Evaluation of simulation engines for crowdsensing activities

Imre Lendák
Faculty of technical sciences
University of Novi Sad
Novi Sad, Serbia
lendak@uns.ac.rs

Károly Farkas
Department of Networked Systems and Services
Budapest University of Technology and Economics
Budapest, Hungary
farkask@hit.bme.hu

***Abstract*—The goal of this paper is to analyze existing simulation engines and assess how well-suited they are for simulating the formation, existence and dissolution of dynamic social networks, with a special emphasis on networks formed around crowdsensing efforts. The crowd in this context is a loosely-coupled social network of people, who use their mobile devices to collect and share data and receive some sort of service or satisfaction in return. Often it is hard to predict whether users would like a certain future crowdsensing application, therefore it is necessary to simulate the expected behavior of the crowd in a pre-specified simulation environment. This paper proposes an urban parking scenario, in which the drivers collect and share parking related events. The main part of this research is the analysis of three simulation engines, which will show which is the best suited for simulating dynamic social networks formed around crowdsensing efforts. The results will show that there are generic simulation environments capable of simulating large crowds, which also possess suitable visualization tools and integration with geospatial data.**

Keywords*—crowdsensing; simulation; comparison; geospatial integration; discrete events

I. Introduction

Modern mobile devices have rich sensing capabilities. The sensors integrated into our mobile devices range from the obvious ones, e.g. camera and microphone, to the more obscure ones like GPS, accelerometer, proximity sensor and gyroscope. The latest offerings in the smart phone arena might come equipped with additional sensors, like a barometer or a compass. Beside the obvious uses of these sensors, like making a call using the microphone, or taking a photograph with the camera, smartphones are also able to detect their orientation through the analysis of gyroscope readings and they can use their GPS readings to report their whereabouts in case they are stolen. There are even less obvious uses of the embedded sensors, e.g. the devices can detect the activity currently undertaken by their holder, i.e. whether the user is walking, driving or running [15]. These use cases are supported by the mobile device's operating system itself, or via publicly available libraries and applications.

Having in mind the above listed rich and varied sensing capabilities of our mobile devices, the challenge explored by many is how to develop services which would utilize the numerous sensors in novel ways. There are well-known success stories in which companies and/or researchers developed such solutions, and either became rich, or (just) improved one aspect of the way we live. One notable success story is Waze, a GPS navigation application, which apart from allowing users to share road-side events (e.g. police with a radar, roadworks) with other users, is also capable of automating some of its readings. For example by cleverly analyzing the position and acceleration readings, it might detect that the vehicle being navigated enters stop-and-go traffic and will ask its user to confirm the fact that there is a traffic jam ahead. Applications, which utilize the numerous sensors in our mobile devices in order to improve one or more aspects of our lives, are abundant and they allow users to send in reports about natural disasters, thereby helping relief efforts, or they might collect information about oil-spills or litter left on beaches.

One might argue that the people involved in collecting and sharing such information are a loosely-coupled social network with shared ideals and goals. These social networks are also dynamic as users come and go. They also change in the function of geographic position and time, i.e. some applications are more often used in certain parts of the world, e.g. Waze is used by many in the USA and in Hungary, and there are a lot less users in other European countries. The dependence on time might be linked to the popularity and necessity of their formation, i.e. in case a natural disaster happens, a local social and sensing network might be formed in order to assist the members of the emergency services and it might be dissolved after the relief efforts are finished.

The people who participate in these dynamic social networks (DSN) might be regarded as a crowd of social sensors. The goal of this paper is to analyze which tools and services are available for analyzing and simulating the emergence, life and dissolution of these DSNs, i.e. the goal of this paper is to assess publicly available simulation environments, which might be utilized to simulate various DSN scenarios. The emphasis is on "various", i.e. the goal is to identify those simulation environments which are generic and thereby capable of simulating a wide variety of DSNs.

