Computing and Informatics, Vol. 30, 2011, 733–747

SEARCH TREE GENERATION FOR THE EXCEPTION HANDLING OF E-COMMERCE DELIVERY PROCESS

Jin-Gyu Shin, Doug-Won Choi

Department of Systems Management Engineering
Sungkyunkwan University
300 Cheoncheon-dong, Jangan-gu, Suwon, Gyeonggi-do
440-746, Korea
e-mail: {sjg0311, dougch01}@paran.com

Dong-Cheol Lee, Yung-Cheol Byun*

Department of Computer Engineering
Jeju National University
66 Jejudaehakno, Jeju-si, Jeju-do
690-756, Korea
e-mail: {dchlee, ycb}@jejunu.kr

Communicated by Jozef Hvorecký

Abstract. A business process management system (BPMS) offers the facility to define new processes or update the existing processes. However, exceptional or non-routine tasks require the intervention of domain experts or generation of the situation specific resolution process. This paper assumes that sufficient amount of business process exception handling cases are stored in the process repository. Since the retrieval of the best exception handling process requires good understanding about the exceptional situation, context awareness is an important issue. To facilitate the representation of the exceptional situation and to enable the selection of the best exception handling process, we adopted the 'situation variable' and 'decision variable' construct. A case example for exception handling in the e-commerce delivery process is provided to illustrate how the proposed construct works. We applied the C5.0 algorithm to build the optimum search tree.

Keywords: Exception handling, process repository management, systematic context description, situation variable, decision variable, search tree generation

* corresponding author

1 INTRODUCTION

Manufacturing processes usually do not involve so many exceptional situations; and this property explains part of the reason why we can automate control and management of manufacturing processes. However, the recent progress in software technology enable us to extend the process automation technology into the area of service and business processes. Currently, there are many software packages available for the management of various business processes. Business process management system (BPMS) is a typical example [1]. Many organizations have implemented this system and are reported reaching good results [2].

It is a difficult job to automate non-routine or exceptional processes. In order to make a BPMS handle this kind of non-routine or exceptional process, we must have a process predefined and implemented into the BPMS such that it can be retrieved and applied to the resolution of the exceptional situation at the time of need [3]. An exceptional situation implies a task which occurs occasionally and has a poorly defined or undefined rules and procedures. Therefore, it usually requires the subjective judgement of the decision maker to resolve the problem.

Exceptional situation falls into the category of semi-structured or unstructured problems as discussed in [4, 5]. We can improve the task efficiency if we store the exception handling knowledge into the knowledge base and have an articulated infrastructure for sharing the knowledge. Selecting the right model for the problem situation is an important issue in a decision support system research [6]. In the same vein, selecting the appropriate process that can best handle the exceptional situation is an important issue in BPMS research.

In this paper, we introduce the two variable sets, i.e., situation variable set $S = (s_1, s_2, \ldots, s_m)$ and decision variable set $D = (d_1, d_2, \ldots, d_n)$, to enable systematic context description of exceptional problem situation and to render a useful data structure for optimum search tree generation. The situation variable set describes the customer requirements, traffic condition, etc. and is used to depict the given or uncontrollable aspects of the problem context. Decision variable set portrays the selection of alternative course of action which the problem solver can adopt to resolve the exception. The situation variable may circumscribe the scope of the decision space and the decision maker has to choose a series of action from the alternative decision space. Therefore, the specific value assigned to each decision variable explains which course of action the decision maker has chosen to resolve the exception problem. In this paper, the data structure which is composed of the situation variable and decision variable plays the key role in designing the process repository architecture for exception handling.

Section 2 discusses the review of related literature about exception problems and the corresponding resolution approaches [7–14]. Basic idea about inductive approach to the selection of exception handling process is provided in the same section. In Section 3 we demonstrate a profile of exceptional situations that could be encountered in the e-commerce delivery process. In Section 4 we provide the architecture for exception handling process repository and present a case study of generating the process repository which can also be used as the search tree for selection of the exception handling process. Section 5 discusses the conclusion and issues about the future research.

