Robustness Management with Data Quality in Information Systems

Zhiting Song* and Yanming Sun

South China University of Technology
NO.381 Wushan Road, Guangzhou, Guangdong, China, 510641
szt05_scut@163.com, meymsun@scut.edu.cn

Abstract

The data quality in information systems affects system robustness significantly. Although a large body of research focuses on the issue of data quality, seldom has literature been done to develop effective control policies to manage system robustness associated with data errors. A process-oriented methodology is proposed to manage system robustness measured by performance, control cost and control time, which is achieved through establishing formal model of information systems and mathematical optimization models of system robustness. The proposed methodology captures how structural and functional characteristics of tasks affect system robustness and finds the optimal control policies, which facilitates the robustness-based design and management of information systems respectively. The methodology is illustrated in the case study.

Keywords: Robustness; Data Quality; Information Systems; Control

1. Introduction

Data quality has long been recognized as an important issue in information systems [1-3]. Poor data quality often prevents information systems from yielding satisficing performance [4], which severely impacts system robustness. Although information systems have been extensively studied in information system literature, there is a lack of research on system robustness associated with data errors. In reality, robustness is a desirable property for information systems, while data errors have been the leading cause of business process interruption in information systems [5]. It is necessary to develop effective control policies to ensure that systems are robust to data errors.

To maintain system robustness at a satisficing level in case of data errors, the paper first extends the well-structured WF-net that Van der Aalst proposed via inserting dummy task transitions between two directly connected branches or merges to model information systems more formally. Based on the extended WF-net, the paper then incorporates structural and functional features of tasks into error control, and selects performance, control cost and control time related to data errors to measure robustness. Finally, it establishes optimization models to determine the optimal control policies that achieve target robustness on performance with maximal robustness on cost or maximal robustness on time. The paper provides a feasible and quantitative approach to robustness management for information systems exposed to the perturbation of data errors.

The paper expects to achieve the following objectives: (1) to model real-world information systems more formally, it extends the well-structured WF-net by introducing dummy task transitions; (2) to facilitate the robustness management of existing information systems, robustness optimization models are established to determine the optimal resource allocation policies that yield satisficing robustness on performance with maximal robustness on cost or robustness on cost time; (3) to provide guidance to the

robustness-based design of information systems, the influence that structural and functional features of tasks exert on the optimal input levels of resource across tasks and error types is analyzed.

2. Literature Review

This article tries to develop effective control policies to make information systems robust to data errors. Three streams of literature are relevant to this paper: the first is related to data quality and data errors, the second is related to business process modeling, and the last is related to system robustness.

2.1. Data Quality and Data Errors

Data quality, originally identified as data accuracy, is a multidimensional concept beyond accuracy [6]. Although there is no uniform definition of data quality dimensions [7], the frequently mentioned dimensions are accuracy, completeness, consistency and timeliness which have been widely accepted in information system literature [6] [8-9]. Data quality dimensions in this study refer to these four dimensions. The research of data quality dimensions provides a starting point to define data error types [2]. For example, according to data quality dimensions of completeness, existence, valuation, presentation and disclosure, and rights and obligations in accounting literature, Lea *et al*. define five error types of completeness errors, existence errors, valuation errors, presentation and disclosure errors, and rights and obligations errors [10]. Similarly, error types in the paper are accuracy errors, completeness errors, consistency errors and timeliness errors. Accuracy errors occur when the data are incorrect or even spurious. Completeness errors occur when all relevant information that should be contained does not fully exist. Consistency errors occur when data are not consistently represented, formatted or compatible with previous data. Timeliness errors occur when data are not disposed during a given period of time or processed in a timely manner. Data errors mainly originate from system design and improper execution of tasks in the business processes of information systems, and this study focuses on the latter error source.

2.2. Business Process Modeling

Changes in the environments of information systems have made many information systems processes-oriented [11], which provides an idea to construct ontologies for information systems by business process modeling. There exist many methods for business process modeling. Flow chart is first introduced to model business processes, but it is unable to analyze properties of business processes [12]. Recently, workflow graph [13]，event-driven process chains (EPC) [14], business process modeling notation (BPMN) and unified modeling language (UML) [4] also have been used to model business processes, but they all lack formal semantics. Petri nets (*i.e., classical Petri nets and high-level Petri nets) have been considered as powerful tools to model and analyze workflow [11]. Concretely speaking, Petri nets are not only applied to analyze behavior, attributes and events of workflow, but employed to model reachability, deadlock and other structural features of workflow [11] [15]. In a word, different methods can be applied to model business processes, but Petri nets are the only formal methods that can be used for both structural modeling and a wide range of qualitative and quantitative analysis [11] [16], which is vital to analyze robustness. A Petri net modeling a workflow process is called a Workflow net (WF-net), and the definition of well-structured WF-net is extended to specify the process-oriented ontology of information systems.

2.3. Robustness

There is no agreement on the concept of robustness in information system literature. Tavana *et al*. define robustness as the ability to avoid failure caused by increasing complexity between people and technology in systems [17]. In reality, all perturbations apart from that complexity will influence robustness. Some researchers hold that a system is robust if it operates correctly with all perturbations inside and outside the system [18-20]. It is undesirable if the normal operation consumes a lot of time or money. From the perspective of strategic management, Sawy *et al*. define robustness of information systems as the ability to yield satisficing performance under all environments quickly and inexpensively [21], which is adopted in the study for its comprehensiveness. To quantitatively analyze robustness, measures of robustness are proposed, and performance is a widely used metric of robustness [19-22]. Performance under a certain environment is computed either as the product of event probability and undesirable consequence related to the event [20] or key performance indexes of systems [22]. In the paper, performance is measured by the product of error probability and loss caused by data errors, which avoids subjectivity in selecting performance indexes. Intuitively, robust systems can suffer perturbations at a low cost and in a short time. Thus, robustness can be measured from the perspectives of repair-recovery cost and repair-recovery time associated with perturbations [17] [23] which in the article are control cost and control time associated with data errors. Moreover, standard deviation of system response [19], efficiency in switching to other configurations and the degree of variability of environmental conditions that systems can cope with are also said to be measures of robustness [21], but they are not frequently used.

In summary, above mentioned papers do not involve in robustness management associated with data quality, but provide essential elements to quantitative formulations of robustness management with data quality in information systems.

3. Formal Model for Information Systems

This section first proposes an extended well-structured WF-net to model information systems formally, and then describes the error propagation in the model.

3.1. Business Process Modeling

Definition 1. $PN = (P, T_t \cup T_b \cup T_d; F)$ is an extended well-structured WF-net if and only if : (1) P is a nonempty set of places, T_t is a nonempty set of task transitions, T_b is a set of block transitions and T_d is a set of dummy task transitions with $P \cap (T_t \cup T_b \cup T_d) = \varnothing$, $T_t \cap T_b = \varnothing$, $T_t \cap T_d = \varnothing$ and $T_b \cap T_d = \varnothing$, and F is called the set of flow relations with $F \subseteq \{P \times (T_t \cup T_b \cup T_d)\} \cup \{(T_t \cup T_b \cup T_d) \times P\}$; (2) PN has two special places: i and o. Place i denotes the start of processes: $\cdot i = \varnothing$. Place o means the end of processes: $o \cdot = \varnothing$; (3) the net synthesized by adding a transition t^* to link i with o (*i.e.,* $\cdot t^* = o$ and $t^{*\cdot} = i$) is strongly connected; (4) for any pair of nodes x and y of which one is a place and the other is a transition and for any pair of elementary paths C_1 and C_2 from x to y, $C_1 \cap C_2 = (x, y)$ holds.

Business processes are driven by cases which in the paper are orders, and are accomplished by executing several tasks which are modeled as task transitions in *PN*. Building blocks AND–split (fork) and AND–join (join) are denoted as block transitions, while building blocks XOR–split (branch) and XOR–join (merge) are represented by places in *PN*. Each transition should start and end with a place. Only above mentioned building blocks are considered herein for other advanced blocks such as OR-split can be converted into the combinations of these four [24]. If two branches or merges are directly connected (e.g., branches 1 and 2 or merges 1 and 2 shown in Figure 2), a dummy task transition is inserted between them (e.g., dummy task transition 1 between branches 1 and

2 or dummy task transition 2 between merges 1 and 2). Tokens in the places represent objects which in the paper are data units. The data flow passing through tasks in the business processes is represented by arcs *F* with arrows from places to transitions or from transitions to places. Flow patterns between tasks may be sequence, parallel split, synchronization, exclusive choice or simple merge, of which the latter four are respectively realized by fork, join, branch and merge. Tokens flow along the paths leading from *i* to *o*, which denotes the completion of orders. In *PN*, the four building blocks are well balanced. That is, parallel flows initiated by the same fork (branch) should be joined by the same join (merge). Definition 1 makes business process modeling more formal, and business processes studied herein are supposed to be well-structured.

