Simulating Cities: A Software Engineering Perspective

Cristina V. Lopes(✉)

University of California, Irvine, CA 92697, USA
lopes@uci.edu
http://www.ics.uci.edu/~lopes/

Abstract. Despite all the reasons why complex simulations are desirable for decision and policy making, and despite advances in computing power, large distributed simulations of urban areas are still rarely used, with most of their adoption in military applications. The reality is that developing distributed simulations is much harder than developing non-distributed, specialized ones, and requires a much higher level of software engineering expertise.

This paper looks at urban simulations from a software engineering and systems design perspective, and puts forward the idea that non-traditional decompositions in simulation load management are not just beneficial for these applications, but are likely the only way to move that field forward.

Keywords: Simulation · Programming · Software architecture · Systems design

1 Introduction

Population, environmental, and technological changes are reshaping the infrastructure of our urban environments, and will be a major focus of attention in the near future [36,45]. Cities have always accommodated new technologies in transportation, such as the invention and widespread use of motorized vehicles, mass transportation, and recently, electric vehicles. Yet every new technology brings with it challenges in policy making. With the advent of alternative energy and the Internet of Things, there is suddenly a plethora of new technologies that will require major restructuring of cities. Some examples are electric and autonomous vehicles, solar energy, unmanned aerial vehicles, and smart homes and buildings. In order to establish sound public policies, all these new technologies can greatly benefit from urban simulations, to measure their impact on the city and on the people before the policies are defined. According to Fujimoto [16], the combination of a severe need for investment in infrastructure redevelopment [44], an increased percentage of the world moving to cities [40], and the need for better disaster resilience to reduce both costs and deaths [10], are reasons why the development of effective platforms for urban environment simulations is one of the greatest challenges in the Modeling and Simulation (M&S) field.

P. Thiemann (Ed.): ESOP 2016, LNCS 9632, pp. 1–14, 2016.
DOI: 10.1007/978-3-662-49498-1_1

Increasing the complexity of simulations requires an increased amount of computation power, which has been made available in the past decade through parallel and distributed computing. Barnes et al. [1] provide a chronological history of the computing capacity using the PHOLD benchmark, a benchmark designed for performance evaluation of parallel and distributed discrete event processing. From 2007 to 2013, there was an improvement from 100 million to 500 billion events per second, a 5000x speedup in 6 years. The speedup was mainly due to an increasing number of cores in supercomputers, with the experiment in 2013 using nearly 2 million cores.

Despite all the reasons why complex simulations are desirable for decision and policy making, and despite advances in computing power, distributed simulations are still rarely used, with most of their adoption in military applications [2–4,37,39]. The reality is that developing distributed simulations is much harder than developing non-distributed ones, and requires a much higher level of software engineering expertise, which usually modeling and simulation experts don't have. The difficulties have been well-known by researchers and engineers, and over the years there have been a few efforts to address them. One of the most important efforts was the development of the High Level Architecture (HLA) standard [19]. HLA specifies a powerful model and an API for distributed simulation components (called *federates*) to interact. The adoption of HLA in practice, however, has been very slow. Boer et al. [2–4] published a three-part survey on the topic of distributed simulation in industry, including the use (or lack thereof) of HLA. In their findings, the major cause of low adoption is cost-benefit: it is technically complex to design and build distributed simulations, even with HLA. The reasons given included semantic interoperability, complexity, and efficiency of implementations. Surveyees suggest that domain-specific adaptors and middleware could help bridge the technical gap in software knowledge. We believe that the use of appropriate software engineering concepts and theories can greatly reduce the technical complexity of developing such middleware, and make distrubuted simulations accessible to a much wider audience, including scientists.

This paper explores the domain of distributed simulations, its historical development, its applications in urban planning, and the challenges and opportunities related to software engineering and systems design. It then presents a design experiment that mixes concepts from Aspect-Oriented Programming (AOP), relational queries and dependent types.

2 Workload Partitions in Simulations

One of the main challenges in doing complex systems simulations, including living organisms and cities, is the fact that they embody many different subsystems, each of them relatively, but not completely, independent from the others.