2 THE THEORETICAL BACKGROUNDS

[8] classified the failures and exceptions of business process management system into four types (as shown in Table 1). Based on their work, [12] provided the exceptional situations which can take place at various stages of the business process and presented the corresponding solution framework (Table 2).

Type	Stage	Instances
Unexpected exception	Process execution stage	The predefined process model is unable to handle the exception. Ex) Change the priority of a VIP customer upon his request
Expected exception	Process definition stage	Part of the process cannot be applied. Ex) Customer failed to pay the fee/ Failed to reserve an airline ticket because it was already booked.
Application failure	Application stage	Program failure/ Constraint violation
Basic failure	System stage	System break down, deadlock, network connection failure, printer break down

Table 1. Type of failure and exception

As shown in Tables 1 and 2, exception handling in BPMS can be divided into two types: handling of expected and unexpected exceptions. When the exception is unexpected, it may be resolved by inserting or deleting specific task unit(s) into the process model at the execution stage. In this case the workers are allowed to change the work flow schema dynamically [8] or some sort of exception handling tools are provisioned such that the workers may handle the exceptions for themselves [9].

The main stream approach to handling the expected exceptions is to store the matching solution (sub-process) in the process repository. It is also possible to include the expected exception handling process as a sub-process of the normal process diagram. However, in this case, it is likely to increase the complexity and reduce the legibility of the process diagram [15].

[11] proposed a binary search tree in retrieving the exception handling process. [16] provided a hierarchical structure for storage of various exception handling processes. In this paper we introduce the data structure which is composed of the situation variable S and the decision variable D in order to enable the systematic

Process stage	Exceptions	Solution	Remarks
Strategic	Unexpected exceptions (employee, team organization)	Human intervention	
Tactical	Expected exceptions (Workflow, data, temporary or exogenous problem	Model the workflow adaptive to the situation	Seek solution by shifting it to the strategic stage
Operational	Basic failure, Application failure	Traditional TPS	Shift the problem to tactical stage

Table 2. Exceptions at various process stages and matching solutions

description of the exceptional problem context, and to facilitate the understanding, classification, and retrieval of the exceptional situation and the matching exception handling process. We deploy an e-commerce delivery process as the case example to demonstrate the usability of the data structure and how it can be used in generating the search tree structure which can be applied to the efficient management of the exception handling process repository.

[16] and [11] reported that most of the preceding process retrieval system architecture design was based on the subjective opinion of the domain expert. In this paper we propose to use the inductive approach in designing the process retrieval system architecture. More specifically, we propose to use the induction-based decision tree structure which can be generated by applying the ID3-based C5.0 algorithm. The advantage of using the induction-based decision tree structure is that it provides the logical reasoning regarding the quest why the induced decision tree is the best structure for the storage and retrieval of the exception handling processes. The advantages of using the decision-tree structure over other existing process retrieval system structure in exception handling are summarized as follows.

1. The decision tree is organized so as to maximize the information gain. Therefore, it guarantees the optimal behavior in the storage and retrieval of the exception handling processes (Han, 2004).
2. The decision tree can be updated anytime as there are more exception handling processes added to the process repository, while in [16], they have to convene and hold a domain expert panel meeting in order to update the classification hierarchy structure. The worse part of their scheme is that there is no guarantee of optimality even after the structure was updated.
3. The context description of the exceptional situation in terms of the situation variable (S) and decision variable (D) reflects the implicit knowledge structure of

the domain experts when they make the decision contingent upon the exceptional situation.

4. The process storage and retrieval scheme based on the situation variable (S) and decision variable (D) enables the efficient identification and recognition of the exceptional situation. It also enables the efficient retrieval of the exception handling process that could best resolve the problem.

Methods	Process repository structure	Structure generation
Adams et al. [11]	Binary tree	Expert panel (subjective)
Klein and Dellarocas [16]	Hierarchy tree	Expert panel (subjective)
The proposed method	S & D variable structure for context description and decision tree structure	Expert panel, Tree induction (subjective and objective)

Table 3. Comparison of the handling methods for expected exceptions

Table 3 is the comparison of the methods used by [11, 16], and this paper.