3.2. Error Propagation

Inadequate or failed execution of tasks in business processes will introduce different error types, and errors introduced at a task will propagate to its downstream tasks along data flow. Given that different flow patterns exist between tasks, error propagation between tasks can be realized in different paths with corresponding possibilities. An elementary path from n_1 to n_k in *PN* is a sequence $C = (n_1, n_2, \cdots, n_{k-1}, n_k)$ where $n_i \in P \cup (T_t \cup T_b \cup T_d)$, arc $(n_i, n_{i+1}) \in F$ for $1 \le i \le k$ -1 [11]. The following mentioned paths all refer to elementary paths. Tasks are disconnected if no path exists between them, otherwise they are connected. For connected tasks, two tasks are directly connected if no tasks lie on the paths between them, otherwise they are indirectly connected. Thus, error transition probability between any two tasks is either direct transition probability or indirect transition probability, and the transition probability between two disconnected tasks can be either direct or indirect.

Let $V = [v_{ij}]$ be the direct transition probability matrix where v_{ij} is the probability that errors directly flow from tasks *i* to *j*, which is determined by the building block between them [4]. As shown in Figure 1, in the sequence pattern, error transition probability between tasks 1 and 2 is 1 (*i.e.,* $v_{12} = 1$) for only one path exists between them. In the exclusive choice pattern, either task 2 or task 3 takes place after executing task 1. Data units output by task 1 flow either from task 1 to task 2 or to task 3 with probabilities *p* and 1-*p*. Thus, error transition probabilities from task 1 to tasks 2 and 3 are *p* and 1-*p* (*i.e.,* $v_{12} = p$, $v_{13} = 1 - p$). For simple merge pattern, data go from task 1 or task 2 to task 3, the transition probability between tasks 1 and 3 equals the probability between tasks 2 and 3, with the value of 1 (*i.e.,* $v_{13} = v_{23} = 1$). In the parallel split pattern, tasks 2 and 3 synchronize after finishing task 1 and data with errors output by task 1 flow to tasks 2 and 3 simultaneously. Therefore, error transition probabilities from task 1 to tasks 2 and 3 equal 1 (*i.e.,* $v_{12} = v_{13} = 1$). In synchronization pattern, errors flow from tasks 1 and 2 to task 3 with probabilities 1 (*i.e.,* $v_{13} = v_{23} = 1$). The reason is that task 3 starts after completing tasks 1 and 2, and errors output by tasks 1 and 2 all flow to task 3.
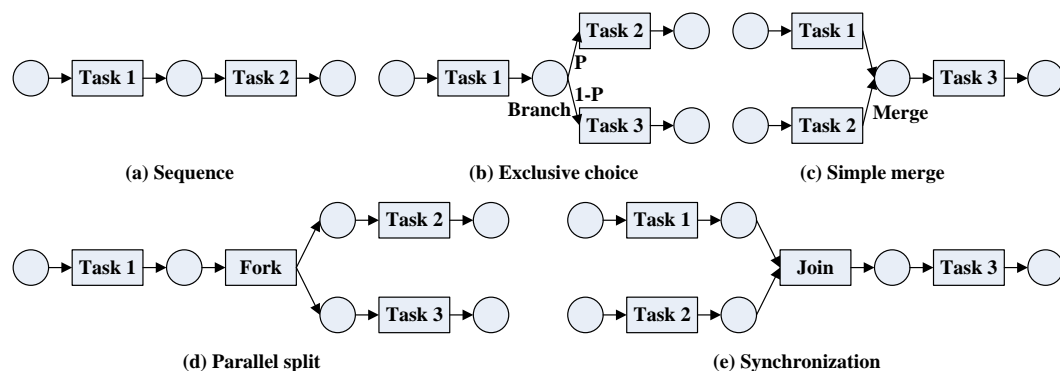

(a) Sequence　　　　**(b) Exclusive choice**　　　　**(c) Simple merge**

(d) Parallel split　　　　**(e) Synchronization**

Figure 1. Five Basic Flow Patterns [24]

Direct transition probability matrix V is also called one-step transition probability matrix V_1. Transition probability between tasks i and j is called two-step transition probability if only task k lies on the path between them, which equals the product of v_{ik} and v_{kj}. If multiple such paths exist, transition probability between them is the sum of the probability of each path. Let V_2 denote the two-step transition probability matrix. By analogy, V_3, V_4, *etc.* can be obtained. Particularly, if there exist no paths between two tasks, the transition probability between them is 0. Let V_k ($k \leq M$-1) be the matrix with maximum step of business processes containing M tasks. Error propagation matrix $E = [\gamma_{ij}]$ is defined as the sum of all transition probability matrices. The sum of the ith column of E (*i.e.*, $\gamma_{+i} = \sum_{j=1}^{M} \gamma_{ji}$) represents the in-degree propagation potential of task i, which means that γ_{+i} copies of errors will reach task i if each of its upstream tasks introduces one copy of errors; the sum of ith row (*i.e.*, $\gamma_{i+} = \sum_{j=1}^{M} \gamma_{ij}$) denotes the out-degree propagation potential of task i, which signifies that γ_{i+} copies of errors will reach its downstream tasks if task i introduces one copy of errors [5].

4. Robustness Optimization of Information Systems

This section presents quantitative formulations of control effectiveness, robustness on performance, robustness on cost and robustness on time based above formal model, and then establishes and analyzes the robustness optimization models.

4.1. Control Effectiveness

Resource such as manpower and equipment is input to handle errors after errors occurred, which is called a control. Effectiveness of a control is the percentage of an error that the control can eliminate, and Bai *et al.* [5] define control effectiveness at task i for error type n as

$$\overline{\alpha}_{in}\left(x_{in}\right) = k_{in}\gamma_{+i}x_{in}^{a_{in}} \tag{1}$$

The parameter x_{in} denotes the input level of resource at task i for error type n and equals the ratio of the actual number of resource input to all available resource. The exponent a_{in} is the marginal effectiveness of per unit input of resource. The parameter γ_{+i} catches the structural role of task i on control effectiveness. The product $k_{in}\gamma_{+i}$ quantifies the maximal effectiveness that a control can achieve at task i for error type n. The standardized factor k_{in} rescales the maximal effectiveness to the interval of [0, 1].

In reality, the functional role of task i also affects control effectiveness. Ceteris paribus, the control applied at task i with greater influence on system function is more effective than that at task j with lower influence on system function, because resource such as human handles errors more careful spontaneously at task i for its powerful influence on system function. Thus, control effectiveness at task i for error type n is rewritten as

$$\alpha_{in}\left(x_{in}\right) = k_{in}\gamma_{+i}\delta_{i}x_{in}^{a_{in}} \tag{2}$$

The parameter δ_i represents the relative weight of task i on system function which can be quantified as the residual closeness centrality [25].

4.2. Robustness Measures

Robustness in the paper is measured from dimensions of robustness on performance, robustness on cost and robustness on time.

4.2.1. Robustness on Performance

Under the perturbation of data errors, robust information systems still yield satisfising robustness on performance which is usually expressed as expected loss

caused by errors. The smaller the expected loss is, the more robust the information systems are. Let $e_{in}(x_{in})$ denote the loss per data unit caused by error type n at task i after control. The mathematical formulation of $e_{in}(x_{in})$ is

$$e_{in}(x_{in}) = l_{in}\gamma_{i+}(1 - \alpha_{in}) \tag{3}$$

The parameter l_{in} represents the average loss per data unit of error type n introduced by task i that reaches its downstream tasks. Given that error type n occurs at task i with probability p_{in}, the total expected loss (*i.e., e*) is given by

$$e = \sum_{i,n} p_{in}l_{in}\gamma_{i+}(1 - \alpha_{in}) \tag{4}$$

The smaller the value of e is, the stronger the robustness on performance (*i.e.,* R_p) is.

