SEARCH TREE GENERATION FOR BUSINESS PROCESS REPOSITORY MANAGEMENT IN THE EXCEPTION HANDLING OF THE e-COMMERCE DELIVERY PROCESS

Jin Gyu Shin and Doug Won Choi
Systems Management Engineering, Sungkyunkwan University
300 Cheoncheon-dong, Jangan-gu, Suwon, Gyeonggi-do 440-746, Korea

Keywords: Exception handling, Process repository management, Systematic context description, Situation variable, Decision variable, Search tree generation.

Abstract: BPMS (business process management system) 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.

1 INTRODUCTION

Manufacturing processes usually do not involve so many exceptional situations. And this property explains part of the reason why we can automate the control and management of manufacturing processes. However, the recent progresses 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 (http://www.wfmc.org). Many organizations have implemented this system and are reported to be reaping good results (Park, 2004).

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 (Weske et al., 2004). An exceptional situation implies a task which occurs infrequently 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 Simon (1960) and Hermann et. al. (2000). 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 decision support system research (Banerjee and Basu, 1993). 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, … , s_m) and decision variable set D= (d_1, d_2, … , d_n), to enable the systematic context description of the 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

Gyu Shin J. and Won Choi D. (2009).
SEARCH TREE GENERATION FOR BUSINESS PROCESS REPOSITORY MANAGEMENT IN THE EXCEPTION HANDLING OF THE e-COMMERCE DELIVERY PROCESS.
In *Proceedings of the International Conference on e-Business*, pages 121-128
DOI: 10.5220/0002237401210128

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 related literature review about exception problems and the corresponding resolution approaches (Christopher and Lee, 2002; Eder and Liebhart, 1995; Kappel, et al., 1995; Lee and Park, 2001; Adams, et al., 2005; Mourão and Antunes, 2003; Keen and Mcdonald, 2000; Gaonkar and Viswanadham, 2003) 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 the issues about future research.

2 THE THEORETICAL BACKGROUNDS

Eder and Liebhart (1995) classified the failures and exceptions of business process management system into four types (Table 1). Based on their work, Mourão and Antunes (2003) provided the exceptional situations which can take place at various stages of the business process and presented the corresponding solution framework (Table 2).

As is shown in tables 1 and 2, exception handling in BPMS can be divided into two types: handling of expected exceptions 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 (Eder and Liebhart, 1995) or some sort of exception handling tools are provisioned such that the workers may handle the exceptions for themselves (Kappel, et al., 1995).

Table 1: Type of failure and exception.

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 2: Exceptions at various process stages and matching solutions.

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

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 (Müller, et al., 2004).

Adams et. al. (2005) proposed a binary search tree in retrieving the exception handling process. Klein and Dellarocas (2000) 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 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.

Klein and Dellarocas (2000) and Adams et. al. (2005) 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 algorithm C5.0. 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 Klein and Dellarocas (2000), 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.

Table 3 is the comparison of the process search methods used by Adams et al.(2005), Klein and Dellarocas(2000), and this paper .

Table 3: Comparison of the handling methods for expected exceptions.

Methods	Process repository structure	Structure generation
Adams et. al. (2005)	Binary tree	Expert panel (subjective)
Klein and Dellarocas (2000)	Hierarchy tree	
This paper	**S** & **D** variable structure for context description and decision tree structure	Expert panel, Tree induction (subjective and objective)

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 (http://www.kiec.or.kr).

- 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 the 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 the 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 these sort of extraordinary exceptions requires the involvement of the problem domain specialists or needs 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 the 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 the 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., identify the exception, retrieve the matching exception handling process, and then resolve the exception (Vojevodin, 2005). 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 one 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.

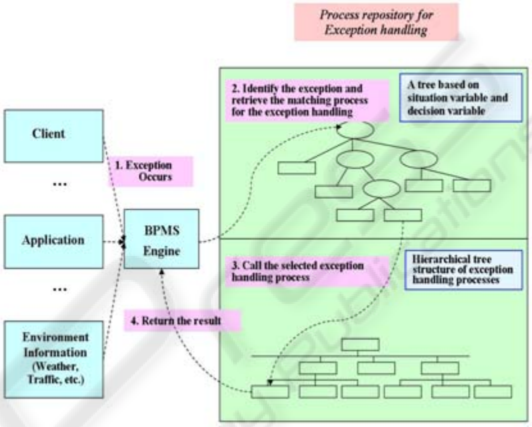

Process repository for Exception handling
Client
...
Application
...
Environment Information (Weather, Traffic, etc.)
1. Exception Occurs
BPMS Engine
4. Return the result
2. Identify the exception and retrieve the matching process for the exception handling
A tree based on situation variable and decision variable
3. Call the selected exception handling process
Hierarchical tree structure of exception handling processes

Figure 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 (Han, 2004). 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 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. Christopher and Lee's (2002) previous work was referenced in the variable definition. They grouped the delivery exceptions into two categories: 'customer originated' and 'system originated' exceptions. Some modifications

Table 4: Situation variable and decision variable.

