

CONCEPTUAL SIMULATION MODELING OF WAREHOUSING OPERATIONS

Ming Zhou
Kitti Setavoraphan

Center for System Modeling and Simulation
College of Technology
Indiana State University
Terre Haute, IN 47809, U.S.A.

Zhimin Chen

College of Management
Shenzhen University
Shen Zhen, Guangdong, 518060, P. R. CHINA

ABSTRACT

This study focuses on the simulation modeling of warehousing operations commonly seen at general merchandize distribution centers. The key processes and their structural and behavioral characteristics were identified and analyzed. Robust representations and patterns were developed to represent the elements and logic of the simulation models for these processes to facilitate efficient and effective construction of the models. The issues of conceptual modeling such as synthesis, abstraction and specialization were also discussed.

1 INTRODUCTION

Distribution centers (DC) have been playing a more important role in modern logistics systems. It improves logistics system performance by working with other channel members (e.g. manufacturers, transportation carriers and retailers) to provide customers with better time and place utilities (Lambert 1998). In general, DC may encompass service functions such as packaging, warehousing and transporting. In this study we consider “distribution” as the set of activities related to the move, control and storage of finished products, and distribution center as a system that provides services related to the distribution. The main facility of DC is a warehouse. In order to reduce inventory cost, and costs of ordering and transportation, warehousing plays an integral role of logistics functions in terms of storage of products as inventory as well as an important link between the producer and the customer during all phases of the logistics process. Needless to say that the operations performed at the warehouse have significant impact on the performance of product distribution.

Studying warehousing operations via computer simulation has been considered an effective and powerful approach to improve the performance or design more efficient warehouse (Banks, 1998). However building a good simulation model for warehousing operations is not trivial.

It is generally a time-consuming and knowledge intensive process that requires domain, simulation and implementation knowledge (Arons 1999, 2000; McLean 2001). Such cross-domain communication has caused great difficulties in simulation modeling, and the cost for training and skill development is very high. Most models developed with the current technology are customized “rigid” models that cannot be reused or easily adapted to other even similar problems (Pidd, 1998, Zhou et al, 2004). With current technology, modeling is still an *ad-hoc* process, and modeling quality and efficiency depend largely on the skill and experience of individual modelers (McLean 2001). This research proposes a pattern-based modeling approach, i.e. identify and represent the key operational processes of warehousing as robust and reusable patterns, and use these patterns to build conceptual simulation models more efficiently and effectively. Issues in pattern-based modeling, such as structural and behavioral representations, model element definitions, classification and abstraction, are addressed in the research. A small example was presented to illustrate the proposed concepts.

2 PATTERNS OF KEY PROCESS FLOWS

In this section we identify the patterns that can be used to characterize the key process flows of warehousing operations. Particularly we focus on the three type of flows that are commonly seen among the physical distribution (finished products) systems: inbound (or receiving) process, outbound (i.e. part of order cycle process), and truck-docking processes. Each process contains a sequence of activities performed at a number of locations using a set of dedicated or shared resources. There is a defined logic that controls the flow of materials through these activities. This set of activities and the logic that configures the activities together form a pattern for the process, i.e. they can be reused in the modeling of similar applications. From a modeling perspective, these patterns are defined at an appropriate abstract level so that they are independent of the

application and implementation specifics (e.g. materials being handled and tools used for handling). Since these patterns are identified and developed to capture the structure and dynamics associated with the processes and represent them in a logical form for simulation model construction. Graphically a flow pattern is a network of activities connected through logical links (representing either a precedent or consequential relationship, and also implying the transfer of entities). First we propose a general in-bound process flow model (Figure1). The diagram shows not only the flow logic, but also the structural components (i.e. activities) involved in the flow.

