DESIGN AND IMPLEMENTATION OF A FRAMEWORK FOR THE INTEGRATED SIMULATION OF TRAFFIC PARTICIPANTS OF ALL TYPES

Ulf Lotzmann
Institute of Information Systems Research
University of Koblenz
Universitätsstraße 1, Koblenz 56070, Germany
E-mail: ulf@uni-koblenz.de

KEYWORDS

Traffic simulation, multi-agent based model, pedestrian model, Java

ABSTRACT

The paper describes a multi-agent system based simulation tool, designed with the aim of exploring behavior of and interaction between various traffic participants in heterogeneous and complex environments. The emphasis lies in the ability to configure and extend the system on a broad basis, especially in terms of embedding different agent-behavior models.

The modular architecture consists of a simulation kernel designed as an open framework, supplemented by visualization components and an import function for geographical data.

The paper focuses on issues related to modeling agents and topographic elements including methods for agent perception and communication. Also notes on adding visualisation and handling the entire system are mentioned.

OVERVIEW

The simulation of the behaviour of traffic participants has a tremendous influence on the analysis and planning of traffic systems, as can be seen from the large number of papers devoted to this topic and from the wide range of simulation tools. The majority of these contributions present approaches which try to solve specialized problems, either pedestrian movements or car simulation. Pedestrian simulation is dominated by microscopic approaches (e.g. in evacuation scenarios, most frequently with cellular automata; Burstedde et al. 2002), while in street traffic scenarios (analysing e.g. traffic jams) a wide range of approaches, both on a macro and a micro level can be observed (Nagel and Schreckenberg 1992, Campari et al. 2003; Jost and Nagel 2003; Lebacque 2003).

An example of a domain-specific tool was presented by Krajzewicz (2005) who described the SUMO ("Simulation of Urban MObility") traffic simulation which allows for the simulation of car movements on the base of a microscopic "car-following" model and is focussed on high processing speed for a large number of cars.

On the other hand, SeSAm ("Shell for Simulated Agent Systems", Klügl and Puppe 1998) is a representative of domain-independent tool which, as a multi-agent simulation environment, uses two-dimensional "gridmaps" to model the world around the agents. It can be customised with the help of several different plug-ins, e.g. as interfaces to geographical information systems (GIS).

The tool which we present here and which was developed as part of the project "Experimental studies for the design of multi-agent systems to simulate traffic participants' behaviour (TRAFFIC)" is midway between the two examples mentioned above. On one hand, it offers facilities for shaping the world surrounding the agents, on the other hand the multi-agent approach supports the simulation of behaviour of people in heterogeneous and complex scenarios and the application of different models of human behaviour. The tool will provide agents which represent all conceivable types of traffic participants and which can interact with each other and with their environment and jointly react on events. An example of an application of the tool is the simulation of a relevant situation within an urban region with pedestrians, cyclists, cars, buses, trams and underground railways.

Our tool can be seen as a tool for evaluating behavioural models in so far as it is a platform fort he implementation of microscopic agent model both of the reactive and proactive types. Contrary to the majority of comparable platforms, our tool is not bases on cellular automata (CA) for modelling the spatial structure. Instead we use a continuous approach with two-dimensional regions in which agents can move freely. This allows for a more "natural" modelling and a higher precision in the simulation of agent behaviour.

The strategy for modelling agent behaviour follows the two-step approach of Bazzan et al. (2000). Whereas tactical behaviour takes account of recognition and reaction on the situation in the environment in short time steps (rounds), the strategic behaviour dominates the way information is collected and hence the way strategic decisions such as route choices are made.

It will be possible to read input data for the simulated scenarios from existing data which will make it necessary to interface the tool with geographical information systems.

APPROACH

A simulation tool designed to fulfil the requirements mentioned above should at least have the following elements:

- models representing traffic participants with their physical attributes and with the ability to recognise their environment and interfaces for the flexible integration of behaviour strategies,
- models of topographies,
- a communication infrastructure and
- an infrastructure for collecting, presenting and visualisation of simulation results.

