Task Transfer in Software Agent Community with Sincerity Merit Point

Nur Huda Jaafar[1(✉)], Mohd Sharifuddin Ahmad[2], and Azhana Ahmad[2]

[1] Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, KM 12, Jalan Muar, 85009 Segamat, Johor, Malaysia
nurhu378@johor.uitm.edu.my
[2] College of Computer Science and Information Technology, Universiti Tenaga Nasional, Putrajaya Campus, Jalan IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia
{sharif,azhana}@uniten.edu.my

Abstract. The software agent technology is one of the human assistive technologies that enables team working. In the process of achieving the team goals, an agent may need help from its teammate to perform its remaining task's activities in order to meet the task's deadline. However, certain conditions are needed to be fulfilled for the task transfer even though these would be a burden to the teammate. This paper shows the use of a Workload Manager for handling signals to allow the agent getting help from its teammate. It is also used to identify the available teammate agents that can really help. In this paper also, we simulate the transfer of the remaining task's activities from one agent to another and demonstrate the process of awarding merit points to the agent that sincerely helps its teammate.

Keywords: Task · Task transfer · Teammate agent · Workload · Sincerity

1 Introduction

The existence of intelligent technologies such as drones and robots which are authorized to make decisions at a certain level eases many burdens of humans especially in performing tasks. With these technologies, task performances are accelerated and at the same time are maintained at high quality of delivery and output. One of the autonomous technologies that support these intelligent technologies is the software agent technology. A software agent is able to work individually or in a team to accomplish its assigned objectives [1]. The ability of a software agent to collaborate with its teammate and react to its environment is one of the keys that accelerate and improve task performance. It is also capable of performing multiple tasks at a certain time. This autonomous technology has proven its advantages in many fields such as healthcare, energy consumption, manufacturing and military [2–5].

While the advantages of software agent technology have been proven in many fields, the mechanism to avoid imbalance of task delegation among agents in a community should be taken into account in designing the intelligent system. The situation

M.S. Mohamed Ali et al. (Eds.): AsiaSim 2017, Part I, CCIS 751, pp. 180–188, 2017.
DOI: 10.1007/978-981-10-6463-0_16

is worst if the tasks involve deadlines which cause problems to an agent. One solution to this problem is for the agent to transfer some of its tasks to a teammate agent. However, in certain situation this can be a burden to the teammate agent, especially if it is having a heavy workload. The agent may face problem to identify a suitable teammate agent which is willing to cooperate in completing the task.

This paper discusses the method of designing the process of task transfer in a software agent community by considering the total workload of each agent. We propose a Workload Manager that calculates an agent's workload but also determines the right time for the agent to get help from a teammate agent. The Workload Manager also identifies a suitable agent which can offer help to the agent. This paper also shows the process of awarding merit points to an agent which helps its teammate to sincerely perform the task.

This paper presents the work-in-progress of our research in modeling sincerity for software agents. The rest of this paper is organized as follows: The next section discusses the related work in this area. We show the design of task transfer in a software agent community. We then animate the process as a simulation to validate the design. Finally, we conclude the paper.

2 Related Work

A software agent is a component that supports autonomous technology. It can work individually or in a group to meet some assigned objectives [1, 6]. The characteristics of software agent such as reactive, autonomous, communicative, goal-oriented, learning, mobile and flexible supports the software agent to work in dynamic environment [1, 6–9]. It also interacts with its community via negotiation, coordination and cooperation activities.

The ability of a software agent to work in a team is an advantage of this technology to deliver a much faster output and with higher quality. It can also build teams in dynamic environments. These advantages provide humans with the confidence to authorize agents in making decisions at certain levels and assist humans [4, 10].

Although working as a team brings many benefits to software agent in achieving the goals, the coordination among agents in a community should be designed properly. Since agents may have individual and shared goals, designers in agent-based systems should design the coordination of agents to avoid conflicts in achieving the individual or shared goals [1]. The situations of multi-agent systems' organization should be carefully studied because collaboration within the community depends so much on it [11].

