Simulation-based Evaluation of Workflow Escalation Strategies

Ka-Leong Chan[1], Yain-Whar Si[1], Marlon Dumas[2]
[1]*Faculty of Science and Technology, University of Macau*
[2]*Institute of Computer Science, University of Tartu, Estonia*
{m36658,fstasp}@umac.mo, marlon.dumas@ut.ee

Abstract

Workflows in the service industry sometimes need to deal with multi-fold increases in customer demand within a short period of time. Such spikes in service demand may be caused by a variety of events including promotional deals, launching of new products, major news or natural disasters. Escalation strategies can be incorporated into the design of a workflow so that it can cope with sudden spikes in the number of service requests while providing acceptable execution times. In this paper, we propose a method for evaluating escalation strategies using simulation technology. The effectiveness of the proposed method is demonstrated on a workflow from an insurance company.

1. Introduction

Business processes in the service industry are often exposed to sudden spikes in demand. For example, in the insurance industry, natural disasters may lead to sudden spikes in insurance claims. In order to cope with such demand spikes while maintaining response times consistent with their service level agreements, insurance companies need to be ready to rapidly adapt their operations, for example by incorporating escalation strategies into their workflows.

In the context of workflow management, an *escalation* is an action that is performed when a workflow execution is delayed to such an extent that it misses a deadline or it is not on track to meet its deadline [1]. For example, in the context of an insurance claim workflow, a possible escalation is to re-deploy staff from other departments into the call centers in order to handle incoming claims, or to require less information from callers when lodging insurance claims so that more claims can be recorded. The decision on whether or not to escalate, and how, needs to consider the cost of missing the deadline (e.g. lower customer services standards) and the cost of escalating (e.g. additional staff costs).

A number of escalation strategies have been proposed in the literature [1] [2] [3], each one with its own tradeoffs. However, the problem of which strategies to choose in a particular context has received little attention. In this paper, we show how simulation technology can be effectively used to evaluate escalation strategies previously proposed in [1, 2, 3].

The main contribution of the paper is a method for evaluating escalation strategies based on simulation technology. The method is applied to a case study from the insurance industry. We consider four escalation strategies and we show how to set up simulation experiments to evaluate these strategies. Based on the results of these experiments, we show the impact of various escalation strategies on a number of performance indicators. In addition, we demonstrate the importance of balanced decision making in selecting appropriate escalation strategies.

The paper is structured as follows. In Section 2, we discuss related work on workflow escalation and time management in workflows. In Section 3 we introduce the escalation strategies considered in this study. In Sections 4, 5 and 6 we describe the proposed simulation method and illustrate it with a case study. Finally, we draw conclusions in Section 7.

2. Background and related work

Paganos et al. [2, 3] have proposed “Dynamic Deadline Adjustment” and “Early Escalation” as complementary strategies to minimize the number of escalations needed during workflow execution and to mitigate their associated costs. Dynamic Deadline Adjustment aims at minimizing the number of escalations by attaching an expected time duration to each task, and by continuously monitoring each workflow execution in order to detect delays as soon as possible. Each task is assigned an expected execution time. When a task takes less than expected to complete, the difference between the expected and the actual execution time is accumulated in a *slack time* variable. When the slack time is negative it means that the workflow execution is delayed. This is where a second strategy, namely Early Escalation kicks in. In the Early Escalation strategy, an algorithm is used to predict whether a case is going to miss a deadline. When a potential deadline violation is detected, the case is escalated. Escalations are defined as actions executed in parallel to the normal activities of the workflow, in order to reduce the risk of a deadline violation, or to negotiate an extended deadline with the customer. Paganos et al. evaluated their strategies using simulation technology, but only from a temporal perspective (without considering resource costs).

van der Aalst et al. [1] analyze various deadline escalation strategies using a so-called *3D approach*, (Detect, Decide and Do). In the 3D approach, potential deadline violations are first detected. Then, suitable escalation strategies are selected and applied. In their work, escalation strategies are classified into three perspectives: the process perspective, the data perspective, and the resource perspective. van der Aalst et al. also evaluate the effectiveness of some sample escalation strategies using simulation experiments. The strategies considered in this study are alternative path selection (performing an alternative task when the execution is delayed), resource redeployment (bringing in more resources into the workflow execution) and data degradation (requiring less data input in order to move faster). Similar to Paganos et al. [2, 3], van der Aalst et al. [1] only evaluate escalation strategies from the time perspective. These limitations of the work reported in [1, 2, 3] are summarized in Table 1. In contrast, in this paper we propose a simulation-based method for evaluating escalation strategies that takes into account the cost of task execution, the cost of resources, and compensation costs. Also, our work covers both the Early Escalation strategy of [2, 3] and the 3D strategies of [1], which are complementary.