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

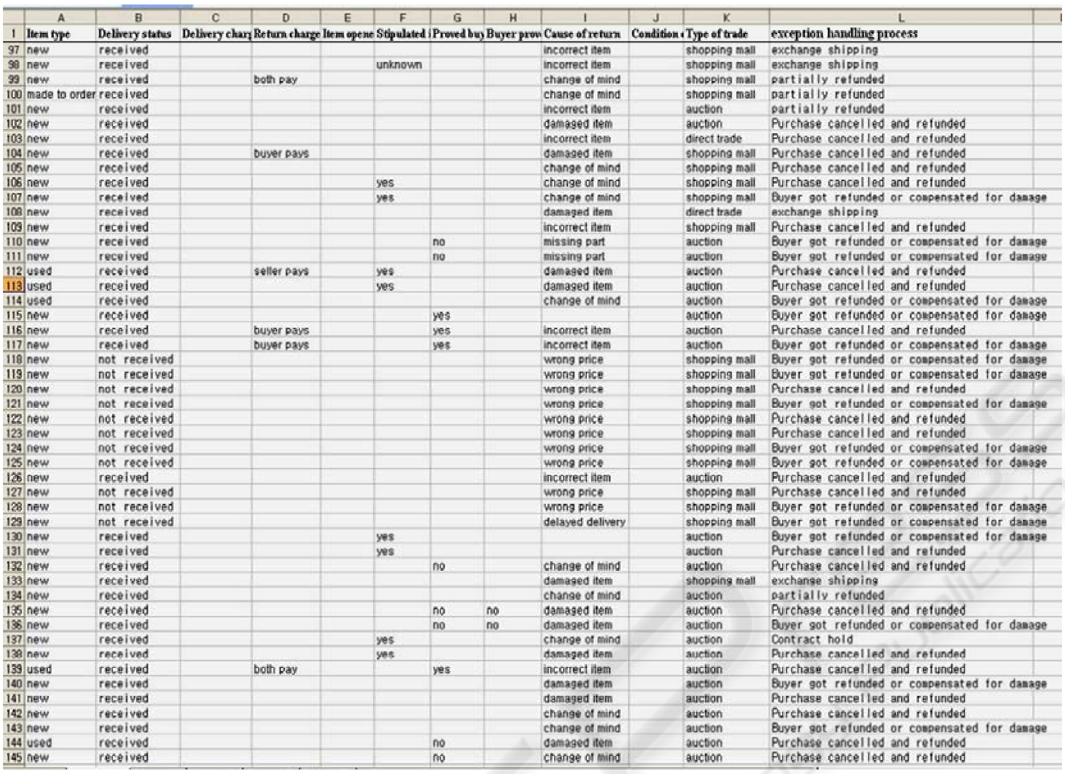
have been made to fit the sample example situation. Table 5 and Table 6 are the modified version of the situation variable and decision variable with reference to the variable definition of Table 4.

Most BPM systems have the facility to monitor the system behavior and store the log data of theiness activities. When substantial amount of log data are 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 lots of 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 data set used in generating the process search tree. Since this is an advanced research, no real field data is available as of this paper writing.

Table 5: Situation variable – example.

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



	A	B	C	D	E	F	G	H	I	J	K	L
1	Item type	Delivery status	Delivery char	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

Figure 2: Sample usage of the situation variable and decision variable used in generating.

Table 6: Decision variable – example.

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

The sample data shown in Figure 2 are compilation of the sample data in the KIEC case book of e-trade dispute arbitration (http://www.kiec.or.kr).

As the summary of the variable definition, we had 8 situation variables, 9 decision variables, and one out variable which is equivalent to the matching exception handling process.

