

ON RISK MANAGEMENT IN BUSINESS PROCESS DESIGN

Xue Bai
Rema Padman
Ramayya Krishnan

The H. John Heinz III School of Public Policy and Management
Carnegie Mellon University, Pittsburgh Pennsylvania 15213, USA
{xbai, rpadman, rk2x}@andrew.cmu.edu

Abstract

With the development of technologies such as Business Process Execution Language (BPEL), a business-process centric approach to design of information systems has emerged. This approach calls for the modeling and analysis of the business process— both to document it as well as to analyze its risk characteristics – specifically, the risk of errors in the data produced by the process. This focus on risk has been driven by recent legislative mandates such as the Sarbanes Oxley Act. Using a graphical model of a business process, we develop a probabilistic approach to assess the risk of errors in the data produced by a business process. We propose an optimization-based method to determine cost-effective ways of embedding controls – procedures that mitigate such risk – in a business process to meet desired risk thresholds. The method lends itself to implementation within process modeling workbenches offered by leading software vendors. We illustrate our work through the case of order fulfillment process in a functioning, online pharmacy.

1. Introduction

The importance of well designed business processes in organizations has long been recognized (Anupindi et al. 2005). A process consists of a collection of tasks or activities that transform a set of inputs to produce a set of outputs. There is a large and diverse literature on business processes including work reported in the Information Systems literature. These include work from the standpoint of process modeling and analysis (Basu and Blanning 2003), process data reliability (Krishnan et al. 2005), implementation of processes using workflow management technologies (Basu and Kumar 2002; Zhao and Kumar 2001) and more recently, work on process centric-design of information systems using technologies such as business process execution language (BPEL) that permit orchestration of web services (Chakraborty and Lei 2004).

Our work contributes to this literature by focusing on the risk assessment of errors in the information produced by a business process. We develop a probabilistic framework for risk assessment, and propose an optimization-based approach to determine the manner in which control procedures can be embedded in the business process to mitigate risk.

Addressing the risk associated with errors in the data produced by business processes has usually been dealt with by auditors. However, auditors focus on evaluating risk of data errors in systems *ex-post* i.e., after they have been built and deployed with a particular system of control procedures (Krishnan et al. 2005). In contrast, we are interested in the design of the control

procedures and the manner in which they should be embedded in business processes *ex-ante* (i.e., during analysis and design phases of business processes prior to their implementation). Our objective is to develop a well-founded quantitative approach to risk assessment and mitigation which can be implemented and deployed as part of a tool kit or workbench (e.g., IBM WebSphere Business Process Modeler) to facilitate an iterative approach to process-centric analysis and design (Please see <http://www.redbooks.ibm.com/redbooks/SG247024/> for a description of how process-mapping is supplemented with analysis in this approach).

In this paper, we first model business processes as a directed, attributed graph, and introduce a hierarchical probabilistic model of error distribution and propagation through the process graph. Then, we model control procedures and their roles in risk mitigation. Finally, we assess the loss due to errors in the data produced by the process and propose an optimization-based approach to tradeoff the cost of applying controls and the reduction in loss brought about by the use of controls. We illustrate our model using order fulfillment processes in an online pharmacy.

2. Business Processes (BP) as a Graph

There is a large literature on graph-based models of business processes. Basu and Blanning (1994a and b) proposed metagraphs, and demonstrated its capacity to model processes, workflows and decision support resources (Basu and Blanning 2002). Kumar and Zhao (2000) proposed a workflow-centric approach for organizational information distribution, and in (Kumar and Zhao 1999) described a general framework for implementing dynamic routing and control mechanisms in workflow management. There is also a literature that emphasizes well formedness of workflows and analyzes their properties using Petri net models (Russell et al. 2005). Krishnan et al. (2005) describe a logic-based approach to modeling and reasoning about processes that translate directly into a graphical process model. In this paper, we work with an abbreviated graphical version of the process model introduced in Krishnan et al. (2005).

