Third 2008 International Conference on Convergence and Hybrid Information Technology

U-Worker Scheduling System: a Case Study at KT

Dong Cheul Lee, Ki Eung Kim, Jae Jin Lee
KT Technology Laboratory
{jackdclee, gekim, jaejin}@kt.com

Abstract

In a telecom, field workers visit many customer sites to provide services and repair faults. Scheduling the workers is hard because there are many tasks and the number of the workers is not sufficient. To tackle this problem, KT had transformed the field workers into ubiquitous workers and used work team based scheduling system. However, this approach was not sufficient to reduce frequent worker's arrival time changes and to increase operational efficiency. Therefore, we proposed individual worker scheduling system for managing the worker's schedule efficiently. We also developed an optimal worker selection algorithm based on statistical information when an operator assigns a task to the worker.

1. Introduction

In a telecom, field workers had many tasks to visit customers and provide services. In the meantime, the amount of task and the diversity of customer needs are overwhelming and the workers available in the telecom's human resource were not sufficient. To solve this problem, KT had transformed field workers into ubiquitous workers(u-Workers). U-Workers carry a hand held device so that they can communicate with operators at call centers or work managers at local offices anywhere. By using the device, they can get detail information of the tasks and transmit the result of the task after they finished it. Furthermore, KT had adopted work team based appointment reservation system to manage the worker's schedule. By using this system, call center operators could take customer calls, arrange appointments, and assign the tasks to work teams for providing services. And each worker could retrieve the task which was assigned to their team.

However, this system has several disadvantages. First, the operators can not know precise time when the worker visits the customer site. Second, the workers retrieve tasks which were assigned to their team by themselves so that the tasks can not be distributed evenly. Third, if an urgent task has assigned to a team which should be done quickly, work managers should find suitable workers on their own even though the system does not provide each worker's schedule.

There have been many studies about worker scheduling. Since a worker should visit a customer site, modeling this problem can be treated as a vehicle routing problem[1]. To find a near optimal solution for this, CVRPTW(Capacitated-Vehicle Routing Program with Time Window)[2] and the fast local search and guided local search[3] which simplifies a complicated real world problem to a similar simple well-known theoretical model were suggested. However, these static scheduling approaches were not efficient in dynamic environment. There had been several dynamic scheduling methods, such as dynamic workforce scheduling[4] and dynamic scheduler for work manager[5] and automated scheduling system[6], but they were not suitable for the operators who should arrange appointments and assign tasks in real time. An individual scheduling system[7] was suitable for real time environment, but estimating working time was not accurate enough.

We propose an individual worker scheduling system especially to address KT's u-Worker scheduling challenge in dynamic environment. KT is the largest telecom in Korea so that we can consider various environment factors.

2. KT's u-Worker scheduling problem

Previously, KT had used work team based scheduling system which was assigning tasks to work teams as described in Figure 1. When customers requested a service, an operator in call center searched suitable work teams. Those work teams should have been in charge of the work regions which included the customer site because work regions had been managed by work teams. Also, the total number of allocated

978-0-7695-3407-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ICCIT.2008.143