Other work in the field of workflow escalation includes that of Georgakopoulos et al. [5], who outline an approach to support dynamic changes in workflows in crisis situations (e.g. for rescue operations during natural disasters). Their focus is on enabling decision makers to escalate at runtime by changing the course of the workflow execution as required, while retaining some level of control. In contrast, our work focuses on analyzing the effectiveness and costs of different escalation strategies at design-time.

A related topic is that of specifying and analyzing time constraints in workflows. Eder et al. [6] propose PERT-like techniques for analyzing time constraints attached to workflows. Bettini et al [7] propose algorithms for checking time constraint satisfiability at design-time, while Chen et al. [8] propose techniques for efficiently checking time constraints at runtime. This body of work is complementary to ours since we do not deal with checking time constraints, but we focus on evaluating escalation strategies for dealing with sudden spikes in service demand.

Previous work also addresses the issue of determining the (minimum) amount of resources needed in a workflow in order to ensure that temporal constraints are met with a certain probability [9]. The reverse analysis is done in [10] where based on the available resources, estimates of average execution time per workflow instance are derived. This work is complementary to ours: the estimates obtained using such techniques can be used to implement escalation strategies based on resource redeployment.

	Strategy	**Workflow time**	**Escalation Cost**	**Cost (Task, Resource, Compensation)**
S1	Alternative Path Selection [1]	√	×	×
S2	Resource Redeployment [1]	√	×	×
S3	Data Degradation [1]	√	×	×
S4	Early Escalation [2,3]	√	√	×

Table 1. Escalation strategies analyzed in [1,2]

3. Escalation strategies

A *case* is an execution of a process model. For example, in an insurance claim process, a case is the set of activities performed in order to handle a given insurance claim. An escalation strategy describes actions to be taken when a case or a set of cases miss their expected completion time (deadline) or are predicted to miss the deadline. In this paper, we consider the following escalation strategies:

S1. Alternative Path Selection: In Alternative Path Selection, certain paths of the workflow are revised or skipped in order to reduce the workflow time.

S2. Resource Redeployment: Each task in a workflow is associated to a resource pool. A resource pool contains a number of resources (e.g. staff members) that may perform the task. In the Resource Redeployment strategy, the capacity of some resource pools is increased either by: a) adding more units (more staff); b) re-deploying units from one resource pool to another (e.g. staff from the sales department move to a call centre to handle incoming claims); or c) increasing the working time of resources in the pool.

S3. Data Degradation: Each task in the workflow model takes certain data as input and produces certain data as output. Data Degradation allows such tasks to be executed with less or different input data resulting reduction in time for gathering and transmission.

S4. Early Escalation: In the Early Escalation strategy, an algorithm is used to predict that a case will miss the deadline. If the algorithm predicts that the case will miss the deadline, a task is triggered. In the general case, this could be any task, like for example a task aimed at re-negotiating the deadline, or a task that sets an escalation variable to true, forcing the case to follow an “escalated path” (as in alternative path selection). The prediction algorithm (see Algorithm 1) is based on the one presented in [3]. However, the algorithm in [3] does not take into the tradeoff between escalating immediately and escalating later. The algorithm outlined here takes this into account.

```
EarlyEscalation(W,T):Boolean
     RETURN(NextStep(W,T,T));
END;

NextStep(W,T,TC): Boolean
  pred_completion(TC) = cur_queue(TC)*
                              avg_completion_overqueue (TC)

  IF TC = T AND deadline(TC)<pred_completion(TC)
       RETURN(“YES”);
  ENDIF
  IF TC ≠ T AND ((1-(escal_cost(T)/escal_cost(TC)))*
                     pred_completion(TC))>deadline(TC)
        RETURN(“YES”);
  ENDIF
  IF (the total number of tasks on the path from T to TC is less than n)
   AND
   ((T = TC) OR (escal_cost(TC)>escal_cost(T)))
      FOR each successor TC’ of TC in W
            IF NextStep(W, T, TC’)=”YES”
                RETURN(“YES”);
            ENDIF
         ENDFOR
   ENDIF
   RETURN(“NO”);
END;
```

Algorithm 1. Early Escalation – Prediction Algorithm

Symbols, Functions	Description
T_k	Task k in the workflow
$deadline(T_k)$	The deadline of task k
$cur\_queue(T_k)$	The total number of cases waiting to be processed by available resources
$pred\_completion(T_k)$	The predicted completion time of task k
$avg\_completion\_overqueue\ (T_k)$	The average completion time of task k per case when the total number of cases waiting to be processed exceed the number of available resources at task k
$escal\_cost(T_k)$	The escalation cost of task k
n	Maximum number of tasks to be tested after T_C

