Negotiation Protocol in Traffic Routing for Priority Vehicle

Krishna Priawan Hardinda, Yuichi Sei, Yasuyuki Tahara, Akihiko Ohsuga

自己適応システムは，ソフトウェアの複雑さを管理するアプローチの 1 つとして提案されている．経路作成問題は，マルチエージェントシステムとしてモデル化できる複雑なソフトウェアの 1 例である．経路作成問題は，多くの場合，不確実性に伴う課題を有する．道路の閉鎖，システムの障害，および目的地の変更は，そのような課題の例である．本論文では，優先車両に関連する経路作成問題で発生した課題を提示する．そのためにまず，実行時の不確実性を表すシナリオにおいて課題を定義する．次に，実験中に発生した問題に対処するためのネゴシエーションプロトコルを提案する．都市部の現実的な交通状況を描写した交通シミュレーターを使用して実験を行い，適応のための計画を交通管理システムが実施することにより適応を実行する．

The self-adaptive systems have been proposed by researchers as one of the approaches to manage the complexity of the software. The traffic routing is an example of complex software which can be modeled as a multi-agent system. The traffic routing often raises several problems that require a solution to tackle the uncertainties. Road closure, failures in systems, and destination changes are a few illustrations for problems in traffic routing. In this paper, we present the problems occurred in traffic routing related to priority vehicle. We define the problem with a number of scenarios which express the uncertainties during the runtime. Then, we propose our negotiation protocol to address the issues occurred during the experiments. We conduct the experiments by using traffic simulator that portrays realistic traffic situation in the urban area, then apply the adaptation plan to its traffic controller to adapt the adaptation plan.

1 Introduction

The complexity of current software systems has brought a challenge for the researcher to be able to manage the software in a proper way. There are several existing approaches to tackle these problems, one of those is a self- adaptive system. Self-adaptive systems can manage the complexity by building more self- managed systems that transforming complexity from users to the software itself. Furthermore, self-adaptive systems have an ability to organize themselves with specified goals in advance without conscious awareness from the users [1, 2].

A self-adaptive system also comprises of several capabilities that support the system to achieve the goals. Those capabilities are often called as self-* properties. The following properties are self-configuration (automatically reconfiguring itself when components are changed); self-optimization (constantly tuning the parameter for optimization); self-healing (recover from the failures); and self-protection (protect from external attacks) [1].

Considering the capabilities of the self- adaptive systems, it is believed that the self- adaptive systems can do automation to reconfiguring itself in response to changes in the environment during the

優先車両の経路作成における交渉プロトコル

Krishna Priawan Hardinda, Yuichi Sei, Yasuyuki Tahara, Akihiko Ohsuga, 電気通信大学大学院情報理工学研究科, Graduate School of Informatics and Engineering, The University of Electro-Communications.

runtime [2]. The core concept of self-adaptive systems relies on adaptation logic that implements a feedback structure: MAPE (Monitoring Analyzing Planning Execute) model.

One of the real problems that depict the complexity of software is traffic systems. The implementation of self-adaptive systems is applicable to this problem since traffic systems contain numbers of large agents, limited information and large numbers of uncertainties [4]. Specifically, the problem that is used in [4], Automated Traffic Routing Problem (ATRP), has demonstrated the needs of self- adaptive approach to tackle the problem. Then with their proposed approach, the problem was solved with a solution and later its solution was evaluated along with a set of comparable dimensions.

However, the scenarios in ATRP related to priority vehicles such as ambulance remain. To the best of our knowledge, existing research papers that address this issue are limited. Specifically, we are facing the lack of techniques that has the incorporation between the self-adaptive approach with the ATRP in the priority vehicle problem. Although several types of research have been conducted related to ambulance routing problem [3], most of them are the focus on hardware and network layer.

In the previous research [4], the priority vehicles are not considered as the problem in the simulation. The vehicles on simulation are not guaranteed to arrive at the destination during the simulation. There are possibilities that those vehicles halt at the road due to the situation on the road such as road closure. The main challenge of this problem is that we aim to maintain both the objective of completing the mission of priority vehicles as well as optimizing the traffic of other vehicles.

