International Journal of Management Science
Vol 9, No 2, November 2003

Business Process Change Design from Decision Model Perspective*

Hyun-Soo Han**
College of Information and Communications, Hanyang University,
Haengdang Dong, Seongdong Gu, Seoul, Korea

(Received Jan 2003, Revised Aug. 2003, Accepted Sep. 2003)

ABSTRACT

Various organizational factors effect successful implementation of IT enabled business transformation. Among them, the most critical success factor is deemed to overcoming change management problem. Lots of studies have been made on implementation methodologies and business process formalizations to encourage organizational members to accept new business process changes. However, the logic of process redesign still depends on qualitative problem solving techniques mostly depending on basically human intuition such as brainstorming, cause-and-effect analysis, and so on. In this paper, we develop algorithmic procedure applicable to designing various business process changes such as process automation, business process resequencing, and more radical process integration. The framework is employed from dynamic programming approach in the literature, which is based on the decision making paradigm of organizations to abstract business processes as quantitative decision models. As such, our research can fill the gap of limited development of theory based analytic methodologies for business process design, by providing objective rationale to reach the consensus among the organizational members including senior management.

1. INTRODUCTION

Business process engineering is perhaps one of the most significant efforts in the 1990's to capturing competitiveness using IT. With the rapid propagation of internet and information technologies on the web, the scope of business process re-

---

* This work was supported by Korea Research Foundation Grant (KRF-2002-003-B00046).

** Email: hahan@hanyang.ac.kr

design has been expanded across firms from internal business process optimization [5, 11]. The importance of business transformation as an extended enterprise has been addressed earlier [10, 24], and internet expedites expanding the scope of process change to across the firms [4, 25]. In the rapidly changing e-business environment, the practical usefulness of business process perspective and change management concept appears unassailable [12, 17, 41, 46].

However, as was reported that 70 percent of BPR projects failed [15, 46], the effective use of IT as well as rational process redesign is still a challenging subject. Various organizational factors effect successful implementation of IT enabled business transformation [6, 16, 23, 39]. Among them, the most critical success factor is deemed to overcoming change management problem that is defined as "potential problems due to failure to manage change from the old process to the new process" [14, 15]. The core of change management is in getting consensus on to-be model from the organizational members. Resistance to changes will necessarily occur since process oriented business transformation can restructure more familiar as-is organizational practices [36].

To encourage organizational members to accept new business process changes, practical change management techniques are widely used. Lots of studies have been made on implementation methodologies and business process formalizations [23, 46]. Process modeling tools are widely used to figure out as-is business structure and to-be design alternatives. Petri-Net [1], Architecture of Integrated Information Systems (ARIS) [13, 40], and other systematic process modeling techniques [8, 30, 37] are prevalent for effective process modeling. They are effective to allow sharing and evaluating redesign alternatives to relevant organization members by pictorially demonstrating the to-be process.

However, the logic of process redesign still depends on qualitative problem solving techniques mostly depending on basically human intuition such as brainstorming, cause-and-effect diagram analysis, and so on [38]. Compared to extensive empirical studies concentrated on the behavioral aspects of BPR implementation, development of theory based analytic framework for business process design has been rather limited [3, 18, 32, 35, 45]. In this paper, we focused on developing algorithmic framework to design to-be business process structure, which can complement qualitative problem solving procedures. As such, our research can fill the gap of limited development of analytic methodologies compared to ample literatures on qualitative implementation methodologies. Quantitative model to design and evaluate to-be process alternatives can help overcoming change management problem by providing objective rationale to reach the consensus among the organizational members including senior management.

In section 2, we analyze business process change patterns and associated IT implementation focuses. Along with general discussion on quantitative approach to business process design, decision model based approach to business process design is outlined in section 3. Decision model based algorithmic procedures to design business process changes are developed in the next section 4. Limitations and further research directions are discussed in section 5.

2. IT ENABLED BUSINESS CHANGE

In the early days of IT implementation, cost reduction from automating business activities is the most significant IT effects on organizations. Information systems allow reducing man-hour cost by automating routine work or eliminating non-value adding activities. Resistance to changes will be minimal if process change is limited to simple automation without restructuring business process. For more effective use of IT as a means to strengthen competitiveness, more radical business process changes are sought in BPR. Patterns of business process change can be categorized as simple automation for process streamlining, linear resequencing, resequencing involving process parallelization, and more radical process integration. While process streamlining aims at improving process efficiency from eliminating less value adding work and waiting time between processes, business process resequencing exploits the advantages of reducing decision making efforts required to perform the task by the introduction of IT. By resequencing the task processing order of the process chain, the advantages of reducing process cost from computerization of a particular business process, which is measured in terms of uncertainty reduction, can be fully exploited [35]. Orman's model [35] deals with linear resequencing of process order to exploit the process cost changes effected after the introduction of IT.

Significant gains can be obtained by adding business process parallelization option in addition to linear resequencing. Organizational restructuring to improve performance sometimes incurs concurrent processing or process parallelization. Process parallelization is usually found in reducing planning cycle times for intra business planning or process synchronization across the firms for effective business process expansion. The industrial applications of cycle time reduction through parallelization are manifold. For instance, concurrent processing of planning activities among supplier, manufacturer, and distributors can reduce total supply chain cycle time required from raw material conversion to finished goods, and selling through distributor [25]. Similarly, total lead time for new product development can be radically shortened by parallelizing new product development processes among the firms, which were carried out sequentially and subsequently linked after the pre-processes are completed [22]. As an enabler to concurrent processing, web based information technologies play a crucial role to allow information sharing among them. Figure 1illustrates concurrent processing of new product development using web based information technologies.

Business process integration affecting organizational structure is more complicated issue. From decision aspects of the organization process, it seeks global optimal through aggregating local processes which runs in local optimal status due to the separation. For example, integrated single organizational unit can manage both production planning and sales planning to overcome functional local optimization. The effective use of decision support systems can support business process integration. As illustrated in Figure 2, the business process integration evolves from functional integration to interface integration usually for coordinating trade-off goals among functional departments, and more extended business units across the supply chain. Business process change patterns and associated IT focus for performance improvement are summarized in Table 1.

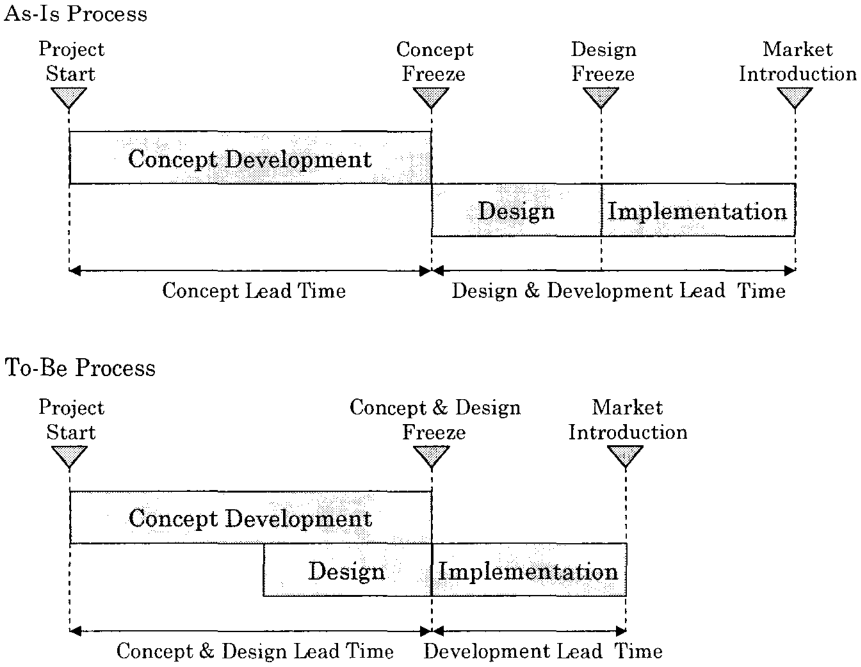


Figure 1. Business process parallelization example [22]

