Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Development of a Case-Rule-Based Reasoning Engine for the Double Agent Convoying Simulation

Makoto Katoh
Department of Mechanical Engineering,
Faculty of Engineering
Osaka Institute of Technology, OIT
Osaka, Japan
katoh@med.oit.ac.jp

Natsuki Imura
Department of Mechanical Engineering,
Faculty of Engineering
Graduate School of Osaka Institute of Technology, OIT
Osaka, Japan

***Abstract*— The present paper describes the simulation of a new intelligent transport system using a newly developed simple two-cycle case-rule-based reasoning engine (CRBRE) for double agent convoying. The CRBRE is driven by many multi-triggers (mm-triggers) and reasons through action allocation. A state transition graph-like double ring is used for the scenario sequence of double agent convoying. Storyboarding and operation diagram analysis (ODA) are performed for redevelopment.**

***Keywords*— case-rule-based reasoning, artificial intelligence, agent, convoying, state transition graph, operation diagram analysis**

I. INTRODUCTION

A. Background and Purpose

The present study is based on our previous study on the design of a case-rule-based reasoning engine (CRBRE) for the double-agent convoying simulation stage [1]. In this development stage, a comprehensible and easy-to-use framework with which to execute a scenario sequence arranged around many multi-triggers (mm-triggers) was necessary for our application. In the present study, we describe the development of a new arrangement for a scenario sequence simulation using a newly developed CRBRE for double-agent convoying with stations.

B. Comparison with Previous Studies

In this section, newly developed CRBRE is compared with previous research in this area [1]:

1) State Transition Network (STN) [2][3]
A planning agent with an STN connected to a situation monitor and an action planner in a cruising planner for autonomous highway vehicle (AHV) selects a suitable driving program based on driving conditions after making the decision to follow or overtake a vehicle.

2) Multi-objective Behavior Coordination [4][5]
Cancellation of behavior competition by an agent or multi-agent in higher-priority behaviors or agents.

3) Perceiving the Acting Cycle [4]
Two-cycle perception from the environment and action to one system is adapted for intelligent behavior through the coordination of robots. An action system executes multi-objective behavior coordination based on fuzzy rules for a logical inference. Modular neural networks for realizing the action control based on the cycle have been applied.

4) Asynchronous Production System (APS) [6]
An APS has a three-cycle inference engine that consists of making a match with an input module, and selecting and executing the output module. These functions match the changes in an external input and the changes in working memory, while selecting the rules for the resolution of changes in a conflict data set and executing the selected rules and I/O operations and external procedures. The APS has a production memory for the storage of rule sets, and a working memory for the storage of active data sets.

5) SIAM and MAMA [7]
Case based-reasoning (CBR) is a four-cycle engine that consists of four steps: retrieve, reuse, revise, and retain.

SIAM is a global case-based reasoning framework that consists of four steps: setup, initialization, application, and maintenance.

MAMA stands for the maintenance management manual for case-based reasoning. The role of MAMA is to direct the planning and execution of maintenance operations using event-condition-action (ECA) rules, where "event" refers to the SIAM context, "condition" refers to the review step, and "action" refers to the restore step.

6) Case-based Reasoning (CBR) with GP (Genetic Programming) in a real-time system [8]
Genetic programming is used in an evolutionary stage based on abstract behavior by simulation, and CBR is applied in the adaptive stage to a real-world environment by means of experiments for a humanoid robot.

7) Ensemble Case-based Reasoning
Collaboration Policies for Multi-agent Cooperative CBR [9]
This is an important concept in convoying research.

Generally speaking, the rule-base system is comprehensive, but lacks the ability to arrange many multi-triggers. The CBR system has excellent classification and arrangement ability, but it is not good at movement through the scenario sequence, although it is good at reasoning from many cases inductively.

C. *Development Concepts*

Simulation development should be performed according to the following multi-stage procedure (left-hand side of Fig. 1) based on the snowman hypothesis. That is, development increases the number of neighbor functions after the simplest winner [10] kernel has been developed.

