

Providing Fairness to Mobile Workforces in an Automated Task Allocation Process: A Semantic Multi-agent Approach

Arash Mousavi, Riza Sulaiman,
Md. Jan Nordin, Zulaiha Ali Othman and Syaimak Abdul Shukor
Department of Industrial Computing,
Faculty of Information Science and Technology,
National University of Malaysia 43600, Bangi, Malaysia

Abstract: Problem statement: Mobile Workforces (MW) unlike computational resources of an automated system are active but not passive entities. Therefore, an automated resource allocation system that deals with MWs should assign tasks to them fairly and in a comparatively equal manner. An unfair task allocation in a group will cause dissatisfaction, which in turn demotivates MWs who are supposed to work as a team. **Approach:** In an automated Mobile Workforce Brokering System (MWBS) tasks are automatically assigned to MWs at Run-Time phase of the system's run. However, the environmental risks specifically risk of disconnection disrupts the task allocation process. Disconnection causes unfair task allocation when an MW must carry the next upcoming task according to a rotator work schedule, but he is disconnected. In this situation another MW has to perform the task in order to satisfy a pre-planned daily workload. **Results:** In this study we explore through the Run-Time phase of MWBS and explain how its underpinning ontology-driven coordination model tackles the risk of disconnection and improves the fairness in the task allocation process. **Conclusion:** Moreover, fairness rates in task allocation processes are compared between an existing system and MWBS and improvement in fairness rate is shown and analyzed for 4 consecutive periods (months) of the system's run.

Key words: Risk of disconnection, fairness in task allocation, mobile workforce, mobile workforce management system, mobile workforce brokering system, ontology-driven dynamic coordination model, semantic multi-agent systems

INTRODUCTION

In a Mobile Workforce Management (MWM) system, the resources of the system are humanitarian workers and therefore, they are not passive but active. Since human workers are entitled to certain labor rights, tasks have to be fairly distributed amongst them. Although fairness has more complicated meaning, in this research we assume that fairness is the equality in assigning tasks to different mobile workforces during a working period, therefore the more equal tasks are distributed the fairer the system will be. However, no system can be found that is absolutely fair or unfair, thus our main objective is to increase the fairness rate in an automated task allocation system up to a satisfactory level.

Procedural Fairness and work satisfaction: According to (Bos and Miedema, 2000) people are concerned about fairness because it protects them against uncertainty. Task distribution, undoubtedly is an uncertain matter in workplaces, which require a team

and group work. If there is no mechanism to assure the trustworthiness of the authorities who are responsible for task distribution, then the resulting uncertainty will create tension and dissatisfaction amongst the workforces of an organization, which in general term creates conflict between workforces and employers. Procedural fairness on the other hand, is the fairness toward how the things happen and thus work attitude of the workforces tightly depends on it. Recent research shows that procedural fairness is a convincing matter for employees and workforces of an organization that the authorities of the organization are trustworthy, which in turn causes more organizational commitment (Siegel *et al.*, 2005). In addition, according to (Tyler, 1997) people obey the law better when procedural fairness is high. Therefore, in an organization with high procedural fairness, the compliance with the organizational law is higher too. Another important factor, which has a positive impact on the overall performance of an organization, is group work satisfaction. According to the research done by (Burdett and Brianne, 2009), a major analyzing indicator for

Corresponding Author: Arash Mousavi, Department of Industrial Computing, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600, Bangi, Malaysia

group members towards group work is the fairness in workload distribution.