For example, there are many similarities between the concept of aspect (as given by AOP [21]) and the general concept of *aspect of a city* that urban planning researchers routinely use. Figure 1 shows an informal model of a city; similar models depicting infrastructure interactions are found in urban planning

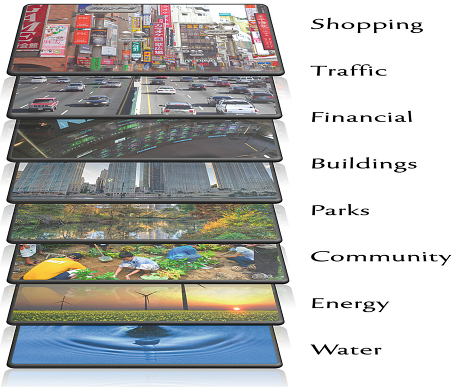

Fig. 1. Aspects of cities.

literature, such as Xu et al. [45]. All these *aspects* are conceptually distinct, but interact with each other over essentially the same objects (data) of the simulation. Moreover, each of these subsystems tends to fall under different expertises, different groups of people who understand them. Aggregating that knowledge under one large simulation is a daunting task.

A model such as the one in Fig. 1 is ripe for decentralized and distributed simulation: we can envision each of these urban aspects being developed by a different team (because they require completely different expertise) and being simulated by a different machine, and yet cooperate for the global simulation of the city. Distributing a single simulation run where the workload is too large to be computed in a single processor, or worse, a single machine, is a challenging research problem in software architecture and design. In such cases, how does one divide and distribute the workload?

One way is to distribute the data. Simulators partition the workload by simulating different objects at each time frame. While scalable in terms of number of objects, this approach is not scalable in terms of simulation complexity: many different subsystems need to access the same data, so each simulation node needs to simulate all subsystems. Another approach is to divide the load by subsystem: functions such as physics simulation, scripts processing, and client management, are run in distributed simulators, and object attributes are synchronized when updated. This a good solution for synchronizing updates, but has some performance penalties, because data needs to be shared among simulators.

Much more can be done to help tame the complexity of these systems and improve the workflow of collaborating groups. Part of it consists in experimenting with the myriad of interesting concepts that exist in the small world of programming languages. Recently, we have been experimenting with Aspect-Oriented Programming (AOP) concepts, but other concepts can be used.

3 Historical Context

3.1 Distributed Virtual Environments

Distributed Virtual Environments (DVEs) are online multi-user interactive systems that simulate shared 3D spaces. The first successful standard for distributed simulation was SIMNET [29], funded by the DoD and developed by DARPA. With SIMNET, it became possible to link hundreds of simulators to produce a virtual world, used for real-time, man-in-the-loop, coordination and tactic simulations. SIMNET eventually evolved into the Distributed Interactive Simulation (DIS) standard, becoming an IEEE standard (IEEE 1278–1993) [9].

Strassburger et al. [37] conducted a peer study of future trends in DVEs, in which experts saw high potential for training sessions, joining computer resources, and integrating heterogenous resources for distributed simulations. We see the same opportunity to use DVEs for our urban platform, while additionally providing visualization and interaction of the environment for any future simulations purposes. Two foundational papers in DVE are DIVE [8] and Diamond Park [43], which both proposes VEs that scale by partitioning the workload in terms of space. Each process handles an area of the virtual world, achieving optimal scalability in terms of virtual land area, and improved scalability of users and objects. Most DVEs used today utilize the same scalable approach of partitioning the world by space.

There are many widely used DVE implementations, most in games such as World of Warcraft, Second Life, and Minecraft. Second Life is particular in its lack of an end goal. It is meant to be an experience of living a different life, with no particular goal other than experience the VE with other users. OpenSimulator [32] is an open-sourced VE that uses the same protocol from Second Life. Both Second Life and OpenSimulator scale through the same approach from DIVE and Diamond Park: object-based space partitioning: the world is divided in blocks of 256 meters squared, and each region is simulated on a different simulator. Other known VE open-sourced distributed simulator implementations are OpenWonderland [20] and Meru [18], but are far less popular than OpenSimulator and Second Life.

The Distributed Scene Graph (DSG) architecture is a novel approach to partitioning the virtual world simulation by functionality instead of space [23,26,27]. In certain workloads, DSG proved to be a much more effective load balancing technique than the traditional space partitioning method. DSG was highly influential in our aspect-based proposal approach for distributed simulations in general. Our group collaborated with Intel on the DSG architecture, to reintroduce space-partitioning to DSG.

3.2 Distributed Simulations

Concurrently to the effort of developing DVEs, another effort unfolded to generalize the concept of distributed simulations beyond interactive environments. The idea here was that simulations don't always have an interactive component, and even if they do, it is important to generalize the model by which the different simulators work together to produce the collaborative, complex simulation.