Conflicts could become worst if the tasks that are handled by the agents involve tight schedules with deadlines. The problem of tasks with the deadlines involve issues such as resources problems that need to be managed to avoid conflicts in performing tasks individually or in groups [12]. The agents may face with issues in managing to achieve individual or team goals. In this situation, the agents should find ways to balance the effort of fulfilling the individual and the team goals [11]. Here, agents' workloads could be considered in solving this problem. The workloads could be an indicator to identify the percentage of burden for each agent, which enforces community awareness between the agents and their environments [13].

In order to motivate the agents to take action ethically, the ethical behaviour such as human sincerity should be instilled in software agent environments. Machine ethic is one of the applied ethics that is created to ensure the machine such as robot, drone or autonomous system to function ethically. The increasing of autonomous machines usage in taking over humans' tasks is a signal to us to include the machine ethics in autonomous machines, especially in the role of decision making [4, 14, 15]. Adapting human behaviour to the machine environment is a solution to develop ethical machine behaviour that is more human user-friendly. Previous researches had proven that adapting human behaviour to machine environment brought a lot of advantages to human livelihoods [16, 17].

3 Task Transfer Between Agents

The process of task transfer involves a group of agents that consist of worker agents and their teammate agents. In this environment, we propose a Workload Manager (WM) that computes a workload for each agent. This Workload Manager will monitor the each task under every agent and also the performance of overall tasks. The monitoring activity of each task is to identify the needs of favour completing it before the deadline. From here, the Workload Manager will be able to use the signal to worker agent to get help from another teammate agent. While monitoring the overall task performance of all agents, the Workload Manager, at the same time, will be able to identify the teammates that could offer help to worker agent. The details of Workload Manager processes have been discussed in the previous research [18].

Figure 1 shows the process of transferring a remaining task from a worker agent to its teammate agent. Based on the workload of each agent, the Workload Manager broadcasts a signal when a worker agent is allowed to get help from its teammate

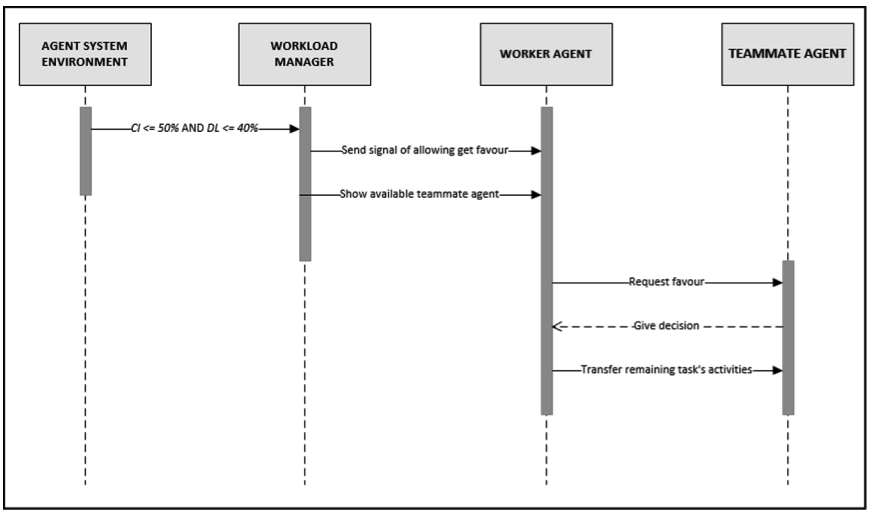

Fig. 1. Sequence diagram of transferring remaining task from one agent to another

agents. We constrain this signal to avoid a worker agent from overly depending on its teammate agents in performing tasks. In this research, the system broadcasts the signal when a worker agent achieves the percentage of a task completion; *CT <= 50%* and the remaining deadline of the task; *DL <= 40%*.

In order to avoid the burden to the other agents which are having heavy workloads, the Workload Manager calculates all agents' workloads to identify their availabilities. In this system, a worker agent can only get help from a teammate agent which does not have a heavy workload. The system considers an agent as having a heavy workload if the total workload of all tasks assigned to it exceeds 100%. The current task, T_c, and the incoming task, T_i, (the task that will be transferred to the agent) is summed up to determine whether the workload exceeds 100%. The agent with a heavy workload is considered as an indispensable agent.