Fairness in mobile workforce brokering systems: In our previous studies (Mousavi *et al.*, 2010a; 2010b; 2010c; Mousavi and Nordin, 2007) an ontology-driven and multiagent-based Mobile Workforce Brokering System (MWBS) has been proposed. Basically, MWBS is an automated system that automatically assigns tasks to a set of MWs according to a prescheduled plan. A set of MWs, who are capable of performing a specific task are grouped in a cluster. The task handling plan for each cluster is made in a periodical manner and a working period in our proposed model lasts for one month. Prior to starting a work period, MWBS generates a monthly plan, which determines the number of tasks that a cluster has to perform in the coming month and a monthly deal, which is a legal agreement to enforce the cluster for being committed to the initial plan. The actual task allocation however, is made based on a daily plan that complies with the monthly deal in order to eventually fulfill the deal. Therefore, MWBS in each work period goes through two main phases; Initial phase and Run-Time phase. The latter is described in details in (Mousavi *et al.*, 2010a; 2010b; 2010c) and the former is the subject of this article.

During the Run-time phase, which is divided into sub-periods (days of the month), available MWs are put in a queue structure and upon receiving a new task it will be assigned to an MW located in the front of the queue. Since MWs are mobile workforces, the regular task allocation process can be disrupted by an environmental risk called the risk of disconnection or disconnected operation (Satyanarayanan *et al.*, 1993; Conan *et al.*, 2002; Abawajy and Deris, 2006). Apparently, the risk of disconnection disrupts the task allocation process when a new task arrives and has to be assigned to a specific MW from the front of the queue but that particular MW is disconnected and thus the system has no access to it. Since tasks have to be performed according to a legal deal, the system has to assign the task to the next available MW in the queue in order to fulfill the deal. The frequent disconnection of some MWs causes an unfair situation as tasks always will be assigned to the MWs who are always available and connected. Therefore, one can clearly conclude that the risk of disconnection is a potential threat to the procedural fairness in an automated task allocation system particularly MWBS.

Research hypothesis: We believe that an automated task allocation system utilizing an appropriate multi-agent architecture and an ontology-driven coordination model can satisfactorily address the risk of unexpected-

-disconnection in a mobile environment, which in turn increases the fairness in the system.

Organization of the study: The remaining in this study is organized as follows. In Materials and Methods section the Run-time phase of MWBS, critical disconnection, our proposed disconnection resolving algorithm and method of incorporating it into an ontology being used as a coordination medium are described. Results section illustrates the result of a simulation that compares an existing system with MWBS based on the percentage of fairness. Discussion section shows the improvement in percentage of fairness of MWBS in comparison with an existing system using no mechanism to resolve critical disconnections. Finally, Conclusion section sums up this study with some recommendations for future work.

MATERIALS AND METHODS

MWBS at Run-Time phase: The first step to describe the MWBS in Run-Time phase is to have a clear picture of its physical configuration that is depicted in Fig. 1. MWBS and Task Allocation System (TAS) (Mousavi *et al.*, 2010b; 2010c) are installed in two separated machines, which are connected to the same communication network. Figure 1 also shows two classes of human users; TAS monitoring staffs and MWBS monitoring staffs, which have access to TAS and MWBS respectively and monitor the operations of these subsystems and they can intervene in their process flow at any time due to any malfunctioning cases.

Moreover, as Fig. 2 illustrates, the life-line of the system in Run-Time phase is called a Period, which is one working month and defined as a duration during which a monthly plan for a cluster has to be executed. Within a period, a certain number of tasks that have been identified during the Initialization phase have to be performed by an MW Cluster. The second divider of the life-line of the system is a Session, which is one working day. Before a session starts, a daily plan has to be made, based on the monthly schedule and the availability of the MWs. A session itself is divided into a set of Rounds. A Round is a period of time during which the entire MWs, who are involved in a daily schedule, perform their tasks at least once. For example if there are four MWs in a cluster (MW1-MW4) then a round is completed when four tasks are performed by the cluster. Since in MWBS, tasks should be fairly allocated to all MWs, it utilizes a Queue data structure to organize the MWs who are working in a certain session (day) so that it can assign the tasks to them in a specific order. This Queue is called MWs Queue.

Figure 3 demonstrates the process flow of the task allocation process in MWBS. The process starts when TAS sends a new task to the system.