For this purpose, another standard was developed with the participation of industry and academia partners: the High Level Architecture (HLA) [19]. The HLA is, as the name implies, higher level when compared to DIS. In DIS, most of the simulation design decisions, such as networking protocols, are fixed. The HLA is flexible, and enforces an API that simulation designers can use to make decisions for their own simulation. The HLA defines two separate entities: the federates and the runtime infrastructure (RTI). The RTI is the communication bus for simulators to share events and updates. It is also used to specify and share the object model template (OMT), a shared specification of what objects are available to be instatiated. New objects can be defined and shared between simulators. The federates are the simulators, and together with the RTI, they form the federation. The API from the HLA defines how federates should share object models, updates, events, and interactions.

The RTI has many commercial implementations, including VT Mak RTI and Pitch pRTI. In our project, we wish to use an open-sourced implementation so we can modify the functionality for better performance and to better fit the aspect-based approach we propose. Two well known open-source implementations are OpenHLA and Portico. For our implementation, we will use Portico, which we believe has more support and use than OpenHLA. Simulators can be adapted to the HLA protocol through what is called an HLA ambassador: a software module that translates events from the RTI to the simulator, and interprets internal events and routes back to RTI. We use ambassadors to leverage existing simulators in our proposal.

Arguably the most important related work to this proposal is NASA's Simulation Exploration Experience (SEE), originally called Smackdown [11]. The SEE was created to encourage education in distributed simulation and the HLA standard. NASA created a scenario, building a moon base, and gave researchers all the tools necessary to start a moon base simulation. The tools include a federate for coordinating the federation and for delivering common data types, documentation, and functionalities, such as reference coordinates from Earth and the moon, a visualization federate, and APIs for the simulation. From there, researchers from many different universities volunteered to help. Many scenarios were proposed and built for the simulation, including communication systems [13], surveillance and defense of base [6] and a lunar mining operation [38]. The success of SEE was presented at several simulation conferences, with researchers from multiple universities presenting positive experience and results [6,13,38], and positive impact for education in distributed simulation [12]. While the simulation was executed successfully, the product was a toy example: the moon base simulation is not in any predictable way useful to real world NASA missions or plans for a moon base. On their analysis of SEE, Essilfie-Conduah et al. [12] say that SEE has potential to tackle real world M&S engineering problems. The experience from researchers participating in SEE also showed that HLA was difficult to implement, and the lack of tutorial examples made the learning curve even steeper. Essilfe-Conduah et al. also notice that while HLA is highly functional, it is also dense in content. We believe that the lack of any domain-specific design in HLA makes it powerful, but places a large burden on simulation designers. HLA should remain powerful, but if simulation designers are to use HLA, a form of middleware that bridges the gap of highly technical software engineering knowledge is required.

3.3 Urban Simulations

Urban simulations model and simulate urban environments and they can be both interactive and non-interactive. Wadell [42] interprets urban simulations as "operational models that attempt to represent dynamic processes and interactions of urban development and transportation". He argues that urban models have grown in complexity, and simulations are becoming a vital part of decision making by stakeholders for urban policies. A comprehensive history of urban modeling can be seen in [41]. In a talk presented at the Winter Simulation Conference 2014, Fujimoto [16] reiterates the importance of urban simulations today and in the near future. The advent of new technologies (e.g. smart power grid, smart homes) and new vehicles (e.g. unmanned aerial vehicles, electric vehicles, autonomous vehicles) will require redevelopment the urban infrastructure. One of the grand M&S challenges proposed by Fujimoto is to handle the complexity of so many different interacting simulations by composing distributed simulations.

Transportation simulation is an important part of urban simulations, and has a large body of previous work in traffic, logistics, and business. For the scope of our proposal, we will focus only in traffic simulations. Pursula [34] provides a comprehensive overview of the different models and approaches to traffic simulation. Mainly, traffic simulation is divided in degree of detail: microscopic, mesoscopic, and macroscopic. The macroscopic models were the first simulation models for traffic, treating traffic as a continuous flows that can be modeled mathematically, for instance, as waves [25]. Microscopic simulations attempt to model and simulate vehicles individually, determining starting and ending points, and calculating routes through roads and intersections. Most traffic simulation applications used nowadays are of microscopic nature. The car-following model [17] and cellular-automaton [31] variations are typically used in microscopic simulations. Mesoscopic simulations are a mix of macroscopic and microscopic models. One example for modeling mesoscopic simulations is the gas-kinectic approach for two-dimensional traffic flow [30]. Another example of a mesoscopic approach is CONTRAM (COntinuous TRaffic Assignment Model) [24], where movement of traffic is modeled by grouping vehicles in"packets". We focus on microscopic simulations for its widespread use and the leveraging of computing power increase from parallel and distributed simulations. There are many distributed and parallel models for large microscopic traffic simulations [15,33,46], that can be scaled independently from our urban simulation platform.