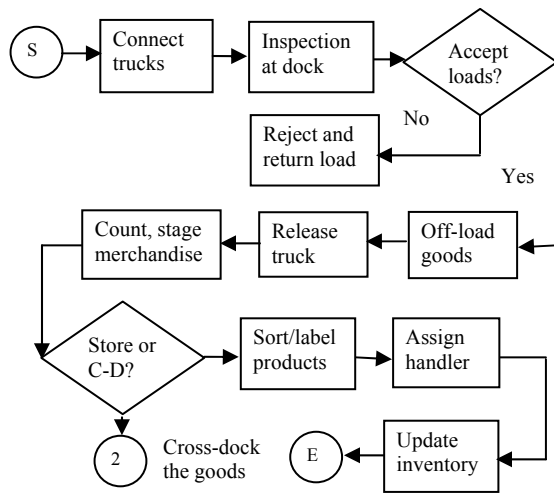


Figure 1: In-bound Flow Pattern

From the figure, inbound operations begin when one of the incoming dock doors is connected with an arriving truck. This information includes the date/time and quantities of trucks (per arrival) and amounts and types of products carried by the trucks. Upon the arrival of trucks, they are assigned to different docks for unloading. The assignment is usually based on the availability of docks, or types of goods or source of supply. If no dock is available upon arrival, trucks have to wait in a queue for available docks. Next the products will be unloaded, inspected and sorted by operators. Products are determined as Stock-Keeping-Units (SKUs). Usually at this point, it will be determined that if the products require storage locations (“direct putaway”) or cross docking (discussed later). In the case of storage, the operators will check the status of locations (availability, size and type of products), assign available locations to inspected products, and call forklifts to move and place SKUs in reserved locations. Some SKUs might be placed in temporary storage areas for waiting for valid available locations. The final step of receiving process is to update database on inventory level after placed SKUs in assigned locations.

The outbound process flow is shown in Figure 2. It generally starts with order processing (OP), including order

entry, checking inventory (backlogging possible), checking customer creditability). In general, order processing function can be performed in or outside warehouse, and the difference affects modeling details or model abstraction. For instance, if OP function is performed outside the warehouse, we can use one single activity to represent it; otherwise we can use a “sub-model” to encapsulate the details. In many cases, the managers of DC are mainly concerned with the warehousing operations (e.g. efficiency of physical handling), in this case, we may “ignore” the OP function in simulation model, and focus only on the order picking/shipping process, which includes generating picking (and packing/shipping) instructions, assign picking jobs (e.g. to forklifts), implementing picking processes (via different modes, e.g. manual, automated or aisle picking), transporting picked items to loading docks or a temporary staging area, sorting items (if necessary), loading them into trucks, updating inventory database and recording finish time and other counts.

