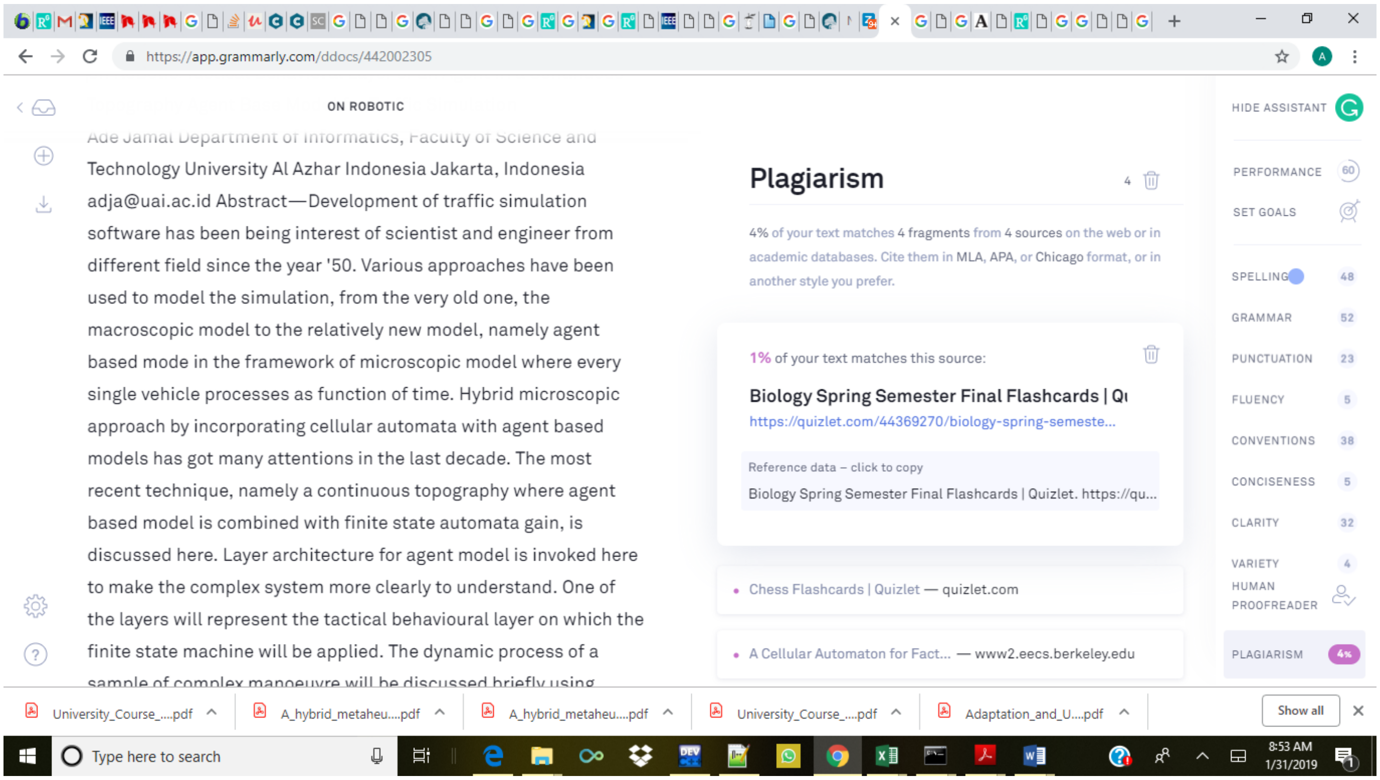

On Robotic/Tactical Behavioral Layer of an Agent in a Continuous Topography Agent Base Model for Traffic Simulation

Ade Jamal Department of Informatics, Faculty of Science and Technology University Al Azhar Indonesia Jakarta, Indonesia adja@uai.ac.id ***Abstract*—Development of traffic simulation software has been being interest of scientist and engineer from different field since the year '50. Various approaches have been used to model the simulation, from the very old one, the macroscopic model to the relatively new model, namely agent based mode in the framework of microscopic model where every single vehicle processes as function of time. Hybrid microscopic approach by incorporating cellular automata with agent based models has got many attentions in the last decade. The most recent technique, namely a continuous topography where agent based model is combined with finite state automata gain, is discussed here. Layer architecture for agent model is invoked here to make the complex system more clearly to understand. One of the layers will represent the tactical behavioural layer on which the finite state machine will be applied. The dynamic process of a sample of complex manoeuvre will be discussed briefly using statechart and sequence diagram.*** ***Keywords—traffic simulation; agent based model; finite state machine;***