Essentially, a process model is assumed to consist of a set of tasks which are organized in precedence order with a directed arc from Task A to Task B implying that A precedes B . Further, directed arcs convey the exchange of information units between the tasks. There are a distinguished set of nodes we refer to as information sources and information repositories which represent “interfaces” to the business process in that information is fed to the process via sources and delivered to other processes via repositories. This simple model is extended to permit attributes, such as error types in the information processed or produced by the task, and later, the presence of control procedures at tasks. For the sake of exposition, we will represent the BP graph as a set of matrices. The expressive power of this representation is sufficient for the purposes of risk modeling of the type discussed in the paper. Since this is work in progress, a thorough discussion of semantics and expressive power is beyond the scope of this paper.

2.1 Process topology

We model the flows in the process as flows of information units. Typically, every information unit is multi-dimensional. In our online pharmacy case (Fig. 1), the input information units are the orders for medication. Each order contains different dimensions such as demographic information, patient history, credit history, prescription etc. As a starting point, we treat different dimensions as equivalent from a risk perspective. However, this constraint can be easily relaxed. Let $\{t_1, \dots, t_N\}$ denote the set of tasks in the business process. We define $T_{N \times N}$ as the *TaskPrecedence* matrix. $T_{N \times N}$ provides a mapping of paths of the information flows in the process:

$$[T]_{iz} = \begin{cases} 1, & \text{if task } t_i \text{ directly precedes } t_z, \\ 0, & \text{otherwise.} \end{cases}$$

The T matrix for our case (Fig. 1) is expressed as in Eq. (1). The process is triggered by the clients ordering the medications. The company interacts with clients, drug manufacturers and insurance companies during the process. As medication orders get processed by Order Management center, the valid orders are sent to the In-house Pharmacies, verified and fulfilled there. Bills are sent out from Billing Management center to the insurance companies or the clients; meanwhile, the medicines are delivered from the In-house Pharmacies by contracted carriers to the clients; the payments are collected at the end of each month at the Billing center and recognized as revenue at the accounting department. Each management center is a module of sub-process, where sequences of tasks are involved to achieve the function of the module. The whole process contains 12 tasks: 1) client orders, 2) order entry, 3) credit check, 4) contract update, 5) check med availability, 6) check quality & price, 7) check patient meds history, 8) supply/alternative-supply, 9) check payment option, 10) prepare accounting information/prepare a bill/collect revenue, 11) update ledgers and 12) insurance company/clients paying bills.

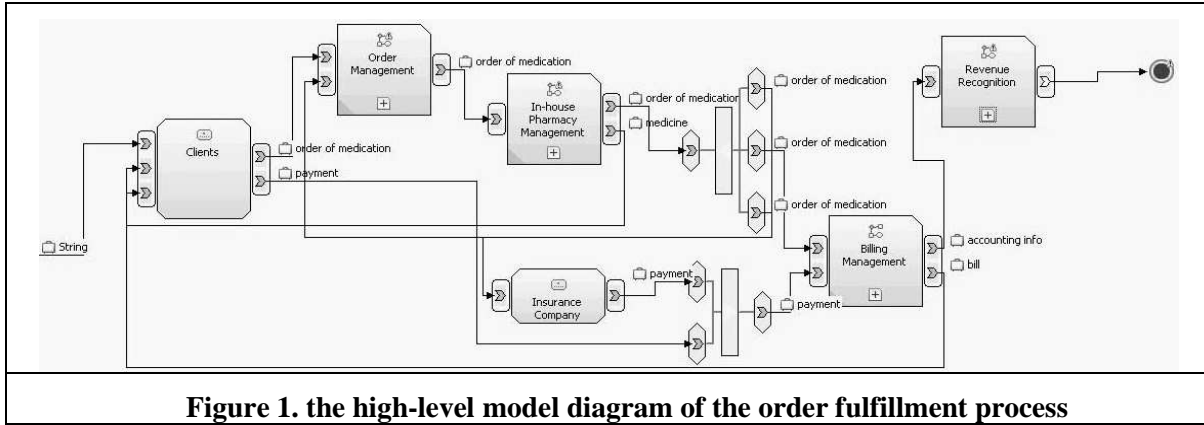


Figure 1. the high-level model diagram of the order fulfillment process

The scope of the main process involves external players: task 1 and task 12 interact with clients and insurance companies, which are not considered in our model. Our model deals with the 10 internal tasks. The maximum number of steps in the process is 10.