From the list of available agents, a worker agent chooses a teammate agent, which would give a supportive response to the request. If the available agent is more than one, the worker agent chooses the one with the lowest workload among the available agents. The worker agent then sends a request and waits for the decision from the selected teammate agent. The teammate agent decides whether to accept or reject the request. Once the teammate agent agrees to help, the worker agent transfers its remaining task's activities to the teammate agent.

In this system, we implement the awarding of merit points to a teammate agent and Fig. 2 shows the sequence diagram of the process. Merit points are used to analyze and gauge the sincerity of a teammate agent when it gives help to a worker agent. We use the point system as a mechanism to formulate the sincerity of agents in the environment [18–20].

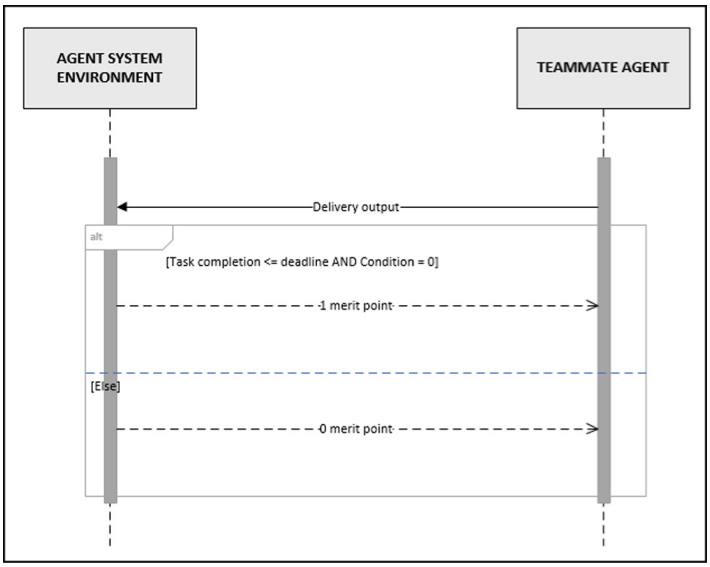

Fig. 2. Sequence diagram of awarding merit point to the agents

The system starts to analyze the merit point gains when a teammate agent delivers the output of a task it has agreed to help. Two criteria are considered when giving the merit point; (i) fulfilling the deadline, and (ii) the condition from the teammate agent. The teammate agent gets 1 merit point if the task is completed on time or earlier and the teammate agent does not give any condition for performing the task completely. The agent is considered as sincere agent if it fulfills these criteria. However, the agent gets 0 merit point if it fails to meet these criteria, which shows the agent's insincerity.

4 Simulating Task Transfer Between Agents

To validate the design, we create a simulation using Java and the JADE platform. We exploit the POSTGRESQL as a database for this system.

In the simulation, we create three agents: Agent_1, Agent_2 and Agent_3 that are involved in production works such as printing and packaging of products. In this simulation Agent_1 and Agent_2 have two tasks while Agent_3 is free. The tasks' information for Agent_1 and Agent _2 is stated in Table 1. Figure 3 is an example of an interface for inserting the task's information.

Table 1. The tasks' information for Agent_1 and Agent_2

Agent	Task	Task deadline	No. of activities
Agent_1	Printing	4/7/2017	3 activities
	Packaging	4/13/2017	5 activities
Agent_2	Printing	4/7/2017	3 activities
	Packaging	4/13/2017	5 activities

Subsequent to the entry of the task assigned to each worker agent, the system shows the total workload for each agent. Currently, the worker agents' information shows that the total workload of Agent_1 and Agent_2 is 17.77% each while the total workload of Agent_3 is 0.00%. Figure 4 shows the total workloads for all agents.

Figure 5 shows that there is a worker agent, which needs help to perform and complete its task. On 2017-04-04, the system detects that Agent_1 needs help to perform the printing task. The deadline for this task is 2017-04-07 and the completion status is 0/3, which shows that the activities are not completed for this task. Agent_1's workload also increases to 44.44%. These situations fulfill the condition for the agent to get help from its teammate.