There are multiple microscopic traffic simulator implementations available for use. Some of the more popular ones are SUMO [22], TRANSIMS [35], PARAMICS [7], and VISSIM [14]. Maciejewski [28] provides a comparison between SUMO, TRANSIMS and VISSIM. PARAMICS is focused in parallel and distributed simulations, optimal for use with thousands of cores, tightly coupled and interconnected by network. For our project, we decided to use SUMO for two reasons: it is open-sourced (PARAMICS and VISSIM are not), and has a remote controlling feature, required to integrate it with our urban simulation. Additionally, we are leveraging previous work done to integrate SUMO and a virtual environment simulator, OpenSimulator, on the Mobdat project [5].

4 An AOP Design Experiment

We have been experimenting with partitioning the simulation load using AOP concepts. The architecture we have is guided by the HLA, but provides concrete design choices and a powerful expression mechanism (inspired by relational queries, in general, and dependent types, in particular) to reduce the technical complexity of distributing an urban simulation.

5 Architecture

The canonical HLA is a generic event distribution system. It can be seen in Fig. 2, where federates (i.e. simulators) communicate through the run-time infrastructure, publishing and subscribing to object attributes. The HLA is a powerful architecture for enabling interoperability between many simulations, but to achieve flexibility, major design decisions are left to the designer. Requirements such as message ordering, software patterns, object modeling, and networking protocols are complex technical decisions that may jeopardize the entire simulation if not planned properly. The HLA enables all such options to be configured and agreed upon between federates, but the excessive number of choices essentially leaves simulation designers with most of the architectural work, requiring highly knowledgeable software engineers and M&S experts to achieve the desired performance and behavior. Simulations designers are often knowledgeable of the simulation concerns and desired results, but not necessarily of building and planning distributed simulations for consistency and scalability. If successful, our effort will reduce this entrance barrier to HLA, allowing simulation designers of different fields to exert their full expertise in their own aspect of the simulation.

Architecturally speaking, we introduce a bridge between the flexible HLA abstractions and the aspect simulators, or federates[1]. Figure 3 illustrates what the resulting architecture will be. Our platform handles composing messages within the HLA standard, but with meaningful functional calls from the simulation federates. The intention is to transform HLA configuration into simulation semantic settings that designers understand. For instance, if the simulation is meant to be human-in-the-loop, our platform makes design choices for interactive real-time simulations, and communicates them with other federates. From the federate (aspect) point of view, there are only two components: itself and the rest of the simulation. The urban platform mediates the interaction of the world with each federate, in terms of publications and subscriptions, and manages synchronization of objects with multiple aspects.

[1] We use the terms “aspect simulator” and “federate” interchangeably.

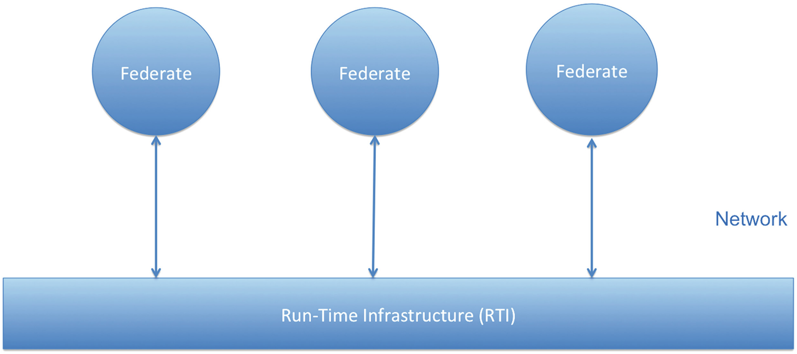

Fig. 2. The High Level Architecture (HLA) representation. Federates are the simulators, that together form the federation. The Run-Time Infrastructure is responsible for handling publish and subscribe events for object attributes.

This urban simulation architecture raises several research questions about building distributed simulations with cross-cutting concerns. How do we represent the same entity in multiple simulations, with each simulation interested in a different aspect of the entity? What kinds of interfaces should the federates expose to the rest of the simulation? In what way will AOP concepts, such as join points and before and after actions, simplify the architecture? How can we specify a universal [base] object model that will scale with new and yet unknown aspects of the urban simulation? Can the RTI, a centralized communication bus, handle real-time simulations at a large scale? Establishing a large-scale urban simulation platform requires addressing such issues, and we hope to reach new insights and knowledge for building distributed real-time simulation models in general.