3 SAMPLE EXCEPTIONS IN E-COMMERCE DELIVERY PROCESS

The following are samples of extraordinary exceptions that might happen in the process of e-commerce delivery. The data are excerpted from the case book of the e-trade dispute arbitration published by the Korean Institute for Electronic Commerce [17].

- The item was delivered to a third party (not an agent) and then got lost.
- An item that exceeds the standard size was accepted for delivery since there was some extra space in the delivery vehicle and the competitor was also accepting such non-standard items under similar conditions. In this case the operator must identify the availability of extra space or extra vehicle and has to follow the complicated procedure to justify the exception handling.
- A buyer ordered an item from an Internet shopping mall and completed the payment process. He received a mail from the seller confirming the order information and notifying that the item was shipped out. However, the item was returned to the seller because of incorrect address. In addition, the buyer was charged for the return shipping.
- An expensive item was deposited for repair at a service center. When the item was shipped back to the owner, he found it damaged due to faulty packaging. So he asked for exchange or compensation. However, the service center refused the claim because the item was already a second handed one and they had no regulation for such a case.

- The seller delayed shipping many times and eventually cancelled the contract because they were unable to procure the inventory. The buyer experienced a big loss due to this contract failure and filed a claim for compensation. A complicated dispute arbitration process is anticipated to resolve this case.
- A perishable item was ordered. However, the package was broken in the delivery process and some other items located adjacent to the package got tainted. The seller asserted that he had made a tight packaging. In this case, the dispute arbitration process must clarify where the responsibility lies. Decisions regarding return, refund, and compensation have to be delineated.

Handling this sort of extraordinary exceptions require the involvement of problem domain specialists or need to go through a series of problem specific decision making processes. The hard part of the task is that it is not easy to automate the entire task and is mostly processed manually. In this paper, we attempt to find an effective methodology for storage and retrieval of the exception handling processes assuming that a sufficient amount of exception handling process data are accumulated over a long period of time.

4 SEARCH TREE GENERATION

In this section we present the process repository architecture for exception handling and also provide a case example for generating the decision tree for storage and retrieval of the exception handling processes. We use the sample exception data excerpted from the KIEC case book (see Section 3) in generation of the decision tree.

4.1 Architecture for Process Repository

In order to handle the exceptions the process must go through three stages, i.e., to identify the exception, retrieve the matching exception handling process, and then resolve the exception [18]. An exception can be identified by monitoring the current status of an ongoing process. At this stage every instance of the ongoing process is checked against exceptionality, and when an exception is perceived, the type of exception is identified. Retrieval of the matching exception handling process is done by looking up the process repository and finding the best fit process for the exception.

At this stage the situation variable S and decision variable D play the key role. If a good exception handling process could be found, then we simply need to apply it for the exception resolution. If an appropriate resolution could not be found, the exception must be resolved by using one of the approaches shown in Tables 2 and 3. When it is resolved, the new exception handling process should be added to the process repository. Figure 1 shows the architecture of the exception handling process repository system.