Table 2. Notations for Early Escalation Algorithm

The symbols and functions used in the algorithm are described in Table 2. The algorithm uses parameter n to limit the prediction scope, and it is invoked when a task T is ready for execution. The algorithm examines activities in depth-first order, and stops exploring the paths in the workflow schema when it finds an activity T_c that has lower escalation cost than T (i.e. $escal\_cost(T)>= escal\_cost(T_c)$). In this case, the algorithm will wait until T_c is ready for execution rather than escalation at task T.

When the algorithm is executed, activity T_c is initiated to T at the very beginning. If the activity T_c is predicted to miss the deadline (i.e. deadline of T_c is less than the predicted completion time of T_c ($deadline(T_c) < pred\_completion\ (T_c)$)), early escalation will be required at T_c. Otherwise, the algorithm uses a heuristics in which the predicted completion time of T_c (when $T_c \neq T$ in the algorithm) is multiplied by a factor and compared with its deadline. The factor approaches its maximum value when the escalation cost of T_c is greater than escalation cost of T. The factor approaches its minimum when the escalation cost of T is greater than the escalation cost of T_c. If the factor is less than 1, the predicted completion time is reduced proportionally. The algorithm returns “Yes” if the next task T_c is predicted to miss the deadline and applying escalation on T is less costly than doing so in T_c.

4. Simulation Method

In this section, we present the simulation method for evaluating escalation strategies. We illustrate the method by means of the insurance claiming process model introduced in [2]. The process model is depicted in Figure 1 in the form of an Event-Driven Process Chain (EPC). The model captures a process in a large Australian insurance company for handling inbound phone calls for lodging different types of insurance claims. Three sub-models are used in this process: the back office sub-process model, the Brisbane call center process and the Sydney call center model. There are two tasks are in each call center: “check if sufficient information is available” and “register claim”. Call center agents handle all tasks in each call center. Each case must be handled by a single resource. There are 90 call center agents assigned on each call center respectively. In normal conditions, approximately 9000 calls are received in each call center. Once the information gathering process in the call center has completed, a claim moves into the back-office process.

In the back-office there are There 150 claims handlers and five tasks: (1) determine likelihood of claim, (2) check total processing time of claim, (3) assess claim, (4) initiate payment, (5) advise claimant on reimbursement, and (6) close claim. An insurance claim will be rejected in call centers or in back office if the claim is not qualified for reimbursement. Similar to call centers, each case in back office is handled by identical resources. In task “determine likelihood of claim”, three data items are required as input.

In the first step of the proposed method, the modeler gathers data to annotate the process model with several *attributes*. These attributes are classified into 4 dimensions: *task*, *case*, *resource* and *data* as discussed below.