To ensure maximum flexibility the tool has to be conceived as a configurable and extendable framework.

The project described in this paper is dedicated to the design and implementation of such a framework whose central component is the simulation kernel which is extended into a simulation suite by various independent output and visualisation components and import filters for geographical data.

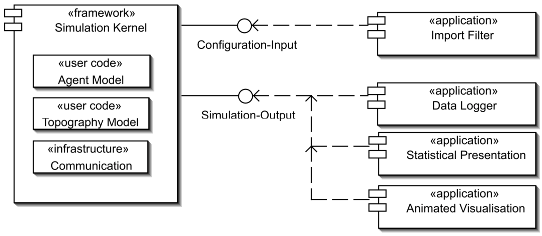

Figure 1: Component diagram with kernel, visualisation and import filter

The desired properties justify the application of a multi-agent system as the base of the simulation kernel although this could be problematic in terms of runtime behaviour in simulation scenarios that could be large and complex (see e.g. Gilbert and Troitzsch 2005). The high effort which is usually necessary to implement simulation environments and models is, however, compensated by a more natural style of modelling (Klügl et al. 2004).

The next section describes the design of the simulation kernel with its agent model, topographical model and communication infrastructure whereas subsequent sections describe the simulation output and its visualisation as well as the planned interface with external geographical data, respectively.

SIMULATION KERNEL

The simulation kernel combines the logically and mathematically described behaviour of the multi-agent system. The design encompasses an object-oriented class hierarchy consisting mainly in an agent model, a topographic model and a communication infrastructure — according to the first three elements required in the section above.

Agent Model of Traffic Participants

The agents representing traffic participants are classified into five basic types which, according to their complexity, form a class hierarchy in terms of the object-oriented paradigm (see Figure 2).

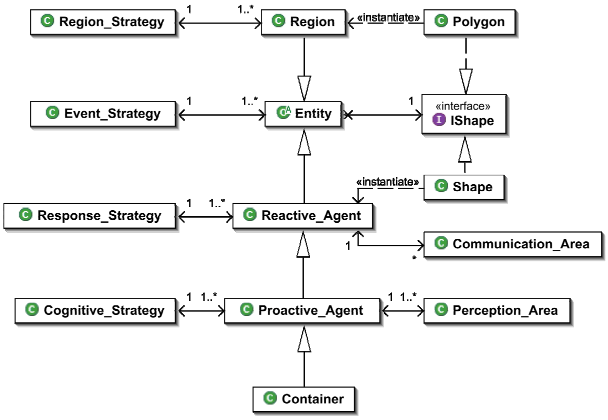

Figure 2: Agent part of the class diagram

- The abstract class **Entity** combines all attributes which are common to all types of agents and regions. These are mainly physical and geometrical attributes such as shape, mass, direction as well as a reference point which defines the position of the agent in a unified manner. Additionally, **Entity** introduces an **Event_Strategy**, which defines the actions which can be performed when events occur. Events can be internal events (e.g. a predefined simulation time step) or external events (e.g. the arrival of a message).

  Generally speaking, the behaviour of agents is defined in **Strategy** objects which come in several types. The inner state of an agent is thus defined by the attribute values and the related strategy objects.
- Agents of class **Region** form the building blocks for the topography of the simulated world.

The geometrical form of a **Region** — as a specialisation of **Shape** — is a **Polygon**.

Region defines a specific **Region_Strategy**, which provides the agents which are currently situated with the same region with topographic information.

The topography abstraction is detailed in the next section.

- **Reactive_Agent** is another specialisation of **Entity**.

  This type of agents has the capability of interacting with other agents; “reactive” means here that these agents cannot recognise their environment, so they are only passive partners in communication.

  As a reactive agent can respond only to queries from other agents, one substrategy — the **Response_Strategy** — is necessary.