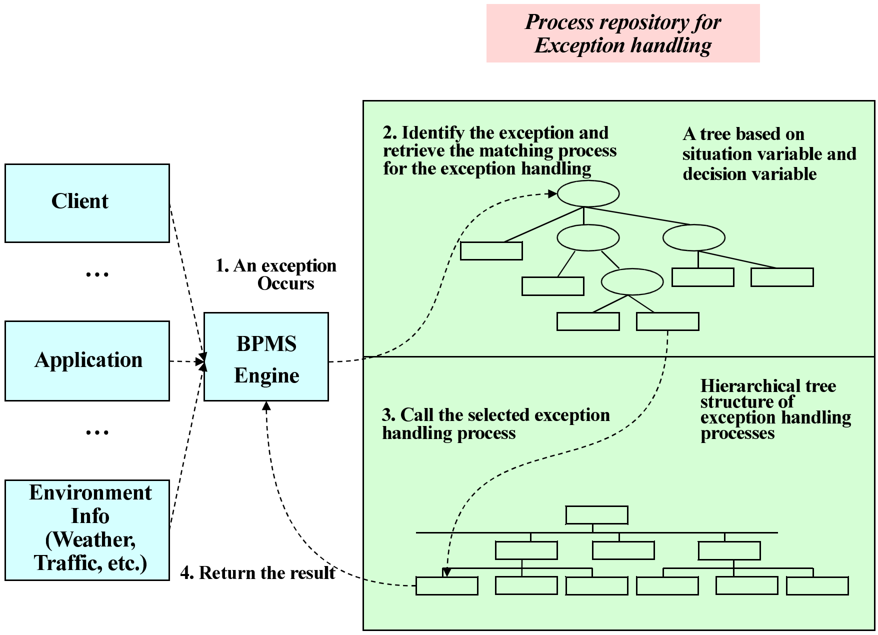

Fig. 1. Architecture for exception handling process repository

4.2 Variable Definition and Data Preparation

The following example explains the process of generating the search tree which can be utilized for selecting the matching process for handling the identified exception. In this paper, the C5.0 algorithm of SPSS Clementine package was used to obtain the induction based decision tree [19]. The overall steps of generating the search tree are as follows.

- Define the situation variable S and decision variable D.
- Collect the case examples of the exception handling process.
- Prepare the input data according to the C5.0 input format.
- Generate the search tree using C5.0.

Table 4 is the sample definition of the situation variables and decision variables that can be used to describe the various exceptions that may occur in the e-commerce delivery process. [7] previous work was referenced in the variable definition. They grouped the delivery exceptions into two categories: 'customer originated' and 'system originated' exceptions. Some modifications have been made to fit the sample example situation. Tables 5 and 6 are the modified versions of the situation variable and decision variable with reference to the variable definition of Table 4.

Origin of exception	Situation variable	Decision variable
Customer	* Order change - Type of order change - Delivery status	* Order change processing - item (name, price, shipping charge) - destination (location, recipient) - delivery date (delayed, expedited) - Shipper - Delivery channel (door to door, seller delivery, special delivery) - Return charge payer (customer, seller) - Urgency (emergency, normal)
	* Cancel order	* Cancellation (allowed/disallowed)
System factor (failure/break down)	* Delayed delivery - Traffic - condition turned worse	* Traffic jam - availability of nearby alternative carrier
	* Traffic accident - type of accident	* Degree of car damage
	* Problem in production stage - abnormal production - abnormal quality	* Alternative supplier - agreement with the original supplier
	* Carrier problem - car break down	* Car damage level - car operational - availability of nearby carrier
	* Problem with the shipped item - damage in item	* Compensation level
	* IT system problem - central control system disorder - transportation system disorder - wireless communication disorder (cell phone, PDA)	* Alternative means of communication - back up server system - public phone
	* Routing problem - Natural disaster (earthquake, typhoon)	* Alternate delivery - adjust schedule - business partner
	* Shipping cost increased - environment change (oil price up, consumer price up)	* change shipping charge
	* Production stopped (fire, power failure)	* Alternate supplier

Table 4. Situation and decision variables

Variable(s_i)	Value
Order change type	item, shipping destination, delivery time, delivery medium
Delivery status	before shipping, In delivery, Delivered, In return(w/RMA), In exchange delivery
Delivery type	normal, bundle, return, exchange, re-exchange, exchanged and cancelled
Priority (schedule)	normal, expedited, special, designated date, delayed
Destination	incomplete address, address changed, moved during delivery
Payment	prepaid, deposit payment & balance payment, deferred pay, escrow
Recipient	buyer, agent, third party, agent of absence, P.O. box
Item description	item name, price, shipping charge, quantity
Condition	new, used, damaged, defective, broken in use, special handling(fragile, perishable, indemnity of damage in delivery, frozen)
Standard	volume, weight, special care
Received	yes, no
Empty vehicle	available number, load factor
Delivery	type door to door, seller delivery, registered mail, regular mail
Type of trade	e-shopping mall, specialty e-store, open market, auction, direct trade

Table 5. An examples of situation variables

Most BPM systems have the facility to monitor the system behaviour and store the log data of the business activities. When substantial amount of log data is collected, the BPMS renders the analysis of the workflow status and analysis of the system performance record. In this regard the log data is a good source of case examples which contain much information about situation variable and decision variable. This observation justifies the fact that constructing a process repository system from system log data is a viable approach.