4.2.2. Robustness on Cost

It is intuitive that robust information systems can handle errors at a low cost. The cost is mainly determined by the wage or compensation for staff members and expenditure on technology and equipment. The lower the cost is, the more robust the information systems are. Let $\pi_{in}(x_{in})$ denote the control cost per data unit of error type n at task i. Bai *et al.* [5] define $\pi_{in}(x_{in})$ as a Cobb-Douglas function

$$\pi_{in}(x_{in}) = c_{in}x_{in}^{b_{in}} \tag{5}$$

The parameter c_{in} is the average control cost if all available resource is allocated at task i for error type n. The exponent b_{in} represents the marginal cost of per unit input of resource. Thus, the total control cost (*i.e., π*) is

$$\pi = \sum_{i,n} \pi_{in}(x_{in}) \tag{6}$$

The smaller the value of π is, the stronger the robustness on cost (*i.e., R_c*) is.

4.2.3. Robustness on Time

Similarly, robust information systems can handle errors in a short time, and the time required equals the total time that all resource spends on disposing errors. The shorter the time is, the more robust the information systems are. Let t_{in} denote the average control time when all available resource is applied at task i for error type n. Let control time be twice as large as t_{in} if no resource is input. Previous literature has illustrated that Cobb-Douglas functions are robust in various applications [26]. The control time per data unit at task i for error type n under the input level of resource x_{in} (*i.e., $\omega_{in}(x_{in})$*) is thus defined as

$$\omega_{in}(x_{in}) = 2t_{in} - t_{in}x_{in}^{d_{in}} \tag{7}$$

The exponent d_{in} measures the time elasticity of resource input. Thus, the total control time (*i.e., ω*) is expressed as

$$\omega = \sum_{i,n} \omega_{in}(x_{in}) \tag{8}$$

The smaller the value of ω is, the stronger the robustness on time (*i.e., R_t*) is.

4.3. Robustness Optimization Models

Robustness defined by Sawy *et al.* [21] reveals that it is worthwhile to pursue strong robustness on cost and robustness on time only when systems yield satisficing robustness on performance. However, formulas (6) and (8) disclose that systems cannot achieve strong robustness on cost and robustness on time simultaneously. This paper thus establishes two optimization models: the first is designed to achieve target robustness on performance with maximal robustness on cost, and the second is set to realize target robustness on performance with maximal robustness on time.

4.3.1. Maximize Robustness on Cost

The robustness-on-cost-optimal model determines the optimal resource allocation policies by maximizing robustness on cost, and the model is specified as

$$P_1 : \max R_c$$
$$s.t. R_p \geq B;$$
$$0 \leq x_{in} \leq 1, \forall i, n.$$

Lemma1. The optimal input level of resource at task i for error type n that maximizes robustness on cost and satisfies target robustness on performance is

$$x_{in/c}^* = \left(1 \Big/ \sum_{i,n} E_{in}^{\frac{b_{in}}{b_{in}-a_{in}}} \left(\frac{a_{in}}{b_{in}c_{in}} \right)^{\frac{a_{in}}{b_{in}-a_{in}}} \right)^{\frac{b_{in}-a_{in}}{a_{in}}\varphi_{in}} \left(\frac{1}{\sum_{i,n} p_{in}l_{in}\gamma_{i+} - B} \right)^{\varphi_{in}} \left(\frac{a_{in}k_{in}p_{in}l_{in}}{b_{in}c_{in}} \right)^{\varphi_{in}} \left(\gamma_{+i}\gamma_{i+}\delta_i \right)^{\varphi_{in}}$$

where $E_{in} = k_{in}\gamma_{+i}\delta_i p_{in}l_{in}\gamma_{i+} \big/ \left(\sum_{i,n} p_{in}l_{in}\gamma_{i+} - B \right)$ and $\varphi_{in} \in \{0, 1/(b_{in} - a_{in})\}$. Proof of lemma 1 is shown in appendix.

Suppose that parameters k_{in}, p_{in}, l_{in}, a_{in}, b_{in} and c_{in} are constant, then $x_{in/c}^*$ is simplified as $x_{in/c}^* = K_1 \left(\gamma_{+i}\gamma_{i+}\delta_i \right)^{\varphi_{in}}$ where K_1 represents the part containing these parameters, which reveals that more resource will be allocated to tasks with greater in-degree (γ_{+i}) and out-degree (γ_{i+}) propagation potentials and greater impact on system function (δ_i). Intuitively, a task with greater in-degree and out-degree propagation potentials receives and emits more errors, errors occurred at a task with greater influence on system function impact system function to a larger extent. Such resource allocation policies detect and correct more errors, better prevent errors from propagation and better protect system function from errors by utilizing resources effectively, which is more likely to achieve target robustness on performance with maximal robustness on cost.

4.3.2. Maximize Robustness on Time

Similarly, the robustness-on-time-optimization model finds the optimal resource allocation policies by maximizing robustness on time, and the model is established as

$$P_2 : \max R_t$$
$$s.t. R_p \geq B;$$
$$0 \leq x_{in} \leq 1, \forall i, n.$$

Lemma2. The optimal input level of resource at task i for error type n that maximizes robustness on time and satisfies target robustness on performance is

$$x_{in/t}^* = \left(1 \Big/ \sum_{i,n} E_{in}^{\frac{d_{in}}{d_{in}-a_{in}}} \left(\frac{a_{in}}{d_{in}t_{in}} \right)^{\frac{a_{in}}{d_{in}-a_{in}}} \right)^{\frac{d_{in}-a_{in}}{a_{in}}\phi_{in}} \left(\frac{1}{\sum_{i,n} p_{in}l_{in}\gamma_{i+} - B} \right)^{\phi_{in}} \left(\frac{a_{in}k_{in}p_{in}l_{in}}{d_{in}t_{in}} \right)^{\phi_{in}} \left(\gamma_{+i}\gamma_{i+}\delta_i \right)^{\phi_{in}}$$

where $E_{in} = k_{in}\gamma_{+i}\delta_i p_{in}l_{in}\gamma_{i+} \big/ \left(\sum_{i,n} p_{in}l_{in}\gamma_{i+} - B \right)$ and $\phi_{in} \in \{0, 1/(d_{in} - a_{in})\}$. Proof of lemma 2 is also presented in appendix.

Analogously, parameters k_{in}, p_{in}, l_{in}, t_{in}, a_{in} and d_{in} are assumed to be constant, and K_2 represents the part containing these parameters. $x_{in/t}^*$ is then altered into $x_{in/t}^* = K_2 \left(\gamma_{+i}\gamma_{i+}\delta_i \right)^{\phi_{in}}$. The strategy to maximize robustness on time also configures more resource to tasks with bigger value of product $\gamma_{+i}\gamma_{i+}\delta_i$. Similarly, such resource

allocation policies are most likely to achieve target robustness on performance with maximal robustness on time.

5. Case Study

This section validates the proposed methodology via a pharmaceutical enterprise in China. Figure 2 presents its order-fulfillment process using extended WF-net. The process includes 11 tasks, and the execution sequence is shown in Figure 2.
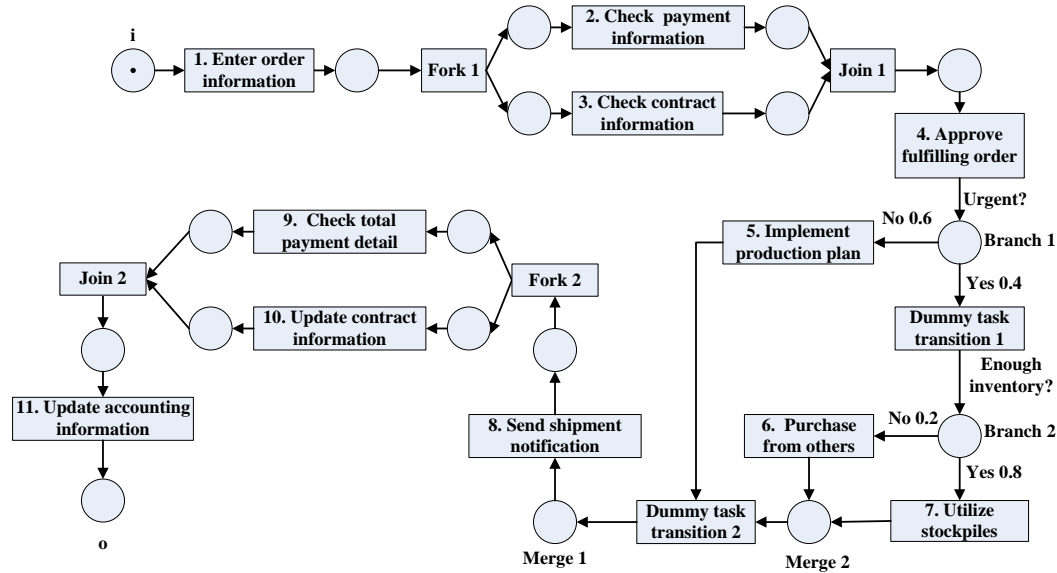