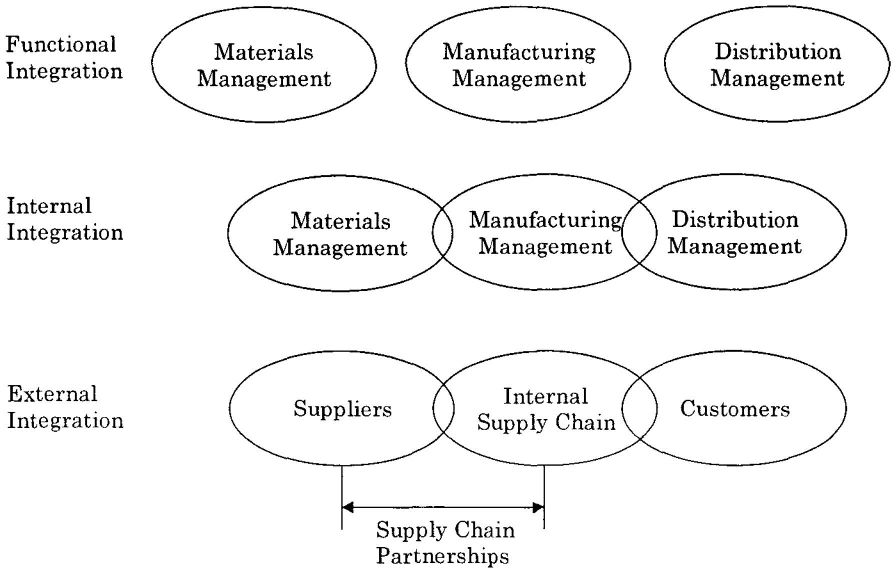
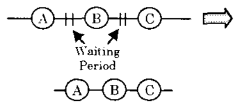
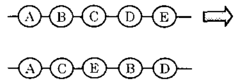
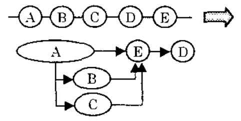
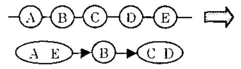


Figure 2. Business process integration stages [29]

Table 1. Business process change patterns and IT focus

Pattern		Change focus
Simple automation		• Streamlining process to eliminate less value adding work and process redundancy • Rudimentary benefit from system integration
Resequencing		• Evolving from business process automation • Effected by process cost reduction from IT implementation • More radical process change than simple automation
Process parallelization		• More effective to cycle time reduction • Applicable to business process expansion such as supply chain synchronization, new product development • Web based IT utilization for information sharIng
Process integration		• Considering benefit from local optimal to global optimal: e.g. sales and production integration • Usually requires rigorous decision support system development

3. DECISION MODEL APPROACH

The advantage of quantitative approaches is in modeling capability to analyze the business operations systematically, and evaluating the effect of business process change. Computer simulation techniques and mathematical programming based algorithmic approaches are attempted to find the optimal business process structure [4, 12, 18, 26, 35, 45]. While mathematical programming techniques have difficulty in incorporating behavioral aspects of business process, decision making process based analytic approaches are well suited in performance appraisal of organizational structure alternatives, organizational design, and information & decision support systems design [2, 19-21, 27, 28, 34, 35].

Based on the decision making paradigm of organizations [20, 21, 43], business processes are abstracted to quantitative decision models. Business process sequencing and restructuring organizational hierarchy problems are examined in light of reducing total cost of implementing decision tasks. Though this approach cannot sufficiently reflect strategic value aspects of process, it is effective as a process redesign rationale for IT intensive business transformation. Decision model based view of process can reflect the advantage of efficiencies created by information technologies. Our algorithmic procedure is based on decision making paradigm of organizations, and extension of the Orman's model management approach [35]. Orman's model management approach to business process reengineering is outlined as follows.

Viewing the firm as a sequence of decision process, business processes are interpreted as collection of decision models. A decision model is a construct that transforms input data into a specific decision that satisfies the constraints of the model. Each decision task t_i is characterized by two parameters, the cost c_i, and the selectivity s^v_i for each input variable ν of task i. The cost of a task is intuitively defined as the resources used to execute the task, such as the value of the decision maker's time for a special decision task. Formally the cost of a decision making is defined by the complexity of the implementing algorithm, which is a function of the size of its input variables. Selectivity is defined as the remaining search space after the execution of the task as a percentage of the initial search space or alternatively $1 - r_i$, where r_i means the percent reduction in search space affected by the execution of the task. This model management approach does not require a utility function, which is a major advantage since utility functions are notoriously difficult to obtain and verify.

A model T with tasks $t_1, t_2, \cdots, t_n$ is given. Each t_i has a complexity function $c_{i,}$

and selectivity s_i. The selectivity s_i is defined as $1 - r_i$. A single model problem is to find the optimum sequence of execution for the tasks $t_1, t_2, \cdots, t_n$ to minimize the cost of information processing. This problem is a special case of the sequential decision making problem [31] that can be formulated as a dynamic programming problem as follows.

$$f(T_i) = MIN_{t\rho \in T - T_i} (\Pi s_i) t_\rho + f(T_i \cup t_\rho)$$

The objective is to find the optimum cost $f(\Phi)$ where $f(T) = 0$ and Φ is the null set. Intuitively $f(T_i)$ represents the optimum cost of the partial model $T - T_{i,}$ that is, the minimum remaining cost after the execution of the tasks $T_{i.}$ It is equal to the cost of the next task t_ρ+ the minimum remaining cost after the execution of the task after $T_i \cup t_\rho$. One of the critical problems in BPR is to determine structure, and the problem is computationally solvable with complexity of $O(n!)$. Considerable computational simplification is possible since some tasks in every process are fixed in sequence because of input-output relationships with other task. This dynamic programming formulation yields reoptimized business process sequences to reflect the process cost reductions after information technology implementation.

4. BUSINESS PROCESS DESIGN

Potential applicability of decision model perspective to business process design is manifold. The primary advantage of adopting quantitative algorithmic framework is that performance improvement can be measured quantitatively. Business performance targets are usually defined quantitatively with respect to firm's internal efficiency and market performance. The practical importance of measurement driven business process reengineering has been emphasized in many literature [33]. Our business process design framework is based on Orman's dynamic programming approach for business process resequencing. We extend the framework by incorporating more practical aspects of process oriented IT-enabled business transformation, which are organized as follows.

Business process restructuring is driven by imperatives to achieve strategic performance targets. It seeks to find process redesign alternatives to enable such performance targets. Performance targets are set from benchmarking the best performances, internal process analysis, and competitor analysis. Practical BPR implementation methodologies suggest that performance target setting stage is followed after as-is analysis stage, and linked to to-be design stage subsequently.

Performance target is more than process cost reduction. As the role of IT is extended as an enabler to capturing strategic competitiveness, IT focus is more centered on its contribution to operational efficiency and market performance [9, 44]. While operational efficiency target pursue process streamlining to reduce total man-hours required for task processing, market performance targets usually includes customer service capability improvements such as lead time reduction. The scope of business process change to reduce lead time includes not only intra-firm business processes, but also total cycle time across the supply chain. Inter-business process collaboration and concurrent processing is one of the focal value creating area from effective exploitation of web based information technologies. Perhaps the most significant improvement effort of BPR has been taken on enhancing strategic performance of delivery performance and inventory through radically reducing lead time.