Figure 2 shows part of the 145 dataset used in generating the process search tree. Since this is an advanced research, no real field data is available as of this paper writing.

The sample data shown in Figure 2 are compilation of the sample data in the KIEC case book of e-trade dispute arbitration [17]. As a summary of the variable definition, we had 8 situation variables, 9 decision variables, and one output variable which is equivalent to the matching exception handling process.

Variable (d_j)	Value
Delivery charge	Seller pay, buyer pay, special contract
Return shipping charge	Seller pay, buyer pay, logistics co., undecided
Deliverer	Current deliverer, new deliverer, substitute
Item opened & used	yes, no
Cause of return	Simple change of mind, wrong item, delayed delivery, not specified, item damaged, wrong price
Returned item condition	Item damaged, good, package damaged
Stipulated in the agreement	clear, unclear
Sufficient info provided	yes, no
Buyer's fault	verified, not verified
Buyer verified defective	verified, not verified
Exception handling process	Change the order item, Adjust delivery priority(schedule), Change destination, Cancel order, Change delivery carrier, Cancel order and refund, Partial refund, Return and exchange ship, Compensation for buyer, Compensation for seller, Hold the contract

Table 6. An example of decision variables

4.3 The Search Tree Generation

The search tree we obtained from C5.0 algorithm with the dataset shown in Figure 2 is provided in Figure 3. The leaf nodes in Figure 3 indicate the processes which fit the given exceptional situation best. The tree diagram in Figure 3 indicates that the situation variables used in the storage and retrieval of the exception handling processes are 'delivery status', 'returned item condition' and 'type of trade'. The decision variables are 'return shipping charge payer', 'cause of return', 'item type' and 'stipulated in the agreement'. The diagram also tells us that the most influential (effective) variable is 'delivery status' (located at the root).

5 CONCLUSION

In the earlier research, the process repository architecture design for exception handling was mostly done by the subjective judgement of the domain experts. This paper presented an alternative approach which utilized the C5.0 algorithm to obtain the decision tree structure that provided the optimal path to store and retrieve