  Before the response strategy is called after a query comes in, a decision process is started to find out whether the querying agent is entitled to contact this reactive agent. There are situations in which the querying agent is in a geometrically appropriate position with respect to the queried agent. To analyse this, the agent uses an arbitrary number of communication area objects — these are sectors with configurable radius and angle which are distributed around the reference point and define the allowable distances and angles of querying agents.

  Modelling the geometrical shape of reactive agents and derived types is particularly important. For the sake of efficiency it is possible to restrict the basic shape to a circle in two-dimensional space, but every agent can be composed of an arbitrary number of circles with different radius around the reference point (see the example in Figure 7).

  Stationary sources of information such as signposts, traffic lights or city maps are examples of this agent class.

- The most important class of agents is the **Proactive_Agent** class. These extend the reactive agent with the capability to recognise their environments and with another type of strategies enabling them to react on the perceived information (**Cognitive_Strategy**), which in turn enables them to represent mobile traffic participants.

  To perceive their environment, agents use a concept which is related to the communication area used by agents of the reactive agent class, namely the perception area which is comparable to the field of vision of a human. Again these are sectors of circles which allow neighbouring agents or topographic objects (streets, squares, sidewalks, ponds etc.) to be accepted as visible of the agent in question. The reason for such a localisation (which has to be connected to the respective communication area objects of the objects which can or cannot be seen by the agent) can easily be demonstrated with the example of a traffic light (see Figure 3).

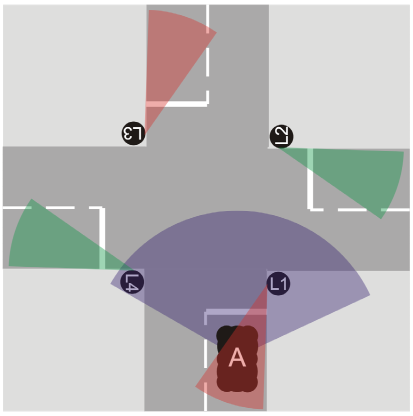

Figure 3: Model of a crossroad controlled by traffic lights: proactive agent A (representing a car) perceives traffic lights L1 and L4 within its own perception area (blue), but only L1 within the latter’s communication area (red) such that only information from L1 will be used for A’s decision making.

  Environment perception on this level of abstraction is reliable and robust, and the results produced are replicable such that stochastic elements are applied only within a cognitive strategy.

  Within this strategy short-lived communication relation are established with respect to the perceived objects, and the information gained from these relations are used as input for the decision which behaviour will be applied.

  As a rule, the strategy of a proactive agent aims at reaching a certain target — a topographic object or another agent. The way towards this target can be made known to the agent with definable precision and extended by the perception and communication processes. The bandwidth of precision is between the imprecise and general direction (such as “about north-northwest”) and a precise representation of the ways to take, including alternatives (such as a map). And it is, of course, possible to have agents that move in a random walk until they reach their target by chance.

  The interface for implementing strategies and the communication infrastructure are designed to allow for a wide range of behaviour models — from simple rule-based models to complex models with a communication language of their own.

  Several predefined and integrated strategies allow agents to act as sources, sinks and transporters for other agents.

- The fourth and last class of agents is the **`Container`** class which allows its instances to add and remove agents and to administer the transported agents.

Modelling the Topography

Modelling the topographic is obviously of decisive importance for any traffic simulation.

In this project we selected two different elements for structuring the topography which we call District and Region, respectively (see the topography part of the class diagram)

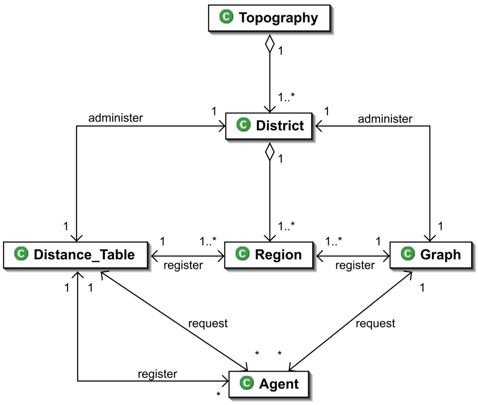

Figure 4: Topography part of the class diagram

Objects of class **`District`** are used to organise the simulated world into units which can easily be manipulated, which depends on the available system (hardware performance, number of nodes in a grid, screen, user preferences).

