

ROBALO: A RISK-ORIENTED JOB DISPATCHING MECHANISM FOR WORKFORCE MANAGEMENT SYSTEM

Shi-Cho Cha, csc@mba.ntu.edu.tw
Hung-Wen Tung, hwtung@iii.org.tw
Han-Chao Lee, garry@iii.org.tw
Tse-Ming Tsai, eric@iii.org.tw
Raymund Lin, raymund@iii.org.tw
and Chih-Hao Hsu, eden@iii.org.tw

Advanced e-Commerce Institute, Institute for Information Industry, Taipei, Taiwan

Abstract We proposed a framework, called *Risk-Oriented job dispatching for mobile workforce management system* (ROBALO), to ease the tension between (a) the reliability requirement to serve a job request, and (b) the cost of the job's assignment. In ROBALO, the risks for workers to execute a job are taken into consideration. Such consideration is especially useful in the scenario of mobile workforce management because mobile workers usually meet unexpected situations in the field. Therefore, we can find the job assignment with the minimum cost under a certain degree of risk. Therefore, the job dispatcher can reserve enough resources and make enough preparations for a incident. In tradition, job dispatching mechanism usually take exception handling processes to deal with the failure of job execution. Compared to this approach, the time to discover the failure can be saved because we try to do things right at the first time.

Keywords: Mobile Workforce Management System, Online Job Dispatching, Risk Management

1. Introduction

As the advances of wireless technologies and hand-held computers, organizations are willing and able to deploy their *Mobile Workforce Management Systems (MWMSs)*, especially when their workers, such as sales, maintenance technicians, house care doctors,, need to deliver services in the field. Generally speaking, MWMSs can help organizations increase productivity by elim-

inating paperwork, arming technicians with more information at the point of performance, and capturing work completion data for enhanced management functions, etc. Hamilton, 2003, Sairamesh et al., 2004.

The main features of MWMSs includes Guido et al., 1998, Appelbald and Lonn, 2004: (1) *Automatic job dispatching*: automatically plan and dispatch job to field technicians or workforce; (2) *Workforce forecasting*: find out what effect a change in future workload; (3) *Schedule optimization*: let the system optimize an already existing and planned schedule; (4) *GIS/GPS support*; (5) *Mobility support*: support the workforce and make the workforce system available via some sort of mobile device; (6) *Monitoring*: monitor the progress of assigned job; (7) *Exception handling*.

Among these features, we focus on the job dispatching (especially the on-line job dispatching model) in this article. The main goal of job dispatching in MWMSs is to assign right people to satisfy a job request in the field under the constraints of time. Traditional job dispatching schemes usually treat failure of job execution as exceptions and take actions after failure is discovered. Obviously, the time to discover the failure is wasted in this case.

In light of this, we propose the Risk-Oriented job dispatching for mobile workforce management system (ROBALO). Simply speaking, ROBALO predicts the risk (or the expected loss) for works to finish a job assignment successfully based on the workers' capacities, statuses, and historical work experiences. When a job request is received, the request will be dispatched to workers so that the risk for the assignment can be controlled under a certain degree. Such consideration is especially useful in the scenario of MWMS because mobile workers usually meet unexpected situations in the field. In addition, while the cost for a person to execute a job is considered, our algorithm can find the assignment with the minimum cost under the degree of risk.

While we take risk into consideration, the following benefits can be achieved:

- The time to discover the failure can be saved because we try to do things right at the first time. It is especially important in the situation that time is critical.
- We provide a systematic approach to ease the tension between reliability and cost. Intuitively, the more number of workers are assigned to execute a job, the lower probability of failure. However, the more number of workers usually means the higher cost. While our approach finds the assignment with the minimum cost under a certain degree of risk, the requirement of reliability and cost can be balanced.
- With the risk information, job dispatcher can reserve some resources and make some preparations for the incident. For example, a dispatching center can have some insurance and transfer the risk to others.

- While we use the workers' experience and other historical data to calculate the risk, our approach can be self-adaptive to the real world. Obviously, the estimation of risk can become more precise while more historical data can be obtained.