Process changes are mutually affected by IT deployment strategy. In most IT enabled business process change projects, IT implementation usually followed after process redesign, instead of process redesign follows after IT implementation. Since the level of process change affects the scale of IT investment, IT investment options need to be evaluated in terms of gains from performance improvement when process change is designed. IT as an enabler to allow required business process changes for performance improvement, our framework aims at providing mechanism to evaluate various IT alternatives in conjunction with corresponding process change alternatives.

In short, in the sense of extending Orman's framework, our proposed model incorporates the following aspects. First, find redesign alternatives to achieve performance goal, which means performance target driven business process change instead of reoptimizing after the introduction of IT. Second, incorporate more diverse process change patterns such as parallel processing, in addition to linear resequencing. Third, evaluate IT enablers and business process change at the same. And finally, we will introduce stage based approach instead of directly introducing continuous time variable.

4.1 Business Process Parallelization

Lead time reduction has been regarded as one of the critical performance improvement target for general IT intensive business process innovation [33]. For example, order processing cycle time reduction from customer service aspects can trigger business process changes. Lead time reduction as well as information processing cost minimization has been regarded as a critical business imperative to restructuring business process.

Extending the model to incorporate lead time necessarily involves continuous time variable to represent process duration time. However, it makes the formulation more complicated to solve, and make it more difficult to understand. Straightforward mathematical programming approach will become impractical since the problem is augmented by including continuous time variables in addition to discrete process sequencing variables. Despite the elimination of infeasible sequencing alternatives, computational complexity will rapidly increase.

In order to avoid this computational difficulty, we adopt 'stage' concept to represent lead time and k-parallelization to represent the degree of business process parallelization. Stage here means the total number of steps to complete one business process. The term k-parallelization means the total number of unit processes performed in parallel. Figure 3 illustrates k-parallelization of business processes. Suppose our target is to reduce the lead time by 50 percent for the process consists of n stages, finding optimal redesign alternative is equivalent to seek optimal solution completed through $n/2$ stages. Using our terminology, it means to find optimal redesign alternative among the set of k-parallelization alternatives, where $k = \lfloor n/2 + 1 \rfloor$. As an extreme case, the lead time can be reduced to one stage through n parallelization of n stage as-is process. We can also treat simple linear resequencing case as finding optimal solution of 1-pararallelzation options. Our stage approach is relatively simple and practical considering cumbersome nature of business process measurement [3]. Measuring exact duration of task processing time is not simple due to the contextual variation of task. For example, the duration of order acceptance task actually shows wide variation as

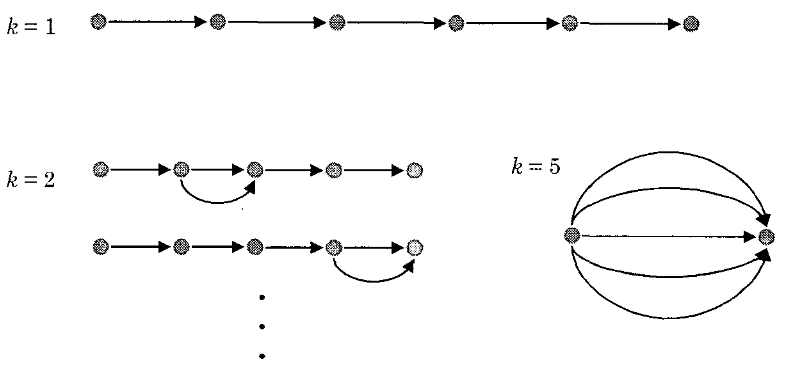


Figure 3. k-parallelization of business process

per the incoming order characteristics. Stage approach can simplify model structure and solution algorithm to make it understandable by manager. Simple abstraction of varying continuous process time to stage concept, we believe, does not fail to reflect the actual situation.

Numerous business process combinations are available for k-parallelization. For ease of computation, we consider all the possible combinations when k-parallelization optimal redesign is sought. When $k = 2$, the total number of parallelization alternatives is ${}_n\mathrm{C}_2$. The possible pattern of parallel combinations becomes more complicated as the degree of parallelization k increases.

Let $M_k = \{M_k^1, \cdots, M_k, \cdots, M_k^L\}$ as the set of possible alternatives available for k-parallelization given task index set $T = \{1,2, \cdots, n\}$. As an example, assume that we have three process parallelization alternatives for $k = 5$ such as $\{1, 2, (3,4), (5,6,7), 8, \cdots, n\}$, $\{1, 2, (3,4,5), (6,7), 8, \cdots, n\}$, and $\{1, (2,3,4,5,6), 7, 8, \cdots, n\}$. Then $M_5^1 = \{1,2, (3,4), (5,6,7), 8, \cdots, n\}$, $M_5^2 = \{1,2, (3,4,5), (6,7), 8, \cdots, n\}$, and $M_5^3 = \{1, (2,3,4, 5,6), 7,8, \cdots, n\}$ are defined each respectively. To represent each parallel task subsets contained in M_k^l, we let Z_m^l as parallel task subsets contained in M_k^l. In our previous example, M_5^1 contains two parallel task subset of (3,4), and (5,6,7), and that Z_1^l, Z_2^l are defined as $Z_1^l = (3,4)$, $Z_2^l = (5,6,7)$.

Then, for l^{th} alternative of k-parallelization, the set of task $T = \{1,2, \cdots, n\}$ can be represented as the union of the subset tasks which are combined in parallel, and the remaining set of non-parallel tasks, i.e. $T = \{ i \mid i \in Z_m^l, Z_m^l \in M_k^l \} \cup \{j \mid j \in T, j \notin Z_m^l \text{ for all } Z_m^l \in M_k^l\}$. When $k = 2$, the total number of parallelization alternatives is ${}_n\mathrm{C}_2$ in case all the possible combinations are available. The possible parallel combination alternatives become more complicated as the degree of parallelization k increases. In our example of 5-parallelization, the total number of possible combinations, $|M_5|$, becomes ${}_nC_2 \times {}_{n\text{-}2}C_3 + {}_nC_0 \times {}_nC_{5.}$, which represents L in M_5^L. In general, the total number of available k-parallelization combinations can be calculated as $|M_k| = \sum (\Pi_{u=1,\ldots,i}\ {}_{n-E_u}C_{ki})$, where $E_u = \sum_{u=1,\ldots,i-1} k_i$, $\sum k_i = k$, $0 \leq k_i \leq k/2$, $k_i \neq 1$, and each k_i is integer.

Let the optimal information processing cost of k-parallelization be $f^*(T_k)$, and optimal information processing cost of l^{th} alternative of k-parallelization corresponds to M_k^l be $f^*(T_k^l)$. The procedure K-PARALLEL (T, k) first extracts set of parallel business process combination alternatives M_k^l, $l = 1, 2, \cdots, L$, and initializes $f^*(T_k)$ as ∞. For each M_k^l, $l = 1, 2, \cdots, L$, the procedure repeats redefining task indexes, finding $f^*(T_k^l)$, and updates $f^*(T_k)$ with $f^*(T_k^l)$.

The set of tasks that are not contained in parallel task subsets is reindexed first using the task counter index counter j, which is incremented sequentially for each task. Next, the procedure redefines task indexes that are contained in parallel task subsets. We treat the set of parallel tasks in Z_m^l as virtually one task during the solution procedure. The set of tasks in Z_m^l are aggregated as virtually one task and new task index is assigned using the counter j. The selectivity and cost of the virtually single task of Z_m^l can be calculated as summation of s_i and c_i, each respectively for all the parallel tasks consisting Z_m^l.

