*, Tbilisi, USSR, 1975.

Srinidhi, B.N. and Vasarhelyi, M.A., 'Auditor judgment concerning establishment of substantive test based on internal control reliability', *Auditing: A Journal of Practice and Theory*, **5**:2, Spring, 1986, pp. 64–76.

Tate, A., *Generating Project Networks*, IJCAI-77, Boston, MA, 1977.

Turner, R.M., *Adaptive Reasoning, for Real World Problems: A Schema-Based Approach*, Lawrence-Erlbaum, Hillsdale, NJ, 1994.

Vasarhelyi, M.A. and Natovich, J., 'Fraud Simulator: An audit approach to evaluate internal control using AI techniques', *Proc. of the 6th Florida Artificial Intelligence Research Symposium*, 1993, pp. 31–36.

Weld, D.S., 'An introduction to least commitment planning', *AI magazine*, Winter, 1994, pp. 27–61.

West, H., *Fraud, The Growth Industry*, Kogan Page, London, 1988.

Wiggins, C.E. Jr and Smith, L.M., 'A generalized audit simulation tool for evaluating the reliability of internal controls', *Contemporary Accounting Research*, **3**:2, 1987, pp. 316–337.

Yu, S. and Neter, J., 'A statistical model for the internal control system', *The Journal of Accounting Research*, Autumn, 1973, pp. 273–295.

Yu, E.S.K. and Mylopoulos, J., 'Using goals rules and methods to support reasoning in business process reengineering', *International Journal of Intelligent Systems in Accounting, Finance and Management*, **5**, 1 March, 1996, pp. 1–14.