5.1 Data Specifications

The architecture we are using is essentially one where data is shared among any number of simulators, each one simulating an aspect of the simulation. It is clear that the different aspects have different perspectives on the shared data. For example, a car might have a rich representation in an aspect decicated to simulating cars, but it is reduced to essentially a bounding box and a position in 3D space for purposes of traffic simulation. We see the data space as a collection of data sets, some of which may be subsets, projections or augmentations of existing sets. Every new aspect that is added to the simulation needs to know about the schema of the existing data, but that is all: it does not need to know about any behavior of other aspects; there is no need for functional APIs between the aspects. Depedencies are purely data dependencies.

As such, we are designing an expressive data specification language based on SQL (relational algebra) that is also inspired by the concept of dependent types. The idea is to allow the different aspects to reduce / augment / change the existing data types so that it better fits their own function.

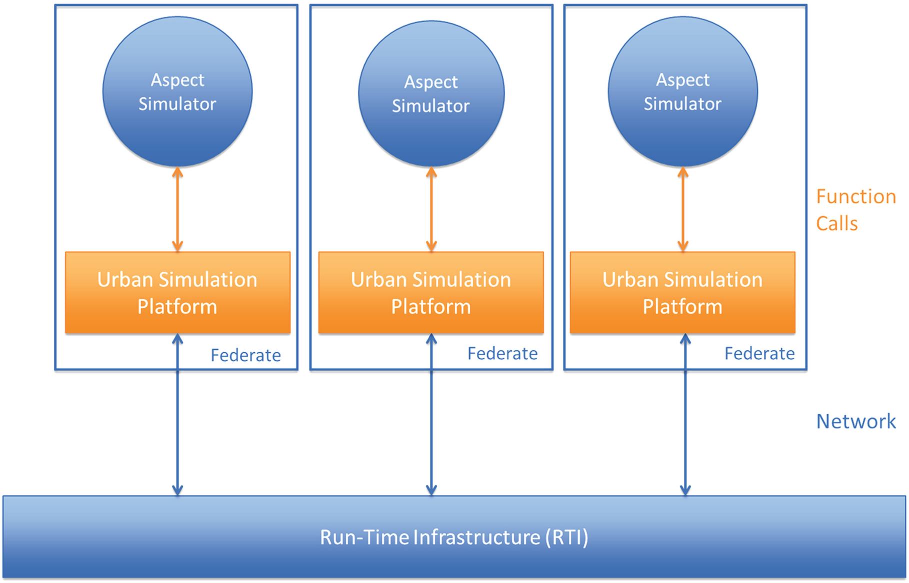
Aspect Simulator
Aspect Simulator
Aspect Simulator
Function Calls
Urban Simulation Platform
Urban Simulation Platform
Urban Simulation Platform
Federate
Federate
Federate
Network
Run-Time Infrastructure (RTI)

Fig. 3. AOP urban simulation architecture, a concrete realization of the canonical HLA.

Consider, for example, two different, but collaborating, simulations: a pedestrian simulation and a traffic simulation. When they are separated in different processes using our AOP architecture, the pedestrian simulation needs to know something about the cars (their position, at least), so that it prevents pedestrians from crossing streets when cars may hit them. But the representation of cars used by the pedestrian simulation isn't necessarily the same as that of the traffic simulation. Without the language that we are designing, car data would necessarily need to be transferred into the pedestrian simulation. With our language, that isn't necessary; the pedestrian simulation can simply issue a query into the data store that returns information about the pedestrians in danger without any car data having to be transferred. The following pseudo-code explains the idea:

```
[Set]
public class Pedestrian {
    int id;
    Position pos;
}

[Subset(Of = typeof(Pedestrian)]
public class EndangeredPedestrian : Pedestrian {
    Query = SELECT Pedestrian.* FROM Pedestrian JOIN Car ON
                   ABS (Pedestrian.pos - Car.pos) < 10;
}
```