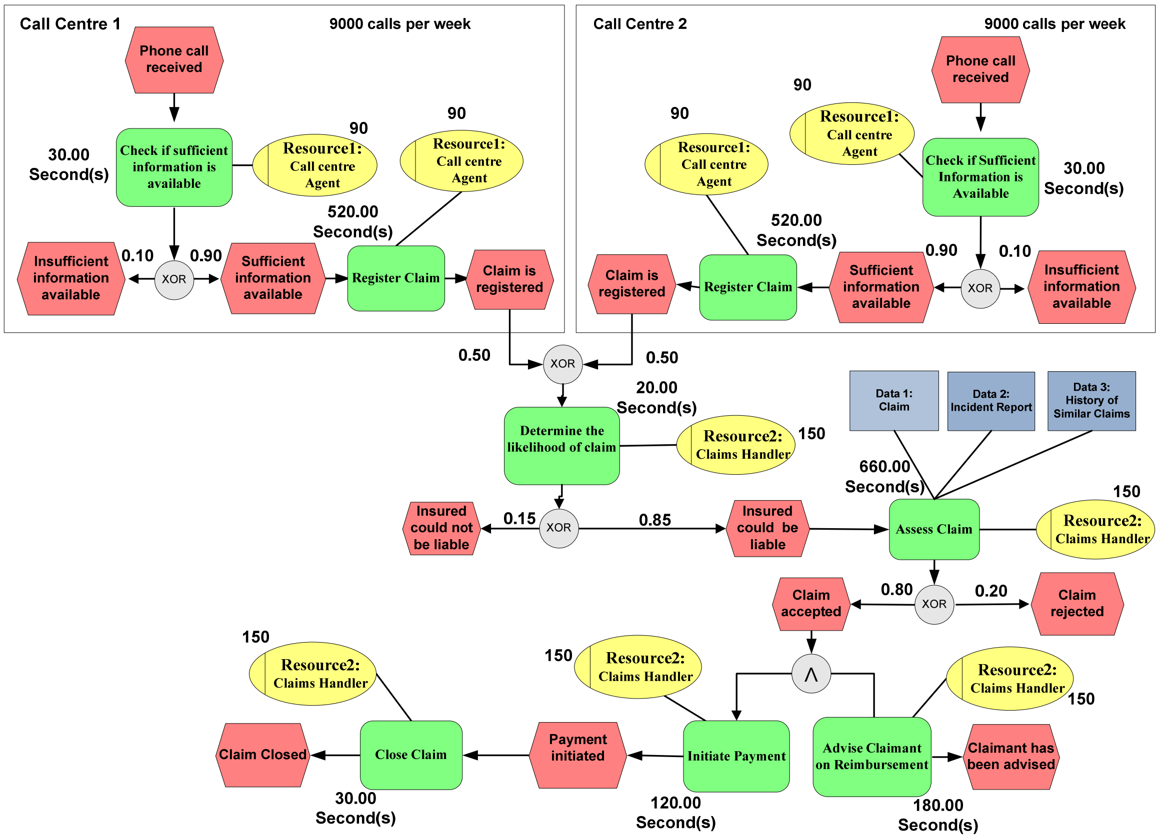

Figure 1. The base model

Attributes of a task: Each task is assigned with an average execution time (shown in seconds in figure Figure 1). In addition, for each task, three attributes are required to assess escalation options: average execution cost, deadline (calculated dynamically), and escalation cost. In Table 3, we give initial values for the attributes of each task in the running example.

Task (T_i)	**Task description**	**Average Execution Cost (C_{Ti})**	**Average Completion Time**	**Deadline**	**Escalation Cost**
T_{B1}	Check if sufficient information is available	C_{TB1}=10	30	54	150
T_{B2}	Register Claim	C_{TB2}=20	520	936	160
T_{S1}	Check if sufficient information is available	C_{TS1}=10	30	54	150
T_{S2}	Register Claim	C_{TS2}=20	520	936	160
T_{O1}	Determine likelihood of claim	C_{TO1}=20	20	36	180
T_{O2}	Access claims	C_{TO2}=30	660	1188	200
T_{O3}	Initiate payment	C_{TO3}=17	120	216	230
T_{O4}	Advise claimant on reimbursement	C_{TO4}=10	180	324	230
T_{O5}	Close claim	C_{TO5}=5	30	54	247

Table 3. Task attribute values for running example

Attributes of a case: Each case is assigned a unique identifier as well as a deadline, which is calculated based on average execution time of the tasks in the critical path of the workflow. In addition, a compensation cost (i.e. the cost of missing the deadline) is assigned to each case. The compensation cost can be a fixed amount, or a value from a certain range, or a function that takes as input the amount of time by which the deadline is missed. In the running example we assume that escalation cost is uniformly drawn from the range [120..170].

Attributes of a resource: Each (human) resource has three attributes: (a) role: used to describe the responsibility of an employee in the workflow (b) amount: number of resources; and, (c) cost: wage of a resource. Two types of resources (Resource1 and Resource2) are defined in the running example. The first type of resources is comprised of 180 call center agents and the second type of resources includes 150 claim handlers. The wage of a call center agent is 4000 (per 2 weeks) and the wage of a claim handler is 6000 (per two weeks). Resource cost represents the cost of utilizing a specific resource for a case (per two weeks).

Attributes of data: Each association (*task*, *data object*) is annotated with an estimated *preparation time* – the time required to retrieve and prepare the data for the task in question. In the running example, three data objects (Data1, Data2 and Data3) are required for task "Determine likelihood of claim". This task can only be executed when all three objects are available. In our experiments, the preparation time of Data1, Data2 and Data3 was 20, 20 and 30 seconds respectively.

The second step of the simulation method is to define the scenarios: one *normal* scenario and one or many *escalated* scenarios. In the running example, we assume there are two scenarios: (i) a normal scenario with approximately 9000 cases per two-weeks period at each call center; (ii) an escalated scenario (*storm season*) where the number of calls increases to 20000 cases. Here, we assume a negative exponential distribution, but other distributions can be adopted.

The third step of the method is to encode the process model and associated attributes for the normal scenario using a discrete event simulation technique. In this paper, we use Colored Petri Nets (CPN) [4], but other techniques/tools could be used instead (e.g. Arena).