Figure 2. Business Process Using Extended WF-net

5.1. Parameter Estimation

To illustrate the proposed methodology, the paper first provides parameter estimation. On average, the enterprise accomplishes 10000 orders yearly (*i.e.,* 10000 data units), of which 4000 orders are urgent. Among the urgent orders, 3200 orders are completed by stockpiles and the remaining orders are finished via purchase. Thus, the transition probabilities from task 4 to tasks 5, 6 and 7 are 0.6, 0.4×0.2 = 0.08 and 0.4×0.8 = 0.32, as shown in Figure 2. The rest transition probabilities between tasks omitted in Figure 2 all equal 1. Based on known transition probabilities, matrices *V* and *E* are worked out, as shown in Figure 3. The *i*th row and column of both matrices represent task

$$
V = \begin{bmatrix} v_{ij} \end{bmatrix} = \begin{bmatrix}
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.6 & 0.08 & 0.32 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

$$E = [\gamma_{ij}] = \begin{bmatrix} 0 & 1 & 1 & 2 & 1.2 & 0.16 & 0.64 & 2 & 2 & 2 & 4 \\ 0 & 0 & 0 & 1 & 0.6 & 0.08 & 0.32 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 & 0.6 & 0.08 & 0.32 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0.6 & 0.08 & 0.32 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 3. Error Transition Probability Matrix and Error Propagation Matrix

The values for all p_{in} are estimated by classifying 1336 error instances occurred in the 10000 orders across error types and tasks, and the values for all l_{in} are estimated by the enterprise, as shown in Figure 4. For both matrices in Figure 4, the ith row denotes task i, while the four columns represent accuracy errors, completeness errors, consistency errors and timeliness errors separately. For example, completeness errors occur at task 1 with probability 0.1%, and such errors per order induce a loss of 100\$.

$$P = [p_{in}] = \begin{bmatrix} 2.5\% & 0.1\% & 0 & 0 \\ 0.3\% & 0 & 0 & 0 \\ 0 & 0.1\% & 0 & 0 \\ 0 & 0 & 0.1\% & 0 \\ 0.03\% & 0 & 0 & 0 \\ 0.02\% & 0 & 0 & 0 \\ 0.3\% & 0 & 0 & 0 \\ 0 & 0 & 0.1\% & 0 \\ 9\% & 0 & 0 & 0 \\ 0 & 0.2\% & 0 & 0 \\ 0.1\% & 0 & 0.5\% & 0.01\% \end{bmatrix} \qquad L = [l_{in}] = \begin{bmatrix} 350 & 100 & 0 & 0 \\ 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 100 & 0 & 0 & 0 \\ 900 & 0 & 0 & 0 \\ 100 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 100 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 925 & 0 & 200 & 50 \end{bmatrix}$$

Figure 4. Error Probability Matrix and Loss Matrix

Numerical values of all c_{in} and t_{in} estimated by the enterprise are listed in Table 1. The values of all δ_i are computed by using software called CodeBlocks, and the result is also presented in Table 1. The values of a_{in}, b_{in} and d_{in} are estimated by taking the average across tasks and error types, and the values in the paper are $a_{in} = 0.6$ ($\forall i$, n), $b_{in} = 0.9$ ($\forall i$, n) and $d_{in} = 1$ ($\forall i$, n). The scale factor k_{in} ($\forall i$, n) is set as 0.35 to ensure that the product $k_{in}\gamma_{+i}\delta_i$ is less than 1 for data errors cannot be eliminated. With the known parameters, the following section applies MATLAB to conduct a series of experiments.

Table 1. Estimation of c_{in}, t_{in} and δ_i

Parameter	Task i ($\forall n$)										
	1	2	3	4	5	6	7	8	9	10	11
c_{in} (\$)	0.76	0.76	0.76	1.85	0.65	0.65	0.65	1.75	0.65	0.65	1.75
t_{in} (\$)	2.75	2	3.5	4	2	2	2	4	2	4	4.5
δ_i (10^{-2})	1.8	3.8	3.8	32.3	5.4	5.4	5.4	32.3	3.8	3.8	1.8

5.2. Numerical Results and Analyses

To analyze whether robustness on cost and robustness on time vary across control strategies and the values of target robustness on performance or not, section 5.2.1

conducts three experiments under random control, robustness-on-cost-optimal control and robustness-on-time-optimal control respectively, and the values of target robustness on performance for the three experiments are achieving 1%, 50% and 78% increment in the robustness on performance compared with the minimal robustness on performance. To analyze trends of robustness on cost and robustness on time as target robustness on performance increases, section 5.2.2 performs 242 experiments under the robustness-on-cost-optimal control and robustness-on-time-optimal control separately, and the values of target robustness on performance for the 242 experiments are distributed uniformly in the interval of [0, 80%] where the element x means x percentage increment in the robustness on performance.

5.2.1. Comparison of Robustness

Table 2 proves that robustness on cost and robustness on time vary across control strategies and the values of target robustness on performance. More specifically, for a given target robustness on performance, the robustness on cost and robustness on time under the random control are worse than those under the other two control strategies; systems under the robustness-on-cost-optimal control possess the strongest robustness on cost, but they are the most robust on time under the robustness-on-time-optimal control, which is intuitive. For a given control strategy, when target robustness on performance increases, robustness on time increases but robustness on cost decreases. To explain above results, the input levels of resource across tasks and error types under different control strategies are worked out, as shown in Figure 5.

Table 2. Robustness Comparison among Different Strategies

Control strategy	R_c ($/10^{-2}$)	R_t (min)	Total resource input level (10^{-4})
	1% increment in the R_p		
Random control	10.11	88.9644	710
Max R_c	5.93	88.8658	660
Max R_t	6.46	88.8584	679
	50% increment in the R_p		
Random control	218.36	87.8005	15485
Max R_c	68.75	86.9942	10204
Max R_t	84.07	86.6608	10422
	78% increment in the R_p		
Random control	301.58	86.6586	22570
Max R_c	202.29	83.9518	17646
Max R_t	254.15	82. 7210	21265

For a given target robustness on performance, Figure 5 shows that random control inputs resource randomly across tasks and error types, while robustness-on-time-optimal control inputs resource preferentially to the points characterized by high error propagation potentials ($\gamma_{+i}\gamma_{i+}$), great influence on system function (δ_i), high error probability (p_{in}) or great loss (l_{in}). For example, the error probabilities of most points presented in Figure 4 (*i.e.,* most p_{in}) take the value of 0; random control still inputs resource at such points, whereas robustness-on-time-optimal control does not and inputs considerable resource at task 8 for consistency errors and task 9 for accuracy errors. The reason is that task 8 propagates the largest amount of errors (*i.e.,* $\gamma_{+8}\gamma_{8+} = 32$ is the biggest), consistency errors per order at task 8 impair system function to the largest extent and cause the hugest loss (*i.e.,* the values of $\delta_8 = 0.323$ and $l_{83} = 1000$ are bigger than those of other points), and task 9 introduces accuracy errors with the highest probability (*i.e.,* $p_{91} = 9\%$ is the biggest).Thus, resource input at such tasks detects and corrects more errors, better prevents errors from propagation, or better protects system function from errors,