The paper is organized as follows: section two presents related works, section three presents one specific sensing scenario which will be used later while assessing the simulation environments, section four discusses the method applied to compare the analyzed simulation environments, section five presents each simulator and section six contains a short discussion and comparison of the analyzed tools.

II. Related works

People involved in crowdsensing efforts form a *crowd*, which might be regarded as a loosely coupled social network of people, usually smart phone users, who collect and share data about their surroundings, and receive some sort of service in return [21][2]. *Sensing* in this context is the collection of some form of data by the practical application of the rich sensing capabilities of modern mobile devices. Crowdsensing is a form of crowdsourcing [1], in which a set of tasks is assigned to a crowd of people, who are often volunteers, i.e. they are usually not getting paid for their contributions. The data collected is known as volunteered geographic information (VGI) [6], which is one form of user generated content (UGC) [1]. VGI contains geospatial dimension, while UGC is not necessarily linked to a specific geographic location.

The challenges faced by researchers and developers building frameworks and services for crowdsensing efforts are discussed in references [2] and [21]. The latter specifically focuses on such activities in urban environments.

The number of crowdsensing efforts focusing on the collection and sharing of data about different aspects of our natural or urban environments is on a rise. The MetroSense project aims to develop a global mobile sensor network consisting of the users' mobile phones, a network capable of societal-scale sensing [11]. The study presented in reference [17] studies the trustworthiness of crowdsensed data and proposes a data validation architecture. Their work is validated on a real-life scenario in which users collect events from the rail network system of Barcelona. The authors differentiate the following three kinds of discrete events generated by the crowd: new event, confirm existing event or mark existing event as invalid. Event invalidation in this context means that other users mark a certain shared event as incorrect, either reported by mistake or by a malicious user.

Crowdsensing simulation is useful to simulate the possible future behavior of the crowd and estimate whether a suitable amount of information might be collected to provide a pre-specified level of service quality, e.g. ascertain that a future crowdsensing application will solve a problem for the end user. Reference [18] presents a simulation environment based on the ns-3 network simulation toolkit, and tries to predict how would the crowd sense events in the railway system of Barcelona, and assesses how many users would be necessary in order to report the majority of significant events, which are otherwise not monitored. Reference [3] describes a simulation environment in which the crowd collects information about parking spots. The underlying engine is custom built. The subject of sensing in both scenarios is discrete events, i.e. the absence or presence of a particular event at a particular location and at a particular time [17]. The principal difference between the above two pioneering crowdsensing simulation efforts lies in the fact that in [17] the crowd might form small clusters of users who are in the vicinity of an event which is sensed, while in [5] the crowd is formed from independent "agents" only circumstantially affected by the actions of other agents in their proximity, i.e. drivers will not go to a parking lot reported to be full.

III. Simulation scenario

The simulation scenario taken into consideration in this paper deals with urban parking. The dynamic social network is formed by people travelling by car in urban areas, who collect and share parking related events and thereby assist other users (i.e. drivers) by enabling them to find optimized routes to empty parking spots or parking lots with empty spaces. Consequentially, the time spent while "cruising for parking" is lowered, the negative impact of the extra exhaust fumes on our natural environment is lowered and the drivers save time, as they find appropriate parking more quickly. Based on the readings from one such system, city administrations might modify urban parking policies, i.e. introduce new parking lots or increase prices where necessary.

The simulations is limited to one urban area, whose boundaries are set as an input of the simulation scenario. A simplified model of the urban area is built from the exact locations of the following points of interest: residential block, commercial/industrial block, shopping area, entertainment facilities (e.g. pub) and religious institutions. The model also includes the parking spots and lots which are in the vicinity of the above listed locations.

People travel by car between the above listed types of locations and free or take parking spots in their vicinity in accordance with availability. Parking availability is calculated in the simulation environment based on (physical) availability and compared to the data shared by crowdsensors. It is assumed that there are no malicious users, i.e. all parking related events reported in simulation will be treated as entirely truthful.