Once the initial process model and its attributes are encoded, we simulate it under each scenario. Table 4 shows the results of simulation in storm season and in normal condition. The results show that the current model is suitable for the normal scenario but not for he storm season. Thus, escalation is needed. Table 4 specifically shows that a bottleneck exists in Back Office at Task TO1 during storm season.

Measurement		Normal condition (9000 cases per week)	Storm season (20000 case per week)
Time	Average workflow time	1213	35951
	Waiting time at Brisbane	0	0
	Waiting time at Sydney	0	0
	Waiting time at Back Office	0	40446
Cost	Average workflow cost	1566000	3480000
	Resource cost	1620000	1620000
	Average compensation cost	756000	4720000
	Total cost	3942000	9820000

Table 4. Initial results in normal and in storm season

5. Introducing Escalations

The next step in the simulation method is to perturb the base process model in order to incorporate different escalation strategies. Below, we show how this is done for the escalation strategies previously introduced, using the running example as a basis.

Alternative Path Selection (S1): We consider three alternative tasks TB2', TS2', and TO2' to replace TB2, TS2, and TO2 respectively (Table 5). These new tasks have higher execution cost, and shorter execution time.

New Task (T_i)	Task description	Tasks to be replaced	Average Executing Cost of task (C_{Ti})	Average Time
$T_{B2'}$	Rapid Register Claim	T_{B2}	$C_{TB2'}$=25	320
$T_{S2'}$	Rapid Register Claim	T_{S2}	$C_{TS2'}$=25	320
$T_{O2'}$	Access claims (Rapid method)	T_{O2}	$C_{TO2'}$=50	400

Table 5. Three alternative tasks and their properties

Given these alternative tasks, we define three instances of the alternative path selection strategy, which are outlined in Table 6.

Strategy	Description
S1.1	Replace T_{B2} with $T_{B2'}$, T_{S2} with $T_{S2'}$
S1.2	Replace T_{O2} with $T_{O2'}$
S1.3	Replace T_{B2} with $T_{B2'}$, T_{S2} with $T_{S2'}$,T_{O2} with $T_{O2'}$

Table 6. Instances of Alternative Path Selection

Resource redeployment (S2): Three instances of this strategy were tested in our experiments (see Table 7). In strategy S2.1, the total number of call center agents is increased from 180 to 240. In strategy S2.2, the total number of claim handlers is increased from 150 to 200. In strategy S2.3, both call center agents and claim handlers are increased to 240 and 200 respectively. We also define the estimated cost of each type of resource (per two weeks) as 4000 and 6000 respectively.

Strategy	Resource Redeployment			
	No of call center agents	Total cost of call center agents	No of claim handlers	Total cost of claim handler
S2.1	240	960000	150	900000
S2.2	180	720000	200	1080000
S2.3	240	960000	200	1080000

Table 7. Instances of Resource Redeployment strategy

Data Degradation (S3) strategy: In normal condition, task TO1 can only be executed when all three data objects are ready. In the data degradation strategy, TO1 is allowed to start if two of the data objects are ready. Thus, the execution time of TO1 is decreased.

Early Escalation (S4) strategy: In this strategy, we add an escalation task (Negotiation with client – TE). with execution cost of 15 and average time of 300. In task TE, a negotiation is conducted with the client to delay his/her claim request. Since the claim is going to be delayed, appropriate compensation is offered to the client. We define the compensation cost of TE as 140.

In addition, a prediction task TP is added to decide if early escalation should be applied. The execution cost and average time of TP are 0 and 5. Tasks TE and TP are executed once in the call center, and once in the back office. Figure 2 shows the workflow model after applying Early Escalation strategy (S4).