Based on the Workload Manager, the available agent that can give help is displayed. In this case, Agent_2 and Agent_3 are available to give help to Agent_1. If the number of available agents that can offer help is more than one, the system compares the total workload among these available agents. Based on the communication between the agents as shown in Fig. 6, the system chooses Agent_3 because its workload is 0.00% while Agent_2's workload is 17.77%.

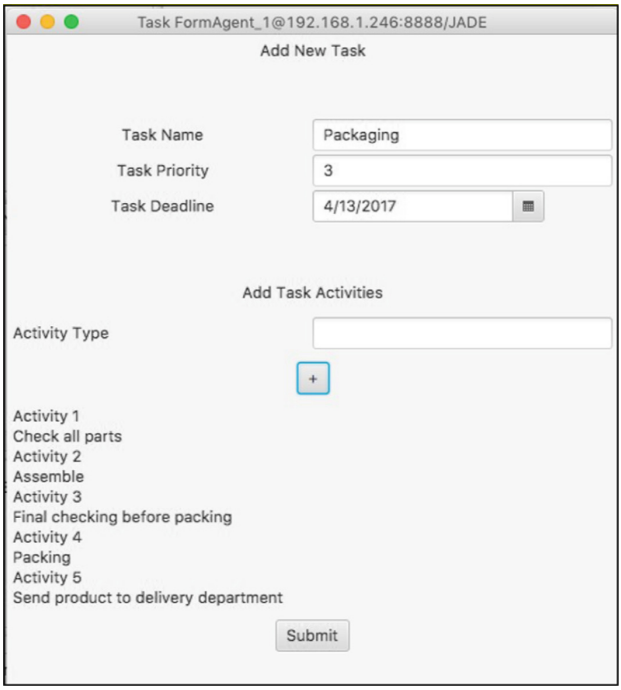

Fig. 3. An example of interface for inserting a task

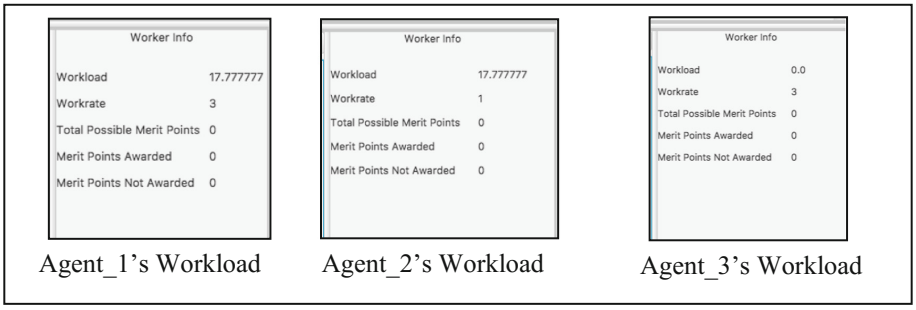

Fig. 4. Workload of agents

Then, the remaining task's activities are transferred to Agent_3. The workload of Agent_1 is reduced and the Agent_3's workload is increased. Figure 7 shows the changes to both workloads of Agent_1 and Agent_3 after the task transfer.

As mentioned earlier, the teammate agent gets 1 merit point if it completes the task on time or earlier without putting any condition and is considered as a sincere agent. In this case, Agent_3 successfully completes the task on time without putting any condition.