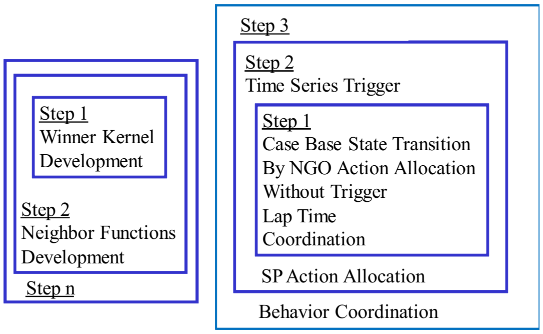

Figure 1. Simulation development procedure

Fig. 2 shows the two-cycle CRBRE with the functions listed on the right in Fig. 1. Unnecessary steps were eliminated by prototyping.

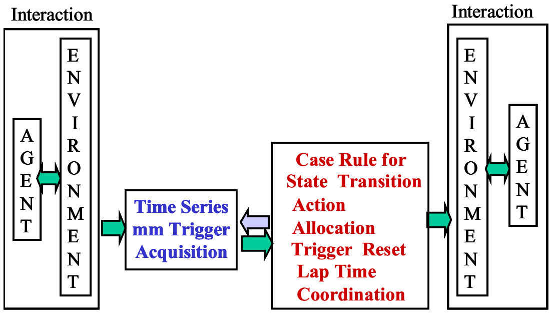

Figure 2. Two-cycle CRBRE system configuration

First, time series mm-triggers are acquisitioned from agents and stations. The time series mm-triggers are then referred from 5*2 stations with two case bases as each agent. Then, reasoning is performed using simple if_then rules for various action allocations and behavior coordination. State transition is executed by next go (NGO) action allocation with destination state check and destination state change. Finally, pass, stop, and wait (WSP) actions are executed. Computation and display of lap time, coordinates and distance are executed in station cases.

The simple structure allows the two-cycle engine to be easily constructed and minimizes mistakes.

II. CONVOYING EXAMPLE USING THE CRBRE

A. *First Step of Development*

Fig. 3 shows a convoying example for which the winner kernel of CRBRE was developed.

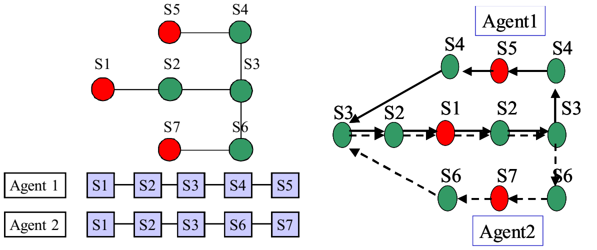

Figure 3. Station arrangement, operation order, and simple transition graph of a double agent

First, seven stations are arranged in the environment as shown in the left-hand side of Fig. 3. Double agents are then operated repeatedly in order as shown at bottom-left in Fig. 3. Second, a state transition graph (STG) of double agents that covers all transitions is shown on the left-hand side of Fig. 3 [1]. This STG-like double ring with a common route is shown for the case of double agent convoying with bifurcated and joined scenarios, although, this STG-like double ring was proposed for a non-bifurcated scenario of double deck elevators in a previous study [3]. The road was expanded according to the order of advance of each agent using a graph having nodes (station) and directed arrows (direction of advance of the agents). Here, the route of Agent 1 is shown by the solid line, and the route of Agent 2 is shown by the broken line.

All sections are assumed to have one lane, so that outstripping is prohibited. Then, the transport section from S1 to S3 is the convoying section.

In the first step, the next station to which the agent should move should be inferred repeatedly after checking the state of the destination and the arrival station using the case rule base shown in the 'next go (NGO)' action allocation table, as follows.

Only NGO action allocation without a trigger or action, as shown in Table 1, is applied for state transition of both agents in each station case. The agents are changed by checking the state of the destination, and the state of the destination is also changed at the end terminal station and the start terminal station.

TABLE 1 NGO ACTION ALLOCATION TABLE WITHOUT A TRIGGER AND ACTION

Agent 1	**S1**	**S2**	**S3**	**S4**	**S5**
Destination S5	**S2**	**S3**	**S4**	**S5**	**Des. S1**
Destination S1	**Des.S5**	**S1**	**S2**	**S3**	**S4**
Agent 2	**S1**	**S2**	**S3**	**S6**	**S7**
Destination S7	**S2**	**S3**	**S6**	**S7**	**Des. S1**
Destination S1	**Des.S7**	**S1**	**S2**	**S3**	**S6**

B. Results of First Step of Development

We propose a storyboard [11] to describe the results of a scenario simulation for event analysis. In this convoying case, the storyboard in the operation scenario bifurcates when the double agents are parallel outbound [12] at Station 3 and joins when the double agents are parallel inbound at Station 3, as shown in Fig. 4.