The source and destination locations, i.e. locations between which the users travel is chosen based on a simple activity model with configurable parameters, e.g. during weekdays in the morning it is likely that people start from residential blocks, free one parking spot, travel by car to a commercial or industrial block, park their cars taking one empty parking spot nearby and spend roughly eight hours in one location. Similarly, during the weekends, people start a bit later and then travel to a shopping area or a religious institution based on a configured probability and spend there a couple of hours.

People driving between the above identified locations might be regarded as autonomous agents. Not all agents are equipped by a crowdsensing application which is used to collect and share parking related events, i.e. a varying percentage of agents is capable and willing to share parking related events. This percentage as well as the total number of agents is set in the input configuration of the simulation scenario.

The dynamic social sensing network is able to generate the following parking related events:

- Parking spot is taken/freed
- Entire parking lot is full/there are empty spaces
- Confirmation of the above events
- Invalidation of the above events
- Cruising for parking.

The model differentiates two typical day types, namely workday and weekend. For simplicity, cost is not taken into consideration, i.e. parking spots and spaces in paid parking lots are treated in the same manner as free parking areas.

As it might become dangerous for the driver to report more complex events while driving a car, it is necessary to either automate sensing, or involve the passengers travelling in the vehicle. Some of the above listed events can be automatically detected with either smart phones, or the in-built systems in modern automobiles. Cars might report when they take or leave a parking spot and they can do the same while cruising for parking, i.e. moving slowly and circling within an area of limited size.

The system might notify nearby users automatically and ask them to confirm events in their vicinity, which were reported by other users. Obviously, this would be limited to event categories which are pre-selected by users, i.e. the system would not bother users with events they are not interested in.

The users of such parking management system could enrich available geospatial databases by reporting in events, which might be detected automatically, e.g. parking a car might be reported as taking an empty spot, as well as confirming that there is a parking spot. Users who confirm the existence of a specific parking spot could improve the contents of existing geospatial databases, e.g. OpenStreetMap (OSM) [9].

IV. Method

If one intends to simulate the activities of thousands of drivers in an urban environment, then the simulation environment has to be able to handle large numbers of events, have visualization capabilities and some form of integration mechanism for importing and exporting geospatial information. Therefore the following set of simulation engine assessment criteria are proposed for those who are seeking a suitable toolkit for crowdsensing simulation:

- Suitable for large simulations – whether the engine can simulate networks of tens of thousands or even more entities?
- Support for discrete events – whether it can simulate events occurring at discrete moments in time?
- Is it a generic simulation engines or is it tied to one simulation domain, e.g. robotics, traffic system?
- Model separation – is the underlying model separated from the simulation and visualization logic?
- Cost – is it free or commercial?
- Visualization capabilities – which visualization capabilities does it possess, e.g. 2D, 3D?
- Geospatial integration – how easy it is to build a model or export results to freely available geographic information management or mapping solutions, e.g. OpenStreetMap [9] or Google Maps.

The analyzed simulation engines will be assigned marks for each of the above listed criteria. The marks will range from one to three, with the following meanings:

- one star (*) – very limited capabilities
- two stars (**) – suitable
- three stars (***) – excellent

A total mark will also be formed which might help prospective dynamic social network/crowdsensing researchers to choose the best simulation solution for their problem.

V. Simulation Engines

This section contains an overview of three simulation engines, namely:

- ns-3 [13]
- MASON [10] and GeoMASON [17]
- TraNS [8] and SUMO [4]

The descriptions of each of the above listed toolkits is followed by a list of similar simulators, which were not analyzed in detail during this work and were not be marked. The above three are deemed as representatives of three different simulator groups and the other simulators are similar to them.

A. ns-3

The ns-3 simulation engine is a discrete-event network simulator [18], designed for networking research and education. Its model is somewhat networking specific, as its key abstractions are nodes, applications and transmission channels. ns-3 is free software licensed under the GNU GPL v2 license and it is publicly available for research, development and use [13]. As far as visualization goes, ns-3 can use external animators and data analysis and visualization tools. The publicly available documentation of ns-3 does not mention built-in support for integration with geospatial data.