The rest of the paper is organized as follows: Section 2 surveys related work on job dispatching on mobile workforce management. Section 3 gives an overview of the framework. Sections 4-6 discuss key components in the framework. Conclusions and future work are offered in Section 7.

2. Related Work

Traditional job dispatching methodologies can be classified into the following two categories: (1) the batch model and (2) the online model.

The batch model assumes that the job requests and available workers are known in advanced. Then the available workers are assigned to deal with these given jobs. For example, the classical job assignment problem in operation research domain tries to find a optimal solution of assigning n jobs to n workers with minimum cost while the cost of assigning worker i to job j is known Taha, 2003.

The batch model is useful for assigning the routine or scheduled jobs. However, in some cases, we cannot know every job requests in priori. For example, an emergency center may receive an emergency call from a injured person and need to dispatch an ambulance to take him/her to a nearby hospital. At this point, we can use online model to assign one or more workers to serve the incoming request.

We focus on the online model in this article (applying our system to the batch model is left to our future work). Current online job dispatching schemes usually assign workers to serve a incoming job request based on the workers' cost, capabilities, or current statuses. For example, HAMS (Healthcare Alert Management System)Chiu et al., 2004,Aydin et al., 2004 classifies staffs in a hospital into different roles based on their capacities. And each kind of tasks (or alerts) can only be assigned to the staffs with appropriate roles. When a alert is triggered, HAMS select a person from the available staff members who can play the roles required for the alert and dispatches the alert to the him/her. The SOS Alarm Normark, 2002 dispatch ambulances for emergency calls based on the proximity to the ambulance stations and the status of each ambulance.

Besides workers' capacities and statuses, we propose to take the probability that a person may fail to finish a job into consideration and use risks as a factor for job assignment in this article. Our ROBALO system assigns just enough workers so that a job can be finished within a specified level of risk. At the point of dealing with the failure of job execution, traditional online job

dispatching systems usually initiate their exception handling processes while assigned workers fail to finish a job. For example, HAMS Chiu et al., 2004, Aydin et al., 2004 sends messages to all staffs with the same role as it finds that a staff does not confirm a job assignment. Intuitively, in comparison with traditional exception handling approach, the time to discover the failure can be saved because we can make some “preparation” (e.g., we can assign another person as backup) while the job is dispatching.

3. System Overview

The architecture of ROBALO can be depicted in Figure 1. First of all, we maintain the following information for job dispatching:

- The *profiles of workforce* contains workers’ basic information, capacities, and other demographic information.
- The *current statuses of the workforce* include workers’ current location, availability information, etc. tracked by the *status monitor* (How to monitor the statuses is beyond the scope of this article).
- The *cost matrix* contains the cost for a person to execute a job.
- The *job execution historical logs* record the results for a worker or a working group to execute a job.
- The *past impact* logs the loss when a job is failed to execute.

Figure 2 shows the process of job dispatching. When a job request is received, the *Request Handler* parses the request to extract the context information from the request and generate constraints of the job. The *Candidate Filter* then uses these constraints to filter out unqualified workers and obtains a candidate list for this job.

In this article, we treat a risk as the expected loss when a job fails. And it can be calculated from a combinational function of loss expectancy and the probability of failure (the formal definition is shown in Section 6). Therefore, after the request context and candidate list are obtained from the above steps, the *Impact Evaluator* predicts the loss if the job cannot be finished on time based on the context of request and past impact history. On the other hand, the *Failure Probability Evaluator* predicts the probability of failure from the request context, the profiles of workforce, the workforce’s current statuses and job execution historical logs.

With the predicted impacts and failure probability, the *Worker Selection* then calculates the risk for a worker or working group to execute and select the workers to execute the job. Finally, the *Job Dispatcher* dispatches the result of job assignment to the selected workers.