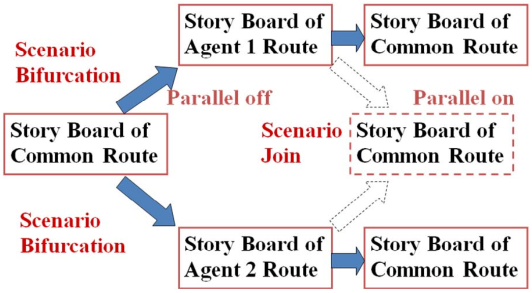

Figure 4. Bifurcation and join configuration of a storyboard

The convoying drive section, which is positioned after joining and before bifurcation of the storyboard, involves the danger of a rear-end collision.

An example of a storyboard that is bifurcated outbound at Station 3 and joined inbound at Station 3 is shown in the first step of this development. The lap time (min) from the start station to the passed station, the action allocation of each agent from current station to next station, and the coordinates of the current station computed using the distance from the start station to the current station are included.

Here, it is assumed that the speed of both agents is 60 (km/h), and Agent 2 starts 1/60 (min) later than Agent 1. That is, the following distance is 16.7 (m). In order to demonstrate the sustainability and accuracy of this engine, nine out of 18 scenes for one round trip, the lap time of which is 72.0 (min), are shown in Table 2. We count the scenes as passing the station in this traffic scenario sequence. The arrows in the action allocation columns denote direction from the current station to next station.

TABLE 2 EXAMPLE OF STORYBOARD BIFURCATED AND JOINED SCENARIOS

Lap Time	Agent1 Allocation	Coordination	Lap Time	Agent2 Allocation	Coordination
0.0	S1->S2	(4.1,9.9)	0.0167	S1->S2	(4.1,9.9)
4.0	S2->S3	(8.1,9.9)	4.0167	S2->S3	(8.1,9.9)
8.0	S3->S4	(12.1,9.9)	8.0167	S3->S6	(12.1,9.9)
13.0	S4->S5	(12.1,14.9)	13.0167	S6->S7	(12.1,4.9)
17.0	S5->S4	(8.1,14.9)	17.0167	S7->S6	(8.1,4.9)
21.0	S4->S3	(12.1,14.9)	21.0167	S6->S3	(12.1,4.9)
26.0	S3->S2	(12.1,9.9)	26.0167	S3->S2	(12.1,9.9)
30.0	S2->S1	(8.1,9.9)	30.0167	S2->S1	(8.1,9.9)
34.0	S1->S2	(4.1,9.9)	34.0167	S1->S2	(4.1,9.9)

C. Second Step of Development

In the second step, the inference from the trigger obtained from each station and the agent, whether stopping at the arrival station or passing the arrival station using the case rule base shown in the following stop or pass (SP) action allocation table is added.

Table 3 shows the SP action allocation for the case of only Agent 2, because Agent 1 has stopped at Station 1 as the result of a break down. The preparation for showing them at the same time was not completed in this step. To simplify the rules, all triggers are assumed to indicate stopping at a station and not only at the destination. A useless stop is generated at least in the convoy section only by the instruction method as the hall calling of a usual elevator.

TABLE 3 SP ACTION ALLOCATION TABLE OF AGENT 2 FOR TRIGGERS FROM STATIONS AND AGENT 2

Agent 2 (Des.S7)	S1	S2	S3	S6	S7
Laptime1=0	Stop	Pass	Pass	Pass	Stop
Laptime1=7 STOPS6inCAR=1	Stop	Pass	Pass	Stop	Stop
Laptime1=14 STOPS3=1	Stop	Pass	Stop	Stop	Stop

D. Results of Second Step of Development

An example of a storyboard is presented in the second step of this development. The lap time (min) includes the stopping time at the station at which the agent has stopped. The storyboard shows the agent action reasoned from the above case rule for the arrival station.

The stopping time upon reaching the station is assumed to be 0.4 (min). In order to demonstrate the sustainability and accuracy of this step, nine out of 18 scenes are shown in Table 4, where the lap time is 74.4 (min). The reasoning engine accurately executes the SP action according to the above action allocation case rules.