Within a district the agents can perceive their environment as a continuous two-dimensional space which — in principle — can be overseen and entered completely. Between two districts, agents can only be transported by transporter agents — such as doors or staircases. This allows for some simple examples of districts: regions which are separated by doors form districts of their own (rooms in flats, storeys in shopping malls, the interior of a bus, a railway or underground station etc.).

The substructure of a district is described by objects of the **`Region`** class. A **`Region`** has a closed polygon as a boundary, and it is given a meaning to agents by a number of readable attributes, such that for instance a **`Region`** can be described by a maximum allowable speed, the preferred direction, or by a list of agents which are allowed or not allowed to use it. In the same manner, particular topographic elements such as crossroads, sidewalks, squares, parks, lakes etc. can be modelled.

Every agent belongs to exactly one region at any time as its reference point is within this region. Additionally, agents of class **`Proactive_Agent`** evaluate the regions within their range of vision (perception area, see Figure 5)

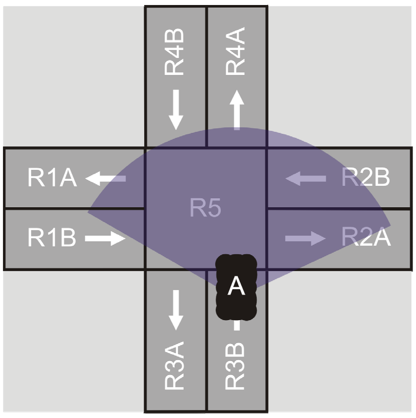

Figure 5: Perception area and region: proactive agent A (representing a car) is about to enter the region R5 (crossing) and perceives all regions R1 to R4 (roads). Only R1A, R2A, R3A and R4A are taken into consideration for route choice because of direction constraints (marked by arrows).

The district administers regions in two different data structures. Positions of the regions, or rather their boundaries, and those of all agents are registered in a distance table which is updated in every time step. Dependencies between agents and regions, however, which are necessary for route planning, are represented in a graph.

Communication Infrastructure

The concept for the communication system is two-tiered. Every district co-operates with a local communication intermediary. This intermediary accepts messages from agents and regions and delivers them to the addressees. Depending on how the agent strategies are implemented, agents, as a rule, communicate among themselves and with regions within the same district by directly passing queries in both directions.

Communication among agents in different districts as well as sending multicast and broadcast messages is supported by communication intermediaries. It is particularly important that multicast messages can be addressed with the help of category objects which associate every entity with one or more categories.

The local intermediaries communicate with a global intermediary which is also the interface with the controlling instances (such as time and round control), with the configuration of the simulation and with the output and visualisation components.

The separation into local and global intermediaries was introduced to allow for the loosest possible coupling between components and to make the internal structure of communication channels as transparent as possible —

also for the sake of a distributed execution of simulation runs.

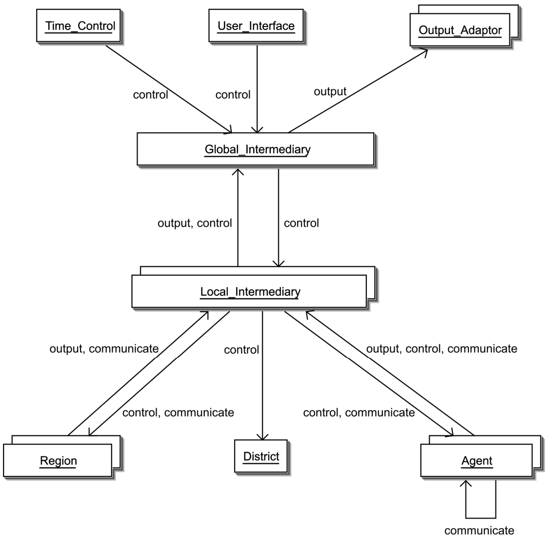

Figure 6: Object diagram with flow of different message types

Communication is managed with message objects which can be classified as follows:

- Unicast inter-agent communication

  These are messages for communication between two concrete agents, and they have dynamic, strategy-dependent contents.