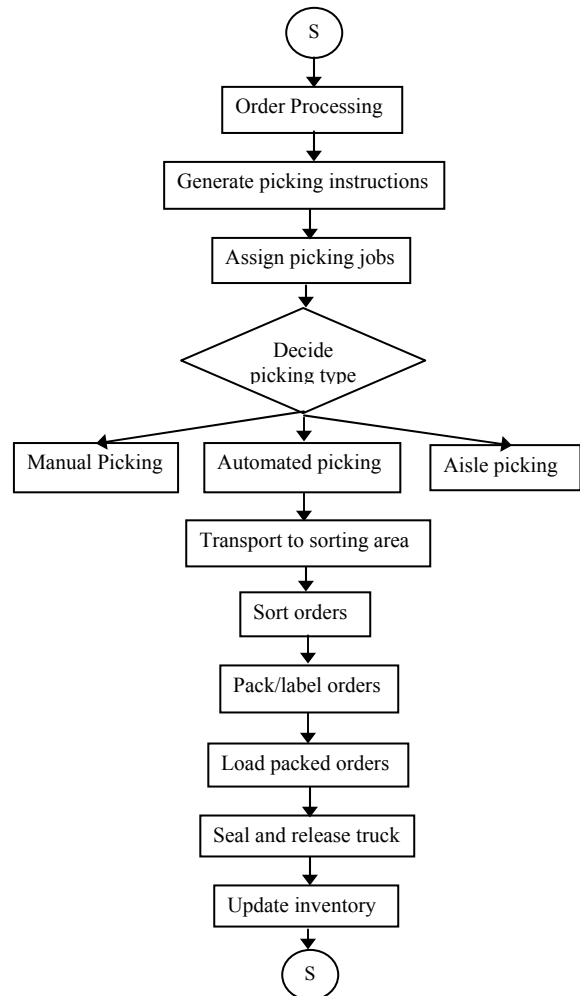


Figure 2: Outbound Flow Pattern

Truck-dock operations: Truck dock operations are involved in both inbound and outbound processes of a distribution center. In this paper we show only the study of truck docking associated with inbound process. The structure of truck dock operations contains several components: a yard for incoming trucks waiting for the doors assigned, a set of off-loading doors, off-loaders and transporters at the doors, and a staging area at the doors for offloaded freight waiting for put-away or shipping. The process flow is shown in Figure 3. When a truck arrives, it has to register at the gate to record its identification and related information. The incoming truck is then assigned to a door for offloading based on some priority or rules. If the door is not available the truck waits in the yard, otherwise it is connected to the door, and offloading process starts. The truck is finally released from the door when the offloading process is finished, and the door is then made available for the next incoming truck or idle.

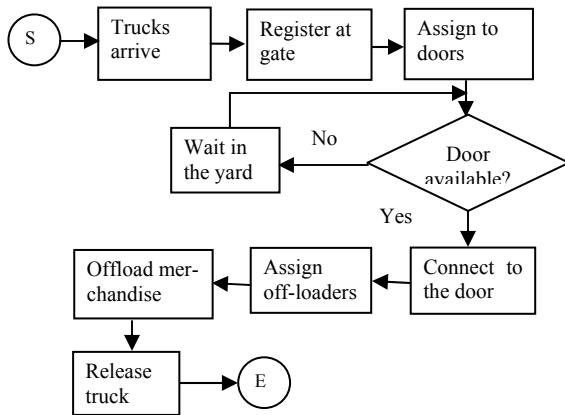


Figure 3: Truck-docking Flow Pattern

3 DECOMPOSITION OF LOGICAL ACTIVITIES

The patterns presented above are robust and can be used to model the key warehousing processes for different applications. However when building the specific models for simulation, we have to specialize these patterns, i.e. we must define the model elements and flow logic specifically with values and parameters according to the specific requirement of the application. In reality it is often necessary to break down the logical steps defined in the flow patterns to capture more detailed activities, i.e. decomposing a logical activity into a set of sub-activities to more explicitly represent the structural and behavioral characteristics of the entity flows at a detailed level. Conceptually this means we must be able to decompose the model logic from a *higher* level (i.e. more abstract level) into a *lower* level (i.e. more detailed level). Two types of decomposition are considered in this study, serial decomposition and parallel de-

composition. In serial decomposition, we decompose a higher-level activity into a sequence of sub-activities, e.g. activity 2 was decomposed into a sequence of sub-activity 2^1 , 2^2 and 2^3 (Figure 4, top). In parallel decomposition, a higher-level activity is decomposed into a set of sub-activities branched out from the predecessor of the higher-level activity (see Figure 4, bottom: activity 2 was decomposed into three parallel placed sub-activities, note that the diamond shape after node 1 indicates that the decomposition is a result of decision making branching).

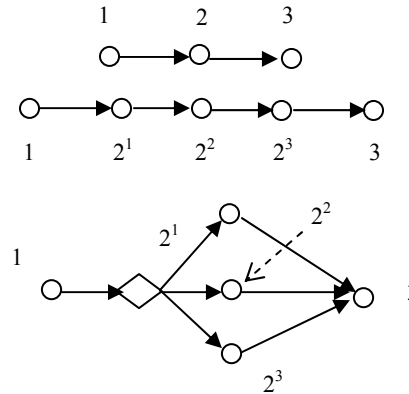


Figure 4: Serial v.s. Parallel Decomposition

We use following set of notation to formalize the decomposition problem. Let a higher level activity denoted by $P(j)$ and its input $I(j)$ and output $O(j)$; and let $T(j) =$ process time of $P(j)$ and $T(j) \sim F(\mu, \sigma^2)$. First we propose a logical model for serial decomposition, in which $P(j)$ is decomposed into a sequence of k sub-activities ($P_1(j), \dots, P_k(j)$), each with its input and output, and each sub-activity has a process time t_i , $1 \leq i \leq k$, and $t_i \sim f(\mu_i, \sigma_i^2)$.

$$I(j) \rightarrow P(j) \rightarrow O(j) \tag{1}$$



$$I_1(j) \rightarrow P_1(j) \rightarrow O_1(j) \rightarrow \dots \rightarrow I_k(j) \rightarrow P_k(j) \rightarrow O_k(j) \tag{2}$$