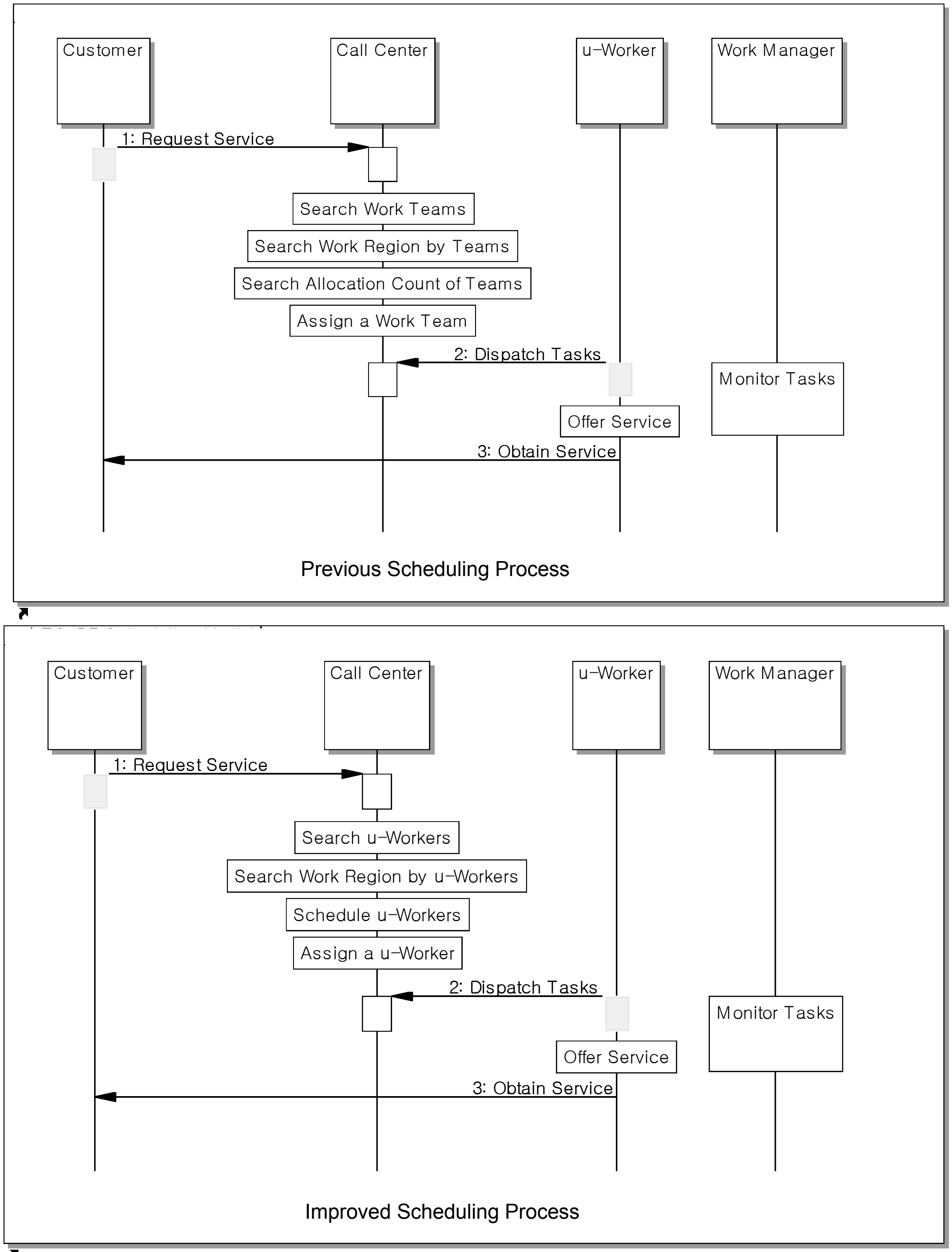

Figure 1. Previous and Improved Scheduling Process

tasks of the teams should not exceed maximum allocation count. To prevent concentrated allocation to a team, maximum allocation count was managed by team hourly. After the operator had selected a suitable team among the teams which satisfying a work region constraint and a maximum allocation count constraint, the task was assigned to the team so that any available u-Workers in the team can dispatch the task and offer the service to the customer. While they were doing their tasks, work managers have been monitoring the progress states of the tasks, and re-scheduled the delayed task.

However, as we mentioned before, this scheduling approach has several disadvantages since the scheduling process is based on the work team. Therefore, we propose individual u-Worker based scheduling process which assigns tasks to the worker directly as described in Figure 1. In this process, when a customer requests a service, the operator searches suitable u-Workers to deliver the service. Since work regions are managed by u-Worker in this process, the selected workers should have been in charge of the work region which includes the customer site. Work regions are small enough not to make the workers travel for a long time. Also, the workers who have available time slot can receive the tasks and the time slot is managed per 15 minutes.