which enables systems to resume normal operation in a shorter time and at a lower cost, that is, systems under the robustness-on-time-optimal control possess stronger robustness on time and robustness on cost than random control. The same pattern occurs when comparing random control with robustness-on-cost-optimal control. Moreover, Figure 5 shows that the input levels of resource across tasks and error types under the robustness-on-cost-optimal control are less than those under the robustness-on-time-optimal control, which accounts for the reality that the former control strategy yields better robustness on cost and worse robustness on time than the latter control strategy. For a given control strategy, Figure 5 indirectly illustrates that the input levels of resource across tasks increase when target robustness on performance increases, which explain the fact that increasing target robustness on performance cause decreasing robustness on cost and increasing robustness on time.
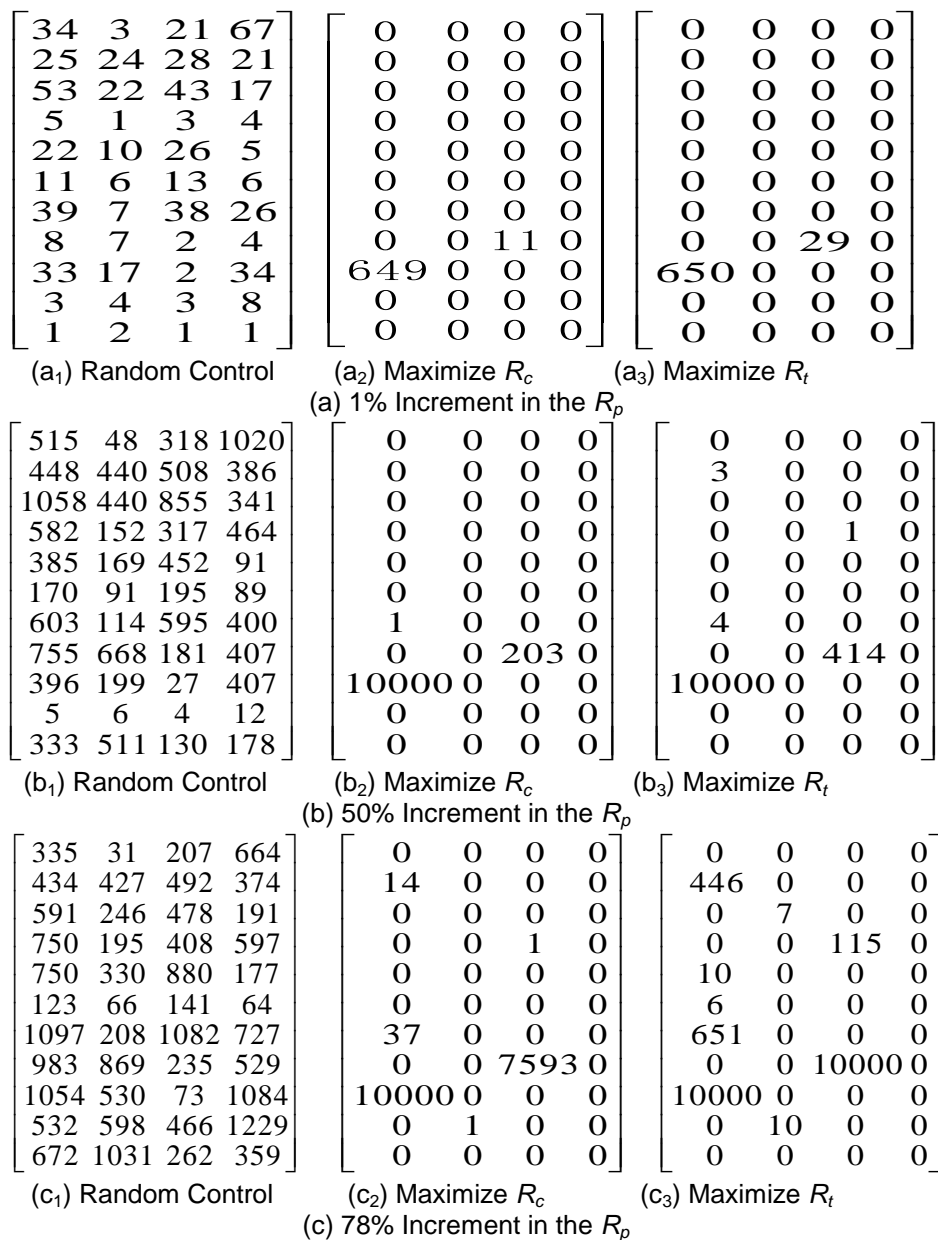
$$
\begin{bmatrix}
34 & 3 & 21 & 67 \\
25 & 24 & 28 & 21 \\
53 & 22 & 43 & 17 \\
5 & 1 & 3 & 4 \\
22 & 10 & 26 & 5 \\
11 & 6 & 13 & 6 \\
39 & 7 & 38 & 26 \\
8 & 7 & 2 & 4 \\
33 & 17 & 2 & 34 \\
3 & 4 & 3 & 8 \\
1 & 2 & 1 & 1
\end{bmatrix}
\quad
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 11 & 0 \\
649 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\quad
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 29 & 0 \\
650 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
$$

(a_1) Random Control (a_2) Maximize R_c (a_3) Maximize R_t

(a) 1% Increment in the R_p

$$
\begin{bmatrix}
515 & 48 & 318 & 1020 \\
448 & 440 & 508 & 386 \\
1058 & 440 & 855 & 341 \\
582 & 152 & 317 & 464 \\
385 & 169 & 452 & 91 \\
170 & 91 & 195 & 89 \\
603 & 114 & 595 & 400 \\
755 & 668 & 181 & 407 \\
396 & 199 & 27 & 407 \\
5 & 6 & 4 & 12 \\
333 & 511 & 130 & 178
\end{bmatrix}
\quad
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 203 & 0 \\
10000 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\quad
\begin{bmatrix}
0 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
0 & 0 & 414 & 0 \\
10000 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
$$

(b_1) Random Control (b_2) Maximize R_c (b_3) Maximize R_t

(b) 50% Increment in the R_p

$$
\begin{bmatrix}
335 & 31 & 207 & 664 \\
434 & 427 & 492 & 374 \\
591 & 246 & 478 & 191 \\
750 & 195 & 408 & 597 \\
750 & 330 & 880 & 177 \\
123 & 66 & 141 & 64 \\
1097 & 208 & 1082 & 727 \\
983 & 869 & 235 & 529 \\
1054 & 530 & 73 & 1084 \\
532 & 598 & 466 & 1229 \\
672 & 1031 & 262 & 359
\end{bmatrix}
\quad
\begin{bmatrix}
0 & 0 & 0 & 0 \\
14 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
37 & 0 & 0 & 0 \\
0 & 0 & 7593 & 0 \\
10000 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\quad
\begin{bmatrix}
0 & 0 & 0 & 0 \\
446 & 0 & 0 & 0 \\
0 & 7 & 0 & 0 \\
0 & 0 & 115 & 0 \\
10 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 \\
651 & 0 & 0 & 0 \\
0 & 0 & 10000 & 0 \\
10000 & 0 & 0 & 0 \\
0 & 10 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
$$

(c_1) Random Control (c_2) Maximize R_c (c_3) Maximize R_t

(c) 78% Increment in the R_p

Figure 5. Input Levels of Resource across Tasks and Error Types for a Given Target Robustness on Performance under Different Control Strategies (10^{-4})