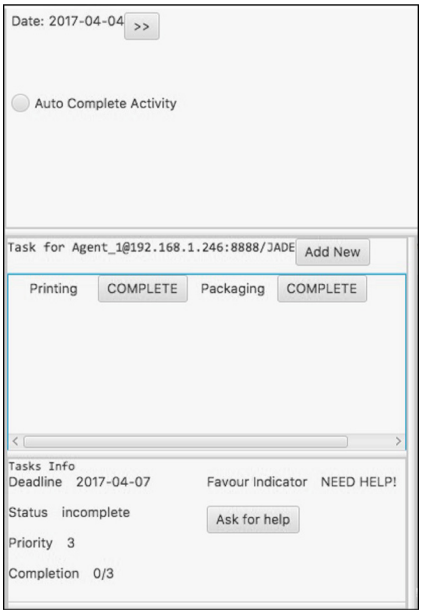

Fig. 5. Agent_1 needs help for performing its task

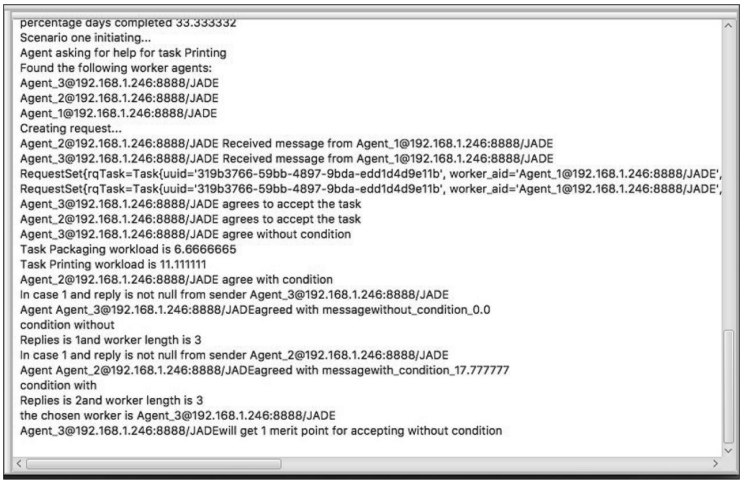

Fig. 6. The communication of agents at the time of choosing the available agent

Thus, Agent_3 gets 1 merit point because it actions shows that it is a sincere agent. Figure 8 shows the merit point earned by Agent_3.

The simulation shows that the Workload Manager works well in delegating tasks and managing help. The simulation also shows the validity of the task transfer design to streamline the imbalances of workloads among agents.

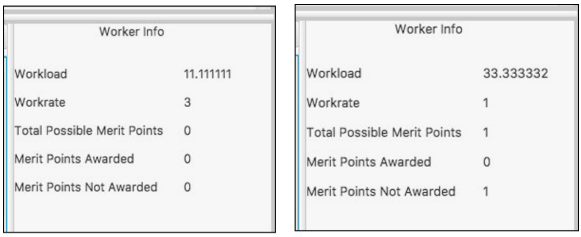

Fig. 7. The changes of workload for Agent_1 and Agent_2

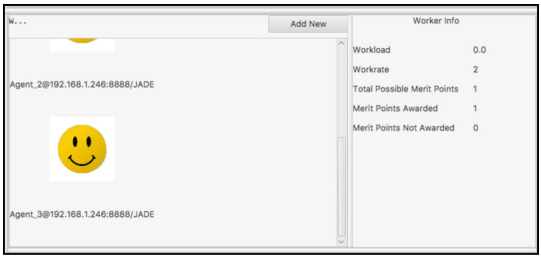

Fig. 8. The Agent_3 gets 1 merit point from the system

5 Conclusion and Future Work

Team working is one of the keys for agents in accomplishing shared objectives. It provides many benefits in supporting autonomous agents to perform tasks faster and with higher quality. However, the problem of workload imbalances among agents can be a serious one. Indirectly, it would negatively affect the accomplishment of the objectives to perform scheduled tasks. Consequently, the transfer of the remaining task's activities from one agent to another should be crucially designed, which this research has attempted to achieve.

The implementation of the Workload Manager in this process somewhat eases this problem and at the same time ensures that the agents are able to deliver the tasks' output as scheduled. The use of merit points for awarding teammate agents which help the worker agents sincerely presents a positive strategy in motivating agents to help each other in its community.

However, the enforcement of sincerity behavior in software agents should be formulated properly to ensure that it works in its environment. As a perspective for further research, we plan to integrate all formulation of instilling sincerity in software agents and simulate it to study the effectiveness of this formulation. In future, the cumulative of sincerity merit point will be used to calculate the sincerity level of agent. This sincerity level will be used to identify how the agent will react to its environment when its teammate facing problems while performing the task.