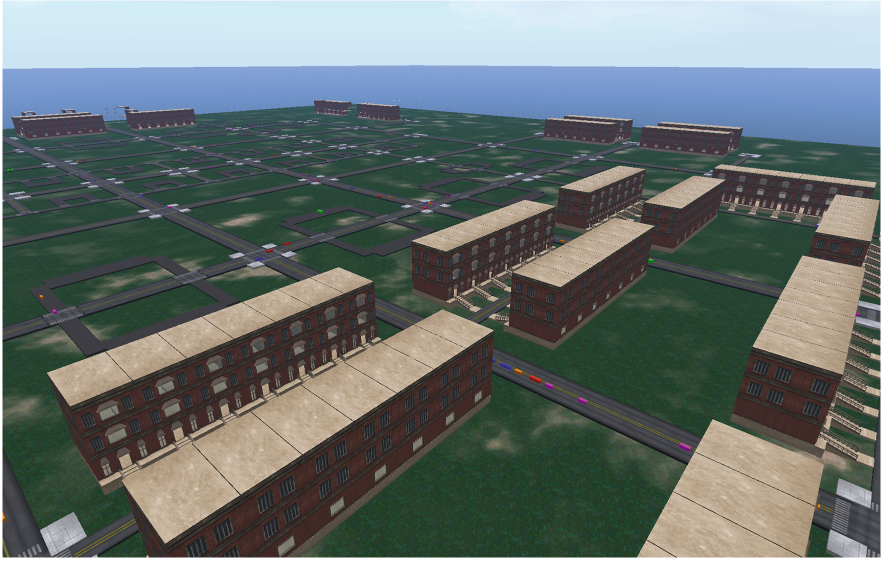

Fig. 4. View of the urban simulation platform prototype. The colored rectangles are vehicles, with each color indicating a different type of vehicle (Color figure online).

We are still in the early stages of prototyping this idea, but the results are encouraging. We have tested a working urban simulation with 3 aspects: socioeconomic, traffic, and virtual environment. Our work is an adaptation of a project called Mobdat, which is available open-sourced[2], and integrates the OpenSimulator VE platform, the SUMO traffic simulator, and its own implementation of socioeconomic factors, including daily routines (e.g. sleep, lunch, dinner, coffee), job, residence, and vehicles. Figure 4 shows a screenshot from an OpenSimulator viewer.

Our work so far is a proof of concept for the urban simulation platform. The ideas here will allow us to support independent teams with different expertise, each one being responsible for their own partial simulation without having to coordinate with the other teams other than on the knowledge of what kind of the data is shared.

6 Final Remarks

If AOP with data-only dependencies proves to be a good fit for distributed simulations, this work can have a tremendouns influence on the accessibility of distributed simulations for a large number of modeling and simulation experts, therefore also having a potential impact on policy making. While the Department of Defense (DoD) has been the primary force behind the development of distributed

[2] https://github.com/cmickeyb/mobdat.

simulations, nowadays, simulations are being used in a much wider spectrum of applications; science, logistics, social behavioral, transportation management, health, and land use are some domains where simulations play an increasingly important role. Parallel and distributed simulations, in particular, are becoming a necessity for complex simulations that don't fit one computer's hardware limits, not only in the military, but in all branches of Science and Engineering. However, as mentioned before, domain experts tend to stay away from distributed simulations, because of their software engineering complexity. Better and easier models and middleware are needed if we are to take advantage of the power of complex simulations.

References

1. Barnes, P.D., Carothers, C.D., Jefferson, D.R., LaPre, J.M.: Warp speed: executing time warp on 1,966,080 cores. In: Proceedings of the 2013 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation - SIGSIM-PADS 2013, New York, USA, p. 327. ACM, New York (2013). http://dl.acm.org/citation.cfm?id=2486134dl.acm.org/citation.cfm?d=2486092.2486134
2. Boer, C., Bruin, A., Verbraeck, A.: Distributed simulation in industry - a survey part 2 - experts on distributed simulation. In: Proceedings of the 2006 Winter Simulation Conference, pp. 1061–1068. IEEE, December 2006. http://dl.acm.org/citation.cfm?id=1218306, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4117719
3. Boer, C., De Bruin, A., Verbraeck, A.: Distributed simulation in industry - a survey part 1 - the cots vendors. In: Proceedings of the 2006 Winter Simulation Conference, pp. 1053–1060. IEEE, December 2006. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4117718, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4117718
4. Boer, C.A., de Bruin, A., Verbraeck, A.: Distributed simulation in industry - a survey part 3 - the HLA standard in industry. In: Proceedings of the 2008 Winter Simulation Conference, pp. 1094–1102. IEEE, December 2008. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4736178, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4736178
5. Bowman, M.: Mobdat. https://github.com/cmickeyb/mobdat
6. Bruzzone, A.G., Dato, L., Ferrando, A.: Simulation exploration experience: providing effective surveillance and defense for a moon base against threats from outer space. In: 2014 IEEE/ACM 18th International Symposium on Distributed Simulation and Real Time Applications, pp. 121–126, October 2014. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6957184
7. Cameron, G.D., Duncan, G.I.: Paramicsparallel microscopic simulation of road traffic. J. Supercomput. **10**(1), 25–53 (1996)
8. Carlsson, C., Hagsand, O.: DIVE a multi-user virtual reality system. In: Proceedings of IEEE Virtual Reality Annual International Symposium - VRAIS 1993, pp. 394–400 (1993). http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=380753
9. Committee, D.S.: IEEE Standard for Distributed Interactive Simulation-Application Protocols. IEEE Standard 1995 (1998)
10. Cutter, S.L., Ahearn, J.A., Amadei, B., Crawford, P., Eide, E.A., Galloway, G.E., Goodchild, M.F., Kunreuther, H.C., Li-Vollmer, M., Schoch-Spana, M., et al.: Disaster resilience: a national imperative. Environ. Sci. Policy Sustain. Dev. **55**(2), 25–29 (2013)