3. System architecture

To tackle KT's u-Worker scheduling challenge, we designed the system architecture as shown in Figure 2. The Schedule Visualizer works as an interface between operators and the scheduling system and visualizes the schedules of u-Workers. It gets schedule data from the Schedule Manager after the Schedule Manager computes an optimal solution for selecting the workers using the Human Resource Manager, the Task Duration Estimator, and the Driving Time Estimator. The workers input their day off, holiday duty, night shift, and business trip information to the Human Resource Manager. The Task Duration Estimator estimates the duration of a task before the worker actually does the task. Also, the Driving Time Estimator estimates the workers' moving time from one customer site to another before they actually move.

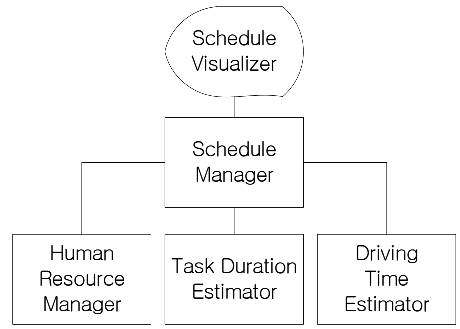

Figure 2. System Architecture

3.1. Task Duration Estimator

When a customer requests a service to a call center operator, the operator should be able to estimate the duration of the task to reserve the worker's schedule. Arrival time of the worker can be easily known since a customer wants specific visit time. However, the finish time of the task is hard to estimate since the duration of a task can vary with the worker's experience, service type, and provided network facilities.

The Task Duration Estimator can estimate the duration of a task by using statistical data. It calculates and stores the mean duration of previous tasks for each task type and u-Worker. The task types can be classified by the service and the network facility. Currently, KT provides following services: POTS, broadband, IP TV, VoIP, and WiBro. Also, each service is divided into provision and repair. And the network facilities are classified into FTTH, hybrid FTTH, xDSL, IP-xDSL, and Fast Ethernet. Furthermore, the duration can be varied by the day of the week and a time zone. The time zone is classified into forenoon, afternoon, and night. As a result, the mean duration $mdur(n)$ is managed by u-Worker ID, the service type, the facility type, the day of the week, and the time zone. And it is calculated as follows:

$$mdur(n) = \frac{\sum_{i=1}^{n} finishtime_i - arrivaltime_i}{n} \tag{1}$$

$finishtime_i$ means the time when the worker had completed the ith task and it can be obtained when the workers had finished their task and sent the result to the work managers. $arrivaltime_i$ means the time when the worker had arrived at a customer site for the ith task and it can be obtained when the workers had arrived at a customer site and notify work managers of their arrival time. Whereas n is the total number of tasks with same worker ID, service type, facility type, day of the week, and time zone for latest one month.

3.2. Driving Time Estimator

Even though call center operators can estimate the duration of the task by using the Task Duration Estimator, they should also consider departure time of a u-Worker from a previous customer site. Therefore, they should also estimate the time to travel from a previous customer site to a next customer site. However, the driving time is hard to estimate since it can vary with moving distance, moving velocity. Also, there can be many uncertainties such as weather and traffic conditions.

The Driving Time Estimator estimates the driving time of the worker from one customer site to another by using statistical data. It calculates and stores mean velocity for each worker. Like the Task Duration Estimator, the driving time can be varied by the day of the week and a time zone. As a result, the mean velocity *mvel*(n) is managed by u-Worker ID, the day of the week, and the time zone. And it is calculated as follows:

$$mvel(n) = \frac{\sum_{i=1}^{n} \frac{dist_i}{arrivaltime_i - finishtime_{i-1}}}{n} \quad (2)$$

$dist_i$ means the previous worker's driving distance from i-1th customer site to ith site. To find $dist_i$, we used GIS with traffic function enabled. Since we can get the addresses of customers, the Driving Time Estimator converts the addresses to coordinates and computes the driving distance between two points by using the GIS. n is the total number of tasks with same worker ID, day of the week, and a time zone for latest one month. If the workers started the first task on a day, $finishtime_0$ means nine O'clock since they had started their work at that time. Also, $dist_1$ means the distance from their local office to a customer site. Likewise, the GIS also provide the driving distance from the previous customer site for each worker to the new task's location when the operator assigns a new task. Because the Driving Time Estimator can get the driving distance and the mean velocity of the worker, it can estimate the driving time by dividing the mean velocity into the driving distance.