5.2.2. Trend of Robustness

As mentioned before, target robustness on performance is negatively correlated with robustness on cost and positively related with robustness on time. It is more interesting and important to find out the optimal target robustness on performance where robustness on cost and robustness on time do not change dramatically. To realize the target, 242 experiments are performed respectively under the robustness-on-cost-optimal control and robustness-on-time-optimal control. Figure 6 plots the change curves of robustness on cost and robustness on time as target robustness on performance increases. The optimal target robustness on performance under the robustness-on-cost-optimal control is achieving 76.7% percentage increment in robustness on performance, whereas under the robustness-on-time-optimal control is 75.6%. Figure 6 shows that robustness on time is positively correlated with target robustness on performance—that is the good news, as the realization of stronger robustness on performance needs less time. Unfortunately, a substantially high cost is needed to satisfy the target robustness on performance exceeding the optimal value. For example, under both control strategies, if current target robustness on performance is optimal, then the cost is 2.4012$ and 2.6000$ under the two control strategies, and the time is 82.9986mins and 82.5901mins. To achieve 80% increment in the robustness on performance, cost of 4.0223$ and 6.3549$ is required respectively, and the corresponding time needed is 78.6389mins and 70.8554mins, which would decrease the robustness on cost by 67.5% and 144.4%, yet increase the robustness on time only by 5.3% and 14.2%. The conclusion from this observation is that it is unnecessary to input additional resource when target robustness on performance exceeds its optimal value. As to which control strategy to be adopted, it depends on the decision criteria of enterprises. For example, robustness-on-cost-optimal control is selected if enterprises purse strong robustness on cost and robustness on performance.
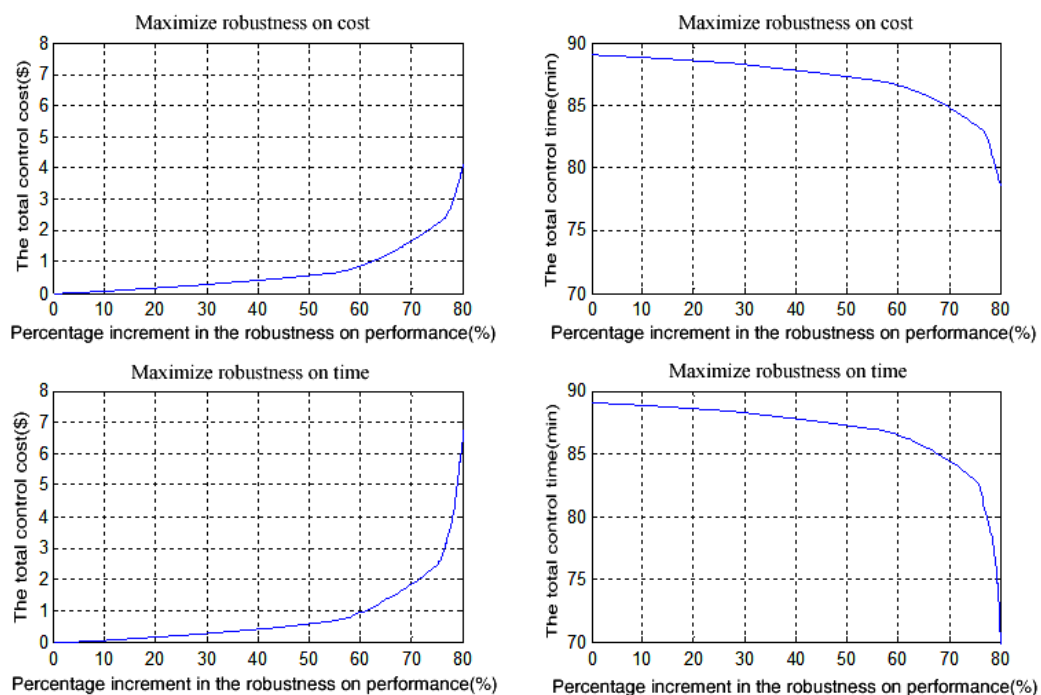

Figure 6. Trends of Robustness on Cost and Robustness on Time under the Two Control Strategies

6. Conclusions

This paper has developed a process-oriented methodology for quantitative modeling of robustness management with data quality in information systems. This methodology extends well-structured WF-net to model information systems more formally, provides necessary elements to analyze robustness quantitatively, and establishes optimization models to find the optimal control policies that achieve target robustness on performance with maximal robustness on cost or robustness on time.

Results both from theoretical and numerical analysis yield several practical implications for robustness management with data quality of information systems. For system design, the research provides guide to design system that are robust to data errors through repeatedly adjusting the structural parameters γ_{+i} and γ_{i+} and the functional parameter δ_i in numerical simulation. For system operation, this study provides feasible resource allocation policies that achieve satisfying robustness after data errors occurred. As to which resource allocation policy to be adopted, it depends on the decision criteria of enterprises (e.g., control cost, control time and performance). In addition, improving parameters p_{in}, l_{in}, c_{in}, *etc.* before occurrence of errors is an effective feed-forward control strategy to increasing system robustness.

This study has two main limitations. First, the proposed optimization models are static, while parameters such as c_{in} may change over time. In this situation, the availability of the proposed models will be greatly impaired. Second, the disruptive factor considered herein is data quality. In fact, many other factors (e.g., unreliable resource) can influence system robustness. The robustness management under several factors is more complex and difficult, and the proposed methodology thus cannot be directly applied. Future work includes developing dynamic policies and establishing robustness-optimization models incorporating several factors for robustness management of information systems.

Appendix

Proof of Lemma1

The objective to maximize robustness on cost equals the objective to minimize total control cost, and to achieve robustness on performance exceeding target B equals to make expected loss under the target B, that is,

$$
\begin{array}{ll}
\mathrm{P}_1 : \max R_c & \mathrm{P}_{11} : \min \pi = \sum_{i,n} c_{in} x_{in}^{b_{in}} \\
s.t. R_p \geq B; & \Leftrightarrow \quad s.t. e = \sum_{i,n} p_{in} l_{in} \gamma_{i+} \left(1 - k_{in} \gamma_{+i} \delta_i x_{in}^{a_{in}}\right) \leq B; \\
0 \leq x_{in} \leq 1, \forall i, n. & 0 \leq x_{in} \leq 1, \forall i, n.
\end{array}
$$

Let $E_{in} = k_{in} \gamma_{+i} \delta_i p_{in} l_{in} \gamma_{i+} / \left(\sum_{i,n} p_{in} l_{in} \gamma_{i+} - B\right)$; $\rho_{in} = x_{in}^{a_{in}}$; $\theta = b_{in}/a_{in}$. P_{11} is transformed as

$$
\begin{array}{l}
\mathrm{P}_{12} : \min R_c = \sum_{i,n} c_{in} \rho_{in}^{\theta_{in}} \\
s.t. \sum_{i,n} E_{in} \rho_{in} \geq 1; \\
0 \leq \rho_{in} \leq 1, \forall i, n.
\end{array}
$$

The Kuhn-Tucker conditions of P_{12} are

$$
\begin{cases}
\nabla \sum_{i,n} c_{in} \rho_{in}^{\theta_{in}} - \lambda \nabla \left(E_{in} \rho_{in} - 1\right) - u \nabla \rho_{in} - \eta \nabla \left(1 - \rho_{in}\right) = 0; \\
\lambda \left(E_{in} \rho_{in} - 1\right) = 0; u \rho_{in} = 0; \eta \left(1 - \rho_{in}\right) = 0; \lambda, u, \eta \geq 0.
\end{cases}
$$

The Kuhn-Tucker point that satisfies above conditions is

$$\rho_{in/c}^* = \begin{cases} \left(\dfrac{\lambda E_{in}}{\theta c_{in}}\right)^{\frac{1}{\theta-1}}, & (\mu = 0 \wedge \eta = 0) \wedge \left(\left(\dfrac{\lambda E_{in}}{\theta c_{in}} < 1 \wedge \theta > 1\right) \vee \left(\dfrac{\lambda E_{in}}{\theta c_{in}} > 1 \wedge \theta < 1\right)\right) \wedge \lambda = \left(1 \Big/ E_{in}^{\frac{\theta}{\theta-1}}\left(\dfrac{1}{\theta c_{in}}\right)^{\frac{1}{\theta-1}}\right)^{\theta-1} \\ 1 , & \left(\theta c_{in} - \lambda E_{in} + \eta = 0 \wedge \mu = 0 \wedge \sum_{i,n} E_{in} = 1\right) \vee \left(\eta = \theta c_{in} \wedge \lambda = 0 \wedge \mu = 0\right) \end{cases}$$

The optimal solution of problem P$_1$ is

$$x_{in/c}^* = \begin{cases} \left(\dfrac{\lambda a_{in} E_{in}}{b_{in} c_{in}}\right)^{\frac{1}{b_{in}-a_{in}}}, & (\mu = 0 \wedge \eta = 0) \wedge \left(\left(\dfrac{\lambda a_{in} E_{in}}{b_{in} c_{in}} < 1 \wedge \dfrac{b_{in}}{a_{in}} > 1\right) \vee \left(\dfrac{\lambda a_{in} E_{in}}{b_{in} c_{in}} > 1 \wedge \dfrac{b_{in}}{a_{in}} < 1\right)\right) \wedge \lambda = \left(1 \Big/ E_{in}^{\frac{b_{in}}{b_{in}-a_{in}}}\left(\dfrac{a_{in}}{b_{in} c_{in}}\right)^{\frac{a_{in}}{b_{in}-a_{in}}}\right)^{\frac{b_{in}-a_{in}}{a_{in}}} \\ 1 , & \left(\dfrac{b_{in} c_{in}}{a_{in}} - \lambda E_{in} + \eta = 0 \wedge \mu = 0 \wedge \sum_{i,n} E_{in} = 1\right) \vee \left(\eta = \dfrac{b_{in} c_{in}}{a_{in}} \wedge \lambda = 0 \wedge \mu = 0\right) \end{cases}$$