In this paper, we address a number of problems. Traffic congestion is one of the problems in the urban area. The congestion often hinders the priority vehicle to travel in the area. The emerging area of smart cyber- physical systems requires a decentralized approach to handle uncertainties between independent system agent. We focus on specific priority vehicle, the ambulance since it carried a critical mission and required fast response. In addition, the ambulance has multiple goals to be satisfied that often has a conflict between other agents (other vehicles).

To tackle those issues, we propose an adaptation policy in our approach. Our approach comprises of three policies: removing obstacles, overriding traffic controller, and information sharing. Each policy is incorporated with MAPE feedback loop [2] such that actualize the concept of self- adaptive in the approach.

The purpose of this paper can be elaborated into at least three points. Achieving the main goal of the ambulance: arrive at the target location of the patient and then travel to the hospital as soon as possible. Moreover, maintain the flow of other traffic in the whole system, then manage and handle the uncertainties which might occur during the runtime. Lastly, define the set of the adaptation plan to handle uncertainties.

The rest of this paper is structured as follows. Section 2 highlights the motivating problem for traffic routing with priority vehicle. Section 3 features our approach to deal with the problem and its implementation on the simulation, Section 4 highlights our experiment with the number of scenarios. Section 5 presents our results and discussion regarding the experiment that we have been conducted. Then, Section 6 compares our work to prior research. Finally, Section 7 concludes our contribution.

2 Motivating Problem

Traffic routing arises several problems in the urban area. The vehicle must travel to one point to another point by going through intersections and traffic lights. Since the traffic congestion is inevitable events in this traffic routing, the vehicle must find a solution to minimize the impact of this traffic congestion to their traveling time. Initially, the vehicle computes the estimated time to reach the destination by calculating the distance from the initial position to the destination. However, the congestion might occur during the journey of the vehicle. In response to that situation, the vehicle must recompute in every single time the vehicle facing this situation.

The cause of traffic jam can be varied from several reasons, such as a high volume of vehicles, road closures due to constructions, and traffic light malfunction. The implementation of an agent that monitors the causes of traffic jam has indicated a satisfactory approach to overcome the situation [4]. Nevertheless, the action that handles specific priority vehicle problem might require a different approach.

In our paper, we are addressing the issue when the simulator called ADASIM [4] was failed to make the priority vehicle or ambulance to arrive at the destination with speedy movement. The inability to achieve this goal has raised the problem for an ambulance to fulfill the initial mission of the ambulance. Since the ambulance carries a critical mission for saving a life, it is mandatory to ensure that the ambulance arrives at the destination on time. In addition to that, the level of uncertainties in the simulation impacts the traveling time of the ambulance to the destination. In such a situation, the vehicle and other agents (traffic agent and road closure agent) only have partial knowledge. The incapability of grasping the knowledge are drawbacks to satisfy the initial goals. In contrast, they are expected to have an accurate decision-making mechanism to achieve their goals.

Furthermore, the other issue is that the capability of sharing information between the ambulances. We think this is important because it will support the ambulance when it has a force majeure such as natural disaster or failure of an engine in the car. In such of scenario, the ambulance is not able to find a path to a destination then the other ambulance will be dispatched to cover for that unfinished mission. In [4], the simulator already implemented the abstract definition of privacy filter. Thus, the routing problem related to priority vehicle remain unsolved and then the implementation of privacy filter function is necessary to tackle this issue.

3 Approach

The motivating problem we illustrated in Section 2 has encouraged us to propose a novel approach to this problem. The problem in traffic routing related to a priority vehicle that we want to tackle is the lack of an adaptation strategy corresponding to the priority vehicle in [4]. Here, we defined several adaptation plans and strategies then incorporated with the ADASIM simulator to support the routing for priority vehicle. Thus, our approach on this paper is Priority Routing Algorithm with predetermined adaption plan and strategy.