Then procedure RESEQUENCE (T, k, l) is invoked with updated task index set of T, degree of parallelization k, and parallelization alternative index l. Each time procedure RESEQUENCE (T, k, l) is invoked, it returns optimal information processing cost $f^*(T_k^l)$ associated with M_k^l. The optimal solution of k-parallelization $f^*(T_k)$ is selected from the minimum of $f^*(T_k^l)$, $l = 1,2,\cdots,L$.

```
Procedure K-PARALLEL $(T, k)$
  Extract $M_k^l$ for all $l = 1,2,\cdots L$.
                                  ; Extract set of all k-parallelization alternatives.
  $f^*(T_k) \leftarrow \infty$                                   ; Initialize the optimal cost.
  Do  $l = 1, L$           ; Repeat the procedure for all the parallel alternative.
    Initialize  $j \leftarrow 0$;                          ; Reset new task counter index.
    For all nor-parallel task  $i \notin M_k^l$             ; Select the non-parallel tasks.
        do  $j \leftarrow j+1$; redefine task index $t_i \leftarrow j$
                                  ; Assign new task index to all non-parallel tasks.
    For each parallel task subset  $Z_m^l \in M_k^l$, repeat
                                            ; Select each parallel task subset.
        do  $j \leftarrow j+1$; redefine $t_i \leftarrow j$, $s_j \leftarrow \sum s_i$, $c_j \leftarrow \sum c_i$  for all $i \in Z_m^l$
                                                     ; Redefine parallel task.
    RESEQUENCE $(T, k, l)$
                  ; Invoke optimal solution finding routine with updated task indexes.
    If  $f^*(T_k^l) \geq f^*(T_k)$  then $f^*(T_k) \leftarrow f^*(T_k^l)$     ; Compare the minimal cost.
  Return $(f^*(T_k^l))$
```

Procedure RESEQUENCE(T, k, l) calculates optimal information processing cost $f^*(T_k^l)$ through enumeration by exploiting computational gain captured from optimal substructure of dynamic programming [9]. For ease of explanation, we use the term "level" as the distance from the root node instead of the height from leaves in the enumeration tree. At level 1, each single task i is associated with each node, so that no computation is required and the cost c_i and selectivity s_i is stored. As the tree expands downwards, the number of tasks examined for optimal sequencing are increased one by one.

For each node at level 1, descendant nodes are generated by adding single task not considered at the present node. For example, from the node corresponding to task index 1, descendant nodes of (1,2),(1,3),$\cdots$,(1,n) are generated. At level h, each node examines optimal sequencing of tasks consisting of h tasks. In order to avoid redundancy in examining the sequencing permutations, simple rule is applied in generating descendant nodes from a parent node.

Without loss of generality, we assume that nodes at level 1 are created in sequential ascending order of tasks $1,2,\cdots,n$ from left to right. Then descendant nodes at level 2 generated from r^{th} node of level 1 consists of nodes that examine combination of (r, j) where task $j \geq r+1$ and $j \leq n$. This rule enables to avoid reexamining (j, r) at level 2, which can be generated from parent node j, $j \geq r+1$. Using this rule, the number of descendant nodes from r^{th} node at level 1 becomes n-r and no descendant node is generated from the n^{th} node, so that the total number of nodes at level 2 becomes ${}_nC_2$. And the number of tasks examined for each node at level h is h. Given the total number of tasks n, the height of enumeration tree is n and the total number of nodes at level h is ${}_nC_h$. Hence, the total number of nodes in the enumeration tree becomes ${}_nC_1 + {}_nC_2 + {}_nC_3,\cdots + {}_nC_n = 2^n - 1$. The entire node traversal can be done in less than 2^n steps. The enumeration tree example with the task index set {1,2,3,4,5} is illustrated in Figure 4.

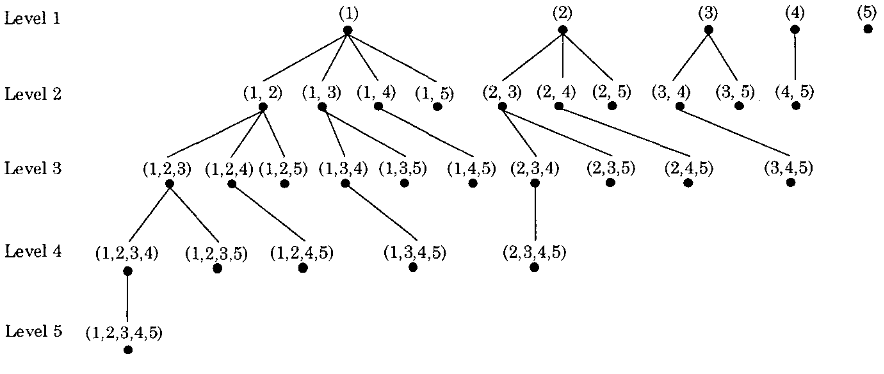


Figure 4. Enumeration tree example

Let $P_r^h = \{p_1, p_2,\cdots, p_h\} \subseteq T$ be the index set of partial tasks extracted from T to be examined for optimal sequencing at the node which is located at level h and r^{th} position from the left in the enumeration tree. Without loss of generality, we can assume p_h is the new task index added at the present r^{th} node at level h branched from parent node where optimal sequence with $\{p_1, p_2,\cdots, p_{h-1}\}$ is retained. We define optimal cost and corresponding selectivity as $R^*(p_1, p_2,\cdots, p_h)$, and $Q^*(p_1, p_2, \cdots, p_h)$ each respectively for the given set of tasks $P_r^h = \{p_1, p_2, \cdots, p_h\}$.

At present node, $R^*(p_1, p_2, \cdots, p_h)$ can be obtained by exploiting the computational result from the previous nodes at level $h-1$ or less than that. To reuse the computation result from the previous nodes, enumeration tree traversal shall be made in width first order. The optimal sequencing of P_r^h is sought through h times of comparison for all the cases from positioning p_h prior to p_1, next at between p_1 and p_2, between p_2 and p_3, subsequently up to positioning after p_{h-1}. At q^{th} comparison, by inserting p_h at q^{th} position of the parent node sequence $\{p_1, p_2, \cdots, p_{h-1}\}$, the set P_r^h is divided into three components, $\{p_1, p_2, \cdots, p_{q-1}\}$, $\{p_h\}$, $\{p_q, p_{q+1}, \cdots, p_{h-1}\}$. The computational gain comes from reusing the optimal cost of $\{p_1, p_2, \cdots, p_{q-1}\}$ and $\{p_q, p_{q+1}, \cdots, p_{h-1}\}$ calculated from the node at level q-1 and h-q-1 each respectively. The optimal substructures for the subset $\{p_1, p_2, \cdots, p_{q-1}\}$ and $\{p_q, p_{q+1}, \cdots, p_{h-1}\}$ are retained at the previous nodes, so that principle of optimality can be applicable.

At q^{th} comparison, we can compare only the minimum of two sequencing alternative, the one is $\{p_1, p_2, \cdots, p_{q-1}\}$, $\{p_h\}$, $\{p_q, p_{q+1}, \cdots, p_{h-1}\}$, and the other is $\{p_q, p_{q+1}, \cdots, p_{h-1}\}$, $\{p_h\}$, $\{p_1, p_2, \cdots, p_{q-1}\}$. Although these three divided parts can yield 3! sequencing alternatives, we don't have to recompute other four alternatives such as $\{p_h\}$, $\{p_q, p_{q+1}, \cdots, p_{h-1}\}$, $\{p_1, p_2, \cdots, p_{q-1}\}$, and $\{p_1, p_2, \cdots, p_{q-1}\}$, $\{p_q, p_{q+1}, \cdots, p_{h-1}\}$, $\{p_h\}$. They are examined when p_h is inserted in the front such as $\{p_h\}$, $\{p_1, p_2, \cdots, p_{h-1}\}$, in this case $q = 1$, and p_h is inserted at the end in case of $q = h$. We can only consider the two cases that $\{p_h\}$ is in the middle.