4.3 The Search Tree Generation

The search tree we obtained from C5.0 algorithm with the data set shown in Figure 2 is provided in Figure 3. The leaf nodes in Figure 3 indicate the processes which best fit the given exceptional situation. 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.' And 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)

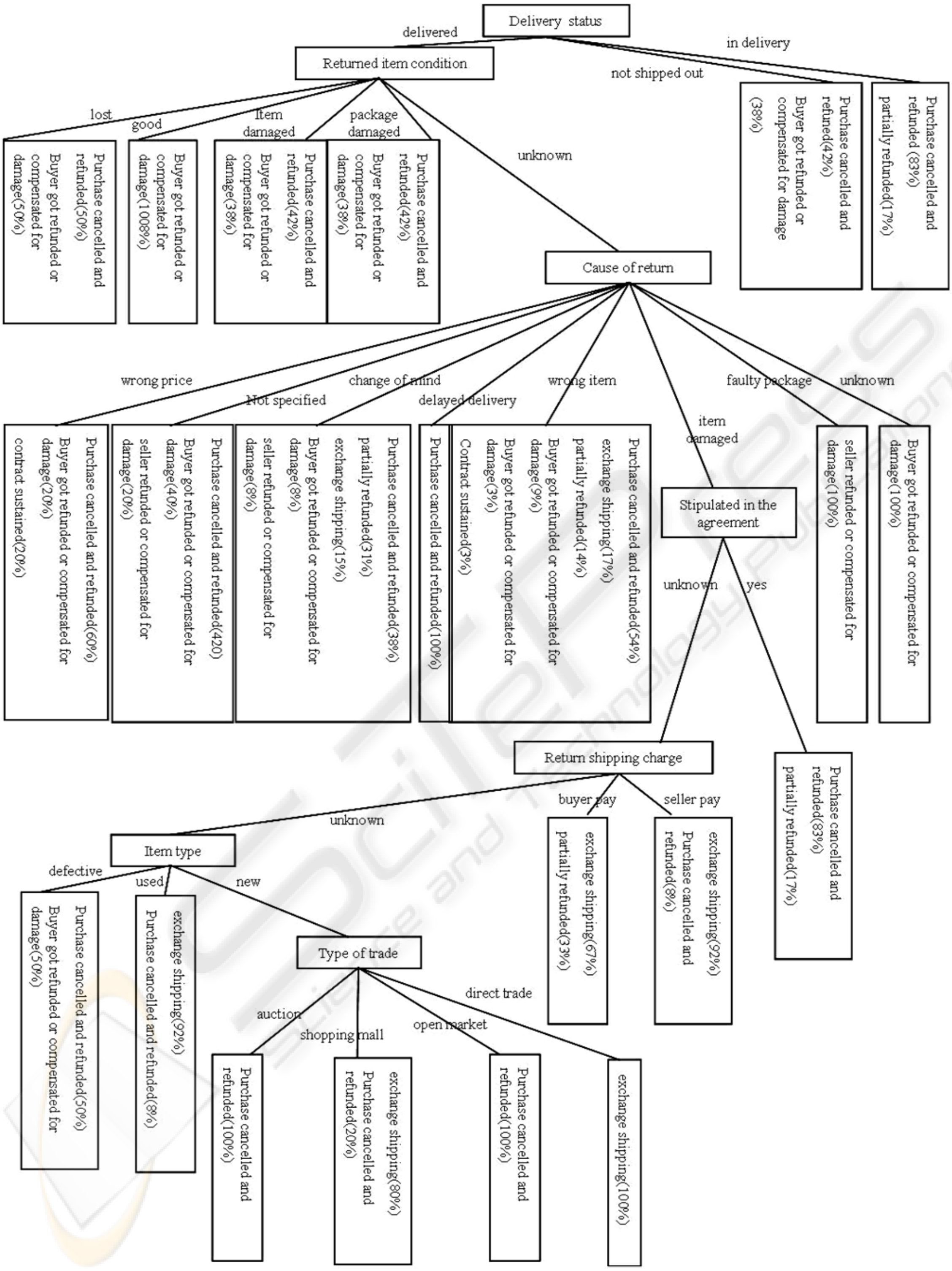

Delivery status
delivered
in delivery
not shipped out
Returned item condition
lost
good
Item damaged
package damaged
unknown
Purchase cancelled and refunded(50%) Buyer got refunded or compensated for damage(50%)
Buyer got refunded or compensated for damage(1008%)
Purchase cancelled and refunded(42%) Buyer got refunded or compensated for damage(38%)
Purchase cancelled and refunded(42%) Buyer got refunded or compensated for damage(38%)
Purchase cancelled and refuned(42%) Buyer got refunded or compensated for damage (38%)
Purchase cancelled and refunded (83%) partially refunded(17%)
Cause of return
wrong price
change of mind
Not specified
delayed delivery
wrong item
faulty package
unknown
item damaged
Purchase cancelled and refunded(60%) Buyer got refunded or compensated for damage(20%) contract sustained(20%)
Purchase cancelled and refunded(420) Buyer got refunded or compensated for damage(40%) seller refunded or compensated for damage(20%)
Purchase cancelled and refunded(38%) partially refunded(31%) exchange shipping(15%) Buyer got refunded or compensated for damage(8%) seller refunded or compensated for damage(8%)
Purchase cancelled and refunded(100%)
Purchase cancelled and refunded(54%) exchange shipping(17%) partially refunded(14%) Buyer got refunded or compensated for damage(9%) Buyer got refunded or compensated for damage(3%) Contract sustained(3%)
seller refunded or compensated for damage(100%)
Buyer got refunded or compensated for damage(100%)
Stipulated in the agreement
unknown
yes
Purchase cancelled and refunded(83%) partially refunded(17%)
Return shipping charge
buyer pay
seller pay
unknown
exchange shipping(67%) partially refunded(33%)
exchange shipping(92%) Purchase cancelled and refunded(8%)
Item type
defective
used
new
Purchase cancelled and refunded(50%) Buyer got refunded or compensated for damage(50%)
exchange shipping(92%) Purchase cancelled and refunded(8%)
Type of trade
auction
shopping mall
open market
direct trade
Purchase cancelled and refunded(100%)
exchange shipping(80%) Purchase cancelled and refunded(20%)
Purchase cancelled and refunded(100%)
exchange shipping(100%)

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

5 CONCLUSIONS

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 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 proceeded in such a fashion to maximize the information gain.

As more and more exception handling processes are added to the repository, 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 data set. 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 topic for future research.