11. Elfrey, P.R., Zacharewicz, G., Ni, M.: Smackdown: adventures in simulation standards and interoperability. In: Winter Simulation Conference Proceedings of the Winter Simulation Conference, pp. 3963–3967 (2011)
12. Essilfie-Conduah, N., Grogan, P., Cunio, P.M., McLinko, R., de Weck, O.L.: A university perspective on the nasa/siso smackdown modeling and simulation outreach event (2011)
13. Falcone, A., Garro, A., Longo, F., Spadafora, F.: Simulation exploration experience: a communication system and a 3D real time visualization for a moon base simulated scenario. In: 2014 IEEE/ACM 18th International Symposium on Distributed Simulation and Real Time Applications, pp. 113–120, October 2014. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6957183
14. Fellendorf, M.: Vissim: a microscopic simulation tool to evaluate actuated signal control including bus priority. In: 64th Institute of Transportation Engineers Annual Meeting, pp. 1–9 (1994)
15. Fernandes, R., Vieira, F., Ferreira, M.: Parallel microscopic simulation of metropolitan-scale traffic. In: Proceedings of the 46th Annual Simulation Symposium, ANSS 2013, pp. 10:1–10:8. Society for Computer Simulation International, San Diego (2013). http://dl.acm.org/citation.cfm?id=2499604.2499614
16. Fujimoto, R.M.: Parallel and distributed simulation. In: Proceedings of the 2014 Winter Simulation Conference, WSC 2014, p. 5. IEEE Press, Piscataway (2014). http://dl.acm.org/citation.cfm?id=2693848.2693854
17. Gipps, P.G.: A behavioural car-following model for computer simulation. Transp. Res. Part B Methodological **15**(2), 105–111 (1981)
18. Horn, D., Cheslack-Postava, E., Mistree, B.F., Azim, T., Terrace, J., Freedman, M.J., Levis, P.: To infinity and not beyond: scaling communication in virtual worlds with Meru. Technical report, Stanford University (2010). http://ewencp.net/papers/to_infinity_and_not_beyond_tr.pdf
19. IEEE Standards Association: 1516–2000 IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Framework and Rules (2000)
20. Kaplan, J., Yankelovich, N.: Open wonderland: an extensible virtual world architecture. IEEE Internet Computing **15**(5), 38–45 (2011). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5871568
21. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C.V., Loingtier, J.-M., Irwin, J.: Aspect-oriented programming. In: Akşit, M., Matsuoka, S. (eds.) ECOOP 1997. LNCS, vol. 1241, pp. 220–242. Springer, Heidelberg (1997). http://dx.org/10.1007/BFb0053381
22. Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L.: Recent development and applications of SUMO - simulation of urban mobility. Int. J. Adv. Syst. Meas. **5**(3&4), 128–138 (2012)
23. Lake, D., Bowman, M., Liu, H.: Distributed scene graph to enable thousands of interacting users in a virtual environment. In: Proceedings of the 9th Annual Workshop on Network and Systems Support for Games - NetGames 2010, pp. 1–6. IEEE Computer Society, November 2010. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5679669
24. Leonard, D., Tough, J., Baguley, P.: Contram: a traffic assignment model for predicting flows and queues during peak periods. Technical report, Transport and Road Research Laboratory (1978)
25. Lighthill, M.J., Whitham, G.B.: On kinematic waves. II. a theory of traffic flow on long crowded roads. Proc. R. Soc. Lond. Ser. A Math. Phys. Sci. **229**(1178), 317–345 (1955)