3.1 Removing obstacles

road. This action can be achieved by unblocking the road that has been closed due to constructions or accidents. In the case of the accidents, the priority vehicles become the priority to pass this road to handle this situation. An action can be achieved by monitoring the closed road during this cycle, then the Road Closure Agent will open the road if the ambulance needs to pass this road.

Monitor: roadblock due to construction or accidents are being monitored frequently.

Analysis: the impact of unblocking the road are calculated to give the effect of enacting this action.

Planning: finding the shortest path to the destination if the road is unblocked.

Execution: recompute and update the estimated time arrival at the destination.

3. 2 Overriding traffic control

Overriding traffic control is a policy to reduce traffic road on a specific segment of the road which is passed by the ambulance. There are two options that we apply for this simulation. First, we create a counter-flow of the traffic so that the ambulance will have easy access to pass the road. Second, the traffic light turns into gree immediately after passing through the intersection. The Traffic Agent then handle this action in the simulation.

Monitor: the current location of the ambulance, the severity of traffic congestion nearby the intersection and the number of other vehicles are supervised. Those factors have a significant impact on congestion.

Analysis: After the ambulance passing the traffic light, we measure the consequences if other vehicles nearby the ambulance are stopped during this action.

Planning: considering the duration of traffic light turns into green when the ambulance is passing by and others vehicle turns into red.

Execution: changing the traffic light into green and red at the same time for different sides of intersections

3. 3 Information sharing policy

Information sharing between ambulances and hospitals. This policy controls the information between agents and their negotiation protocol. Vehicle Agent operates this action by selecting which vehicle is categorized as a priority vehicle and then applying the policy to the selected vehicles. In this process, we are carefully implementing this method since the existing simulator already specified the data privacy policy on its agent. This situation was considered to prevent improper use of the data. For example, the information about its current location of the vehicles might not be shared to all vehicles and it is only limited to Traffic Agent.

Monitor: the indicator of engine failure in the ambulance and the emergency status of the city are being watched.

Analysis: we analyze the effectiveness if the destination of the ambulance is changed or the mission of ambulance is being taken over by other ambulance.

Planning: a new destination and a new ambulance are set.

Execution: recompute the travel route to the new destination.

At the implementation level, removing the obstacle and overriding the traffic control are incorporated in PriorityRoutingAlgorithm class. The base class implementing a modified version of Dijkstra's algorithm. This class comes with two constructors. The zero- argument constructor (used when this class is instantiated from a configuration file) creates an instance of the strategy that will compute shortest paths based on node weights alone and will never recompute a path. The second constructor takes two arguments: lookahead and recompute. These two parameters determine how far away from its current node a car can perceive traffic, and how often to recompute the path, respectively. Each of the policy was referred to as a function in the PriorityRouting class.

Information sharing policy is installed into each of priority vehicle. In Vehicle class, we added a new criterion for privacy filter which filters only for priority vehicle. After filtering the eligible criterion for priority vehicle, we specify which information should be shared between this priority vehicle. This function then has to store the information and share to other priority vehicle agents. This filter is applied when the priority vehicle is facing an unexpected situation which might lead to an unsuccessful mission of the ambulance. For example, an ambulance is failed to continue the journey due to engine failure and the only way to transport this is to transfer the mission to the other ambulances.

4 Related Work

A recent survey points out that self-adaptive software systems have been proposed in some related research. Research has discussed the strategic layer for selecting the routes and adaptive learning behavior in a multi-agent framework based on JADE [5]. This JADE framework interacted with traffic simulator called SUMO that supplies the tactic layer of simulation: vehicle changing the lane, breaking, and accelerating. On the other hand, this paper focus on finding the best route to reach the destination while monitoring the current situation of the traffic. This action is necessary to maintain the primary goal of the priority vehicles as well as other vehicles in the traffic system. Further, the adaptive learning process in [5] might lead to another computation process hence require extra time and resource consumption.