3.3. Human Resource Manager

The Human Resource Manager manages a day off, a holiday duty, a night shift, and business trip information of u-Workers. It is used for searching the available workers at a specific date by the Schedule Manager. If they had a day off or a business trip on a certain day, they are excluded from a resource pool at that time. On the other hand, if they put in a holiday duty or the night shift, they are included in the pool at that time.

3.4. Schedule Manager

Call center operators should search a suitable u-Worker when they assign a task. If they choose a worker without any consideration at their convenience, they can not manage worker's schedule efficiently. However, finding a suitable worker by themselves is hard due to there are many factors to consider.

The Schedule Manager searches a near optimal worker for a task. It uses *find_worker*() algorithm to find a solution. The input value of the algorithm consists of arrival time, service type, and work region. These values can be obtained while the operator talks over with a customer. It uses the Driving Time Estimator and the Task Duration Estimator to estimate the departure time and finish time for each worker. Also, the Human Resource Manager is used for excluding workers who don't work on weekday and including workers who work on holiday or at night.

3.5. Schedule Visualizer

Call center operators should be able to identify and monitor u-Worker's schedule. The Schedule Visualizer display worker's schedule and interacts with the operators. It represents the schedule as a Gantt chart or on GIS map so that the operator can identify the distribution of the tasks. It also helps the work managers identify the task that is not likely to be completed on time so that they can rearrange the next task of the worker.

4. Implementation

The u-Worker scheduling system is implemented in KT's NeOSS-WM(New Operations Support System-Workforce Management). NeOSS-WM consists of PC client, PDA client, application server, EAI server, and database server as shown in Figure 3.

The PC client is used by call center operators and work managers at local offices. They can get a suitable u-Worker when assigning a task to a worker. They can also monitor the current status of the tasks whether there are delayed tasks. It is implemented as C/S method and works on .NET framework.

U-Workers use the PDA client when they are doing their tasks. They notify the application server of their

Algorithm *find_worker*(*arrivaltime*, *servicetype*, *workregion*)
BEGIN
1. DEFINE *wlist* , *templist* as worker's list;
2. IF(*arrivaltime*==holiday) THEN store workers who are responsible for a holiday duty into *wlist*;
   ELSE IF(*arrivaltime*==night) THEN store workers who are responsible for a night shift into *wlist*;
   ELSE store workers who don't have a day off or are not on a business trip into *wlist*;
3. exclude workers who are not in charge of the *servicetype* from *wlist*;
4. exclude workers who are not responsible for the *workregion* from *wlist*;
5. *templist*=*wlist*;
6. DEFINE *stime* as the departure time for the task, *ftime* as the finish time of the task;
7. FOR(each worker *t* in *templist*)
   BEGIN
   stime=*arrivaltime*-*DrivingTimeEstimator*();
   ftime=*arrivaltime*+*TaskDurationEstimator()*;
   IF time between *stime* and *ftime* is overlapped with other schedules of *t* THEN exclude *t* from *templist*;
   END
8. IF *templist* is NULL THEN RETURN *wlist*;
9. IF *templist* has workers having no task THEN exclude workers who have tasks from *templist*;
10. delete all workers from *templist* except top 5 workers who have shortest (driving time+task duration);
11. IF(gap between (driving time+task duration) of workers in *templist* is within 10 min) THEN delete all workers from *templist* except a worker who has minimum number of tasks;
12. ELSE delete all workers from *templist* except a worker who has shortest (driving time+task duration);
13. RETURN *templist*;

Algorithm. *find_worker*()

arrival time and finish time by using the client. If an operator has assigned a task to a worker, the application server sends a short message to the worker so that they can identify their schedule and retrieve detail information from the server by using the client. The PDA client communicates with the application server on WiBro or CDMA2000 1x or EVDO network. It uses Windows Mobile .NET platform and MS SQL