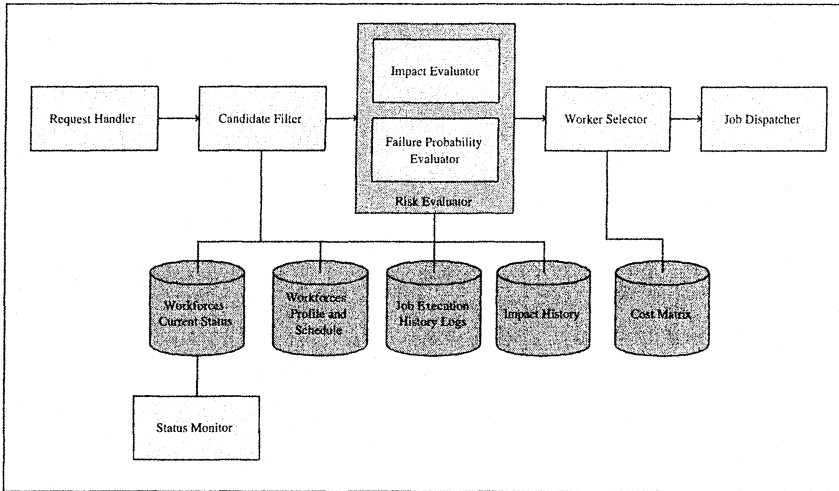


Figure 1. ROBALO Architecture

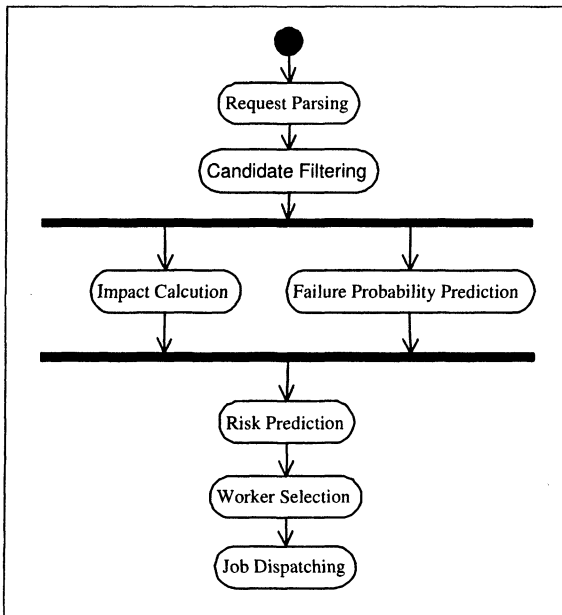


Figure 2. The Job Dispatching Process

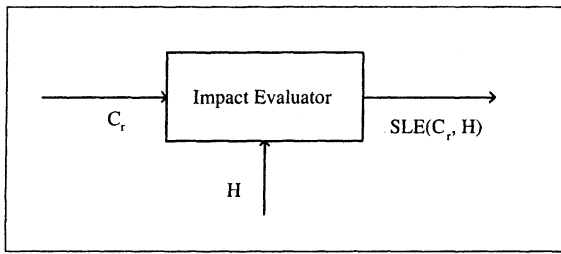


Figure 3. The I/O of the Impact Evaluator

4. Impact Evaluation

Figure 3 shows the input and output of the Impact Evaluator. First of all, the context of a request can be defined as follows:

DEFINITION 1 (CONTEXT) *The context of a request r (we denote it as C_r) is represented by a m -ary tuple: (c_1, c_2, \dots, c_m) . Each c_i represents an attribute or a feature of the context.*

With the definition of context, we can further define the impact history:

DEFINITION 2 (IMPACT HISTORY) *The impact history H can be defined as a set of tuple (C_i, I_i) . For a past failure request R_i , C_i is the context of this request. And I_i is the loss incurred from the failure execution of the job request.*

The output of the Impact Evaluator is the prediction of the loss when the request r cannot be finished on time (we denote it as $SLE(C_r, H)$). The traditional linear discriminant analysis methodology Fisher, 1936, Devijver and Kittler, 1982 can be used as follows:

- First, we classify the amount of loss into several classes. For example, we can divide the amount of loss into five classes as Table 1. For each value in $Class_i$ and $Class_j$, the value in $Class_i$ is less than the value in $Class_j$ if i is less than j .
- Secondly, for each adjacent class, find the linear discriminant function $g_{i,i+1}$ with the impact history H , where $g_{i,i+1}(X) = \sum_{1 \leq k \leq m} (v_k \times x_k + v_0)$ and X is a m -ary tuple: (x_1, x_2, \dots, x_m) . These linear discriminant functions are re-calculated every some period of time so that these functions can reflect the up-to-date scenarios. For a tuple X , if $g_{i,i+1}(X) \leq 0$, X belongs to $Class_k$ where $k \leq i$. Otherwise, X belongs to $Class_l$ where $l > i$.