I. _Introduction_

In the earlier phase of research and development of traffic simulation software, a macroscopic model based on continuum hydrodynamic kinematic law was used where traffic flow is described as continuum. Modern traffic simulation has been being studied and developed based on microscopic model where trajectory of every single vehicle is calculated as function of time. Since the influences of mathematicians and computer scientist on the traffic simulation research are getting more significant, two important techniques, namely the cellular automata (CA) and the agent based model (ABM) have been incorporated in this research area. CA was first time used in the microscopic traffic model by introducing the two lane model with periodic boundary conditions [1]. The combination of CA and ABM was later on incorporated to model pedestrian movement using regular square structured multi-grid topography [2]. Traffic participant movements are directed by topography model of the traffics environment. In the CA based two- or multi-lane model, vehicle's movement can take place in two ways, first completely unrealistic sideway and forward [1]. On the more complex regular grid structured topography using CA and ABM combination, traffic participants basically can move in any direction depending on the grid cell shape and size. Using hexagonal shaped cell gives more possibilities of movement direction than using square shaped cell. Furthermore, the smaller the cell size, the more accurate the movement trajectory can be modeled. Making more complex cell shape and smaller cell size requires significantly extra computational resources in term of power and memory. The CA based traffic model is received many interests during 90's as it can run large and relatively complex traffic simulations such urban traffic with only comparatively low computational resources [3]. As the strength of CA model is efficiently used of computer resources, then the previously improvement by increasing cell shape complexity and using smaller cell size should not be followed. Ulf Lotzmann [4] has proposed a completely other approach by introducing a continuous topography model. In this model, agent based system is still retained but cellular automaton principle is thrown away. The finite state automata takes place the role of CA to control the agent behavior. The present research will study this new model of topography as has been shown in the previously published paper that the model of lane changing movement in the new approach needs more complex mathematical model comparing that the old CA based model [5]. This paper will focus more in the finite state machine of the tactical agent's behavioral layer where all physical layer of agent will be determined autonomously.

II. A_Continuos_Topography_ Agent_Based_Model_Traffic_ Simulator_

The traditional approach of microscopic model based for traffic simulation was modelled as single lane, i.e. the system consists of a one dimensional grid with periodic boundary. Two basic models are very often used i.e. car-following model [6, 7, 8, and 9] and lane-changing model [6] in the microscopic single lane model. Later on a mathematicians and physician scientist [1] has incorporated cellular automata (CA) to introduce a two lane model consisting of two parallel single lane models with periodic boundary conditions and additional rules defining the exchange of vehicles between the lanes. This CA approach lasts not longer than one decade in the year 90's, until software scientists introduced the so called agent based model (ABM) to traffic simulation software [10].The ABM approach has been a successful technique in the recent traffic simulator software in the last decade [11,12]. Dijkstra [2] has combined CA and ABM to model pedestrian movement using a regular grid based topography model. While in the single or multi-lane microscopic model, traffic participant can only move forward or lateral only in every (sub)-step, in the regular grid structure CA based topography, any traffic participant can move in any direction as long as no obstacles blocking the movement depending on the grid shape and size. In contrary to CA model where its topography is characterized by regular grids in various forms e.g. square or hexagonal shape, the current approach takes a continuous topography hence the agent is able to move more freely and independent from any restriction such as grid shape or grid size. An agent is an autonomous entity that cannot be controlled by external interference but and it perceives events from its environment and then react accordingly. How agent perceives and reacts to this perception must be governed by a logical system. In the case CA and ABM combination, a simple CA algorithm is invoked since the perception area covers only the neighbouring grid cell of the agent. In the continuous topography model, a communication system likes a sensor system is used to receive any event's data only in the perception area as shown in Fig. 1 two agents are out of communication areas. In this case the simple algorithm incorporated by CA must be replaced by more complex finite state automata or finite state machine (FSM) as explained in [13]. Fig. 1.Agent's communication system Following the footprint of works in [4, 11, 14], the on- going research [5, 15] distinguishes the agent model in two abstract layers, i.e. a physical layer and behavioural layers. The physical layer depicts the real-world attributes of an agent such as location coordinates, velocity, direction and other physical parameters attached to the agent's entity. This physical layer is the only layer that has interface to outside world of agent, either for dynamic process or for visualization. The behavioural layers model how the agent will behave in response to its environment. This layer is further separated into two functional layers, namely the operational behavioural layer and the strategic behavioural layer. The first layer is also called the robotic operative behavioural layer in [4, 14] or tactical layer in [11]. This tactical behavioural layer will interact directly with the physical layer describing the perception and reaction of the agent in a very short time. The second layer is the strategic layer which has an intelligent behaviour for decision making such as route selection. The strategic layer will perceive the robotics perception and then will provide an intelligent action to be performed by tactical layer accordingly. This intelligent layer is called Artificial Intelligent (AI) layer in [4, 14]. Figure 2 shows the agent model and the interaction within agent's layers and the topography. This article as described in the title will focus only in the tactical behavioural layer. Fig. 2.Layer wised agent model and interaction with each layer and topography