Although ns-3 is somewhat biased towards networking research, reference [18] presents a crowdsensing simulation study based on ns-3.

The most notable similar simulation toolkits are GloMoSim a network protocol simulation software [22], JANE, a platform for network application and protocol design [7], as well as OMNeT++ [14], a library and framework for building network simulators.

B. MASON

The Multi-Agent Simulator of Neighborhoods... or Networks (MASON) is a discrete-event multi-agent simulation toolkit developed in Java [10]. It is freely available.

Based on the claims of the authors and the available demos, MASON is quite capable of handling large simulations. The toolkit's simulation engine is well separated from its (optional) visualization modules, which are capable of rendering both 2D and 3D views.

Another important positive aspect of MASON is the fact that it is a generic simulation toolkit as opposed to other tools which focus on one specific domain, e.g. robotics, traffic systems.

Through GeoMASON, this toolkit was extended with support for vector and raster geospatial data [17]. Built-in integration with OpenStreetMap and/or Google Maps is not available.

The most notable similar tools and environments are SWARM [12], a toolkit for multi-agent simulation of complex systems, Ascape [3], a generic agent interaction framework, and NetLogo a multi-agent modeling environment [20].

C. *SUMO and TraNS*

The Simulation of Urban Mobility (SUMO) [4] is a specialized traffic system simulation suite. It is suitable to model systems involving large numbers (no upper limit!) of road vehicles, public transport and pedestrians. Its goal is to simulate the flow of vehicles for a certain pre-configured model of roads. SUMO supports OpenStreetMap integration.

The Traffic and Network Simulation Environment (TraNS) [8] is a graphical user interface (GUI) tool for simulating Vehicular Ad hoc NETworks (VANET). It integrates ns-2 (the network simulator preceding ns-3) and SUMO in order to create and visualize realistic traffic system simulations. TraNS might be regarded as a visualization tool for SUMO.

Both SUMO and TraNS are free. Unfortunately the development of TraNS is suspended [8].

Zero similar tools were analyzed during this research.

VI. Discussion

The previous section of this paper contains a detailed description of three simulation environments, which might be used in dynamic social network analysis. A quick overview of six more engines was also given (three network simulators and three multi-agent simulators), but those tools were not analyzed in detail because of their similarities with the chosen three. This section contains a short overview and comparison of the three simulation environments.

A. *General assessment*

Based on the short descriptions of the three analyzed simulators, it was concluded that they share the capability to handle simulations of large numbers of entities and that all are free to use. While ns-3 lacks in the geospatial integration area, SUMO comes with OpenStreetMap integration which makes it the clear winner in this category. MASON is somewhere in between, with somewhat limited integration capabilities.

Based on the analysis of the three simulation engines, it can be concluded that if a simulation engine is generic, then it probably also has a well separated model. MASON performed best in these domains with applications from widely different problem domains, and SUMO was the underdog as it is a highly specialized traffic simulation engine.

MASON is the clear winner on the visualization capabilities front, SUMO coming in second through its own and TraNS' UI support. ns-3 is somewhat limited in this domain as it needs external visualizers.

Table 1 contains the comparison of the three simulation toolkits presented in this paper. The cumulative marks of each tool are similar, with a slight advantage on the side of MASON.

Table 1 Simulation engine comparison

	ns-3	MASON	SUMO
Large simulations	***	***	***
Discrete events	***	***	**
Generic engine	**	***	*
Model separation	***	***	*
Cost	***	***	***
Visualization capabilities	**	***	***
Geospatial integration	*	**	***
Total	17*	20*	16*

Based on the detailed review of the above listed three simulation environments, and on the quick recapitulation of the characteristics of an additional six toolkits, it was found that researchers or developers who would like to simulate dynamic social networks formed around crowdsensing efforts, can choose from three flavors of simulation environments:

- network simulators developed for the analysis of wireless and sensor networks,
- multi-agent simulators developed for simulating complex agent interactions, and/or
- simulation toolkits specialized for one specific problem domain.