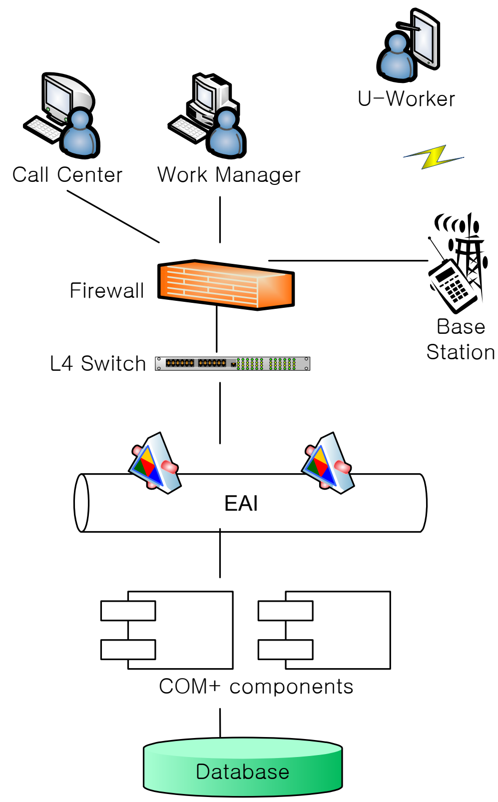

Figure 3. Implementation

CE 2.0 DBMS. And it is implemented with C# .NET on .NET framework.

The Driving Time Estimator and the Task Duration Estimator and the Schedule Manager are implemented in the application server as COM+ components. The server communicates with other systems or the clients via the EAI server. BizTalk Server 2004 is used for the EAI server. Both servers operate on Windows Server 2003 platform and load balanced by using L4 switch and protected by a firewall. Server applications are implemented with C# .NET.

The database server stores tasks, schedules, and u-Workers information from the application server and it uses MS SQL Server 2000 as a DBMS. Periodically executed stored procedures are registered as SQL Server jobs for calculating and storing the average durations of the tasks and the average driving velocities of the workers.

4. Conclusion

We proposed the scheduling system for u-Workers especially to address KT's worker scheduling challenge. To solve this problem, we designed five modules which are the Task Duration Estimator, the Driving Time Estimator, the Schedule Visualizer, the Human Resource Manager, and Schedule Manager. They were implemented in NeOSS-WM and the system has been used for scheduling KT's u-Workers successfully.

As a future work, we will improve NeOSS-WM to solve not only KT's workforce scheduling problems but also other resource scheduling problems.

5. References

[1] G.Laporte, M. Gendreau, J.Y. Potvin, and F. Semet, "Classical and modern heuristics for the Vehicle Routing Problem", *Intl. Trans. in Op. Res.* 7, 2000, pp. 285-300.

[2] L.J. Schumitt, J. Aflaki, S.T. Pitts, and R.H. Kamery, "An Emphasis on Heuristics Combined with GA to Improve the Quality of the Solutions: Some Methods Used to Solve VRPs and VRPTCs", *Allied Academies International Conference*, 2004, pp. 53-58.

[3] E. Tsang and C. Voudouris, "Fast Local Search and Guided Local Search and Their Application to British Telecom's Workforce Scheduling Problem", *Operations Research Letters*, 1997, pp. 119-127.

[4] D. Lesant, C. Voudouris, and N. Azarmi, "Dynamic Workforce Scheduling for British Telecommunications plc", *INTERFACES, 30(1)*, January-February 2000, pp. 45-56.

[5] D. Lesaint, N. Azarmi, R. Laithwaite and P. Walker, "Engineering Dynamic Scheduler for Work Manager", *BT Technology Journal Vol 16(3)*, July 1998, pp. 16-29.

[6] D.C. Lee, S.I. Kim, S.S. Kim, D.S. Yun, "The Scheduling Model and Algorithm for Workforce Management: A case study at KT", *Asia-Pacific Network Operations and Management Symposium*, 2005, pp. 552-562.

[7] D.C. Lee, "Individual Workforce Scheduling Architecture: a case study at KT", *International Association of Science and Technology for Development - Communication Systems, Networks, and Applications*, 2007, pp. 42-47.