REFERENCES

Adams, M., Arthur, A. H. M. ter Hofstede, E. David and W. M. P. van der Aalst, 2005, Facilitating Flexibility and Dynamic Exception Handling in Workflows through Worklets, *In The 17th Conference on Advanced Information Systems Engineering Forum (CAiSE05 Forum)*.

Armin, W., W. Oliver, M. J. Josef and C. H. Siu, 2003, Data Mining for ontology Building, *IEEE Intelligent Systems*.

Banerjee, S. and A. Basu, 1993, Model type selection in an integrated DSS environment, *Decision Support Systems, No. 9 (1993), 75~89.*

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

Eder, J. and W. Liebhart, 1995, The workflow activity model WAMO, *Proceedings of the 3rd International Conference on Cooperative Information Systems (CoopIs).*

Gaonkar, R., N. Viswanadham, 2003, Robust Supply Chain Design: A Strategic Approach for Exception Handling, *International Conference on Robotics & Automation, (2003), 1762~1767.*

Han, J., 2004, *Data mining: Concepts and techniques*, 2nd ed. Morgan Kaufmann Publishers.

Hermann, T., M. Hofmann, K. U. Loser and K. Moysich, 2000, Semistructured models are surprisingly useful for user-centered design, In G. De Michelis, Giboin, A., Karsenty, L., Dieng, R., *Design Cooperative Systems (Coop 2000),* IOS Press, Amsterdam, 159~174.

Kappel, G., P. Lang, S. Rausch-Schott and W. Retschitzegger, 1995, Workflow Management Based on Object, Rules and Roles, *Bulletin of the Technical Committee on Data Engineering, Vol.18, No.1(1995), 11~19*.

Klein, M. and C. Dellarocas, 2000, Knowledge-based Approach to Handling Exceptions in Workflow Systems, *The Journal of Computer Supported Cooperative Work, Vol.9, No.3-4(2000), 399~412.*

Keen, P. and M. Mcdonald, 2000, *eProcess Edge*, McGrowHill.

Lee, H. B. and S.J. Park, 2001, Intelligent Workflow Automation System Flexible to Organization Change: K-WFMS, Management Information System Research, *Korea Operations Research and Management Science Society, Vol.11, No.3(2001),150~164.*

Mourão, H. R. and P. Antunes, 2003, Supporting Direct User Interventions in Exception Handling in Workflow Management Systems, *9th CRIWG 2003, Springer-Verlag, France, 159~167.*

Müller, R., U. Greiner and E. Rahm, 2004, AgentWork: A workflow system supporting rule-based workflow adaptation, *Data & Knowledge Engineering, Vol.51, No.2 (2004), 223~256.*

Park, J. H., 2004, Process Innovation and BPM, *IE Magazine, Korea Institute of Industrial Engineering, Vol.11, No.1 (2004), 19~24.*

Simon, H.A, 1960, *The New Science of Management Decision*, NY. Harper & Row.

Vojevodina, D., 2005, Exception Handling Automation in E-business Workflow Process, *Proceeding of Conference on Advanced Information Systems Engineering.*

Weske, M., W.M.P. van der Aalst and H.M.W. Verbeek, 2004, Advances in Business Process Management, *Data & Knowledge Engineering, Vol.50 (2004), 1~8.*

http://www.kiec.or.kr

http://www.wfmc.org