TABLE 4 EXAMPLE OF A STORYBOARD WITH THE ACTION OF AGENT 2 FOR TRIGGERS FROM STATIONS AND AGENT 2

Lap Time	Agent2 Allocation	Agent2 Action	Coordination
0.4167	S1->S2	Stop_S1	(4.1,9.9)
4.4167	S2->S3	Pass_S2	(8.1,9.9)
8.4167	S3->S6	Stop_S3	(12.1,9.9)
13.8167	S6->S7	Stop_S6	(12.1,4.9)
18.2167	S7->S6	Stop_S7	(8.1,4.9)
22.2167	S6->S3	Pass_S6	(12.1,4.9)
27.2167	S3->S2	Pass_S3	(12.1,9.9)
31.2167	S2->S1	Pass_S2	(8.1,9.9)
35.6167	S1->S2	Stop_S1	(4.1,9.9)

E. Third Step of Development

The system development shown in Fig. 5 is performed by installing a waiting place in front of the station, at which agents compete to stop when two agents are directed toward the same direction. Simple behavior coordination to avoid a collision is performed, where the leader agent stops at the station and the follower waits at the waiting position when both agents are called from the station. Then, the action of the follower agent changes from stop to pass the station. The full car passing, for example of passenger cars, is not considered.

The action allocation of this step is 'wait, stop, and pass' (WSP). The waiting action in front of a competing station for a follower agent is used for behavior coordination. The waiting action was processed as the same scene in the same case in order to simplify the program structure.

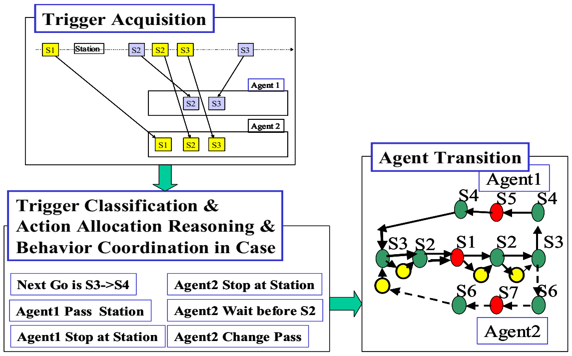

Figure 5. System configuration of Step 3

If the agent stops at a station, all action allocations are reset upon stopping.

Here, a simple behavior coordination based on the 'stop priority' between the two agents is used for the station trigger. That is, in the case of behavior competition between the leader agent and the follower agent when stopping at S2, cancellation of competition is performed based on priority coordination. Then, the behavior of the follower agent for Station 2 changes from stopping to passing.

Table 5 shows behavior coordination and the actions of both agents when Agents 1 and 2 are called from a competing station S2 (or S3) and are coordinated based on order priority. The priority of the leader agent (Agent 1) is assumed to higher than that of follower agent (Agent 2).

When the action of Agent 1 (leader) is stop at S2 (or S3), the stop trigger for Agent 2 is reset, and the action of Agent 2 (follower) becomes wait to avoid collision with the leader by using not only the conditioned sequence but also the time limit sequence at the waiting place in front of the station. Then, Agent 2 passes the station. Since the condition logic does not prevent mistakes, schedule management by time is important.

When the action of Agent 1 is pass S2, the action of Agent 2 is stop at the station rather than wait at the waiting place. However, Agent 2 might wait at the waiting place for the sake of safety in the case of unmanned operation in which Agent 2 can not visually confirm being passed by Agent 1. When the limit switch, which informs the follower that the leader has stopped, is broken, a high-speed rear-end collision occurs at the station.

TABLE 5 BEHAVIOR COORDINATION OF THE DOUBLE AGENT AT COMPETING STATIONS

Call both Agent 1 and Agent 2 from Competing Station	Station S2 or S3	Station S2 or S3
Agent 1 (leader) Action Trigger for Agent 2	Stop &Trigger Reset	Pass
Agent 2 (follower) Action	Time-Cond Wait &Pass	Stop

Table 6 shows an example of the storyboard with this security precaution. To draw an operation diagram, stop actions for the stations were divided into two parts, that is, arrival and departure.

This example shows that the follower stopped at the waiting place once when the leader is not only stopping but also passing S2 and S3, which are competing stations.