1. m = the number of classes
2. for $i=1, i \leq m-1$
3. if $g_{i,i+1}(C_R) \leq 0$ then C_R belongs to class $_i$
4. next i
5. C_R belongs to class $_m$

Figure 4. The Algorithm for Context Classification.

- Finally, when a context C_r is received, we calculate with class C_r as shown in Figure 4. And the ceiling of the range of the class is returned as $SLE(C_r, H)$.

Table 1. Classification of Impacts

Class	Range
Class ₁	\$0 – \$10K
Class ₂	\$10K – \$1M
Class ₃	\$1M – \$100M
Class ₄	\$100M – \$10G
Class ₅	\$10G –

5. Probability Evaluation

As shown in Figure 5, the Probability Evaluator predicts the probability that a working unit fails to finish a job under the context of the received request. Besides context, we have the following definitions:

DEFINITION 3 (WORKING UNIT) *A working unit W_r is the minimum set that can be assigned to serve a request r . Supposed that the universal set of workers is U ($U = \{u_i \mid u_i \text{ is a worker}\}$), $W_r \in U^*$.*

DEFINITION 4 (JOB EXECUTION HISTORIES) *The job execution history EH can be defined as a set of tuple (C_i, W_i, S_i) . For a past job request R_i , C_i is the context of this request, W_i is the work unit assigned to execute the job, and S_i is its result (either success or failure).*

DEFINITION 5 (PROFILES OF WORKFORCE) *The current profiles of workforce WP is represented by a set of x -ary tuple: $(wp_{i1}, wp_{i2}, \dots, wp_{ix})$. Each wp_{ij} represents an attribute or a feature of the worker u_i 's profile.*

DEFINITION 6 (WORKFORCE STATUSES) *Similar to the profiles of workforce, the current statuses of workers WS is represented by a set of y -ary tuple: $(ws_{i1}, ws_{i2}, \dots, ws_{iy})$. Each ws_{ij} represents a kind of status of the worker u_i .*

To make it simple, we assume that the failure probabilities for each worker to execute a job is independent on another. This means that the failure probability of a work unit $P(W_r, C_r, EH, WP, WS)$ can be calculated from the product of $P(\{u_k\}, C_r, EH, WP, WS)$ where $u_k \in W_r$. Before computing $P(\{u_k\}, C_r, EH, WP, WS)$, a linear discriminant function d is found as follows:

- For each element (C_i, W_i, S_i) in EH , we extract the members of the working unit.
- For each worker u_k in the working unit W_i , we obtain the worker's profiles WP_{kt} and statuses WS_{kt} at that point of time t .
- We concatenate WP_{kt} , WS_{kt} , and C_i into a new $(x + y + m)$ -ary tuple (we call it as a working features vector (wf_{kti})). And use the tuple to find out the linear discriminant function d .
- After the linear discriminant function d is found, we calculate wf_{kti} for each working features vector and generate a set D . In the set D , we have tuples of the value of $d(wf_{kti})$ and its related result, e.g., $(0.12, \text{success})$, $(-1, \text{failure})$, $(0.01, \text{failure})$, etc..... And we denote it as (d_{wf_i}, s_{wf_i})

Then, $P(\{u_k\}, C_r, EH, WP, WS)$ can be calculated as follows:

- First of all, we extract the user's profiles and statuses and concatenate them with C_r into a working features vector wf_k .
- Compute the value of $d(wf_k)$.
- Compare the value of each d_{wf_i} in D and find N -nearest tuples.
- Compute the number of tuples where the value of s_{wf_i} is failure. If the number is n , the probability of failure is predicted as n/N .