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{10 \times 10}$$

(1)

$$e = \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \end{matrix}_{10 \times 5}$$

(2)

$$\Gamma = \begin{bmatrix} 0 & 1 & 3 & 2 & 2 & 2 & 2 & 2 & 1 & 1 \\ 0 & 0 & 3 & 3 & 2 & 2 & 2 & 2 & 2 & 1 \\ 0 & 0 & 2 & 3 & 3 & 2 & 3 & 2 & 2 & 2 \\ 0 & 0 & 2 & 2 & 3 & 3 & 3 & 2 & 2 & 2 \\ 0 & 0 & 3 & 2 & 2 & 3 & 2 & 3 & 2 & 2 \\ 0 & 0 & 3 & 3 & 2 & 2 & 2 & 3 & 3 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{10 \times 10}$$

(3)

2.2 Error sources and error correlations

Each task may introduce errors when it is performed. Errors may be caused by a number of reasons such as mistakes, omissions, delays, and software glitches. Let there be M types of errors

that can possibly occur during the process, $\{e_1, \dots, e_M\}$. We use a hierarchical probabilistic model to define the distribution and correlation of error types at each task. Let \vec{p}_i be the M -dimensional random vector of the probabilities of task t_i introducing the M types of errors. Each element of \vec{p}_i , p_{ij} , is the probability of error type j being introduced by task t_i ($j=1, \dots, M; i=1, \dots, N$). \vec{p}_i follows a *Generalized Logistic Normal (GLN)* distribution. *GLN* distribution is an extension of the *Logistic Normal* distribution introduced by (Aitchison and Shen 1980):

$$1. \quad \vec{p}_i | \vec{\mu}, \Sigma \sim \text{GeneralizedLogisticNormal}_M(\vec{\mu}, \Sigma); \quad i=1, \dots, N \quad (4)$$

Σ is the M -dimensional variance-covariance matrix of \vec{p}_i ; Σ encodes the correlations among error types at each task; the mean vector $\vec{\mu}$ represents the average value of \vec{p}_i . Let e_{ij} be the random binary variable, encoding the presence or absence of an error type j by task i . e_{ij} follows a *Bernoulli* distribution with parameter p_{ij} where p_{ij} is the j th element of \vec{p}_i :

$$2. \quad e_{ij} | p_{ij} \sim \text{Bernoulli}(p_{ij}), \quad j=1, \dots, M \quad (5)$$

A snapshot of the error samples among tasks in our case is shown in Eq. (2).

GLN requires the prior information to estimate the mean and the variance-covariance matrix of the error probabilities ($M(M+3)/2$ parameters). In our case study, the parameters are calibrated using the company's proprietary data. A summary of the data is in (Bai et al. 2006).

2.3 Path relationships and error propagation

Errors introduced by tasks are propagated along paths in the process graph. We assume if an error is generated at t_i , it will be carried along to the succeeding tasks of t_i , unless it is detected and fixed by controls. The error propagation model is specified as follows: Note that matrix T encodes the direct adjacency of the tasks. Similarly, $T^k = \underbrace{(T \times T) \times T \dots \times T}_k$ is the k -step adjacency matrix, where the element at the i th row and z th column of T^k represents the number of k -step paths from t_i to t_z .

Usually, the number of steps to execute a process varies. For example, the orders may subject to iterative validation checks, or be terminated as an invalid. Let $p(k)$ be the empirical (or the estimated *ex-ante*) distribution of k ; k is the number of steps for process execution. The expected sum of the adjacency matrices, $\Gamma = \sum_{k=1}^K (T^k \cdot p(k))$, yields a matrix in which each cell represents the average number of pathways between a pair of nodes (K represents the length of the longest path in the process graph). We use this as "weight" to model the impact of an error that arises in one task and propagates to other tasks downstream. The intuition is that if error e_j arises at t_i with probability p_{ij} , it affects t_z , which can be reached from t_i , in τ_{iz} different pathways (τ_{iz} is the entry at the i th row and z th column in Γ), τ_{iz} times more than to a task that can be reached from t_i through only one path.

$\tau_i = \sum_{z=1}^N \tau_{iz}$, is the sum of the elements in the i th row of Γ . τ_i represents the total number of pathways through which an error caused by t_i can affect all the succeeding tasks of t_i in the business process. Later, we will use this property to discuss loss functions. The computational complexity of computing Γ is $O(KN^3)$. In our case, Γ is expressed as in Eq. (3).