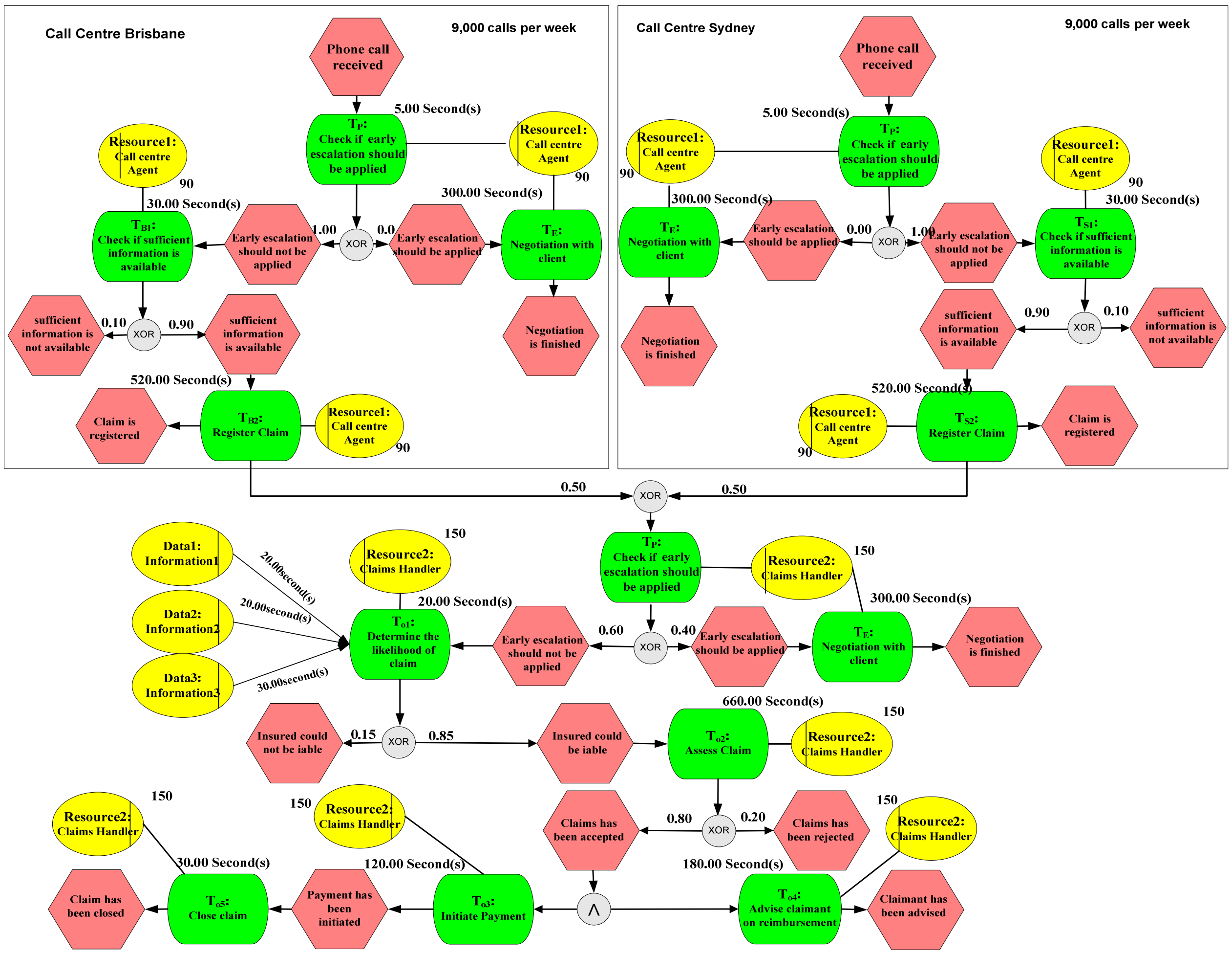

Figure 2. Running example after applying the Early Escalation strategy (S4)

6. Experimental Results

The final step in the proposed method is to analyze the simulation results in terms of time and cost.

Time. In the initial simulation, we found that average waiting time during storm season at the call centers is zero and at the back office is 4000. Thus, applying escalation strategies at the back office should be more effective than applying at the call centers. Figure 3 shows the average workflow time after applying the escalation strategies. Results indicate that Alternative Path Selection strategy (S1.1), Resource Redeployment strategy (S2.1) and Data Degradation strategy (S3) have less influence on the average time. One reason for their low effectiveness is that in the case study, these strategies are applied where there are no bottlenecks. For instance, Alternative Path Selection strategy S1.1 (see Table 6) is only applied on call centers where there is no waiting time. Meanwhile, Resource Redeployment strategy S2.1 (see Table 7) only increases the call center agents and therefore does not relieve the back-office bottleneck.

Figure 3 shows that Alternative Path Selection strategies (S1.2 and S1.3) are more effective in reducing the average workflow time, followed by Resource redeployment (S2.2, S2.3) and Early Escalation (S4). In Resource Redeployment (S2.2 and S2.3), increased numbers of resources in Back Office contributes to a reduction in average workflow time.