	A	B	C	D	E	F	G	H	I	J	K	L
1	Item type	Delivery status	Delivery charg	Return charge	Item opene	Stipulated i	Proved buy	Buyer prov	Cause of return	Condition (	Type of trade	exception handling process
97	new	received							incorrect item		shopping mall	exchange shipping
98	new	received				unknown			incorrect item		shopping mall	exchange shipping
99	new	received		both pay					change of mind		shopping mall	partially refunded
100	made to order	received							change of mind		shopping mall	partially refunded
101	new	received							incorrect item		auction	partially refunded
102	new	received							damaged item		auction	Purchase cancelled and refunded
103	new	received							incorrect item		direct trade	Purchase cancelled and refunded
104	new	received		buyer pays					damaged item		shopping mall	Purchase cancelled and refunded
105	new	received							change of mind		shopping mall	Purchase cancelled and refunded
106	new	received				yes			change of mind		shopping mall	Purchase cancelled and refunded
107	new	received				yes			change of mind		shopping mall	Buyer got refunded or compensated for damage
108	new	received							damaged item		direct trade	exchange shipping
109	new	received							incorrect item		shopping mall	Purchase cancelled and refunded
110	new	received					no		missing part		auction	Buyer got refunded or compensated for damage
111	new	received					no		missing part		auction	Buyer got refunded or compensated for damage
112	used	received		seller pays		yes			damaged item		auction	Purchase cancelled and refunded
113	used	received				yes			damaged item		auction	Purchase cancelled and refunded
114	used	received							change of mind		auction	Buyer got refunded or compensated for damage
115	new	received					yes				auction	Buyer got refunded or compensated for damage
116	new	received		buyer pays			yes		incorrect item		auction	Purchase cancelled and refunded
117	new	received		buyer pays			yes		incorrect item		auction	Buyer got refunded or compensated for damage
118	new	not received							wrong price		shopping mall	Buyer got refunded or compensated for damage
119	new	not received							wrong price		shopping mall	Buyer got refunded or compensated for damage
120	new	not received							wrong price		shopping mall	Purchase cancelled and refunded
121	new	not received							wrong price		shopping mall	Buyer got refunded or compensated for damage
122	new	not received							wrong price		shopping mall	Purchase cancelled and refunded
123	new	not received							wrong price		shopping mall	Purchase cancelled and refunded
124	new	not received							wrong price		shopping mall	Buyer got refunded or compensated for damage
125	new	not received							wrong price		shopping mall	Buyer got refunded or compensated for damage
126	new	received							incorrect item		auction	Purchase cancelled and refunded
127	new	not received							wrong price		shopping mall	Purchase cancelled and refunded
128	new	not received							wrong price		shopping mall	Buyer got refunded or compensated for damage
129	new	not received							delayed delivery		shopping mall	Buyer got refunded or compensated for damage
130	new	received				yes					auction	Buyer got refunded or compensated for damage
131	new	received				yes					auction	Purchase cancelled and refunded
132	new	received					no		change of mind		auction	Purchase cancelled and refunded
133	new	received							damaged item		shopping mall	exchange shipping
134	new	received							change of mind		auction	partially refunded
135	new	received					no	no	damaged item		auction	Purchase cancelled and refunded
136	new	received					no	no	damaged item		auction	Buyer got refunded or compensated for damage
137	new	received				yes			change of mind		auction	Contract hold
138	new	received				yes			damaged item		auction	Purchase cancelled and refunded
139	used	received		both pay			yes		incorrect item		auction	Purchase cancelled and refunded
140	new	received							damaged item		auction	Buyer got refunded or compensated for damage
141	new	received							damaged item		auction	Purchase cancelled and refunded
142	new	received							change of mind		auction	Purchase cancelled and refunded
143	new	received							change of mind		auction	Buyer got refunded or compensated for damage
144	used	received					no		damaged item		auction	Purchase cancelled and refunded
145	new	received					no		change of mind		auction	Purchase cancelled and refunded

Fig. 2. Sample usage of the situation and decision variable used in generating the decision tree

the exception handling processes. The use of 'situation variable' and 'decision variable' structure for the context description of the exceptional problem is an efficient way to identify the problem context and to find the best fit exception handling process. Since the search tree is constructed based on the ID-3 algorithm, each step of the tree traversal from the root down to the leaf node was adapted so as to maximize the information gain.

As more exception handling processes are added to the repository, then we need to update the search tree. As long as we keep describing the exceptions in terms of the situation variable and decision variable, updating the search tree for renewal of the optimality will be a handy task since we simply have to run the C5.0 algorithm with the updated dataset. Since we can anticipate a substantial change in the search tree organization every time we update the search tree, we have to provide the facility to accommodate the tree structure change into the database implementation. And this should be the subject for future study.

Acknowledgements

This work was supported by the research grant of the Jeju National University in 2006.