26. Liu, H., Bowman, M.: Scale virtual worlds through dynamic load balancing. In: 2010 IEEE/ACM 14th International Symposium on Distributed Simulation and Real Time Applications, pp. 43–52, October 2010
27. Liu, H., Bowman, M., Adams, R., Hurliman, J., Lake, D.: Scaling virtual worlds: simulation requirements and challenges. In: Johansson, B., Jain, S., Montoya-Torres, J., Hugan, J., Yücesan, E. (eds.) Proceedings of the 2010 Winter Simulation Conference (WSC), WSC 2010, pp. 778–790. IEEE Computer Society, Baltimore, December 2010. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5679112
28. Maciejewski, M.: A comparison of microscopic traffic flow simulation systems for an urban area. Probl. Transportu **5**, 27–38 (2010)
29. Miller, D., Thorpe, J.: SIMNET: the advent of simulator networking. Proc. IEEE **83**(8), 1114–1123 (1995)
30. Nagatani, T.: Gas kinetic approach to two-dimensional traffic flow. J. Phys. Soc. Jpn. **65**(10), 3150–3152 (1996)
31. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. J. Phys. **2**(12), 2221–2229 (1992)
32. OpenSimulator: OpenSimulator. http://opensimulator.org
33. Potuzak, T.: Distributed-parallel road traffic simulator for clusters of multi-core computers. In: 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications, pp. 195–201, October 2012. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6365074
34. Pursula, M.: Simulation of traffic systems-an overview. J. Geogr. Inf. Decis. Anal. **3**(1), 1–8 (1999)
35. Smith, L., Beckman, R., Anson, D., Nagel, K., Williams, M.E.: Transims: transportation analysis and simulation system. In: Fifth National Conference on Transportation Planning Methods Applications-Volume II: A Compendium of Papers Based on a Conference Held in Seattle, Washington (1995)
36. Stafford, S.G., Bartels, D.M., Begay-Campbell, S., Bubier, J.L., Crittenden, J.C., Cutter, S.L., Delaney, J.R., Jordan, T.E., Kay, A.C., Libecap, G.D., et al.: Now is the time for action: transitions and tipping points in complex environmental systems. Environ. Sci. Policy Sustain. Dev. **52**(1), 38–45 (2010)
37. Strassburger, S., Schulze, T., Fujimoto, R.: Future trends in distributed simulation and distributed virtual environments: results of a peer study. In: Proceedings of the 2008 Winter Simulation Conference, pp. 777–785. IEEE, December 2008. http://dl.acm.org/citation.cfm?id=1516887, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4736140
38. Taylor, S.J., Revagar, N., Chambers, J., Yero, M., Anagnostou, A., Nouman, A., Chaudhry, N.R., Elfrey, P.R.: Simulation exploration experience: a distributed hybrid simulation of a lunar mining operation. In: 2014 IEEE/ACM 18th International Symposium on Distributed Simulation and Real Time Applications, pp. 107–112, October 2014. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6957182
39. Tolk, A.: Avoiding another Green Elephant - a proposal for the next generation HLA based on the model driven architecture. In: 2002 Fall Simulation Interoperability Workshop, Orlando, Florida, pp. 1–12, November 2010. http://arxiv.org/abs/1011.6671
40. United Nations: World Urbanization Prospects (2014). http://esa.un.org/unpd/wup/Highlights/WUP2014-Highlights.pdf

41. Waddell, P.: UrbanSim: modeling urban development for land use, transportation, and environmental planning. J. Am. Plan. Assoc. **68**(3), 297–314 (2002). http://www.tandfonline.com//abs/10.1080/01944360208976274
42. Waddell, P., Ulfarsson, G.: Introduction to urban simulation: design and developmentof operational models. In: Haynes, K., Stopher, P., Button, K., Hensher, D. (eds.) Handbooks in Transport, vol. 5. Elsevier, Oxford (2004)
43. Waters, R.C., Anderson, D.B., Barrus, J.W., Brogan, D.C., Casey, M.A., McKeown, S.G., Nitta, T., Sterns, I.B., Yerazunis, W.S.: Diamond park and spline: a social virtual reality system with 3D animation, spoken interaction, and runtime modifiability. Presence Teleoperators Virtual Environ. **6**(4), 461–480 (1997). http://citeseerx.ist.psu.edu/viewdoc/summary?=10.1.1.2.9500
44. White, J., Stevens, B., Holden, R.: Strategic Transport Infrastructure Needs to 2030. OECD, Paris (2012). http://trid.trb.org/view.aspx?id=1214312
45. Xu, M., Crittenden, J.C., Chen, Y., Thomas, V.M., Noonan, D.S., DesRoches, R., Brown, M.A., French, S.P.: Gigaton problems need gigaton solutions 1. Environ. Sci. Technol. **44**(11), 4037–4041 (2010)
46. Xu, Y., Tan, G.: An offline road network partitioning solution in distributed transportation simulation. In: 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), pp. 210–217. IEEE (2012)