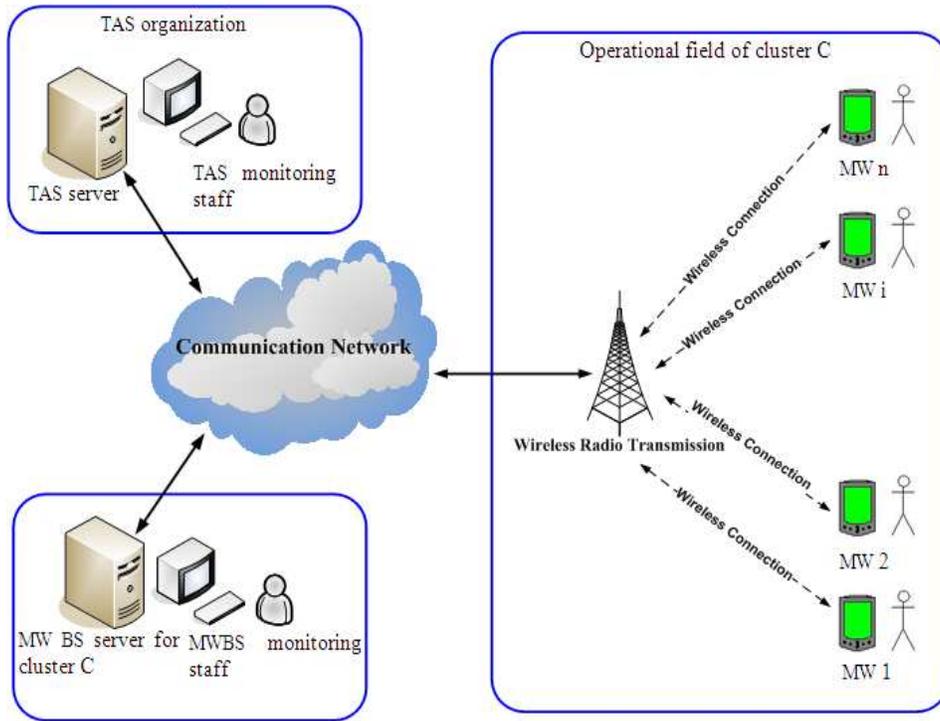


Fig. 1: Physical configuration of MWBS in Run-Time phase

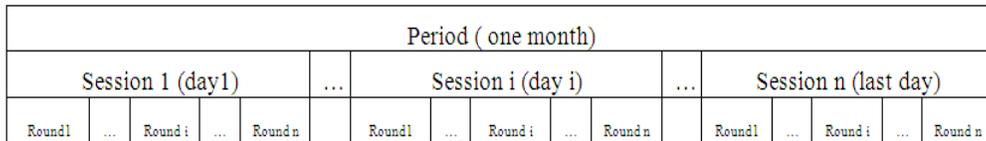


Fig. 2: The Life-line of the system in Run-time phase

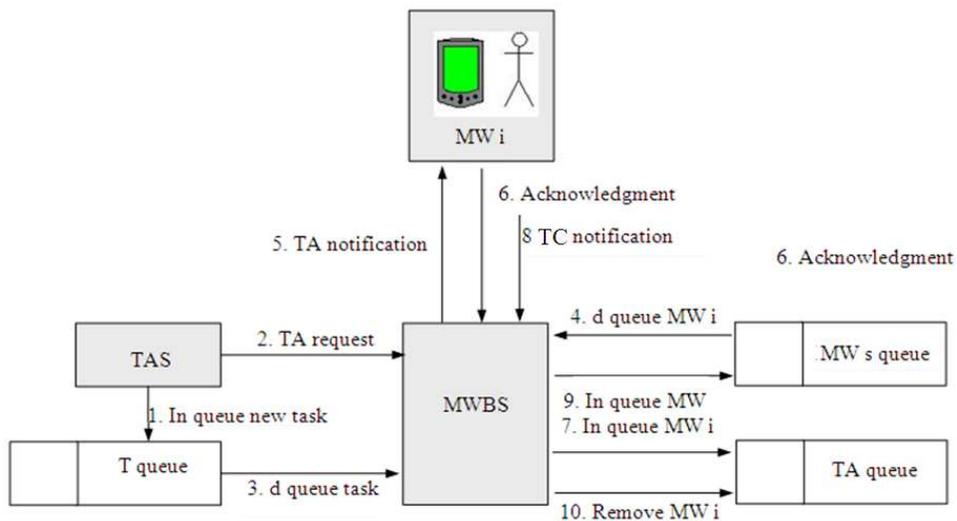


Fig. 3: The process flow of the task allocation process in MWBS at Run-Time phase

In doing that, TAS first adds the new task into a Queue structure called T Queue (Task Queue) and sends a Task Allocation Request (TA Request) message to MWBS. Upon receiving a TA Request, if there is any free MW exists in the system, MWBS fetches the task from the T Queue and then fetches an MW from MWs Queue, which is a Queue that keeps track of the available and free MWs in the system (it is obvious that if the entire MWs are engaged, task allocation process will be delayed until an MW gets free). Next, the task will be allocated to the MW by sending a Task Allocation Notification (TA Notification) message to that MW (e.g., MW_i) which in turn has to acknowledge the notification with MWBS. Upon receiving the acknowledgement, MWBS adds the MW_i to the task allocation queue (TAQueue), which is a structure that keeps track of the MWs who are currently engaged and to whom a task has been assigned. MWBS then waits until it receives a Task Completion Notification (TC Notification) from MW_i. Upon receiving a TC Notification message, which means that MW_i has finished its task, MWBS firstly, adds MW_i to MWs Queue and then removes MW_i from TA Queue meaning that MW_i is free now and the task assigned to it has already been completed.

Problem to address: As mentioned earlier, the main challenge of the process shown in Fig. 3 is the risk of disconnection. When an MW who has to perform the next task gets disconnected, the upcoming task has to be assigned to the next available MW in MWs Queue. If these are missing-assigning tasks are not recorded and taken care of, the fairest rate of the whole system will drop radically. Thus, providing a solution to tackle the reduction of fairness rate caused by the risk of unexpected and critical disconnection is the major goal of this study.