3. Modeling Control Procedures for Risk Mitigation

Control procedures detect and fix errors introduced to the process when the tasks are executed. Fundamentally, controls provide a means to mitigate risk. In our paper, we model controls as classifiers that fix errors by applying them at all or a subset of task locations. Each task may be covered by one or more controls. Controls can be applied at different levels of utilization that have associated cost and error detection implications. A concrete example of a control may involve sampling of business documents such as purchase orders, receiving reports and invoices to detect valuation, existence and completeness errors. The larger the sample size chosen, the greater is the cost to execute the control and the more likely the errors being caught.

3.1 Error detection capabilities and effectiveness of controls

Suppose we have Q control units available for use in a process. A control unit is deemed applicable if it has the capacity to fix errors that might arise at a task. Every task has a set of available control units to monitor and/or correct the performance of a task. Let Q_i be the set of available control units for task t_i , the *power set* of Q_i , $\Lambda(Q_i)$, is the set of all possible combinations of control units in Q_i . Each element of $\Lambda(Q_i)$ represents a combination of control units that can be employed for monitoring the process for errors. Within each combination, each control unit has its utilization capacity, which may or may not be fully put to use. Every combination of control units along with their utilization level represents a certain level of control *effectiveness*, measured by two parameters: $0 \leq \alpha_1 \leq 1$ and $0 \leq \alpha_2 \leq 1$:

- a. α_1 : the *true positive rate*, i.e. the probability that a control system detects and fixes an error, given that the error exists; the larger the α_1 is, the more effective is the control;
- b. α_2 : the *false positive rate*, i.e. the probability that a control system detects and fixes an error, given the error does not exist; the smaller the α_2 is, the more effective is the control.

Note that by abstracting a set of control units as a “control system”, we associate effectiveness with the system. Let $x = \{x_i : 0 \leq x_i \leq 1\}_{i=1, \dots, N}$ be the set of the control utilization allocation factors at each task. For each task t_i , $x_i=0$ means controls are not in use; $x_i=1$ means controls are used at their peak level; $0 < x_i < 1$ means controls are running at some intermediate level between 0 and 1. We model α_1 and α_2 as continuous, nonnegative functions of x_i . The forms of $\alpha_1(x_i)$ and $\alpha_2(x_i)$ may be specified for specific applications based on prior knowledge. We assume controls are equally effective over all error types at task locations in our current model. However, this assumption can be relaxed by extended our modeling framework. The probability of error e_j being introduced by t_i , with control systems in use, is given by p'_{ij} :

$$p'_{ij} = p_{ij} \cdot (1 - \alpha_1(x_i)) + (1 - p_{ij}) \cdot \alpha_2(x_i) \quad (6)$$

4. Risk Management Model

Risk is the expected loss due to the errors in the information produced by the process. The loss can be characterized as legal penalties or additional business cost due to operating with incorrect information.

4.1 Loss due to errors

Each task contributes to the total loss by introducing errors of various types to the information flows (specified by \bar{p}_i) and by propagating incorrect information to the downstream

tasks (represented by τ_i). Let $c_{j=1,\dots,M} > 0$ be the *cost* of an error of type j in an information unit. The loss of task t_i introducing error e_j into an information unit is specified as (Berger 1980)¹:

$$L(e_{ij}) = \tau_i \cdot \sum_{j=1}^M (c_j \cdot e_{ij})^2. \quad (7)$$

The risk of the process with information volume² v , with controls being applied, is specified as:

$$R = v \mathbb{E}_{p(e_{ij})} L(e_{ij}) = v \int L(e_{ij}) P(de_{ij}) = v \sum_{i=1}^N \tau_i \sum_{j=1}^M c_j^2 [p_{ij}(1 - \alpha_1(x_i) - \alpha_2(x_i)) + \alpha_2(x_i)] \quad (8)^3$$

4.2 Cost of applying controls

We extend the approach in Krishnan et al. (2005) by requiring the cost of a control to be a function of the control utilization factor, x_i , $x_i \in x$. Let $\omega_i(x_i)$ be the cost of applying control to t_i . We impose the most commonly used conditions for cost functions from system engineering (Kuo et al. 2001): $\omega_i(x_i)$ is a continuous, nonnegative, convex and non-decreasing function of x_i .