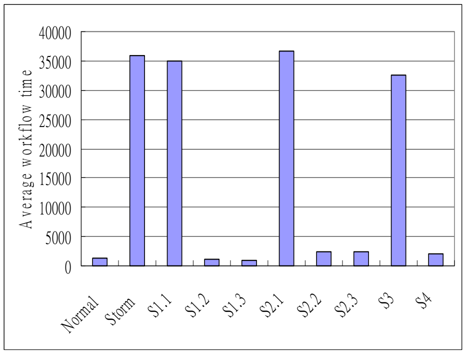

Figure 3. Average workflow time

With respect to cost, we differentiate between *workflow cost*, *resource cost* and *compensation cost*.

Workflow Cost. The workflow cost is equal to the total execution cost of all cases during one week period. The execution cost of a case can be calculated by summing the average execution cost of tasks involved in that particular case (see Table 3).

Figure 4 shows the average weekly workflow cost for the escalation options considered. As expected, high workflow cost is recorded during the storm season due to a larger number of claims. Alternative Path Selection options (S1.1, S1.2, and S1.3) increase the workflow cost since they add tasks. Early Escalation (S4) lowers the workflow cost since it escalates as early as possible. The remaining escalation strategies have little effect on workflow cost.

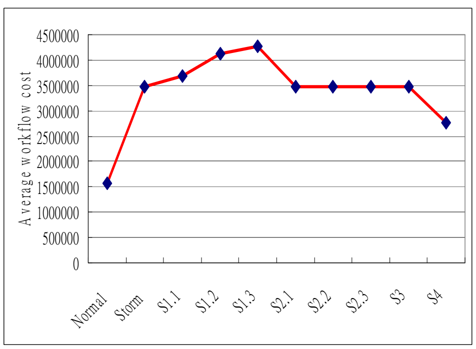

Figure 4. Comparison on average workflow cost

Resource Cost. Figure 5 shows the weekly average resource cost for the various escalation strategies. Only Resource Redeployment strategies (S2.1, S2.2, S2.3) increase the resource cost since these strategies alter the resource allocations in the workflow model. Among these strategies, strategy S2.3 has the highest resource cost due to the increase allocation in both call center agents and claim handlers.

Compensation Cost Compensation cost is incurred when workflow cases miss their deadlines. We assume that compensation cost is normally determined when the case starts, but can be re-negotiated (cf. Early Escalation Strategy). Figure 6 shows the average compensation cost (per week) of the experiment model for the eight escalation options considered.

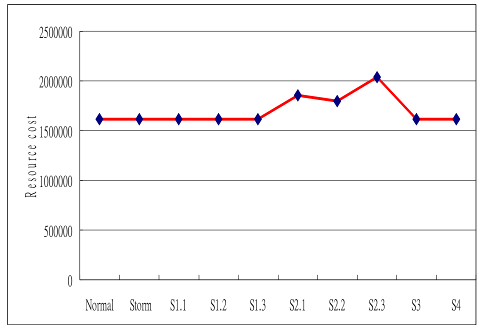

Figure 5. Average resource cost

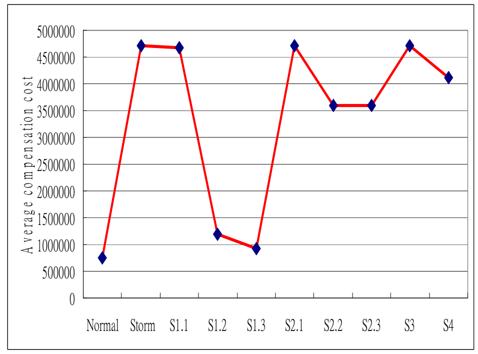

Figure 6. Average compensation cost

Comparison between Figure 6 and Figure 3 reveals that the escalation strategies with shorter average workflow time also have lower compensation cost, except in Early Escalation (S4). In general, shorter average workflow time implies that fewer cases will miss the deadline, resulting in lower compensation cost. For instance, Alternative Path Selection options (S1.2 and S1.3) which have short workflow time (see Figure 3) have also low compensation cost. Early Escalation strategy (S4) has shorter time but higher compensation cost since clients are compensated after a negotiation aimed at postponing the case deadline.

Total cost. The total cost is the sum of workflow cost, resource cost and compensation cost. Figure 7 shows the average weekly total cost in various conditions. The total cost of applying Alternative Path Selection (S1.1) and Resource Redeployment (S2.1) exceeds the total cost of the experiment model in storm condition in which no escalation has been performed. These strategies are unable to decrease the number of cases missing their deadlines. Thus, a large number of clients need to be compensated. Alternative Path Selection Strategies S1.2 has the lowest total cost among all strategies thanks to its low compensation cost.