Solution description: Disconnection occurrence in mobile environments has no specific pattern. It may happen at any time and thus makes the Run-Time phase a highly unpredictable process. Although such unpredictability creates a high risk, it surely happens in specific ranges. In order to address this risk, one has to specify those ranges and examine the system in each possible range. Since in this article the main aim is to show the increment in fairness, we avoid exploring through all the possible ranges of disconnection. For the sake of simplicity and clarity, we choose the simplest situation and then develop a solution for it. Our solution consists of describing the conditions, variables that play effective roles in resolving the risk of disconnection and a disconnection resolving algorithm.

Next, we will be describing how to incorporate the solution into the body of our Ontology-Driven Coordination model (O-DC) described in (Mousavi *et al.*, 2010c).

Conditions of the problem domain:

- In each Period, an MW can be disconnected only once
- In each Round, only one MW and for only one time can be disconnected.
- No critical disconnection happens until an existing critical disconnection is resolved.
- Let t_T be the time required by MW to perform a task, t_{TA} be the time required for the next task to arrive and t_{RC} be the time required for reconnecting a disconnected MW, then: $t_T = t_{TA} = t_{RC}$

The variables involved in the disconnection resolving algorithm:

Loc_dc: A location in TAQueue containing the MW who is critically disconnected.

Loc_replaced: A location in TAQueue containing an MW who replaced the critically disconnected MW.

No-Of-MWs: Number of available MWs in the system in current period (working day).

Round_No: Current Round number.

Max_Round_No: Maximum number of Rounds that can be performed in a period.

Front: Variable indicating the front of the TAQueue.

The functions involved in the disconnection resolving process: Swap (int, int): swaps two locations of the TA Queue.