Subject to:

$$P(j) = \langle P_1(j), \dots, P_k(j) \rangle \tag{3}$$

$$P_1(j) \rightarrow P_2(j) \rightarrow \dots \rightarrow P_k(j) \tag{4}$$

$$\{A_{i1}(j) \oplus A_{i2}(j) \oplus \dots \oplus A_{ik}(j)\} \Leftrightarrow A(j) \quad \forall i \tag{4}$$

$$I(j) = I_1(j) \text{ and } O_k(j) = O(j) \tag{5}$$

$$T(j) = \sum_{i=1}^k t_i \tag{6}$$

$$\mu = \sum_{i=1}^k \mu_i \text{ and } \sigma^2 = \sum_{i=1}^k \sigma_i^2 \tag{7}$$

The decomposition should satisfy above constraints. Constraint (3) states that the set of sub-activities is a partition of $P(j)$, i.e. this set completely divides the functionality of $P(j)$. Constraint (4) presents a precedence requirement among the activities. Constraint (5) requires that the attributes defined for sub-activities and the aggregation of these definitions should be consistent with the attributes defined for $P(j)$ at higher level, where $A_i(j) = i$ th attribute of $P(j)$ and $A_{ih}(j) = i$ th attribute of sub-activity $P_h(j)$, $1 \leq h \leq k$. Symbol “ \oplus ” is used as a general aggregation operator, and “ \Leftrightarrow ” as a consistency operator. In many cases operator “ \Leftrightarrow ” reduces to a simple “ \in ” operation, e.g. a resource element assigned to a sub-activity \in resource set defined at a higher level. Constraint (5) defines boundary conditions, i.e. the input and output “external” to the set of sub-activities should match the original input and output associated with $P(j)$. Constraints (6) states that total process time is a sum of sub-activity times, and (7) shows the relationship between the statistical expectations of random variable $T(j)$ and t_i 's. When $T(j)$ and t_i 's follow Triangle distribution (an empirical distribution popularly used by practitioners), i.e.

$$T(j) \sim TRIA(T_{min}, T_{mode}, T_{max}), t_i \sim TRIA(t_{i_{min}}, t_{i_{mode}}, t_{i_{max}}) \quad \forall i$$

Then we can replace (7) with more useful constraints as follows:

$$\sum_{i=1}^k t_{i_{max}} \leq T_{max} \quad \text{and} \quad \sum_{i=1}^k t_{i_{min}} \geq T_{min}$$

IDEF3 (Mayer et al. 1995) has been used by a number of modelers as an effective tool to implement the decomposition of process activities. It also has a set of notation that can be used to represent the logic of decomposition to facilitate modeling process. In this study we proposed to use an IDEF3 type of procedure to implement the decomposition.

4 DEFINITIONS OF MODEL ELEMENTS

Model elements contain the definitions of those **objects** required by simulation models, including logical relationship among the objects. While many objects do represent physical things (e.g. products, pallets, orders, trucks, labors), some are used for constructing or evaluating the simulation models. We present these as follows:

Entities: those objects that flow through the system to receive services provided by a sequence of activities. Different process models require different forms of entities, e.g. for inbound and outbound process, the entities can be pallet-load, caseload or SKU unit, but for truck-docking, entity form usually is the truck or truck-load (TL). In addition to product-related entities, information-oriented enti-

ties (customer orders and replenishment orders) are also commonly used.

Resources: those objects that are mobile or placed at fixed locations according to some configuration to provide means of services, e.g. labors, transporters, staging/storage space, sorting or packing equipment. Although we define resources independently for each process flow, some resources are shared between the processes, e.g. loading/off-loading crew/equipment.

Logical Activities: an activity is a logical step that performs a defined function required by the simulation of a target process. The activities process entities, control the process flow, and collect the data generated for analysis. For efficient modeling the activities are defined at abstract levels, and each may be decomposed into a set of sub-activities. This composite nature makes it possible to represent and encapsulate the activities through a hierarchical structure. A classification of the top-level activities based on their functions is shown in the following:

- Create entities;
- Change/assign entity;
- Process/service entities (processing activities, including inspection);
- Store/hold entities;
- Aggregate/disaggregate entities (e.g. assemble/disassemble batch parts);
- Transport entities (move from one location to another);
- Branch the flow of entities (condition-based versus probability-based);
- Control the movement (temporal) of entities;
- Dispose entities;
- Collect data/statistics (counts, tallies).