- Multicast and broadcast inter-agent communication

  These have the same structure as the unicast messages, but are used to inform many or all agents about occurring events.

- Control

  These are messages to initialise the simulation (e.g. generate agents) and to control the ongoing simulation (time and rounds, transporting agents between districts, configuring agents).

- Output

  These are messages containing data about the state of objects in each round and sent to the output and visualisation components.

VISUALISATION AND OUTPUT

Implementing output and visualisation as an independent component opens the possibility to adapt the presentation of the results gained from the simulation flexibly to current needs.

The simulation kernel elements relevant for output can be configured easily for their output behaviour. This is done on-line, which means that output parameters can be changed during a simulation run. The output messages are generated as multiples of the simulation rounds and — depending on the configuration — can contain both generic data (positions of reference points, directions, shapes, colours) and information generated by the strategy objects during the simulation

The output system interprets the messages and transforms them for presentation.

Beside raw data logging and their statistical transformation visualisation methods are particularly interesting. A simple approach generates graphical primitives from output data such as position, agent shape and region boundaries and visualises them on screen. More expensive visualisations can be generated by an object structure administered by the output component, in this case each object represents a complex graphical element, and the object states are set by the output date.

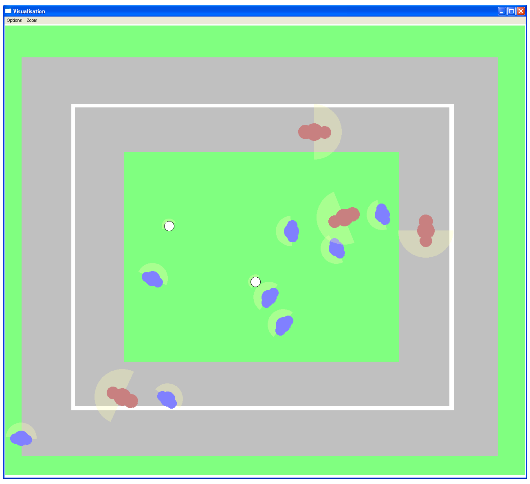

Figure 7: Screenshot from a running simulation where a number of agents representing human beings (blue) play with balls (white) in the presence of bicyclists (red)

USAGE

The tool is used in three steps requiring different activities:

The first step consists of modelling the properties of traffic participants and topographic scenarios and of implementing the respective attributes and strategies of the agents, in most cases by specialising the predefined classes. As the framework is implemented in Java this necessitates writing Java programs and methods although in principle it seems possible to provide adapters which allow the integration of code in other languages.

The second step includes the initialisation of the simulation environment, i.e. the topography is mapped on districts and regions, agents are positioned, and agents which generate and consume other agents (as sources and sinks) are defined. For this configuration, an XML-based data format is being designed in order that data from geographic information systems (e.g. in GML as a typical GIS tool or SVG as pure vector format data) can be imported (OpenGIS Consortium. 2002). DOM is used within the simulation kernel for the representation of the XML configuration document.

The third step, finally, encompasses the experiments with the simulation environment. During simulation runs, the behaviour of agents can be observed, the behaviour of agents and of regions can be changed (with the help of the respective elements of the GUI), and the time and rounds control can be reconfigured. Depending on the output and visualisation component currently used the simulation animation can be observed, or simulation produced data can be analysed in tables or diagrams

CONCLUSION

Configuring the simulation environment with the help of a proprietary XML data format is only an intermediate step on the way to the initially mentioned functionality of integrating topographical information from geographic information systems. This is why we plan to develop filter programs which add simulation specific semantics to imported data in a partly automated manner and convert all information into the proprietary format of the simulation tool. During the current project only one filter will be built as an example for a widely used format such that other conversion tools can be used to make even more formats indirectly available.