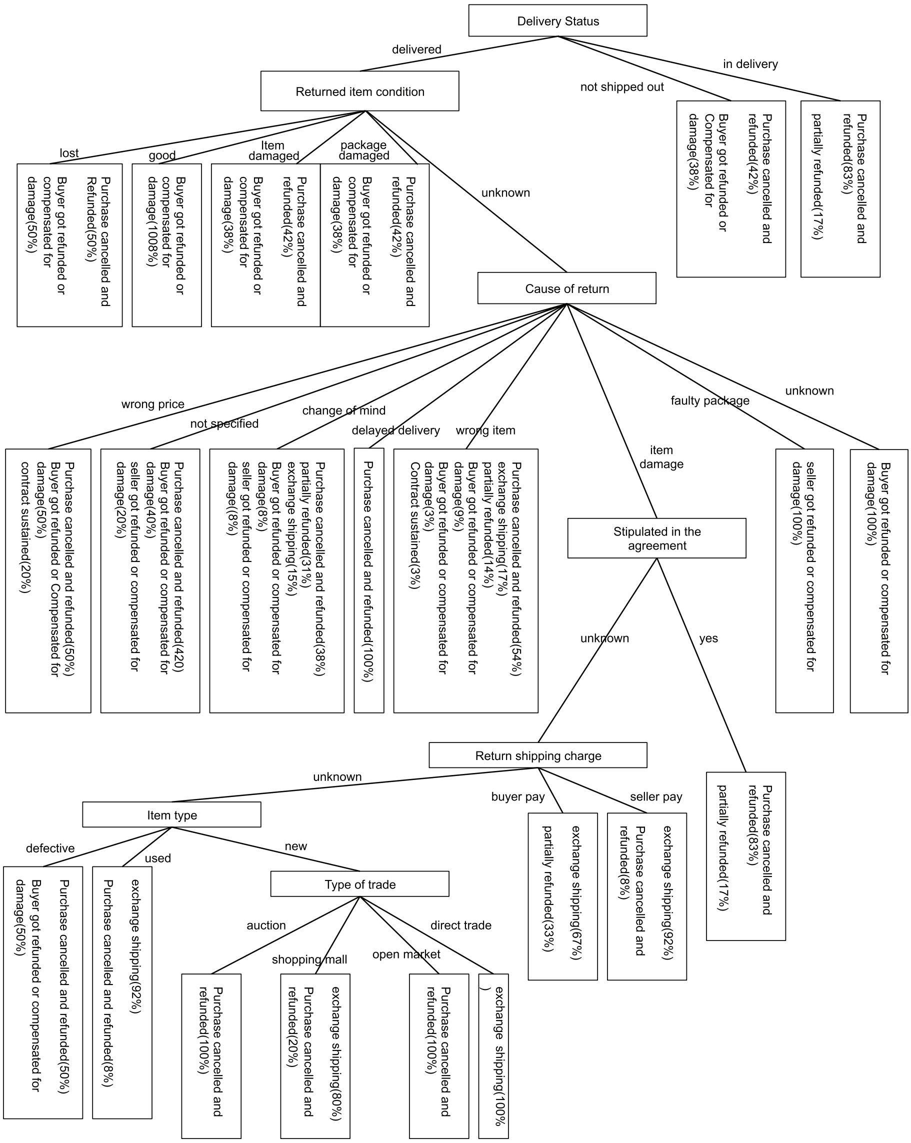

Fig. 3. Decision tree for storage and retrieval of the exception handling processes in e-commerce delivery

REFERENCES

[1] http://www.wfmc.org.

[2] Park, J. H.: Process Innovation and BPM. IE Magazine, Korea Institute of Industrial Engineering, Vol. 11, 2004, No. 1, pp. 19–24.

[3] Weske, M.—van der Aalst, W. M. P.—Verbreek, H. M. W.: Advances in Business Process Management. Data and Knowledge Engineering, Vol. 50, 2004, pp. 1–8.

[4] Simon, H. A.: The New Science of Management Decision. Harper and Row, N. Y. 1960.

[5] Hermann, T.—Hofmann, M.—Loser, K. U.—Moysich, K.: Semistructured Models Are Surprisingly Useful for User-Centered Design. In G. De Michelis, A. Giboin, L. Karsenty, R. Dieng: Design cooperative systems, IOS Press, Amsterdam 2000, pp. 159–174.

[6] Banerjee, S.—Basu, A.: Model Type Selection in an Integrated DSS Environment. Decision Support Systems, 1993, No. 9, pp. 75–89.

[7] Christopher, M.—Lee, H. L.: Supply Chain Confidence: The Key to Effective Supply Chains Through Visibility and Reliability. Stanford Global Supply Chain Management Forum 2002.

[8] Eder, J.—Liebhart, W.: The Workflow Activity Model WAMO. Proceedings of the 3rd International Conference on Cooperative Information Systems 1995.