A process flow (or control) logic: it defines the order of activities in and the control of an entity type flow. This logic serves as a basis for construction and specialization of the process flow patterns (to build simulation models) The types of control are described as follows.

- **Control the direction of the movement**, i.e. routing entities. It does not stop the entity flow in simulated time. Three approaches are: **Sequential:** specified by a predetermined sequence defined in a process plan; **Condition:** determined by testing a predetermined condition; **Random sampling:** determined by the result from a random sampling.
- **Control the timing of the movement**, e.g. holding entities until a certain event occurs. It may stop entity flow in simulated time. Two approaches are: **Control by condition:** hold entities until a prescribed condition becomes true; **Control by stimulus:** hold entities until a predetermined type of signal is received.

- **Control the quantity of entity flow**, e.g. allowing a limited number of entities per release (of movement).

Data/input requirements:

- Numerical attributes or properties associated with objects and activities, e.g. input distributions of entity types and entity arrival, processing time, transportation distance or transfer time.
- Global variables and expressions used to define and implement the process flow logic, including control elements.

Output requirements:

- Goals and objectives can be classified by the purpose of analysis, e.g. comparing alternatives; estimating/predicting performance measures; analyzing sensitivity of input variables; optimizing system performance. Output measures, transformed from the domain/application objectives, are usually specified in three aspects: productivity, utilization, and performance of warehousing operations:
 - *Productivity measures*: pallets, cases or SKU-units handled per labor-hour or lines selected per equipment-hour.
 - *Utilization measures*: the percent of pallet spaces filled in a warehouse and employee-hours worked versus employee-hours available, etc.
 - *Performance measures*: the ratio of actual output to standard output, e.g. cases picked per hour versus a standard picking rate, and actual return on assets employed versus budgeted return on assets employed.
- Experimental requirements: strategies and procedures to achieve what are specified by the output requirements, e.g. how to collect and analyze the data through simulation experiments? For instance:
 - Type of statistical analysis
 - Plan for data collection
 - Procedure of the analysis
- Other special requirements: e.g. schedules to represent planned capacity changes of resources, control or review policy for special type of inventory, usually depending on the type of application.

5 AN EXAMPLE

In this section we use a simple example to illustrate how to specialize the process flow pattern discussed earlier to cre-

ate conceptual simulation model for specific application. In this case we consider an *outbound process* in which picking operation is manually done, and the order processing (OP) activity needs to be decomposed to capture the details of several “sub-activities”, such as in-stock inventory checking, backloging, and customer credit checking. Limited by the space, we only show the decomposition of OP (Figure 5) and the logical flow model generated (Figure 6). Note that we have used the notation of IDEF3 to present the decomposition.