Let the optimal cost of former sequencing be $R_q^+(p_1, p_2, \cdots, p_h)$, which is associated with $\{p_1, p_2, \cdots, p_{q-1}\}$, $\{p_h\}$, $\{p_q, p_{q+1}, \cdots, p_{h-1}\}$. Similarly $R_q^-(p_1, p_2, \cdots, p_h)$ is defined as the optimal cost of the latter sequencing $\{p_q, p_{q+1}, \cdots, p_{h-1}\}$, $\{p_h\}$, $\{p_1, p_2, \cdots, p_{q-1}\}$. Based on the definition of selectivity, $R_q^+(p_1, p_2, \cdots, p_h)$ is calculated straightforwardly as $R^*(p_1, p_2, \cdots, p_{q-1}) + Q^*(p_1, p_2, \cdots, p_{q-1}) \times c_h + \{Q^*(p_1, p_2, \cdots, p_{q-1}) \times s_h\} \times R^*(p_q, p_{q+1}, \cdots, p_{h-1})$. Note that $Q^*(p_1, p_2, \cdots, p_{q-1})$ is the selectivity associated with optimal cost with $\{p_1, p_2, \cdots, p_{q-1}\}$, and c_h and s_h is the cost and selectivity of the new task each respectively. And that the optimal cost of $R^*(p_q, p_{q+1}, \cdots, p_{h-1})$ associated $\{p_q, p_{q+1}, \cdots, p_{h-1}\}$ shall be multiplied by the input uncertainty of $\{Q^*(p_1, p_2, \cdots, p_{q-1}) \times s_h\}$. In the same way, $R_q^-(p_1, p_2, \cdots, p_h)$, the latter sequencing cost can be calculated as $R^*(p_q, p_{q+1}, \cdots, p_{h-1}) + Q^*(p_q, p_{q+1}, \cdots, p_{h-1}) \times c_h + \{Q^*(p_q, p_{q+1}, \cdots, p_{h-1}) \times s_h\} \times R^*(p_1, p_2, \cdots, p_{q-1})$. For the given set of tasks $P_r^h =$

$\{p_1, p_2, \cdots, p_h\}$, optimal information processing cost of the case that p_h is inserted at q^{th} position of the parent node sequence $\{p_1, p_2, \cdots, p_{h-1}\}$ can be calculated from the minimum of the $R_q^+(p_1, p_2, \cdots, p_h)$ and $R_q^-(p_1, p_2, \cdots, p_h)$.

Therefore, the optimal sequence of $P_r^h = \{p_1, p_2, \cdots, p_h\}$ is obtained by examining all the cases when p_h is positioned prior to p_1, which is equivalent to $q = 1$, up to positioning after p_{h-1}, which corresponds to $q = h$. Each time, $R_q^+(p_1, p_2, \cdots, p_h)$ and $R_q^-(p_1, p_2, \cdots, p_h)$ is calculated and the minimum of the two is selected. The optimal solution finding at the r^{th} node at level h is summarized as the following formula.

$$R^*(p_1, p_2, \cdots, p_h) = Minimum_{q=1,2,\ldots,h} \{R_q^+(p_1, p_2, \cdots, p_h), R_q^-(p_1, p_2, \cdots, p_h)\}$$

Following procedure RESEQUENCE (T, k, l) algorithmically summarizes our enumeration based dynamic programming solution finding process for the given task set T and parallelization parameter k. Note that the set of parallel tasks Z_m^l are defined and task indexes and costs are updated to treat them as virtually single task in procedure K-PARALLEL (T, k).

```
Procedure RESEQUENCE ($T, k, l$)
  $n \leftarrow n-k+1$                                  ; Define enumeration tree height.
  Do  $i = 1, n$   $P_i^l \leftarrow i$;  $R^*(i) \leftarrow c_i$;  $Q^*(i) \leftarrow s_i$    ; Initialize nodes at level 1.
  Do  $h = 1, n$               ; Sequentially increase the level of the enumeration tree.
    Do  $w = 1, {}_nC_h$                         ; Traverse the tree in width first order.
      $u \leftarrow$ max $|p_i|$   where  $p_i \in P_l^h$
                                   ; Find highest task index to avoid redundancy.
      Do  $j = u+1, n$                      ; Generate descendant nodes at the next level.
        $r \leftarrow r + 1$;  $m \leftarrow j$
                    ; Increment node counter and define the newly added task index.
        Update $P_r^{h+1} \leftarrow P_w^h \cup \{m\}$
                                        ; Update the task index at the present node.
        Calculate $R^*(P_r^{h+1}) = Minimum_{q=1,2,\ldots,h+1} \{R_q^+(P_r^{h+1}), R_q^-(P_r^{h+1})\}$
                                                                   ; Find optimal cost.
    $r \leftarrow 0$                    ; Reset descendant nodes counter for the next level.
  $f^*(T_k^l) \leftarrow R^*(P_r^h)$
                    ; Optimal solution is obtained at the bottom first node in the tree.
  Return  $(f^*(T_k^l))$
```

The procedure begins with defining height of the enumeration tree, which is the number of tasks that has been adjusted to reflect the parallel tasks in $Z_m{}^l$. Since the set of parallel tasks contained in $Z_m{}^l$ is treated as virtually one task, the total number of tasks, which is the same as the height of the enumeration tree, becomes n-k+1. At level one, total n-k+1 nodes are initialized. Each node is associated with single task, and cost and selectivity is stored straightforwardly.

Node traversal is made in width first order because the computational results are exploited at the nodes generated afterwards. The total number of nodes at level h is ${}_nC_h$, and descendant nodes are generated for each node. Let the new task index added to each descendant node be m. In general, the task index set of the descendant node r at level h+1 can be defined as $P_r{}^{h+1} = P_w{}^h \cup \{m\}$, where h and w is the parent node's level and node counter each respectively. Note that the new task index m of the first generated descendant node becomes the next to the largest task index of the parent node. Newly added task index will be incremented sequentially for the descendant nodes generated successively.

The optimal sequencing of $P_r{}^{h+1}$ is sought through h+1 times of comparison for all the cases from positioning m within the task set of P_l^h successively. Not that $Q^*(P_r{}^{h+1})$ is obtained simultaneously when $R^*(P_r{}^{h+1})$ is calculated using the formula. After optimal sequencing solution is calculated, the next descendant node from the parent node is generated and the procedure is repeated.

When all the descendant nodes are generated from the parent node, the next node at the same level is visited, and the same descendant nodes generation and optimal solution finding procedures are repeated. The descendant node counter is reset after all the parents nodes at level h is traversed, and the procedure continues with incremented level counter. The optimal solution $f^*(T_k{}^l)$ is found at the first node at level n since only one node can be generated at level n. All the possible descendant nodes that can be generated from the parent nodes at level n-1 will be redundant except the one. The procedure returns $f^*(T_k{}^l)$, and procedure K-PARALLEL (T, k) sequentially compares all the $f^*(T_k{}^l)$ for all $l = 1, 2, \cdots, L$, to find the optimal solution of $f^*(T_k)$ for k-parallelization.