III. Agent's_Tactical_Layer_

In [4] an agent can be taught as a reactive agent who has ability to interact with other agent but can only passively communicate. Hence the name "reactive" means that this type of agent cannot recognize its environment actively. The specialization of this reactive agent is a proactive agent that defined as subclass of reactive agent. The sub-class proactive agent is the most important agent as the most of traffic vehicles fall in this agent category. This type extends the reactive agent with capability to perceive events from the agent's environment and then react accordingly. In the present research, an agent will be distinguished in different manner,

Agent A Sensor's perception area B
Sensor's perception area A Agent B
Agent
Behavioral layer
Topography
Physical Layer
Strategic Layer (AI layer)
Tactical layer (operative / robotical layer)
Action Perception

i.e. passively immobile agent and actively mobile agent. The difference lies in the structural relationship where these two type agents are both sub-classes of a common abstract agent. Some of physical attribute will belong to the abstract class of agent, such as shape, position and sensors. Their capability will be different for each type, for sure the passive agents will not occupy a strategic behavior, but do have a tactical behavior however much simpler than the actively mobile agent. The more complex tactical layer of actively mobile agent requires Finite State Machine (FSM) approach to model the autonomous dynamical process as can be represented by nested state chart diagrams [13]. *A. Finite State machine* Finite State machine is an abstract machine that can be in one of a finite number of states but in only one state at a time, called current state. By a triggering event or condition, the state can be changed from one to another state; this process is called a transition. A particular FSM is defined by a number of its states and triggering event for each transition. For a complex machine, a particular state can consist of many sub states which form complete FSM in a hierarchy level. The state activity and action in the more complex state (super state machine), will trigger an event for a transition on the state machine bellow (sub-state machine) by supplying the state transition function and its input parameters. However, when the transition function cannot handle the input information for instance because of missing or unknown data, a trigger signal for state change on the automaton at the level above (its super state machine [13]) might be fired by the sub state machine. The tactical layer of actively mobile agent is further split into three sub-layers, namely: (Level one (the lowest level) represents basic actions which are usually conducted without thinking by humans (e.g. turning the steering wheel). (Level two deals with activities composed of basic actions (e.g. hold the center of a lane). (Level three subsumes all required plans for complex activities a human is aware of when executing (e.g. lane change operation). The state activities on the lowest level directly modify the physical agent attributes during the lapse of time. Due to the time-discrete simulation model, the activities are adjusted for the duration of a discrete time step. Furthermore, state activity and transition function are executed at every time step. The transition functions for level three are provided by the AI strategic behaviour layer. Each layer of robotic behaviour is equipped with a sensor communication and a perception filter processor. The filter is used to reduce the input data that are not significant to the state machine and in order to decide whether the information from the perception input data is not sufficient for underlying level. *B. Level One FSM* The first level of tactical layer consists of states that represent four basic actions which are normally done without thinking by human driver, namely: (Idle: the vehicle remains steady in the current state. Speed ***v*** and heading direction _ remain constant. (Accelerated: the driver accelerates or decelerates the vehicle by accelerating intensity ***a*** to the desired speed ***v***. (Bent: the driver turns the steering wheel with the angular radius m in order to gain a new direction _ where linear speed ***v*** remains constant and the heading direction _ is modified accordingly. (Unsteady bent (accelerated and bent): combination of accelerate and bend where both linear speed ***v*** and the heading direction _ are modified. Fig. 3 shows the state chart diagram of the level one FSM for the four basic action, where for each underlying movement action a mathematical model based on Newtonian governing equation are given in [5, 15].