[9] Kappel, G.—Lang, P.—Rausch-Schott, S.—Trtschitzegger, W.: Workflow Management Based on Object, Rules and Roles. Bulletin of Technical Committee on Data Engineering, Vol. 18, 1995, pp. 11–19.

[10] Lee, H. B.—Park, S. J.: Intelligent Workflow Automation System Flexible to Organization Change: WFMS. Management Information System Research, Korea Operations Research and Management Science Society, Vol. 11, 2001, No. 3, pp.150–164.

[11] Adams, M.—ter Hofstede, A. A. H. M.—David, E.—van der Aalst, W. M. P.: Facilitating Flexibility and Dynamic Exception Handling in Workflows Through Worklets. In The 17th Conference on Advanced Information Systems Engineering Forum 2005.

[12] Mourão, H. R.—Antunes, P.: Supporting Direct User Interventions in Exception Handling in Workflow Management Systems. 9th CRIWG 2003, Springer Verlag 2003, pp. 159–167.

[13] Keen, P.—McDonald, M.: eProcess Edge. McGraw Hill 2000.

[14] Gaonkar, R.—Viswanadham, N.: Robust Supply Chain Design: A Strategic Approach for Exception Handling. International Conference on Robotics and Automation 2003, pp. 1762–1767.

[15] Müller, R.—Greiner, U.—Rahm, E.: AgentWork: A Wirjfkiw System Supporting Rule-Based Workflow Adaptation. Data and Knowlwdge Engineering, Vol. 51, 2004, pp. 223–256.

[16] Klein, M.—Dellarocas, C.: Knowledge-Based Approach to Handling Exceptions in Workflow Systems. The Journal of Computer Supported Cooperative Work, Vol. 9, 2000, No. 3-4, pp. 399–412.

[17] http://www.kiec.or.kr.

[18] Vojevodina, D.: Exception Handling Automation in E-Business Workflow Process. Proceedings of Conference on Advanced Information System Engineering, 2005.

[19] Han, J.—Kamberr, M.: Data mixing: Concepts and techniques. 2^{nd} ed., Morgan Kaufmann Publishers 2006.

Jin-Gyu Shin received his Ph. D., M. Sc. and B. Sc. from Sung-KyunKwan University, Korea in 2010, 1997 and 1995, respectively. He has been working as a researcher in University College of the University since 2007. His research interests include business process management, data mining and evaluation.

Doug-Won Choi is a Professor in the Department of System Management Engineering, Sungkyunkwan University, Korea. He received Ph. D. from Temple University, Philadelphia, U.S.A. in 1995 with major in Computer and Information Sciences, M. Sc. degree in Industrial Engineering in 1985 from University of Wisconsin, Madison, U.S.A., and B. Sc. from Seoul National University, Korea in 1973. Before joining Sungkyunkwan University he worked at Korea Institute of Science and Technology as an investigator. His research interests include management information system, data mining, business process management, artificial intelligence, and expert systems.

Dong-Cheol Lee received his Ph. D. from Sungkyunkwan University, Korea in 2002 and 1998, M. Sc. from Kookmin University in 1992, and B. Sc. from Chungnam National University in 1986. He was with Jeju National University as an Assistant Professor in 2003. Now, he is an Associate Professor of the MIS Department at the University. His research interests include e-commerce, Aaents, XML and semantic web, and digital contents.

Yung-Cheol Byun received his Ph. D. and M. Sc. from Yonsei University, Korea in 1995 and 2001, respectively, and B. Sc. from Jeju National University in 1993. He worked as a special lecturer in SAMSUNG Electronics in 2000 and 2001. From 2001 to 2003, he was a senior researcher of Electronics and Telecommunications Research Institute (ETRI). He was promoted to join Jeju National University as an Assistant Professor in 2003. Now, he is an Associate Professor of Computer Engineering Department at the University. His research interests include intelligent computing, semantic web and ontology, home network and ubiquitous computing, RFID middleware and USN, artificial intelligence, and pattern recognition.