In addition, a Belief, Desire, and Intention (BDI) agent have been integrated with MATSim for creating an agent that has a reactive response to the surrounding environment [6]. The self-adaptation approach used in [6] has developed the static adaptation plan in advance to tackle its problem scenario. In contrast to our paper, we present the number of adaptation plan that has the capability to change dynamically during the runtime.

Several adaptation approaches on dependent networks also have been introduced [7], [8]. That paper mainly features the integration between SFINA framework with domain- specific modeling language such as MATSim to mitigate the failures in the networks. We instead focus on the self-adaptation approach in a multi-agent system with decentralized optimization. Moreover, an adaptive traffic management plan has been proposed in [9]. That paper features an adaptive plan to ensure the provision of secure and efficient emergency services in smart cities. In this work, some components in traffic management system such as traffic management controllers (TMC), adaptive traffic light controller, and sensor controller are introduced to collect the information about the severity of the disaster. In case of the emergency condition occurred, the TMC has a responsibility to control the traffic flow for the emergency service vehicle. In contrast to this paper, we simplified the scenario by defining the uncertainties with the probability factor. The goal of this is to create a decentralized approach in a self-adaptive system.

Eventually, the ADASIM simulator [4] itself has presented novel benchmark tools with a set of dimensions along which solutions should be evaluated. The related challenges such as scalability, robustness, and balancing the precision and performance of monitoring has been highlighted. In contrast to our paper, we focus on specific traffic routing problem related to priority vehicle which has not been addressed by prior research. Regarding the results, we have improved the mean travel time for the priority vehicles.

5 Conclusion

This paper has proposed a Priority Routing algorithm with a set of adaptation plans for traffic routing problem related to priority vehicle. We have successfully reduced the average travel time for priority vehicle. We believe that the adaptation plans we have implemented are suitable for such a scenario.

For future work, we will investigate more accurate travel time for an ambulance in the urban area. We plan to create a more realistic simulation of traffic that is able to model the traffic flow in the urban area as well as traffic congestion. We also interested to apply the machine learning technique to make a finer adaptive learning technique to be incorporated in the simulation process.

References

[1] De Lemos, Rogério, et al. "Software engineering for self-adaptive systems: A second research roadmap." Software Engineering for Self-Adaptive Systems II. Springer, Berlin, Heidelberg, 2013. 1-32.

[2] J. O. Kephart and D. M. Chess. The vision of autonomic computing. IEEE Computer Society Press, 36:41–50, 2003.

[3] Derekenaris, Grigoris, et al. "Integrating GIS, GPS and GSM technologies for the effective management of ambulances." Computers, Environment and Urban Systems 25.3 (2001): 267-278.

[4] Wuttke, J., Brun, Y., Gorla, A., & Ramaswamy, J. (2012). Traffic routing for evaluating self-adaptation. ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, 27–32. https://doi.org/10.1109/SEAMS.2012.6224388

[5] G. Soares, Z. Kokkinogenis, J. L. Macedo, and R. J. Rossetti, " Agent-based traffic simulation using sumo and jade: an integrated platform for artificial transportation systems, " in Simulation of Urban MObility User Conference. Springer, 2013, pp. 44–61.

[6] L. Padgham, K. Nagel, D. Singh, and Q. Chen, "Integrating BDI agents into a Matsim simulation," in Proceedings of the Twenty-first European Conference on Artificial Intelligence. IOS Press, 2014, pp. 681–686.

[7] E. Pournaras, B.-E. Brandt, M. Thapa, D. Acharya, J. Espejo-Uribe, M. Ballandies, and D. Helbing, " SFINA-simulation framework for intelligent network adaptations, " Simulation Modelling Practice and Theory, vol. 72, pp. 34–50, 2017.

[8] E. Pournaras, M. Ballandies, D. Acharya, M. Thapa, and B.-E. Brandt, " Prototyping self-managed interdependent networks–self-healing synergies against cascading failures, " in 13th International Symposium on Software Engineering for Adaptive and Self-managing Systems (SEAMS 2018), 2018

[9] Djahel, Soufiene, et al. "Adaptive traffic management for secure and efficient emergency services in smart cities." 2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops). IEEE, 2013.