Assign task to (String): Assigns a task to a specific MW, whose name is passed to the function.

Disconnection resolves algorithm:

Definition 1:

Let: A Normal Round is a Round in which a critical disconnection can be resolved. If the TA Queue has four locations from 0 to 3, then according to the conditions of the problem domain, if a critical disconnection happens for locations 3-1 it is resolved during the same period. But if the critical disconnection happens for location 0 (rear of the queue) then it cannot be resolved in the same Round. Therefore, a normal round is the round in which either no critical disconnection happens or it happens for MWs located in locations 3-1 of the TA Queue.

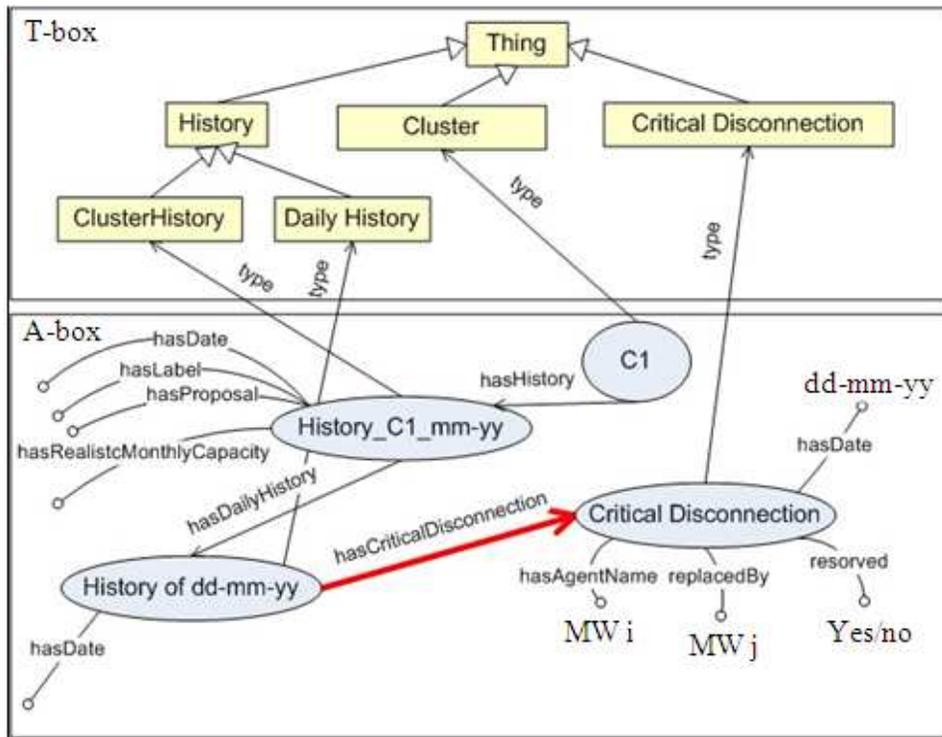


Fig. 4: Incorporating daily history and critical disconnection into the coordination medium

Table 1: Pseudocode codes for disconnection resolving algorithm

```

/*Disconnection can be resolved in a Normal Round*/
IF (Loc_dc < (No_Of_MWs - 1)) THEN
BEGIN
Swap (TAQueue[Loc_dc] , TAQueue [Loc_replaced])
Front = Loc_replaced
END
/*Disconnection can be resolved in an Extended Round */
IF (Loc_dc == (No_Of_MWs - 1)) THEN
IF (Round_No < Max_Round_No) THEN
BEGIN
Append (TAQueue, TAQueue [3])
TAQueue [3] = TAQueue [0]
Front = 4
Round_No = Round_No +1
END
/*Disconnection cannot be Resolved */
ELSE IF (Round_No == Max_Round_No) THEN
Print ("Disconnection Not Resolved ")

```

Definition 2:

Let: An Extended Round is around that has been extended to resolve a critical disconnection, which happened to an MW located in the rear of the TA Queue (location 0). An extended round according to the conditions of the problem domain is double the size of a normal round.