B. *Simulation scenario related assessment*

In the crowdsensing simulation scenario people in cars use mobile devices to share parking related events.

When it is modeled in MASON, the model of an urban area is loaded and built from geospatial data. Each driver is represented with an agent. At discrete moments in time, the agents move between points of interest (POIs) and look for parking, until they find one in the vicinity. For simplicity, the driver agents might "hop" between POIs instantaneously, i.e. travel time is not taken into consideration. The choice of the next POI is made in accordance with their programming and based on the probabilities set in the model. When leaving a POI each agent creates a "parking free" event, and when arriving at a POI, they raise a "parking taken" event. Cruising for parking might be modeled as a "hop" from a full parking spot/lot to the next nearest spot/lot. The events "fired" are visualized in 2D within MASON.

When the urban parking scenario is modeled in ns-3, then the drivers are represented with network nodes, which run crowdsensing applications. It is labor intensive to build a suitable model of an urban area because of the limited geospatial capabilities. The nodes in the vicinity of an important event, e.g. empty parking spot, use their applications to share information about its occurrence. ns-3 is also capable of modeling the interconnections between nodes located near to each other. The events raised by the "applications" run on the nodes are visualized in an external tool.

If the crowdsensing dynamic social network is modeled in SUMO, then it is fairly easy to load a suitable model of the urban area directly from OSM. Before each simulation cycle exact start and end points should be calculated for each of the entities (i.e. drivers). These calculations should be executed externally and fed into SUMO. With those data, SUMO can simulate the entire route between source and destination, with exact travel times. Crowdsensed events might be raised when the vehicles driven leave their sources or arrive at their destinations. Cruising for parking, as one of the scenario events is somewhat tricky to cover, because during that activity the drivers do not have a clearly defined destination where they will park their cars. SUMO can visualize one simulation step at a time and needs a new set of source and destination points to be calculated externally for each vehicle (i.e. member of the crowd) at the beginning of each cycle.

It was concluded that as far as the simulation scenario proposed in this paper is concerned, the optimal starting point for its simulation would be MASON, i.e. a generic multi-agent simulation toolkit. The main reasons for this are its clear model and visualization separation, which is important for researchers who plan to simulate various kinds of dynamic social networks, geospatial integration capabilities and rich visualization subsystem.

VII. Conclusion

This paper presents an analysis of three simulation engines and assesses how well-suited they are for simulating crowdsensing efforts. The crowd in this context might be regarded as a dynamic social network of users who utilize the rich sensing capabilities of their mobile devices to collect and share information about certain phenomena in their surroundings. The simulation toolkits were compared by analyzing various characteristics, ranging from architecture to geospatial integration. The analysis also utilized a hypothetical simulation scenario, in which drivers collect and share parking related events in an urban environment.

The results show that the analyzed simulation toolkits fall into three broad categories: network simulators, multi-agent simulators and toolkits developed for specific application domains. It was shown that MASON, a multi-agent simulation toolkit had the best marks in the general assessment and therefore it would be the most suitable for carrying out the proposed (urban parking) simulation scenario. Although MASON might not be the best choice for simulating other crowdsensing scenarios, this paper might be used as a guide for choosing the optimal simulation tool.

As a continuation of this research, the authors plan to develop the proposed urban parking simulation model in MASON and perform various experiments with the final goal of finding out which parking related events should be collected, and with how many users in a successful crowdsensed parking system.

Acknowledgment

This research was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia under grants III-42004 and TR33013, and by the Hungarian National Academy of Sciences under grant Domus 105/6864/HTMT. Károly Farkas has been partially supported by the Hungarian Academy of Sciences through the Bolyai János Research Fellowship.

This work was partially supported by COST Action IC1203 (ENERGIC).