Idle
Accelerated(a, v)
Bent(m,alpha)
Unstedy bent(a,v,m,alpha)

Fig. 3.Statechart diagram FSM Level One *C. Level Two FSM* The state at level two describes various basic driving manoeuvres, namely: (Lane-Centered: the driver steers the vehicle in the center of a lane. (Lane-Bordered: the driver steers the vehicle in the direction of the (left or right) lane border with the angle _. (Off-Lane: the driver steers the vehicle onto a topographic region not marked as a road (e.g. crossing, parking etc.), heading towards a target "***tg"*** decelerated to desired velocity ***v***. (Turned: the driver steers the vehicle into a bend. (Lane-Center & Lane-End-Ahead: the driver steers the vehicle in the center of a lane and reaches the end of the lane.

Fig. 4 shows the state chart diagram of the level two FSM for the five basic manoeuvres that in turns composed of basic actions from the level one FSM. Before each activity from level two will be triggered, a moderate calculation must be performed to determine some input parameters from the level one basic action based on the result of perception data from the environment. For instance Lane-Centred manoeuvres calculation is presented in [5], and the other manoeuvres will be presented in [15].

OffLane(v, tg)
LaneCentered(v)
LaneCenter_LaneEndAhead(v)
Turned(v)
LaneBordered(v,alpha)

Fig. 4.State chart diagram FSM Level Two *D. Level Three FSM* Level three FSM will cover various complex driving manoeuvres, such as: (Go-Ahead: the driver follows the course of the road with intended velocity ***v*** considering traffic rule and road situation. (Lane-Changing: the driver performs a lane change to the left or right side according to parameter "direction" (Cross: the driver passes an intersection with intended velocity ***v***, heading towards outbound lane "out" (Drop-Off: the driver stops the vehicle for short time along shoulder of the road (border lane), usually to drop off or pick up passengers. Fig. 5 shows these four samples of complex manoeuvres in the state chart diagram of level three FSM. The level three manoeuvres can consist of one or more level two activities and will be explained later using sequence diagram.

GoAhead(v)
Cross(v,out)
LaneChanged(direction)
DropOff(tg)

Fig. 5.State chart diagram FSM Level Three

IV. Dynamic_Process_of_tactical_ Behaviour_Layer_

In [14], an example of the complex manoeuvres of level three robotic layer, in this case the crossing capability of an agent. Here the other manoeuvres will be briefly discussed. The Fig. 6 shows a picture of a situation for a car traffic where agent a1 hindered by slower moving agent a2 (v2 < v1). The current state of a1 is {L1: Idle; L2: Lane-Centred; L3: Go- Ahead}. The agent a1 have two possible actions i.e. changing lane to b2-b3 or following car agent a2. Fig. 7 depicts the dynamic process of lane changing movement as sample complex manoeuvres. Fig. 6.Sample traffic situation two agents in one lane

V. Conclusion_

A new approach of agent based model traffic simulation software was discussed in this paper. There are two important differences in the new technique i.e. continuous topography in compare with the traditional cellular grid based topography and utilization of finite state automata in place of cellular automata. These two issues yield more complex mathematical model and computational model. A layered architecture is invoked in the agent model to make the model definition easier to develop and applying nested finite state machine gives a successful result. The layer which describes how the agent will perceive and react to environment's event and situation is named a behavioural layer. This layer is divided into tactically operative behavioural and intelligent strategic behavioural layer. The tactical behavioural layer, also called robotical layer is further split into three level of nested finite state machine, from the simplest movement, e.g. accelerating, in level one up to the most complex manoeuvres in the level three such as lane changing movement. How these three sub layers of tactical behavioural layer interact in the dynamic process of simulation are clearly described using sequence diagram.