Based on the aforementioned definitions, variables, concepts and conditions we propose an algorithm for resolving a critical disconnection as depicted in Table

1. The algorithm divides the problem space into three parts as follows:

- When a critical disconnection can be resolved in a normal round: in this case MW in Loc_dc will be swapped with the MW located in Loc_replaced and therefore the next task will be assigned to the critically disconnected MW
- When a critical disconnection can not be resolved in a normal round: in this case, a new location will be appended to the front of TAQueue and TAQueue[3] will be placed in the new location (location 4). Next TAQueue[0] will be placed in TAQueue[3] so that it will perform 2 tasks in the extended round
- When a critical disconnection cannot be resolved: this is the case when TAQueue[0] will be critically disconnected in the last round of a period (current working day). In this case since no more round remains for resolving the critical disconnection, it will be considered as unresolved

Incorporating the solution into O-DC: In our early work (Mousavi *et al.*, 2010c) we have described how an ontology can be utilized to act as a coordination

medium to synchronize and manage the interdependencies of the activities involved in MWBS. Since managing the risk of disconnection in essence is of coordination problem, we use the same technique to firstly record the disconnection into the coordination medium and secondly retrieve the critical disconnections whenever required.

Figure 4 illustrates a partial view of the MWBSOnto. OWL, which is used as coordination medium. As depicted in this figure, every typical MW cluster (C1) may have many monthly histories and every monthly history has many daily histories.

```

The SPARQL Query to Extract a Critical Disconnection:
Q = cdRetrivalQuery :
PREFIX MWBSOnto:<http://MWBSOnto.owl#>
SELECT ?history ?dailyHistroy ?criticalDisconnection
WHERE {
MWBSOnto:C1 MWBSOnto:hasHistory ?history.
?history MWBSOnto:hasDate ?"mm-yy";
MWBSOnto:hasDailyHistory? ?dailyHistory.
?Daily Histroy MWBSOnto: hasCriticalDisconnection?
CriticalDisconnection.}

O-DS's Query Execution Operation:
query (cdRetrivalQuery)
    
```

Fig. 5: Query to retrieve entire critical disconnections of cluster C1 in a specific Month

```

The SPARQL Query to Extract a Critical Disconnection:
Q = cdRetrivalQuery :
PREFIX MWBSOnto:<http://MWBSOnto.owl#>
SELECT ?history ?dailyHistroy ?criticalDisconnection
WHERE {
MWBSOnto:C1 MWBSOnto:hasHistory ?history.
?history MWBSOnto:hasDate ?"mm-yy";
MWBSOnto:hasDailyHistory ? ?dailyHistory.
?DailyHistroy MWBSOnto: hasCriticalDisconnection?
CriticalDisconnection.
?criticalDisconnection MWBSOnto: resolved?"Yes".}

O-DS's Query Execution Operation:
query (cdRetrivalQuery)
    
```

Fig. 6: Query to retrieve resolved critical disconnections of cluster C1 in a specific Month

When a critical disconnection happens, a new instance of CriticalDisconnection class will be inserted into the ontology, which has to be connected to an appropriate daily history as well. A Critical Disconnection object has some properties such as replacing By and resolved, which describes whether it has been resolved and/or replaced by other agents. Moreover, the information attached to any existing critical disconnection object can be used to detect and then resolve a critical disconnection. Thus, appropriate queries have to be designed to retrieve these sorts of required information. In addition to analyzing the improvement of the fairness, the number of resolving critical disconnections also have to be calculated by retrieving resolved critical disconnection from the ontology. Figure 5 shows the SPARQL (Sirin and Parsia, 2007) query to retrieve the entire critical disconnection of MW Cluster 1 (C1) for a specific month and Fig. 6 shows a SPARQL query to retrieve the entire resolved critical disconnection of C1 for a specific month. Next section, presents the improvement of fairness using these queries.