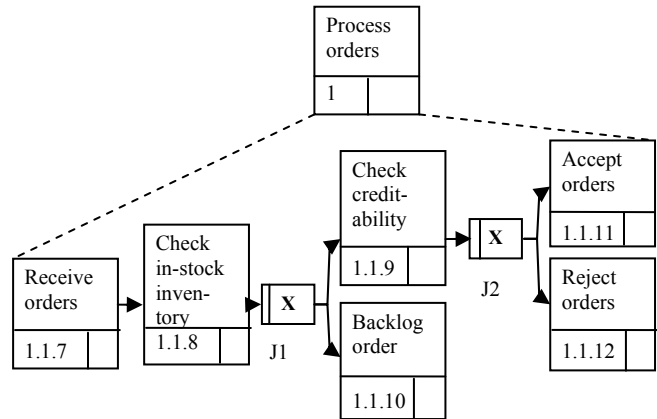


Figure 5: Decomposing Activity OP

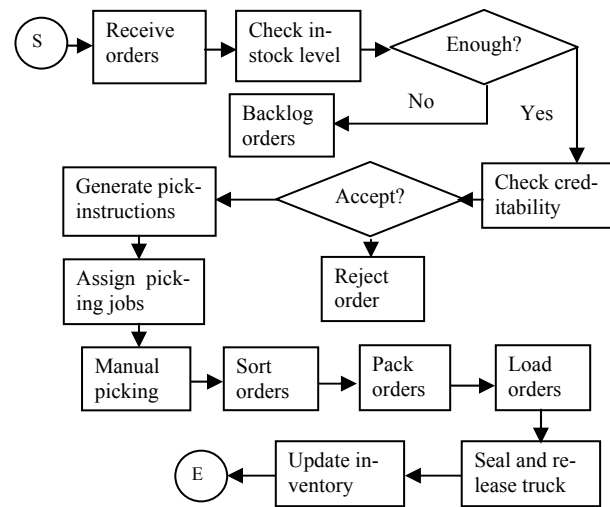


Figure 6: A Model Specified from Inbound Flow Pattern

6 CONCLUSION

In this paper we showed some initial results of a study that focus on pattern-based modeling of warehousing operations at a typical distribution center environment. The key processes, such as in-bound, outbound and truck-dock op-

erations, are identified and represented through defined networks that contain the logic dictating the flow and robust elements required by the simulation of the processes. These patterns can be specialized to create conceptual simulation models more efficiently and effectively.

REFERENCES

- Arons, H.D.S. 1999. Knowledge-based modeling of discrete-event simulation systems. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 591-597. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Arons, H. D. S. and E. V. Asperen. 2000. Computer Assistance for Model Definition. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang and P. A. Fishwick, 399-408. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers. Piscataway.
- Banks, J. (editor). 1998. *Handbook of Simulation, Principles, Methodology, Advances, Applications, and Practice*. John Wiley & Sons, Inc. New York.
- Lambert, D. M., J.R. Stock and L. M. Ellam. 1998. *Fundamentals of logistics management*. Irwin/McGraw-Hill.
- Mayer, J. Richard, C. P. Menzel, M. K. Painter, P. S. deWitte, T. Blinn and B. Perakath. 1995. Information Integration for Concurrent Engineering (IICE) IDEF3 Description Capture Method Report. Available online via http://www.ideal.com/pdf/ideal3_fn.pdf [accessed April 2005].
- McLean, C. and S. Leong. 2001. The Expanding Role of Simulation in Future Manufacturing. In *Proceedings of the 2001 Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros and M. W. Rohrer, 1478-1486. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Pidd, M. and R. B. Castro. 1998. Hierarchical Modular-Modeling in Discrete Simulation. In *Proceedings of the 1998 Winter Simulation Conference*. ed. D. J. Medeiros, E. F. Watson, J. S. Carson and M. S. Manivannan, 383-389. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Zhou, M., Y. J. Son, and Z. Chen. 2004. Knowledge representation for conceptual simulation modeling. In *Proceeding of the 2004 Winter Simulation Conference*, ed., R.G. Ingalls, M.D. Rossetti, J.S. Smith, and B.A. Peters. 450-458. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available online via <http://www.informs-sim.org/wsc04papers/054.pdf> [accessed March 2005].

AUTHOR BIOGRAPHIES

MING ZHOU is a professor and program coordinator of Mechanical Engineering Technology at Indiana State University. He is also the Director and Chief Researcher at the Center for System Modeling and Simulation of the university. He received a Ph.D. in Systems & Industrial Engineering from The University of Arizona in 1995. Dr. Zhou's research interests include knowledge based simulation and intelligent decision support systems for manufacturing, logistics and distribution systems. He has been a member of IIE and the Editorial Board, International Journal of Industrial Engineering since 1997. His email address is imming@isugw.indstate.edu.

KITTI SETAVORAPHAN is a graduate assistant at Indiana State University. Kitti received his BS degree in Electrical Engineering from Thailand, and is working towards to a Master Science degree in Industrial Technology. He is also a research assistant working at the Center for System Modeling and Simulation, Indiana State University, focusing on modeling distribution operations.

ZHIMIN CHEN is a professor and the Associate Dean of the College of Management, Shenzhen University, China. He received a Ph.D. in Engineering from Beijing Aerospace University, and had been a visiting professor at the Boston University during 1999 and 2001.