For simple resequencing problem of 1-parallelization, procedure RESEQUENCE traverses maximal 2^n number of nodes, and maximal n times of comparisons are performed at each node. So that the procedure returns optimal solution in $O(n2^n)$, which is much less than $O(n!)$ of n^n computational complexity suggested by Orman. As the degree of parallelization k increases, the height of the enumeration tree becomes reduced, and that optimal solution can be found in less than $O(n2^n)$. Considerable amount of infeasible parallelization alternatives and process sequencing will prevents exponential increase of computation so that the algorithm can track optimal solution in a reasonable time. With the practical assumption that the total number of plausible alternatives for parallelization is rather limited, the algorithm K-PARALLEL (T, k) can track the optimal solution in $O(cn2^n)$.

Let V_k^l be the IT investment cost as an enabler to process parallelization associated with M_k^l. Then our framework can be easily extended to incorporate IT cost V_k^l for each M_k^l, so that the sum of the information processing cost $f^*(T_k^l)$ and IT investment cost V_k^l can be compared for all the parallelization alternatives M_k^l, $l = 1,2,\cdots, L$. The total sum of the information processing cost and IT investment cost for k-parallelization can be evaluated, and optimal cost of the sum can be obtained from $Minimum_{\, l=1,2,\dots,L} \{f^*(T_k^l) + V_k^l\}$. Since parallel processing of business processes usually requires more sophisticated IT support, we expect that IT investment cost will increases as the degree of parallelization becomes high.

4.2 Business Process Integration

The algorithmic procedure designed for business process parallelization with resequencing can be applicable for business process integration. Instead of k-parallelization, k-integration to represent the degree of business process integration is used similarly. The same algorithmic procedure as finding optimal business process parallelization can be applicable to organize process integration alternatives and optimal solution finding. The stage concept will also be effective to track the optimal solution in reasonable time and efforts while avoiding too complicated computational difficulty. We don't reiterate the algorithmic procedure here since only extracting possible process integration alternatives, equivalent to extract set of parallel process alternatives M_k^l, will be different.

Business process integration is distinguished from concurrent processing in the sense of exponentially increasing uncertainty caused by integrating processes that are run separately. Since decision model perspective of organization explains that task units are designed to manage the uncertainty within the tolerance limit of human information processing and efficiencies from specialization of repetitive work, integrating sequential or parallel processes will incur rapid increase in processing cost. In this respect, the analytic model to show integration mechanism will necessarily include strategic performance enhancements as well as information processing cost reduction. We illustrate the concept using the general multi-products, multi-facility production scheduling process represented in Figure 5.

The sequence of general scheduling process is organized as mill routing, order scheduling, lot formation for each physical manufacturing process, and shipping scheduling. We assume, without loss of generality, that the manufacturing process consists of three consecutively linked distinctive physical processes. Mill routing determines which plant and facility each order is passed through, which we define as determining index k to represent the information. Order schedule determines which date the order is processed for each facility. We use index t to represent processing date. Lot formation determines material composition for unit manufacturing lot using the input materials.

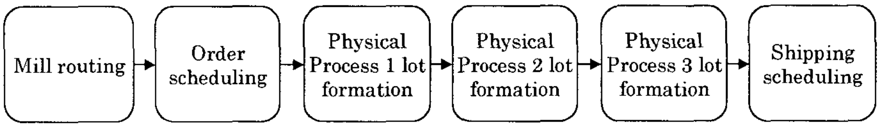


Figure 5. General production scheduling processes

Each business process is interpreted as decision model to reduce the uncertainty. For example, mill routing process reduces the uncertainty from available lots of path combinations to unique path for each order. In the same manner, order scheduling process is interpreted as uncertainty reduction procedure to define particular order processing date of each manufacturing process. The most complicated uncertainty reduction process is the lot formation process since it involves huge number of alternatives. Because these processes are sequentially connected physical processes, any combinations of the integrating them can incur considerable savings in cost and delivery performances. However, the integration in the sense of managing complexity of the uncertainty requires more complicated uncertainty, which increases in multiple orders of magnitudes.

The primary operational goal of mill routing is to maintain appropriate capacity utilization and load balancing. Delivery performance is significantly affected by order scheduling since it determines the processing sequence. Since each manufacturing process requires sufficient input materials having same or close material characteristics for production lot composition, the level of optimization of the lot formation affects productivity and inventory level including work-in-processes. If average manufacturing lot size becomes small, it affects manufacturing cost to increase. Also if we produce dummy material to make unit lot bigger, it increases unnecessary inventory or work-in-process. Delivery failure is also partly due to this sequential lot formation process that could delay the conversion process to wait for arriving similar materials to constitute minimal lot size.

We will examine business process integration of physical lot formation processes of Figure 5, which are operated in three consecutive separate processes. Let **T1** be the material conversion matrix to represent available alternatives to matching raw material **x0** to output material **x1** at physical process 1. Decision variable of $\mathbf{m1}_{ij}$ has only 0, 1 integer value. For example, $\mathbf{m1}_{5,2} = 1$ means raw material of index 5 is used to produce output material having index 2. Since lot formation is followed after mill routing and order scheduling, we use order processing facility index k, and processing time as t to represent lot formation decision variables. Then the lot formation process at plant facility k at time t becomes the matching problem modeled as follows to minimize total cost given material composition alternatives. Note that $\mathbf{x0}_t^k$ and $\mathbf{x1}_t^k$ is constant which is set previously from mill routing and order scheduling processes. The lot formation of physical process 1 is equivalent to the following decision model, and the uncertainty reduction procedure is interpreted as finding mathematical optimal or satisfactory solution of the model.

$$\begin{aligned} &\text{Minimize} && \mathbf{c1}\ \mathbf{m1}_{ij}^{t,k} \\ &\text{Subject to} && (\mathbf{x0}_t^k)\ \mathbf{T1}\ (\mathbf{m1}_{ij}^{t,k}) = \mathbf{x1}_t^k \\ &&& \mathbf{m1}_{ij}^{t,k} = \mathbf{0 \text{ or } 1} \end{aligned}$$

Same formulation is available for lot formation processes of physical process 2 and 3. We use transformation matrix **T2** to represent available matching alternatives in physical process 2. The lot formation process selects one among enormous combinatorial alternatives of matching input $\mathbf{x1}_t^k$ with output $\mathbf{x2}_t^k$. In this problem, input material $\mathbf{x1}_t^k$ is determined from physical lot formation process 1 or from completed work-in-processes at physical process 1, and $\mathbf{x2}_t^k$ is constant set from mill routing and order scheduling processes. Similarly, we can apply the same logic for physical lot formation process 3 with only index modification. Each separated decision models equivalent to distinctive business processes are formalized as follows.

$$\begin{aligned} &\text{Minimize} && \mathbf{c2}\ \mathbf{m2}_{ij}^{t,k} \\ &\text{Subject to} && (\mathbf{x1}_t^k)\ \mathbf{T2}\ (\mathbf{m2}_{ij}^{t,k}) = \mathbf{x2}_t^k \\ &&& \mathbf{m2}_{ij}^{t,k} = \mathbf{0 \text{ or } 1} \end{aligned}$$

$$\begin{aligned} &\text{Minimize} && \mathbf{c3}\ \mathbf{m3}_{ij}^{t,k} \\ &\text{Subject to} && (\mathbf{x2}_t^k)\ \mathbf{T3}\ (\mathbf{m3}_{ij}^{t,k}) = \mathbf{x3}_t^k \\ &&& \mathbf{m3}_{ij}^{t,k} = \mathbf{0 \text{ or } 1} \end{aligned}$$

Order scheduling process determines the sequence of order completion date from expected cold rolling completion date up to steel making process time t. Therefore, order scheduling implicitly determines right hand side output vectors of the model that has been treated as constant.