*Acknowledgment_*

v1 v2 b1 b2 b3
Agen
1

This work is funded by research grant from LP2M UAI.

*References_*

[1] M. Rickerta, K. Nagel, M. Schreckenberg, A. Latour, *Two lane traffic simulations using cellular automata*, LANL Report No. LA-UR 95- 4367, 1995 [2] J. Dijkstra, A. J. Jessurun, H. J. P. Timmermans, *A multi-agent cellular automata mode of pedestrian movement*, in Pedestrian and Evacuation Dynamics (Eds: M. Schrekenberg and S. D. Sharman), Springer-Verlag, 2001 [3] J. Esser, M. Schreckenberg, *Microscopic simulation of urban traffic based on cellular automata*, in International Journal of Modern Physics C, Vol. 8 No. 5, pp. 1025-1036 , 1997 [4] U. Lotzman, *Design and Implementation of a framework for the integrated simulation of traffic participants of all types*, In Proceeding of the 2nd European Modelling and Simulation Symposium, 2006 [5]A. Jamal, *A Continuous Topography Approach for Agent Based Traffic Simulation, Lane Changing Model*, in Proceeding of the 1st International Conference Science, Electronic and Instrumentation, ICCSE2012, University of Gajahmada, Yogyakarta, 2012. [6]H. T. Fritzsche, *A model for traffic simulation*, Traffic Engineering & Control 35(5), pp. 317-321, May 1994 [7] M. Brackstone, M. McDonald, *Car-following: a historical review*, Transportation Research Part F 2, Elsevier Science Ltd., pp. 181-196. 1999 [8] J. J. Olstam, A. Tapani, *Comparison of car-following models*, VTI medelande 960A, Swedish National Road and Transport Institute, 2004 [9]K. Nagel, M. Schreckenberg, *A cellular automaton model for freeway traffic*, In Journal de Physique I, France, 2, pp. 2221-2229, 1992 [10] B. Burmeister, A. Haddadi, G. Matylis, *Application of multi-agent systems in Traffic and transportation*, in Software Engineering, IEE Proceedings, Vol. 144 (1), pp.51-60, 1997 [11] F. Klugl, H. Wahle, A. L. C. Bazzan, M. Schreckenberg, *Towards anticipatory traffic forecast- modelling of route choice behaviour*, in Proceeding of Workshop 'Agents in Traffic Modelling' , 2000 [12] A. L. C. Bazzan, F. Klugl, *Multi Agent Systems for Traffic and Transportation Engineering*, Information Science Reference, 2009 [13] D. Harel, *Statecharts: A Visual Formalism for Complex System*, in Science of Computer Programming 8, North-Hollad pp.231-274, 1987 [14] U. Lotzman, *A Multi-Purpose Agent-Based Simulation Framework for Complex Traffic Simulation Applications*, in Multi Agent Systems for Traffic and Transportation Engineering, (Editor Anna L.C. Bazzan, Franziska Klugl) Information Science Reference, 2009 [15] A. Jamal, *Model Matematis Topografi Kontinu Dari Simulasi Lalu Lintas Berbasiskan Agen*, in Jurnal Al-Azhar Indonesia Seri Sains dan Teknologi (in Indonesian) (to be published) 2013

PhysicalLayer
Timer
RoboticsLyr1
Sensor
RoboticsLyr2
SensorProcL2
RoboticLyr3
SensorProcL3
1 : new TimeStep()
2 : getSensorData()
sensorData: agent: a2 Lane Border: b1,b2,b3
3 : triggerTransisition<sensorData()
4 : process<sensorData>()
perceptionL2 Possible Collision a2 Lane Right: b2, b3
5 : triggerTransition<perceptionL2>()
alternatives: carFollowing a2 laneChange<right>
6 : process<perceptionL2>()
LaneChanged<right>
7 : changeState<LaneChanged<right>()
8 : nextState<LaneBordered<v,alpha>;LaneCenter<v>>()
9 : bend<m,alpha>()
10 : adjustAttribute()
11 : bend<m1,alph1>()
12 : adjustAttribute()
13 : bend<m2,alpha2>()
14 : adjustAttribute()

Fig. 7. Sequence diagram interaction three levels FSM in lanechanging manoeuvres