TABLE 6 EXAMPLE OF A STORYBOARD FOR BOTH AGENTS WITH A WAITING PLACE BEFORE THE COMPETING STATION

Agent1 (Leader)

Laptime	Distance	NGO Alloation	SP Allocation	Action	Coordinates
0.4	0	S1->S2	Stop_S1	Dep_S1	(4.1,9.9)
4.6	4	S2->S3	Stop_S2	Arriv_S2	(8.1,9.9)
4.8	4	S2->S3	Stop_S2	Dep_S2	(8.1,9.9)
9	8	S3->S4	Stop_S3	Arriv_S3	(12.1,9.9)
9.2	8	S3->S4	Stop_S3	Dep_S3	(12.1,9.9)
14.4	13	S4->S5	Stop_S4	Arriv_S4	(12.1,14.9)
14.6	13	S4->S5	Stop_S4	Dep_S4	(12.1,14.9)
18.8	17	S5->S4	Stop_S5	Arriv_S5	(8.1,14.9)
19	17	S5->S4	Stop_S5	Dep_S5	(8.1,14.9)
23.2	13	S4->S3	Pass_S4	Pass_S4	(12.1,14.9)
28.4	8	S3->S2	Pass_S3	Pass_S3	(12.1,9.9)
32.6	4	S2->S1	Pass_S2	Pass_S2	(8.1,9.9)
36.8	0	S1->S2	Stop_S1	Arriv_S1	(4.1,9.9)

Agent2 (Follower)

Laptime	Distance	NGO Alloation	SP Allocation	Action	Coordinates
0.4167	0	S1->S2	Stop_S1	Dep_S1	(4.1,9.9)
4.6167	3.8	S1->S2	Stop_S2	Arriv_PS2	
4.8167	3.8	S1->S2	Stop_S2	Dep_PS2	
5.016	4	S2->S3	Stop_S2	Pass_S2	(8.1,9.9)
9.2167	7.8	S2->S3	Stop_S3	Arriv_PS3	
9.4167	7.8	S2->S3	Stop_S3	Dep_PS3	
9.6167	8	S3->S6	Stop_S3	Pass_S3	(12.1,9.9)
14.8167	13	S6->S7	Stop_S6	Arriv_S6	(12.1,4.9)
15.0167	13	S6->S7	Stop_S6	Dep_S6	(12.1,4.9)
19.2167	17	S7->S6	Stop_S7	Arriv_S7	(8.1,4.9)
19.4167	17	S7->S6	Stop_S7	Dep_S7	(8.1,4.9)
23.6167	13	S6->S3	Pass_S6	Pass_S6	(12.1,4.9)
28.8167	8.2	S6->S3	Pass_S3	Arriv_PS3	
29.0167	8.2	S6->S3	Pass_S3	Dep_PS3	
29.2167	8	S3->S2	Pass_S3	Pass_S3	(12.1,9.9)
33.4167	4.2	S3->S2	Pass_S2	Arriv_PS2	
33.6167	4.2	S3->S2	Pass_S2	Dep_PS2	
33.8167	4	S2->S1	Pass_S2	Pass_S2	(8.1,9.9)
38.0167	0	S1->S2	Stop_S1	Stop_S1	(4.1,9.9)

yellow is estimation

Here, the distances from the start point were estimated because the distance computation was not prepared here.

The operation diagram has been used for traffic scheduling, and, more recently, for internal traffic control planning [13]. In the present study, we perform an operation diagram analysis (ODA) of the above storyboard in order to confirm that the agents do not collide at the convoying sections between the first three stations and between the last three stations, as follows.

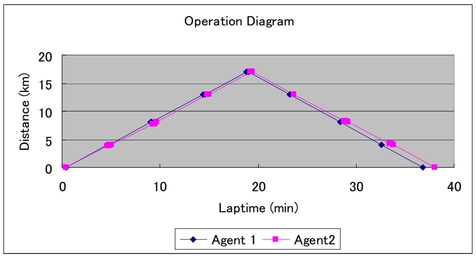

Figure 6. Operation diagram analysis on a storyboard

The distance between the agents increases in last three stations because there are waiting of follower agent in front of competing stations at passing of leader agent.

III. CONCLUSION

Three-step prototype development of the convoy simulation stage using two-cycle CRBRE was described. Since STG could be modeled as a winner function which became a kernel of total system in step 1, a development method like self-organization which added various neighbor functions in steps 2 and 3 could be adopted. As a result, the method of developing various methods in parallel could not be applied to this system.