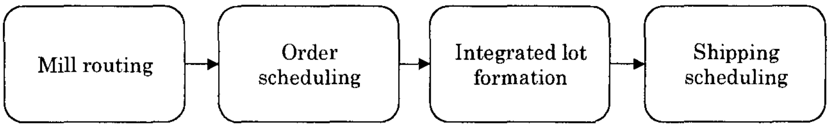


Figure 6. Redesigned processes

Process integration of the separately carried out lot formation processes illustrated in Figure 6 will necessarily improves the decision quality in the sense of finding better objective function value. The gains from integration come from simultaneously determining input and output materials from global optimal perspective by treating them as variables instead of fixed constant. However, as has been demonstrated, it involves managing exponential growth of computational complexity. The decision model of the integrated lot formation process in the redesigned processes of Figure 6 is formulated as follows. Note that $\mathbf{x1}_t^k$ and $\mathbf{x2}_t^k$ are decision variables instead of constants in this model.

$$\begin{array}{ll} \text{Minimize} & \mathbf{c1\ m1}_{ij}^{t,k} + \mathbf{c2\ m2}_{ij}^{t,k} + \mathbf{c3\ m3}_{ij}^{t,k} \\ \text{Subject to} & (\mathbf{x0}_t^k)\ \mathbf{T1}\ (\mathbf{m1}_{ij}^{t,k}) - \mathbf{x1}_t^k = 0 \\ & \qquad (\mathbf{x1}_t^k)\ \mathbf{T2}\ (\mathbf{m2}_{ij}^{t,k}) - \mathbf{x2}_t^k = 0 \\ & \qquad\qquad (\mathbf{x2}_t^k)\ \mathbf{T3}\ (\mathbf{m3}_{ij}^{t,k}) - \mathbf{x3}_t^k = 0 \\ & \mathbf{m1}_{ij}^{t,k}, \mathbf{m2}_{ij}^{t,k}, \mathbf{m3}_{ij}^{t,k} = \mathbf{0\ or\ 1} \end{array}$$

Many integrating process alternatives are available as well as sequencing order of processes. The priority of selecting subset of processes depends on computational difficulty, and gains from integration such as efficiency and delivery compliance. In general, most integrated decision problem has been known to be very difficult to solve, and that considerable efforts has been made to design the mathematical algorithm to find the optimal solution of the partially integrated processes.

Unlike process parallelization, integrating processes affects the performance in different manner so that focus of performance improvement will affect integration alternatives and the gains from integration. In our example of integrating three consecutive manufacturing lot formation processes, the manufacturing efficiency is the most significantly improved performance. If only the latter two physical lot formation processes are integrated, the delivery performance of the finished goods will be enhanced, while efficiency of the upstream manufacturing processes 1 will be less concerned.

Process integration can be extendible to other business processes such as mill routing, order scheduling, and shipping schedule. Up to now, we have assumed that index k and t each representing mill route and time of order processing is fixed for lot formation process. In case mill routing and lot formation processes are integrated, facility index k becomes variable, and new decision model will be shaped. Fragmented small piece of materials across the physical processes are the major cause of delivery failure or surplus materials increase. Waiting for similar materials to make the lot complete can incur delivery delay, and unnecessary inventory will increase if dummy materials are added for lot composition. If mill routing and lot formation process is integrated, then the chances are high to avoid small piece material problem occurred due to the mill separation. Similarly, order scheduling process to determine time t can be integrated with mill routing, and can yield huge savings if the new decision model can be solved. Further, shipping schedule can also be integrated with production processes, which can be formulated as well known integrated production distribution decision problem.

In the general production scheduling process illustrated in Figure 5, the total number of available process integration alternatives is ${}_6C_5 + ({}_6C_4 \times {}_2C_2) + ({}_6C_3 \times {}_3C_2) + ({}_6C_2 \times {}_4C_2 \times {}_2C_2)$. The logic for counting total number of parallelization alternatives and stage based k-parallelization concept can be applicable to business process integration cases in the same manner as k-integration. However, our investigation suggests the insight that strategic performance enhancement gains shall be more emphasized than business process cost efficiency in case of integration. The increase in complexity of process uncertainty from integration will radically increase information processing cost and requires development of more sophisticated analytic IT development. This cost increase should be offset by the strategic performance improvements, and the gains for each multiple distinctive strategic objectives are affected by the integration alternatives each respectively. Though the performance gains are not quantifiable with easy, the algorithmic procedures and IT investment alternative evaluation framework of business process parallelization can also be applicable to integration case.

5. DISCUSSIONS AND CONCLUSIONS

In this paper, we develop algorithmic procedures to design new business process structure. The framework is based on the decision model theory of the firm to view the business process as a collection of the decision model. Each business process activity is interpreted as the uncertainty reduction process, equivalent to solve the decision model. The algorithm is an extension of the dynamic programming approach suggested by Orman [35], which is prescriptive process resequencing framework after the introduction of IT. As is summarized in Table 2, the distinctive features of extensions are incorporating diverse change patterns, target driven approach such as lead time reduction, and adopting stage and *k*-parallelization (integration) concepts. Although the computational complexity for resequencing and alternative generation is not solvable in linear time, the optimal solution can be tractable in much less than $O(cn2^n)$ since practical limitations on feasible alternatives for parallelization and resequencing.

Table 2. Extension focuses and model framework

	Orman's framework	Proposed model	General IT-enabled business transformation
Process change initiative	Prescriptive Process redesign after introduction of IT	Target driven business process change; IT as an enabler	Target driven business process change; IT as an enabler
Performance improvements	Total process cost minimization	Stage based lead time reduction and total process cost minimization Strategic performance enhancements	• Delivery performance: lead time reduction, on-time delivery percentage, etc. • Process cost reduction • Inventory reduction • Customer service improvement
Process redesign options	Linear resequencing	Resequencing, parallel, and integration processing	• Automation • Linear resequencing • Parallel processing • Process integration

The model includes process parallelization option to reduce the cycle time. We introduced *k*-parallelization to manage cycle time reduction, that is, stage based approach to represent cycle time. We believe that stage approach will be effective in that they are easy to understand by avoiding too complicated mathematical programming caused by augmenting the problem with continuous time variable. Process parallelization allows significant lead time reduction more than business process automation or simple linear sequencing. Industrial applications for business process parallelization are manifold, for example, supply chain synchronization and new product development on the web.

Algorithmic procedures of alternative generation and optimal structure finding of parallelization can also be applicable to process integration. Physical processes are considered altogether with business process, which was left for future study in Orman [35]. Physical process can be modeled as mathematical optimization model in the sense of reducing uncertainty to extract best alternative from feasible alternative sets. It explains that process integration can achieve global optimality compared to local optimum operated on separated process that are divided under the manageable uncertainty level and specialization. Therefore, in addition to business process cost reduction, strategic performance enhancement such as on-time delivery and cost efficiency affects the integration alternative selection more significantly. The gains from integration are compared against the cost of managing exponentially growing complexity of uncertainty.

Similar to other information processing cost based organizational design models, our framework has limitations in reflecting value aspects of business process such as customer relationships, collaborations, and cultural aspects of the organization. As such, incorporating with behavioral and value aspects of the organizational theory can be our future research extension area. The procedures are thought to be very similar to heuristic business process redesign processes, so that the proposed framework can be used as an effective decision support system for business process redesign, or as a core engine to evaluate redesign alternatives for business process simulations. Applications to business networking design, supply chain synchronization, can also be potential extension areas. In addition, automating the procedures and shaping it with user interfaces can be used as an embedded decision support module for various graphical process modeling tools and ERP.