4.3 Designing the optimal control system

The design problem we address is the problem of determining an *optimum* set of control utilization allocations (i.e., the effort to be invested in each control in the process) x^* to deploy in a business process that will minimize the expected loss due to errors under the control cost constraints. The mathematical model is specified as follows:

Case I: If the budget is not constrained, the objective is to find the optimal x^* such that the total cost is the minimum:

$$\min C(x) = R + \sum_{i=1}^N \omega_i(x_i), \quad x_i \in x; \quad (9)$$

Case II: If there is a budget constraint B , the objective is to find the optimal x^* such that the risk is the minimum, subject to the budget constraint:

$$\min R, \quad \text{s.t.} \quad \sum_{i=1}^N \omega_i(x_i) \leq B; \quad x_i \in x. \quad (10)$$

5. Work in Progress

We have collected data from an online pharmacy, modeled the process in WebshpereBP Modeler, calibrated the parameters of our hierarchical model and cost functions using the data. We have developed algorithms to solve the optimization problems stated in the previous section and are implementing a simple system to support risk management-based business process design

References

- Aitchison, J., Shen S. M. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67:261--272, 1980
- Anupindi, R., Deshmukh, S., Van Mieghem, J, Zemel E. *Managing Business Process Flows*. Prentice Hall. 1st edition, 1999; 2nd edition 2005.
- Bai, X., Padman, R., Krishnan R. Risk Management in Business Process Design, working paper #HNZ- 2006-8
- Basu, A. and Blanning, R. W. Metagraphs: A Tool for Modeling Decision Support Systems. *Management Science*, Vol. 40, No. 12. 1994(a).
- Basu, A., Blanning R. W. Model Integration Using Metagraphs, *Information Systems Research*, Vol.5-3. 1994(b).
- Basu, A., Kumar, A. Research Commentary: Workflow Management Systems in e-Business. *Information Systems Research*, Vol. 13, No. 1, March 2002, 1-14

¹ Squared error loss, absolute error loss and exponential error loss are most common loss functions in decision theory and economics. We adapted the squared error loss function by adding the cost of errors and path influence.

² Information volume refers to the number of information units in a process for some unit interval of time.

³ The detailed calculation is available in (Bai et al. 2006).

- Basu, A., Blanning R. W. Synthesis and Decomposition of Processes in Organizations, *Information Systems Research*, vol. 14, no. 4, December 2003.
- Berger, J. *Statistical Decision Theory: Foundations, Concepts, and Methods*. Springer-Verlag, New York, 1980
- Chakraborty, D., Lei, H. Pervasive enablement of business processes, Pervasive Computing and Communications, PerCom 2004. *Proc. of the Second IEEE Annual Conference on 2004* Page(s):87 - 97
- Krishnan, R., Peters, J., Padman, R., Kaplan, D. On Data Reliability Assessment in Accounting Information Systems. *Information System Research* Vol. 16, No. 3, September 2005, 307-326.
- Kumar, A., Zhao L. Dynamic Routing and Operational Controls in a Workflow Management System, *Management Science*, Volume 45, No. 2, February 1999, 253-272.
- Kumar, A., Zhao L. Workflow Support for Electronic Commerce Applications. *Decision Support Systems* (32:3), January 2002, pp.265-278.
- Kuo, W., Rajendra Prasad, V. J., Tillman F., Hwang, C. L. *Optimal Reliability Design: Fundamentals and Applications*. Cambridge University Press, 2001
- Pasley, K. Protecting Financial Information: Sarbanes-Oxley. SOA Compliance Whitepaper pp. 1-11, 2002
- Russell, N. Arthur, H. M., ter Hofstede, Edmond, D., Wil, M. P. van der Aalst: Workflow Data Patterns: Identification, Representation and Tool Support. ER 2005: 353-368
- Zhao, L., Kumar, A., Stohr, E. A., Workflow-centric Information Distribution through Email. *Journal of Management Information Systems*, winter 2000-2001, Vol. 17, No. 3, 45-72.