References

[1] D.C. Brabham, “Crowdsourcing the public participation process for planning projects”, Planning Theory, vol. 8(3), pp. 242–262. August 2009.

[2] Cardone, G., Foschini, L., Bellavista, P., Corradi, A., Borcea, C., Talasila, M., Curtmola, R., “Fostering ParticipAction in smart cities: a geo-social platform”, IEEE Communication Magazine, vol 51(6), 2013, pp. 112–119.

[3] D. Centola, “The Ascape Model Developer’s Manual”, http://ascape.sourceforge.net/manual/Overview.html, last access Jan. 31, 2015.

[4] DLR Institute of Transportation Systems, “Simulation of Urban MObility (SUMO)”, http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/, last access on Jan. 30, 2015.

[5] K. Farkas, I. Lendak, “Simulation Environment for Investigating Crowd-sensing Based Urban Parking”, Models and Technologies for Intelligent Transportation Systems (MT-ITS 2015), Budapest, Hungary, June 2015 – abstract accepted for publication.

[6] M.F. Goodchild, “Citizens as sensors: the world of volunteered geography”, GeoJournal, vol. 69(4), 2007, pp. 211–221.

[7] Gorgen, D., Frey, H. and Hiedels, C., “JANE – The Java Ad hoc Network development Environment”, 40th Annual Simulation Symposium, ANSS ’07, Norfolk, USA, 2007, pp. 163–176.

[8] Laboratory for Communications and Applications (LCA), “Traffic and Network Simulation Environment (TraNS)”, http://lca.epfl.ch/projects/trans/, last access on Jan 30, 2015.

[9] M. Haklay, and P. Weber, “OpenStreetMap: User-Generated Street Maps”, IEEE Pervasive Computing, vol. 7(4), 2008, pp. 12–18.

[10] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan and G. Balan, “MASON: A Multi-Agent Simulation Environment”, Simulation: Transactions of the society for Modeling and Simulation International, vol 82 (7), 2005, pp. 517-527.

[11] MetroSense project, http://metrosense.cs.dartmouth.edu, last accessed January 24th, 2015.

[12] N. Minar, R. Burkhart, C. Langton, and M. Askenazi, “The SWARM simulation system”, http://www.swarm.org, last access Jan. 31, 2015.

[13] ns-3 Simulator, http://www.nsnam.org, last access Jan. 30, 2015.

[14] OMNet++ simulation environment, http://omnetpp.org, last access Jan 30, 2015.

[15] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, “Using mobile phones to determine transportation modes”, ACM Transactions on Sensor Networks, vol 6 (2), 2010, pp. 13:1–13:27.

[16] G. Rose, “Mobile phones as traffic probes: practices, prospects and issues”, Transport Reviews: A Transnational Transdisciplinary Journal, vol 26 (3), 2006, pp. 275-291.

[17] Sullivan K., Coletti M., Luke S., “GeoMason: Geospatial Support for MASON”, George Mason University Technical Report GMU-CS-TR-2010-16, 2010.

[18] C. Tanas, and J. Herrera-Joancomart, “Crowdsensing Simulation Using ns-3”, Citizen in Sensor Networks, Lecture Notes in Computer Science, 2014, pp 47-58.

[19] C. Tanas, and J. Herrera-Joancomart, “When users become sensors: Can we trust their readings?”, International Journal of Communication Systems, vol. 28 (4), 2015, pp. 601-614.

[20] U. Wilensky, “Netlogo”, http://ccl.northwestern.edu/netlogo/, last access Jan. 31, 2015.

[21] F. Zambonelli, “Pervasive urban crowdsourcing: Visions and challenge”, 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Seattle, USA, 2011, pp. 578–583.

[22] Zeng, X., Bagrodia, R., Gerla, M., “GloMoSim: a library for parallel simulation of large-scale wireless networks. 12th Workshop on Parallel and Distributed Simulation (PADS 98), Banff, Canada, 1998, pp. 154–161.