The design stage of the three-cycle CRBRE was judged to be labor intensive, and the advantage was comparatively small. Therefore, we developed only a two-cycle CRBRE. The main difference between this two-cycle system and previous systems is that, in the two-cycle system, the action part is case-rule-based so that the rules can be arranged easily for the multi-agent, multi-object (stations) system.

Then, the case base used here was double, and the time series triggers were classified directly in the case base. The reasoning using if_then rules was adopted based on NGO action allocation, SP action allocation, and behavior coordination of double agents at competing stations.

An STG-like double ring with a common route was shown for the case of double agent convoying with bifurcated and joined scenarios.

The storyboard used in the present study had columns indicating lap time, distance, NGO allocation, SP allocation, action, and coordinates. It is important that an allocation of the action and an actual action are necessary to think as another things. In the present study, the storyboard was used as a program checking tool.

The operation diagram analysis, in addition to the storyboard analysis, which was also used as a simulation program check method, was demonstrated. However, the former must be used as an internal traffic control planning tool, as in the problem considered herein, and the latter must be used as an accident analysis tool.

The difficulty in this development is applying correct logic in complex systems and the time limit sequence. And time management is important.

ACKNOWLEDGEMENTS

The authors would like to thank Mr. Ito, Mr. Tokunaga, and Ms. Uetani, who researched STG and the degree of comfort of elevators. Special thanks are due to Ms. Li and Mr. Wang, investigated the queuing of elevators and CBR.

Special thanks are due to Mr. Takarabe (formerly of Otis Elevator Company), and to Mitsubishi Electric Co. Ltd. for providing a tour of their elevator division.

REFERENCES

[1] M. Katoh, “Design of CRBRE for Double Agent Convoying Simulation Stage”, SICE 2009, (in Press), 2009.

[2] K. Fujita, Y. Kagawa, L. Satonobu, and A. Okuno, “A knowledge-based planner for autonomous highway cruising”, Proceedings of the 4th annual conference of artificial intelligence society, pp. 579-582, 1990 (in Japanese).

[3] K. Itoh, M. Katoh, Y. Tokunaga, and X. Li, “A Stochastic Simulation of Sequence Control based on State Transition Graph for Double Deck Elevators”, Transactions of JSME Annual Conference(7), No. 05-1, pp. 305-306, 2005 (in Japanese).

[4] N. Kubota, H. Masuta, F. Kojima, and T. Fukuda, “Perceptual and action system of a mobile robot based on perceiving-acting cycle”, Transactions of Japanese Robot Society, Vol. 24, No. 4, pp. 473-481, 2006 (in Japanese).

[5] N. Aizawa and N. Kubota, “Formation behavior of multi-agent system based on dynamics model”, 2nd Joint Conference of Oukan, pp. 121-124, 2007 (in Japanese).

[6] S. S. Iyengar, A. Sabharwal, F. G. Pin, and C. R. Weisbin, “A synchronous production system for control of an autonomous mobile robot in real-time environment”, Applied Artificial Intelligence, 6, pp. 485-509, 1992.

[7] I. Iglezakis, “Case-based maintenance of case-based reasoning systems in classification domains”, Shaker Verlag Aachen, pp. 23, 2004.

[8] H. Ryu and H. Iba, “A framework for flexible humanoid robot control based on CBR-augmented GP”, Transactions of Japanese Robot Society, Vol. 24, No. 1, pp. 112-123, 2006.

[9] E. Plaza and S. Ontanon, “Ensemble case-based reasoning, Collaboration Policies for Multi-agent Cooperative CBR, ICCBR 2001, LNAI 2080, Springer-Verlag, pp. 437-451, 2001.

[10] T. Kohonen, “Self Organizing Maps”, Springer-Verlag, 1997.

[11] T. Shelley, “Storyboards give whirl to better design”, Eureka, Vol. 21, No. 11, pp. 33-34, 2001.

[12] M.Katoh and N. Imura, “Double-agent Convoying Scenario Changeable by an Emergent Trigger”, ICARA2009, 2009

[13] J. L. Graham and R. Burch, “Internal Traffic Control Plans and Worker Safety Planning Tool”, Transp Res Rec, No. 1948, pp. 58-66, 2006.