In the end, a few ideas for further research and development should be presented which could be realised in later stages. This is mainly about possibilities of other geometries for agents and topographic elements. It might be desirable to use curves of all types (particularly Bezier curves) to achieve a more realistic and/or more exact representation of all these objects. Apart from the higher demand in the computing power, the architecture of the framework will make it easy to realise this idea.

Another field of experimentation concerns the distributed execution of simulation models. The structuration of the tool (particularly the separation of the model world into districts) and the separation between calculation and output will render this easy, too.

ACKNOWLEDGEMENTS

This work is funded by Deutsche Forschungsgemeinschaft within the project on “Experimental studies for the design of multi-agent systems to simulate traffic participants’ behaviour (TRAFFIC)”, Tr 225/10–1. Thanks to Michael Möhring, Klaus G. Troitzsch and three anonymous reviewers for helful comments on an earlier version of this paper.

REFERENCES

Bazzan, A., Wahle, J.; Klügl, F., Schreckenberg, M. 2000. “Anticipatory Traffic Forecast Using Multi-Agent Techniques”. In *Traffic and Granular Flow’99 2000*, D. Helbing, H. J. Hermann, M. Schreckenberg, D. E. Wolf (Eds.). Springer, Berlin, 87-92.

Burstedde, C.; Klauck, K.; Schadschneider, A.; Zittartz, J. 2001. “Simulation of pedestrian dynamics using a two-dimensional cellular automaton”. In *Physica A 295,* 507-525.

Campari, E. G.; Clemente, M.; Levi, G.; Quadrani, L. 2003. “Real Highway Traffic Simulations Based on a Cellular Automata Model”. In *Traffic and Granular Flow 2003*, S. P. Hoogendoorn, S. Luding, P. V. L. Bovy, M. Schreckenberg, D. E. Wolf (Eds.). Springer, Berlin, 73-78.

Epstein, J.-G. 2002. *Modellierung eines Multi-Agenten-Systems mit fuzzy-logischen Regeln auf Mikro-Ebene*. Diploma thesis. University of Koblenz.

Gilbert, N.; Troitzsch, K. G. 2005. *Simulation for the Social Scientist.* Open University Press, McGraw-Hill, Maidenhead.

Jost, D.; Nagel, K. 2003. “Probabilistic Traffic Flow Breakdown in Stochastic Car Following Models”. In *Traffic and Granular Flow 2003,* S. P. Hoogendoorn, S. Luding, P. V. L. Bovy, M. Schreckenberg, D. E. Wolf (Eds.). Springer, Berlin, 87-103.

Klügl, F.; Puppe, F. 1998. “The Multi-Agent Simulation Environment SeSAm”. In *Proceedings of the Workshops "Simulation in Knowledge-based Systems"1998,* H. Kleine Büning (Ed.).

Klügl, F.; Oechslein, C.; Puppe, F.; Dornhaus, A. 2004. “Multi-Agent Modelling in Comparison to Standard Modelling”. In *Simulation News Europe, Issue 40 2004*, 3-9.

Krajzewicz, D. et al. 2003. “Using the Road Traffic Simulation SUMO for Educational Purposes”. In *Traffic and Granular Flow 2003*, S. P. Hoogendoorn, S. Luding, P. V. L. Bovy, M. Schreckenberg, D. E. Wolf (Eds.). Springer, Berlin, 217-222.

Lebacque, J.-P. 2003. “Intersection Modeling, Application to Macroscopic Network Traffic Flow Models and Traffic Management”. In *Traffic and Granular Flow 2003*, S. P. Hoogendoorn, S. Luding, P. V. L. Bovy, M. Schreckenberg, D. E. Wolf (Eds.). Springer, Berlin, 261-278.

Nagel, K.; Schreckenberg, M. 1992. “A Cellular Automaton Model for Freeway Traffic”. In *Journal de Physique I France, 2 1992*, 2221-2229.

OpenGIS Consortium. 2002. *Geography Markup Language.* http://www.opengeospatial.org/specs/?page=specs.