Given that $\left(\lambda a_{in} E_{in} / b_{in} c_{in}\right)^0 = 1$, $x_{in/c}^*$ can be unified as

$$x_{in/c}^* = \left(\lambda a_{in} E_{in} / b_{in} c_{in}\right)^{\varphi_{in}} \text{ where}$$

$$\varphi_{in} = \begin{cases} \dfrac{1}{b_{in}-a_{in}}, & (\mu = 0 \wedge \eta = 0) \wedge \left(\left(\dfrac{\lambda a_{in} E_{in}}{b_{in} c_{in}} < 1 \wedge \dfrac{b_{in}}{a_{in}} > 1\right) \vee \left(\dfrac{\lambda a_{in} E_{in}}{b_{in} c_{in}} > 1 \wedge \dfrac{b_{in}}{a_{in}} < 1\right)\right) \wedge \lambda = \left(1 \Big/ E_{in}^{\frac{b_{in}}{b_{in}-a_{in}}}\left(\dfrac{a_{in}}{b_{in} c_{in}}\right)^{\frac{a_{in}}{b_{in}-a_{in}}}\right)^{\frac{b_{in}-a_{in}}{a_{in}}} \\ 0 , & \left(\dfrac{b_{in} c_{in}}{a_{in}} - \lambda E_{in} + \eta = 0 \wedge \mu = 0 \wedge \sum_{i,n} E_{in} = 1\right) \vee \left(\eta = \dfrac{b_{in} c_{in}}{a_{in}} \wedge \lambda = 0 \wedge \mu = 0\right) \end{cases}$$

Proof of Lemma2

The objective to maximize robustness on time equals the objective to minimize total control time, and to achieve robustness on performance exceeding the target *B* equals to make expected loss under the target *B*, that is,

$$\begin{array}{ll} \text{P}_2 : \max R_t & \text{P}_{21} : \min \omega = \sum_{i,n} 2t_{in} - t_{in} x_{in}^{d_{in}} \\ s.t. R_p \geq B; & \Leftrightarrow \quad s.t. e = \sum_{i,n} p_{in} l_{in} \gamma_{i+}\left(1 - k_{in}\gamma_{+i}\delta_i x_{in}^{a_{in}}\right) \leq B; \\ 0 \leq x_{in} \leq 1, \forall i, n. & 0 \leq x_{in} \leq 1, \forall i, n. \end{array}$$

Let $E_{in} = k_{in}\gamma_{+i}\delta_i p_{in} l_{in} \gamma_{i+} / \left(\sum_{i,n} p_{in} l_{in} \gamma_{i+} - B\right)$; $\rho_{in} = x_{in}^{a_{in}}$; $\sigma = d_{in}/a_{in}$. P$_{21}$ is transformed as

$$\begin{array}{l} \text{P}_{22} : \min R_t = \sum_{i,n} 2t_{in} - t_{in}\rho_{in}^{\sigma} \\ s.t. \sum_{i,n} E_{in}\rho_{in} \geq 1; \\ 0 \leq \rho_{in} \leq 1, \forall i, n. \end{array}$$

The Kuhn-Tucker conditions of P$_{12}$ are

$$\begin{cases} \nabla \sum_{i,n} 2t_{in} - t_{in}\rho_{in}^{\theta_{in}} - \beta\nabla\left(E_{in}\rho_{in} - 1\right) - \varepsilon\nabla\rho_{in} - \tau\nabla\left(1 - \rho_{in}\right) = 0; \\ \beta\left(E_{in}\rho_{in} - 1\right) = 0; \varepsilon\rho_{in} = 0; \tau\left(1 - \rho_{in}\right) = 0; \beta, \varepsilon, \tau \geq 0. \end{cases}$$

The Kuhn-Tucker point meeting above conditions is

$$\rho_{in/t}^* = 1, \left(-\sigma t_{in} - \beta E_{in} + \tau = 0 \wedge \varepsilon = 0 \wedge \sum_{i,n} E_{in} = 1\right) \vee \left(\tau = \sigma t_{in} \wedge \beta = 0 \wedge \varepsilon = 0\right)$$

In the case that $\beta \neq 0$ and $\varepsilon = \tau = 0$, Kuhn-Tucker formulation is ineffective to find out the optimal solution. The case of $\beta \neq 0$ and $\varepsilon = \tau = 0$ indicates that $\sum_{i,n} E_{in}\rho_{in}$ strictly equals 1 and the value of ρ_{in} is within the interval of (0, 1). P$_{22}$ is rewritten as

$$P_{23} : \min R_t = \sum_{i,n} 2t_{in} - t_{in}\rho_{in}^{\sigma}$$
$$s.t. \sum_{i,n} E_{in}\rho_{in} = 1;$$
$$0 < \rho_{in} < 1, \forall i, n.$$

The Lagrangian formulation of P_{23} is

$$L(\rho_{in}, \beta) = \sum_{i,n} 2t_{in} - t_{in}\rho_{in}^{\sigma} + \beta\left(\sum_{i,n} E_{in}\rho_{in} - 1\right)$$

The Lagrangian conditions of P_{23} are

$$\begin{cases} \dfrac{\partial L}{\partial \rho_{in}} = -\sigma t_{in}\rho_{in}^{\sigma-1} + \beta E_{in} = 0; \\ \dfrac{\partial L}{\partial \beta} = \sum_{i,n} E_{in}\rho_{in} - 1 = 0. \end{cases}$$

The optimal solution of P_{23} is

$$\rho_{in/t}^{*} = \left(\frac{\beta E_{in}}{\sigma t_{in}}\right)^{\frac{1}{\sigma-1}}, \left(\varepsilon = 0 \wedge \tau = 0\right) \wedge \left(\left(\frac{\beta E_{in}}{\sigma t_{in}} < 1 \wedge \sigma > 1\right) \vee \left(\frac{\beta E_{in}}{\sigma t_{in}} > 1 \wedge \sigma < 1\right)\right) \wedge \beta = \left(1 \middle/ E_{in}^{\frac{\sigma}{\sigma-1}}\left(\frac{1}{\sigma t_{in}}\right)^{\frac{1}{\sigma-1}}\right)^{\sigma-1}$$

Thus, the optimal solution of P_{21} is

$$\rho_{in/t}^{*} = \begin{cases} \left(\frac{\beta E_{in}}{\sigma t_{in}}\right)^{\frac{1}{\sigma-1}}, \left(\varepsilon = 0 \wedge \tau = 0\right) \wedge \left(\left(\frac{\beta E_{in}}{\sigma t_{in}} < 1 \wedge \sigma > 1\right) \vee \left(\frac{\beta E_{in}}{\sigma t_{in}} > 1 \wedge \sigma < 1\right)\right) \wedge \beta = \left(1 \middle/ E_{in}^{\frac{\sigma}{\sigma-1}}\left(\frac{1}{\sigma t_{in}}\right)^{\frac{1}{\sigma-1}}\right)^{\sigma-1} \\ 1 \qquad , \left(-\sigma t_{in} - \beta E_{in} + \tau = 0 \wedge \varepsilon = 0 \wedge \sum_{i,n} E_{in} = 1\right) \vee \left(\tau = \sigma t_{in} \wedge \beta = 0 \wedge \varepsilon = 0\right) \end{cases}$$