RESULTS

Calculating and Comparing the Percentage of Fairness: We compare the fairness rates between an existing system and MWBS. The data are collected by simulating MWBS for 4 months of its life time. There are 4 variables being used in calculating fairness in this model as follows:

- TPE (Total Performed): Sum of the number of days that each MW performed during the 4 months of simulation
- TFR (Total Fairness Reduction): Each unresolved critical disconnection reduces the fairest rate by one unit; therefore total fairness reduction is the total number of unresolved critical disconnections
- Total Task Performed (TTP): We consider that each MW performs a constant number of tasks (15 tasks) per each working day. Therefore $TTP = TPE \times 15$
- Fairness (Percentage of fairness) = $[(TTP - TFR) \div TTP] \times 100$

Table 2 shows the number of days that each MW performed and the number of critical disconnections for each MW within 4 months for the existing system. In addition TPE, TFR, TTP and Fairness are calculated and depicted in Table 2 for the existing system. Table 3 on the other hand, shows the number of resolvable disconnection and unresolved disconnection for each MW employed by MWBS. We consider that TPE carries the same value in both MWBS and the existing system. In addition, TTP and Fairness for MWBS are calculated and depicted in Table 3.

Table 2: Average Fairness for existing task allocation system for 4 consecutive months

MW	Days performed	Critical disconnections
MW ₁	55	96
MW ₂	66	132
MW ₃	63	79
MW ₄	69	117
TPE=253		TFR = 424
TTP = 253 × 15 = 3795		
Fairness = [(3795-424) ÷ 3795] × 100=88.8%		

Table 3: Average Fairness for MWBS for 4 consecutive months

MW	Resolved disconnections	Unresolved disconnections
MW ₁	90	6
MW ₂	131	1
MW ₃	74	5
MW ₄	115	2
TTP = 253 × 15 = 3795		TFR=14
Fairness = [(3795-14) ÷ 3795] × 100 = 99.63%		

DISCUSSION

Figure 7 compares the Fairness between MWBS, Existing system and Ideal case. An ideal case is the one with 100% of fairness and is shown in Fig. 7 as a norm for graphical comparison. From the other side, Fig. 8 illustrates the monthly comparison between MWBS, existing system and the ideal case. As the results reveal, our technique of managing the risk of critical disconnection improves the percentage of fairness in comparison with the existing system. The average percentage of improvement of fairness in 4 months is: 99.63-88.8 = 10.83 %. This improvement in fairness proves that the technique described in this study is effective and appropriate to improve the fairness of task allocation amongst mobile workforces in an automated task allocation system.

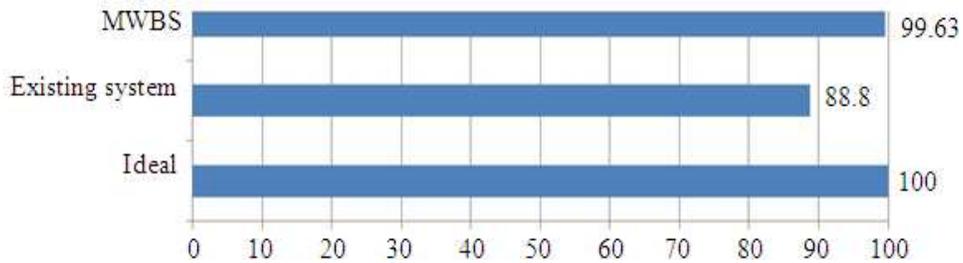


Fig. 7: Overall Fairness comparison between MWBS and existing system for 4 Months



Fig. 8: Monthly fairness comparison between mwbs and existing system during 4 months