Finally, $P(W_r, C_r, EH, WP, WS)$ can be obtained from the product of $P(\{u_k\}, C_r, EH, WP, WS)$ (for each $u_k \in W_r$).

6. Worker Selection

As demonstrated in Figure 6, the Worker Selector select a set of disjointed work units S . That is, $S = \{W_i | W_i \text{ is a working unit and } \neg \exists W_j, W_j \in S, W_i \neq W_j, \text{ and } W_i \cup W_j \neq \phi\}$. Also, S satisfied the following constraints:

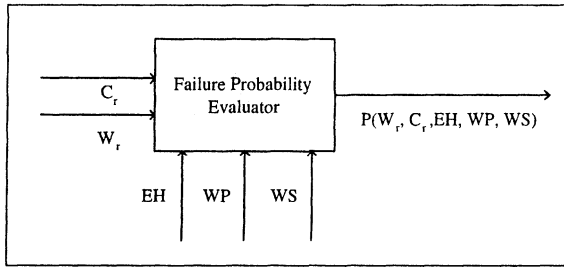


Figure 5. The I/O of the Probability Evaluator

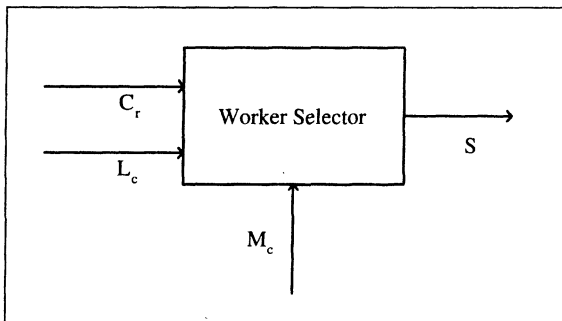


Figure 6. The I/O of the Worker Selector

If the acceptable risk is R_a , S is the assignment with the minimum cost among the possible assignments which risk are less than R_a . In this article, we adopt the risk definition in the domain of information security Tipton and Krause, 2004 and define risks as follows:

DEFINITION 7 (RISK) *The risk of a request R_r is the product of its loss expectancy ($SLE(C_r, H)$)(See Section 4) and the probability that a assignment S fails to serve the request ($P_r(S)$).*

Suppose that the failure probability for a working unit to execute a job is independent to others. We can calculate $P_r(S)$ from the product of $P(W_i, C_r, EH, WP, WS)$ (See Section 5), for each $W_i \in S$.

With the candidate list L_c and cost matrix M_c , we can select S with the pseudo-code shown in Figure 7. Generally speaking, the algorithm wishes to find S with the minimized cost where $P_r(S) \times SLE(C_r, H) \leq R_a$ or $P_r(S) \times SLE(C_r, H) \leq R_a/SLE(C_r, H)$. Obviously, if $R_a/SLE(C_r, H)$ is greater or equal to 1, any job assignment can satisfy the constraint. Therefore, we can select the working unit W in L_c with the least cost. Otherwise, because $0 \leq P(W_i, C_r, EH, WP, WS) \leq 1$, we can apply the logarithmic function

```

1. If  $R_a/SLE(C_r, H) \geq 1$  then
2.   Select the working unit  $W$  in  $L_c$  with the least cost
3.    $S = \{W\}$ 
4.   return  $S$ 
5. Else
6.    $current\_risk = 0$ 
7.    $W' = L_c$ 
8.    $S = \{\}$ 
9.   while  $current\_risk > R_a$  and  $W'$  is not empty
10.    Find the work unit  $W_i$  in  $W'$  with the biggest  $\log P(W_i, C_r, EH, WP, WS)/C_{W_i}$ ,
        where  $W_i$  does not contain any workers in  $S$ 
11.     $W' = W' - W_i$ 
12.     $S = S \cup W_i$ 
13.     $current\_risk = SLE(C_r, H) \times \prod_{W_i \in S} P(W_i, C_r, EH, WP, WS)$ 
14.  End While
15.  If  $current\_risk > R_a$  and  $W'$  is empty then
16.    The result cannot be found
17.    return  $\phi$ 
18.  End If
19.  return  $S$ 
20. End If