The optimal solution of P_2 is

$$x_{in/t}^{*} = \begin{cases} \left(\frac{\beta a_{in} E_{in}}{d_{in} t_{in}}\right)^{\frac{1}{d_{in}-a_{in}}}, \left(\varepsilon = 0 \wedge \tau = 0\right) \wedge \left(\left(\frac{\beta a_{in} E_{in}}{d_{in} t_{in}} < 1 \wedge \frac{d_{in}}{a_{in}} \bullet > 1\right) \vee \left(\frac{\beta a_{in} E_{in}}{d_{in} t_{in}} > 1 \wedge \frac{d_{in}}{a_{in}} < 1\right)\right) \wedge \beta = \left(1 \middle/ E_{in}^{\frac{d_{in}}{d_{in}-a_{in}}}\left(\frac{a_{in}}{d_{in} t_{in}}\right)^{\frac{a_{in}}{d_{in}-a_{in}}}\right)^{\frac{d_{in}-a_{in}}{a_{in}}} \\ 1 \qquad , \left(-\frac{d_{in} t_{in}}{a_{in}} - \beta E_{in} + \tau = 0 \wedge \varepsilon = 0 \wedge \sum_{i,n} E_{in} = 1\right) \vee \left(\tau = \frac{d_{in} t_{in}}{a_{in}} \wedge \beta = 0 \wedge \varepsilon = 0\right) \end{cases}$$

Given that $\left(\beta a_{in} E_{in} / d_{in} t_{in}\right)^{0} = 1$, $x_{in/t}^{*}$ can be unified as $x_{in/t}^{*} = \left(\beta a_{in} E_{in} / d_{in} t_{in}\right)^{\phi_{in}}$ where

$$\phi_{in} = \begin{cases} \frac{1}{d_{in} - a_{in}}, \left(\varepsilon = 0 \wedge \tau = 0\right) \wedge \left(\left(\frac{\beta a_{in} E_{in}}{d_{in} t_{in}} < 1 \wedge \frac{d_{in}}{a_{in}} \bullet > 1\right) \vee \left(\frac{\beta a_{in} E_{in}}{d_{in} t_{in}} > 1 \wedge \frac{d_{in}}{a_{in}} < 1\right)\right) \wedge \beta = \left(1 \middle/ E_{in}^{\frac{d_{in}}{d_{in}-a_{in}}}\left(\frac{a_{in}}{d_{in} t_{in}}\right)^{\frac{a_{in}}{d_{in}-a_{in}}}\right)^{\frac{d_{in}-a_{in}}{a_{in}}} \\ 0 \qquad , \left(-\frac{d_{in} t_{in}}{a_{in}} - \beta E_{in} + \tau = 0 \wedge \varepsilon = 0 \wedge \sum_{i,n} E_{in} = 1\right) \vee \left(\tau = \frac{d_{in} t_{in}}{a_{in}} \wedge \beta = 0 \wedge \varepsilon = 0\right) \end{cases}$$

Acknowledgements

References

[1]   M. L. Brodie, "Data Quality in Information Systems", Information & Management, vol. 3, no. 6, **(1980)**, pp. 245-258.

[2]   R. Krishnan, J. Peters, R. Padman and D. Kaplan, "On Data Reliability Assessment in Accounting Information Systems", Information Systems Research, vol. 16, no. 3, **(2005)**, pp. 307-326.

[3]   B. Heinrich and M. Klier, "Metric-Based Data Quality Assessment-Developing and Evaluating a Probability-Based Currency Metric", Decision Support Systems, vol. 72, **(2015)**, pp. 82–96.

[4]   X. Bai, M. Nunez and J. R. Kalagnanam, "Managing Data Quality Risk in Accounting Information Systems", Information Systems Research, vol. 23, no. 2, **(2012)**, pp. 453-473.

[5]   X. Bai, R. Krishnan, R. Padman and H. J. Wang, "On Risk Management with Information Flows in Business Processes", Information Systems Research, vol. 24, no. 3, **(2013)**, pp. 731-749.

[6]   Y. Wand and R. Y. Wang, "Anchoring Data Quality Dimensions in Ontological Foundations", Communications of the ACM, vol. 39, no. 11, **(1996)**, pp. 86–95.

[7]   R. Y. Wang, V. C. Storey and C. P. Firth, "A Framework for Analysis of Data Quality Research", IEEE Transaction on Knowledge & Data Engineering, vol. 7, no. 4, **(1995)**, pp. 623–640.

[8]   R. Y. Wang and D. M. Strong, "Beyond Accuracy: What Data Quality Means to Data Consumers", Journal of Management Information Systems, vol. 12, no. 4, **(1996)**, pp. 5–33.

[9]   D.M. Strong, Y. W. Lee and R. Y. Wang, "Data Quality in Context", Communications of the ACM, vol. 40, no. 5, **(1997)**, pp. 103-110.

[10]   R. B. Lea, S. J. Adams, R. F. Boykin and W. Smieliauskas, "Modeling of the Audit Risk Assessment Process at the Assertion Level within an Account Balance", Auditing, vol. 11(Supplement), **(1992)**, pp. 152–179.

[11]   W. M. P. Van der Aalst, "The Application of Petri Nets to Workflow Management", Journal of Circuits, Systems & Computers, vol. 8, no. 1, **(1998)**, pp. 21-66.

[12]   R.S. Aguilar-Savén, "Business Process Modelling: Review and Framework", International Journal of Production Economics, vol. 90, no. 2, **(2004)**, pp. 129-149.

[13]   W. Sadiq and M. E. Orlowska, "Analyzing Process Models Using Graph Reduction Techniques", Information Systems, vol. 25, no. 2, **(2000)**, pp. 117-134.

[14]   A. W. Scheer, "ARIS-Business Process Modeling", Springer, Berlin, **(2000)**.

[15]   D. Liu, J. Wang, S. C. F. Chan, J. Sun and L. Zhang, "Modeling Workflow Processes with Colored Petri Nets", Computers in Industry, vol. 49, no. 3, **(2002)**, pp. 267-281.

[16]   K. Salimifard and M. Wright, "Petri Net-Based Modelling of Workflow Systems: An Overview", European Journal of Operational Research, vol. 134, no. 3, **(2001)**, pp. 664-676.

[17]   M. Tavana, D. A. Trevisani and J. L. Dussault, "The Stability Model: An Interactive Framework for Measuring Robustness and Resiliency in Military Command and Control Systems", International Journal of Information Technology Project Management, vol. 4, no. 2, **(2013)**, pp. 18-34.

[18]   S. D. Gribble, "Robustness in Complex Systems", Proceedings of the Eighth Workshop on Hot Topics in Operating Systems, Elmau, Germany, **(2001)** May 20-22.

[19]   R. Malak, B. Baxter and C. Hsiao, "A Decision-Based Perspective on Assessing System Robustness", Procedia Computer Science, vol. 44, **(2015)**, pp. 619-629.

[20]   M. Shah, "Performance as a Measure of Robustness", Structures Congress 2009: Don't Mess with Structural Engineers: Expanding Our Role, Austin, Texas, **(2009)**, April 30-May 2.

[21]   O. A. E. Sawy and B. Nanus, "Toward the Design of Robust Information Systems", Journal of Management Information Systems, vol. 5, no. 4, **(1989)**, pp. 33-54.

[22]   N. Stricker and G. Lanza, "The Concept of Robustness in Production Systems and Its Correlation to Disturbances", Procedia CIRP, vol. 19, **(2014)**, pp. 87-92.

[23]   M. Tavana, D. A. Trevisani and J. L. Dussault, "The Fuzzy Stability Model: An Interactive Framework for Measuring Robustness and Resiliency under Uncertainty", International Journal of Data Analysis Techniques & Strategies, vol. 6, no. 2, **(2014)**, pp. 137-161.

[24]   W. M. P. Van der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski and A. P. Barros, "Workflow Patterns", Distributed & Parallel Databases, vol. 14, no. 1, **(2003)**, pp. 5-51.

[25]   C. Dangalchev, "Residual Closeness in Networks", Physica A: Statistical Mechanics & Its Applications, vol. 365, no. 2, **(2006)**, pp. 556-564.

[26]   C.W. Cobb and P. H. Douglas, "A Theory of Production", The American Economic Review, vol. 18, no. 1, **(1928)**, pp. 139-165.

Authors

Zhiting Song, She is currently a doctoral candidate in the school of business administration of South China University of Technology. Her areas of interest are information systems, service-oriented manufacturing, and complex systems.

Yanming Sun, He is currently a professor and doctoral supervisor in the school of business administration of South China University of Technology. His areas of interest are manufacturing systems, information systems, and system simulation.