CONCLUSION

In conclusion, an ontology-driven and multi-agent approach can be used as an effective way to address environmental risk such as risk of disconnection in an automated task allocation system. In this article we have shown that fairness is an important issue in distributing tasks amongst mobile workforces. Moreover, it has been shown that risk of critical disconnection is a threat to providing fairness in the task allocation process in mobile environments. Next, we propose an algorithm for resolving critical disconnection and incorporated our solution into an ontology, which acts as a coordination medium in MWBS. Finally, we have compared the percentage of fairness between an existing system that does not resolve critical disconnections and MWBS, which utilizes the proposed algorithm to resolve the critical disconnections, in four consecutive months. The result of the comparison showed an increment of 10.83% in the overall fairness in MWBS. However, our proposed algorithm can be still enhanced further as we have examined only a limited situation when the critical disconnection can take place. We believe that improving the proposed algorithm can enhance our technique to cover a wider range of disconnection situations and thus can be considered as future work of this research.

REFERENCES

- Abawajy, J.H. and M.M. Deris, 2006. Supporting disconnected operations in mobile computing. Proceedings of the IEEE International Conference on Computer Systems and Applications, Mar. 8-8, IEEE Xplore Press, pp: 911-918. DOI: 10.1109/AICCSA.2006.205197
- Bos, K.V.D. and J. Miedema, 2000. Toward understanding why fairness matters: The influence of mortality salience on reactions to procedural fairness. *J. Personality Soc. Psychol.*, 79: 355-366. DOI: 10.1037//0022-3514.79.3.355
- Burdett, J. and H. Brianne, 2009. Predicting satisfaction with group work assignments. *J. Univ. Teach. Learn. Practice*.
- Conan, D., S. Chabridon and G. Bernard, 2002. Disconnected operations in mobile environments. Proceedings of the International Parallel and Distributed Processing Symposium, Apr. 15-19, IEEE Xplore Press, France, pp: 192-199. DOI: 10.1109/IPDPS.2002.1016593
- Mousavi, A. and M. Nordin, 2007. An architectural model for a multi-agent mobile workforce brokerage system based on CBR-BDI agent architecture and active shared-data space coordination model. Proceedings of the IEEE International Conference on Electrical Engineering and Informatics, Jun. 17-19, Bandung Indonesia, pp: 294-297.
- Mousavi, A., M.J. Nordin and Z.A. Othman, 2010a. An ontology has driven procedural reasoning system-like agent model for multi-agent based mobile workforce brokering systems. *J. Comput. Sci.*, 6: 557-565. DOI: 10.3844/jcssp.2010.557.565
- Mousavi, A., M.J. Nordin and Z.A. Othman, 2010b. An ontology driven, multi-agent based framework for automated resource allocation in Mobile Workforce Management systems. Proceedings of the International Symposium in Information Technology, Jun. 15-17, IEEE Xplore Press, Kuala Lumpur Malaysia, pp: 1416-1421. DOI: 10.1109/ITSIM.2010.5561520
- Mousavi, A., M.J. Nordin and Z.A. Othman, 2010c. Ontology-driven coordination model for multiagent-based mobile workforce brokering systems. *J. Applied Intell.*, 36: 768-787. DOI: 10.1007/s10489-011-0294-z
- Satyanarayanan, M., J.J. Kistler, L.B. Mummert, M.R. Ebling and P. Kumar *et al.*, 1993. Experience with disconnected operation in a mobile computing environment. Proceedings of the 1993 USENIX Symposium on Mobile and Location-Independent Computing, (MLICS' 93), CiteSeerX, USA., pp: 11-28.
- Siegel, P.A., C. Post, J. Brockner, A.Y. Fishman and C. Garden, 2005. The moderating influence of procedural fairness on the relationship between work-life conflict and organizational commitment. *J. Applied Psychol.*, 90: 13-24. DOI: 10.1037/0021-9010.90.1.13
- Sirin, E. and B. Parsia, 2007. SPARQL-DL: SPARQL query for OWL-DL. Proceedings of the 3rd OWL Experiences and Directions Workshop (OWLED' 07), CiteSeerX, USA.
- Tyler, T.O.M.R., 1997. Procedural fairness and compliance with the law. *J. Econ. Stat.*, 133: 219-240.