```

Figure 7. The Pseudo-code for Worker Selection.

to the inequality $\prod_{W_i \in S} P(W_i, C_r, EH, WP, WS) \leq R_a/SLE(C_r, H)$. The line 6–line 19 in Figure 7 try to find a solution to $\sum_{W_i \in S} \log(P(W_i, C_r, EH, WP, WS)) \geq (\log(R_a) - \log(SLE(C_r, H)))$ with the minimized cost. The greedy algorithm is used here to choose the working unit with the biggest unit cost of $\log(P(W_i, C_r, EH, WP, WS))$.

7. Conclusions and Future Work

We proposed a framework, called *Risk-Oriented job dispatching for mobile workforce system* (ROBALO), to ease the tension between (a) the reliability requirement to service a job request, and (b) the cost of the job's assignment. In ROBALO, the risks for a worker or a working group to execute a job assignment are taken into consideration. Such consideration is especially useful in the scenario of mobile workforce management because mobile workers may usually meet unexpected situations in the field. Therefore, we can use a systematic approach to find the assignment with the minimum cost under a certain degree of risk. Therefore, job dispatcher can reserve enough resources and make enough preparations for a incident. In tradition, job dispatching mechanism usually take exception handling processes to deal with the failure of job

execution. Compared to this approach, the time to discover the failure can be saved because we try to do things right at the first time.

Other than a concrete implementation of ROBALO, there are many interesting things left to be done. First, we observe that this paper focus on the online job dispatching scenarios. Similar methods may be used to develop a framework for batch scenarios. Secondly, current design in ROBALO assumes that the failure events are independent. The relationship between workers may be considered. Finally, our current system uses linear discriminant analysis to predict the risk from history data. More complicated approaches, such as non-linear programming technologies, may be included.

Acknowledgments

This research was supported by the Service Web Technology Research Project of Institute for Information Industry and sponsored by MOEA, ROC.

- Appelbald, Agneta Soderpalm and Lonn, Stefan (2004). A study of workforce management. Master's thesis, IT University of Goteborg.
- Aydin, Nizamettin, Marvasti, F., and Markus, H. S. (2004). Embolic doppler ultrasound signal detection using discrete wavelet transform. *IEEE Transactions on Information Technology in Biomedicine*, 8(2):182–190.
- Chiu, Dickson K. W., Kwok, Benny W. C., Wong, Ray L. S., Cheung, Shing-Chi, and Kafeza, Eleanna (2004). Alert-driven e-service management. In *HICSS*.
- Devijver, Pierre A. and Kittler, Josef (1982). *Pattern Recognition: A Stastical Approach*. Prentice Hall. ISBN 0-13-654236-0.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annal of Eugenics*, 7:179–188.
- Guido, Bruno, Roberto, Gavazzi, Tria, Paolo Di, and Bisio, Rossella (1998). Workforce management (wfm) issues. In *Network Operations and Management Symposium*, pages 473–482.
- Hamilton, Steve (2003). Workforce automation: Mobile computing for asset management. In *Proceedings of the 18th International Maintenance Conference (IMC2003)*.
- Normark, Maria (2002). Sense-making of an emergency call: possibilities and constraints of a computerized case file. In *NordiCHI '02: Proceedings of the second Nordic conference on Human-computer interaction*, pages 81–90, New York, NY, USA. ACM Press.
- Sairamesh, J., Goh, S., Stanoi, I., Padmanabhan, S., and Li, C. S. (2004). Disconnected processes, mechanisms and architecture for mobile e-business. *Mobile Networks and Applications*, 9(6):651–662.
- Taha, Hamdy A. (2003). *Operations Research*. Prentice Hall. ISBN 0-13-048808-9.
- Tipton, Harold F. and Krause, Micki (2004). *Information security management handbook 5-th ed*. CRC Press. ISBN 0-8493-1997-8.