REFERENCES

[1] Aalst, W. M. P. and K. M. Hee, "Business Process Redesign: A Petri-Net-Based Approach," *Computers in Industry* 29 (1996), 15-26.

[2] Blanning, R., "Model Management Systems: An Overview," *Decision Support Systems* 9, 1 (1993), 9-18.

[3] Buzacott, J. A., "Commonalities in Reengineered Business Process – Models and Issues," *Management Science* 42, 5 (1996), 768-782.

[4] Changchien, S. W. and H. -Y. Shen, "Supply Chain Reengineering Using A Core Process Analysis Matrix and Object-Oriented Simulation," *Information & Management* 39 (2002), 345-358.

[5] Clark, T. H. and D. B. Stoddard, "Interorganizational Business Process Redesign: Merging Technological and Process Innovation," *Journal of Man-*

agement Information Systems 13, 2 (1996), 9-28.

[6] Clemons, K., M. E. Thatcher and M. C. Row, "Identifying Sources of Reengineering Failure: A Study of the Behavioral Factors Contributing Reengineering Risks," *Journal of Management Information Systems* 12, 2 (1995), 9-36.

[7] Cormen, T. H., C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, The MIT press, New York 2001.

[8] Curtis, B., M. I. Kellner and J. Over, "Process Modeling," *Communications of the ACM* 35, 9 (1992), 75-90

[9] Davern, M. J. and R. J. Kauffman, "Discovering Potential and Realizing Value from Information Technology Investments," *Journal of Management Information Systems* 16, 4 (2000), 121-143.

[10] Davison, W. H., "Beyond Re-engineering: The Three Phase of Business Transformation," *IBM Systems Journal* 32, 1 (1993), 485-499.

[11] El Sawy, O. A., *Redesigning enterprise processes for e-business*, McGraw-Hill, Boston, 2001.

[12] Fowler, A., "Operations Management and Systemic Modeling as Frameworks for BPR," *International Journal of Operations & Production Management* 18, 9/10 (1998), 1028-1056.

[13] Green, P. and M. Rosemann, "Integrated Process Modeling: an Ontological Evaluation," *Information Systems* 25, 2 (2000), 73-78.

[14] Grover, V., S. R. Jeong, W. J. Kettinger and J. T. C. Teng, "The Implementation of Business Process Reengineering," *Journal of Management Information Systems* 12, 1 (1995), 109-144.

[15] Grover, V., "From Business Reengineering to Business Process Change Management: A Longitudinal Study of Trends and Practices," *IEEE Transactions on Egineering Management* 46, 1 (1999), 36-46.

[16] Guha, S., V. Grover, W. J. Kettinger and J. T. C. Teng, "Business Process Change and Organizational Performance: Exploring An Antecedent Model," *Journal of Management Information Systems* 14, 1 (1997), 119-154.

[17] Hammer, M. and S. Stanton, "How Process Enterprises Really Work," *Harvard Business Review* (November-December 1999), 108-118.

[18] Hofacker, I. and R. Vetschera, "Algorithmic Approaches to Business Process Design," *Computers & Operations Research* 28 (2001), 1253-1275.

[19] Huber, G. P. and R. R. McDaniel, "The Decision-Making Paradigm of Organizational Design," *Management Science* 32, 5 (1986), 572-589.

[20] Huber, G. P., "The Nature of Organizational Decision Making and The Design of Decision Support Systems," *MIS Quarterly* (June 1981), 1-10.

[21] Huber, G. P., "A Theory of The Effects of Advanced Information Technologies on Organizational Design, Intelligence, and Decision Making," *Academy of Management Review* 14, 1 (1990), 47-71.

[22] Isansiti, M. and A. MacCormack, "Product Development on The Internet," in Sense & Respond: Capturing Value in The Network Era, Edit. by Bradley, S. P. and R. L. Nolan, HBS Press, Boston (1998), 175-200.

[23] Kettinger, W. and V. Grover, "Toward a Theory of Business Process Change Management," *Journal of Management Information Systems* 12, 1 (1995), 9-30.

[24] Konsynki, B. R., "Strategic Control in the Extended Enterprise," *IBM Systems Journal* 32, 1 (1993), 111-142.

[25] Lee, H. L., "Synchronized Supply Chains: The New Frontier," *Achieving Supply Chain Excellence through Technology* 1 (1999).

[26] Lin, F. and Y. Pai, "Using Multi-Agent Simulation and Learning to Design New Business Processes," *IEEE Transaction on System, Man and Cybernetics-Part A: Systems and Humans* 30, 3 (2000), 380-384.

[27] Malone, T. W. and S. A. Smith, "Modeling the Performance of Organizational Structure," *Operations Research* 36, 3 (1988), 421-436.

[28] Malone, T. W., "Modeling Coordination in Organizations and Markets," *Management Science* 33, 10 (1987), 1317-1332.

[29] Maloni, M. J. and W. C. Benton, "Supply Chain Partnership: Opportunities for Operations Research," *European Journal of Operational Research* 101 (1997), 419-429.

[30] McKay, A. and Z. Radnor, "A Characterization of Business Process," *International Journal of Operations and Production Management* 18, 9-10 (1998), 924-936.

[31] Moore, T. C. and A. B. Whinston, "A Model of Decision Making with Sequential Information Acquisition," *Decision Support Systems* 2, 4 (1986), 289- 308.

[32] Motwani, J., A. Kumar, J. Jiang, and M. Youssef, "Business Process Reengineering, A Theoretical Framework and an Integrated Model," *International Journal of Operations and Production Management* 18, 9-10 (1998), 964-977.

[33] Nissen, M. E., "Redesigning Reengineering through Measurement-Driven Inference," *MIS Quarterly* (December 1998), 509-534.

[34] Orman, L., "Information Intensive Modeling," *MIS Quarterly* (March 1987), 73-84.

[35] Orman, L., "A Model Management Approach to Business Process Engineering," *Journal of Management Information Systems* 15, 1 (1998), 187-212.

[36] O'Tool, J., *Leading Changes: The Argument for Value Based Leadership*, Ballantine Books, New York 1999.

[37] Papinniemi, J., "Creating a Model of Process Innovation for Reengineering of Business and Manufacturing," *International Journal of Production Economics* 60-61 (1999), 95-101.

[38] Rampersad, H. K., Total Quality Management: An Executive Guide to Continuous Improvement, Springer, Berlin, 2001.

[39] Ranganathan, C. and J. S. Dhaliwal, "A Survey of Business Process Reengineering Practices in Singapore," *Information & Management* 39 (2001), 125-134.

[40] Scheer, A. -W., Business Process Engineering, Springer-Verlag, Berlin, 1994.

[41] Sivestro, R. and C. Westley, "Challenging the Paradigm of the Process Enterprise: A Case-Study Analysis of BPR Implementation," *Omega* 30 (2002), 215-225.

[42] Tang, L., J. Liu, R. Aiying, Z. and Z. Yang, "A Review of Planning and Scheduling Systems and Methods for Integrated Steel Production," *European Journal of Operational Research* 133 (2001), 1-20.

[43] Tushman M. L. and D. A. Nadler, "Information Processing as an Integrating Concept in Organizational Design," *Academy of Management Review* 3, 3 (1978), 613-624.

[44] Venkatraman, N., "IT-Enabled Business Transformation: From Automation to Business Scope Redefinition," *Sloan Management Review* (Winter 1994), 73-87.

[45] Voolkner, P. and B. Werners, "A Decision Support System for Business Process Planning," *European Journal of Operational Research* 125 (2000), 633-647.

[46] Wu, I. L., "A Model for Implementing BPR based on Strategic Perspectives: An Empirical Study," *Information & Management* 39 (2002), 313-324.