References

1. Liu, H., Shao, Z., Li, S., Tan, X.: A study of multiagent systems for resource allocation. In: Advances in Materials, Machinery, Electronics I: Proceedings of the International Conference on Advances in Materials, Machinery, Electronics (AMME 2017). American Institute of Physics, Wuhan, China (2017)
2. Eldabaa, P.N., Bahgat, A., Seddawy, E.L., Ahmed, M.H.: Adaptive framework using intelligent system to enhance total quality management in pharmaceutical industry. Int. J. Adv. Sci. Res. Manag. **2**, 1–15 (2017)
3. Malibari, A.A., Gamlo, A.H., Metro, A.E.: Application of agents for efficient energy consumption: coalition formations approach; a systematic review. Int. J. Comput. Acad. Res. **5**, 279–289 (2016)
4. Hellström, T.: On the moral responsibility of military robots. Ethics Inf. Technol. **15**(2), 99–107 (2013)
5. Wallach, W., Allen, C.: Framing robot arms control. Ethics Inf. Technol. **15**, 125–135 (2013)
6. Wooldridge, M.: An Introduction to Multiagent Systems. Wiley, United Kingdom (2009)
7. Franklin, S., Graesser, A.: Is it an agent, or just a program?: A taxonomy for autonomous agents. In: Müller, J.P., Wooldridge, M.J., Jennings, N.R. (eds.) ATAL 1996. LNCS, vol. 1193, pp. 21–35. Springer, Heidelberg (1997). doi:10.1007/BFb0013570
8. Green, S., Hurst, L., Nangle, B., Cunningham, P., Somers, F., Evans, R.: Software agents: a review (1997)
9. Jennings, N.R., Sycara, K.P., Wooldridge, M.: A roadmap of agent research and development. Auton. Agents Multi-agent Syst. **1**, 7–38 (1998)
10. Anderson, M., Anderson, S.L.: Toward ensuring ethical behavior from autonomous systems: a case-supported principle-based paradigm, pp. 19–28 (2014)
11. Golpayegani, F.: Goal-based multi-agent collaboration community formation: a conceptual model. In: 4th Workshop on Goal Reasoning at IJCAI 2016, New York, USA (2016)
12. Ahuja, P., Sethi, P., Juneja, D., Mukherjee, S.: Task allocation strategy for multi agent: a review. J. Netw. Commun. Emerg. Technol. **7**, 18–21 (2017)
13. Bradshaw, J.M., Hoffman, R.R., Johnson, M., Woods, D.D.: The seven deadly myths of "autonomous systems". IEEE Intell. Syst. **28**, 2–9 (2013)
14. Allen, C., Wallach, W., Smit, I.: Why machine ethics? IEEE Intell. Syst. **21**(4), 12–17 (2006)
15. Anderson, M., Anderson, S.L.: Robot be good. Sci. Am. **303**(4), 72–77 (2010)
16. Liu, W., Zhou, L., Xing, W., Yuan, B.: Modeling human-like autonomous behaviors and movements of virtual humans in real-time virtual environment. In: IEEE Symposium on Computer and Communication, pp. 546–549 (2010)
17. Signoretti, A., Feitosa, A., Campos, A.M., Canuto, A.M., Xavier-Junior, J.C., Fialho, S.V.: Using an affective attention focus for improving the reasoning process and behavior of intelligent agents. In: 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, pp. 97–100. IEEE (2011)
18. Jaafar, N.H., Ahmad, M.S., Ahmad, A.: The rationality of sincere software agent in task completion. In: The International Symposium on Agents, Multi-agent Systems and Robotics 2016 (ISAMSR 2016), Selangor, Malaysia (2016)
19. Jaafar, N.H., Basir, N.M., Ahmad, M.S., Ahmad, A.: Human sincerity factors for adaptation in software agents. In: 2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014). IEEE, Penang, Malaysia (2014)
20. Jaafar, N.H., Ahmad, M.S., Ahmad, A.: The influence of human blaming or bragging behaviour towards software agent sincerity implementation. In: Omatu, S. (ed.) Distributed Computing and Artificial Intelligence, 13th International Conference. AISC, vol. 474, pp. 239–246. Springer, Cham (2016). doi:10.1007/978-3-319-40162-1_26