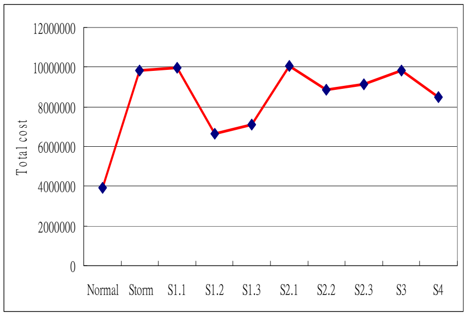

Figure7. Average total cost

8. Conclusion

Table 8 outlines the outcomes of the case study.

Strategy		Rank based on average flow time	Rank based on total cost
Alternative Path Selection	S1,1	6	6
	S1.2	2	1
	S1.3	1	2
Resource Redeployment	S2.1	7	7
	S2.2	4	4
	S2.3	4	4
Data Degradation	S3	5	5
Early Escalation	S4	3	3

Table 8. Summarized analysis result

Although these results are based on one case study only, they allow us to make some observations. First, the results reveal that the strategies decrease workflow time to varying degrees. The efficiency of each strategy is determined by the following factors:

- An escalation strategy is more effective if it is applied on tasks that have longer execution time. For instance, Data Degradation strategy S3 aims to reduce to the executing time of TO1. However, S3 has limited impact since it is applied at task TO1 which has a short execution time anyway.
- When an escalation strategy is designed to increase the number of available resources, the total number of added resources and the resource pools to which new resource are added, are crucial factors in shortening the workflow time. In the case study, resources were added at call centers and at the back office. The experiments showed that increasing resources at the call centers is less effective compared to back office since the degree of congestion is less critical at the call centers.
- The placement of escalations in the process is a key factor. For example, the S1.2 option, which alters Task TO1 (see section 5) located right in a bottleneck, is more effective in decreasing workflow time than options S1.1 and S1.3.
- Escalating in early stages of the process is more effective. However, due to higher cost, escalation should only be performed if it is really necessary.

Regarding cost, Resource Redeployment increases the number of resources and leads to higher resource cost. The other strategies alter the execution cost of some tasks and may therefore increase the workflow cost.

Above all, the analysis reveals that applying one strategy independently may not address all problems associated with a process. Modelers need to select suitable combinations of escalation strategies, taking into account resource allocation policies, budget and time constraints, and the desired service quality.

Acknowledgment

This research is funded by the University of Macau under grant RG056/06-07S/SYW/FST and by the European Regional Development Fund through the Estonian Centre of Excellence in Computer Science.

References

- [1] W.M.P. van der Aalst, M. Rosemann, and M. Dumas. Deadline-based Escalation in Process-Aware Information Systems. *Decision Support Systems* 43(2):492-511, 2007.
- [2] E. Panagos and M. Rabinovich. Escalations in Workflow Management Systems. In Proc. of the Workshop on Databases: Active and Real-Time (DART), pp. 25-29. Rockville, MD, November 1996.
- [3] E. Panagos and M. Rabinovich. Reducing Escalation-Related Costs in WFMSs. In Proc. of NATO Advanced Study Institute on Workflow Management Systems and Interoperability, pp. 107–127, Springer 1998.
- [4] CPN Group, University of Aarhus, Denmark. CPN Tools Home Page. http://wiki.daimi.au.dk/cpntools/.
- [5] D. Georgakopoulos, H. Schuster, D. Baker, A. Cichocki. Managing Escalation of Collaboration Processes in Crisis Mitigation Situations. In *Proc. of the Intl. Conf. on Data Engineering (ICDE)*, pp. 45-56, IEEE Comp. Soc., 2000.
- [6] J. Eder, E. Panagos, M. Rabinovich. Time Constraints in Workflow Systems. In Proc. of the 11th Intl. Conf. on Advanced Information Systems Engineering (CAiSE), pp. 286–300. Springer, 1999.
- [7] C. Bettini, X. Sean Wang, S. Jajodia: Temporal Reasoning in Workflow Systems. *Distributed and Parallel Databases* 11(3):269-306, 2002.
- [8] J. Chen, Y. Yang: Temporal dependency-based checkpoint selection for dynamic verification of fixed-time constraints in grid workflow systems. In *Proc. of the Intl. Conf. on Software Engineering (ICSE)*, pp. 141-150, IEEE Computer Society, 2008.
- [9] J. Hyun Son, M.H. Kim: Improving the performance of time-constrained workflow processing. *Journal of Systems and Software* 58(3): 211-219, 2001.

- [10] J.Q. Li, Y. Fan, M.C. Zhou: Performance modeling and analysis of workflow. *IEEE Transactions on Systems, Man, and Cybernetics, Part A* 34(2): 229-242, 2004