Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

DISTRIBUTED, INTEGRATED AND INTERACTIVE TRAFFIC SIMULATIONS

Jayanth Raghothama
Sebastiaan Meijer

Gaming and Participatory Simulation Labs
School of Technology and Health
KTH Royal Institute of Technology
Stockholm, Sweden

ABSTRACT

Mainstream discourse in urban planning is in transition, due to shifts from a technical to a communicative perspective, and increased scrutiny and criticism of models and simulations. The cognizance of complexity in urban systems is imposing limitations on modeling. The added benefits of today's data and computational power make simulations harder to validate and understand. Reconciling the movements towards a communicative and exploratory approach as compared to a technical and predictive approach requires new methods for planning process and posits new requirements and functions for simulations. Based on distributed simulation and gaming simulation, the paper presents a framework to support the exploration of simulated and realistic virtual worlds in a participatory fashion, enabling new approaches to urban planning. The development and evaluation of the framework casts simulations in a new perspective and explores the context of use of simulations in planning and design.

1 INTRODUCTION

Complexity theory invites us to consider issues in planning as complex systems, which evolve under contextual pluralities and natural tendencies to self-organize. Complex systems such as cities show emergent properties and non-linear relationships between components, making them highly intractable and difficult to predict. Planning theory has in recent years started to recognize and tackle this challenge, whether from the point of view of conceptualizing coherent urban form (Salingaros 2000), articulating planning processes (Healey 1997), relating social theory, self-organization and planning rules (Portugali 1999, Alfasi and Portugali 2007), analyzing and simulating complex urban structures (Batty 2007).

The movements in planning theory have come about largely as a result of two forces: the increasing scrutiny and criticism that the models and processes used by planners have been subject to; and pressure to make the planning process more democratic and inclusive.

Modernist planning has been historically associated with a reduction in complexity. This reductionist technical-rational/rational-comprehensive perspective relies heavily on models and computer simulations, which serve as important toolkits for analysis, description and experimentation. Such technical-rational planning implied more control from the top-down through planners, who were expected to be able to foresee consequences through the use of analytical tools (Roo and Rauws 2012).

Models and simulations used in planning have been subject to constant criticism. With increasing computational power and sophistication in modeling techniques models started representing these systems at greater levels of detail and richness, requiring ever more computational power and data and proving harder to validate. Models tend to assume and define the system with clear boundaries, look for states of equilibrium for the system and for homogeneous elements within the system. However, complexity thinking suggests that such systems are impossible to close, equilibrium is hard to identify or often does not exist at all, and the richness of such systems comes from its heterogeneity. Central criticisms of models

978-1-4673-9743-8/15/$31.00 ©2015 IEEE

are therefore that the assumptions behind a lot of these models are no longer valid, or are changing rapidly; the implausibility of simulating open worlds in a (necessarily) closed simulation; lack of availability of comprehensive datasets that describe such systems and that the world these models describe have become simply too complex to be simulated adequately (Batty 2015, Grüne-Yanoff and Weirich 2010).

For example, in the development of Integrated Land Use and Transport models, scholars are calling for the development of a new generation of models that can address shortcomings in previous models and address new trends in society, such as the influence of information technology on travel patterns and behaviors (Wee 2015). Similar concerns exist in urban planning, where scholars increasingly question the limits of prediction, and are faced with question of increasing diversity and heterogeneity of actors and behaviors (Batty and Torrens 2005), links between urban and regional dynamics (Healey 2006) and the intersections between planning and complexity theory (Byrne 2003).

Further doubts about this rational-comprehensive perspective started towards the end of the 1950s when people like Lindblom (1959) and Davidoff (1965) started to criticize and question this approach. These doubts followed the realization that planning is essentially a political process, and the scientific tools were inadequate (Harvey 1973, Lee 1994). In response, Lindblom tried making the process more realistic and Davidoff tried to make it more democratic.

Such populist movements in urban planning started refuting the hegemony of the expert urban planner to make the process of planning more democratic and inclusive. Stemming from the need to address the specific needs of individuals, local communities started asserting their rights to organize their local environments participate in and influence the urban planning process. This led to a conflict between the traditional top-down model of urban planning and the self-organizing bottom-up influences of different actors. The solution was to envision cities as holistic systems consisting of multiple sub-systems organized under the influence of multiple networked actors (stakeholders), such as contractors, residents, planners, individuals and others (Hajer 2003, Healey 1997, Hughes and Sadler 2013, Innes 1996, Innes and Booher 2010, Jacobs 1961).

These dilemmas posit new requirements and functions for simulations and models in planning processes. Given that predictive ability is suspect, how should planners use models or the results thereof? How should the tensions between the top-down and bottom-up approaches to planning be reconciled, given that neither approach is enough to address planning issues by itself?

Scholars have suggested that integrated frameworks and methods that support participatory processes in the context of realistic explorations of the complex dynamics of systems and of different data sets are the best ways forward for planning support (Dearden and Wilson 2011, Devisch 2012, Bueren 2009). Such frameworks are needed to further develop methodologies. Without technological capabilities, it will be hard to explore successful methods for overcoming the issues in current methods.

In the following sections we present a technical framework that is capable of supporting integrated methods for planning support. Based on distributed simulation methods, the framework presents complex realities for exploration through gaming methods and techniques, backed by real data to enhance fidelity.

2 SIMULATIONS IN URBAN PLANNING

Highly interconnected systems need to be explained and managed across many layers - technological, social, data, human and institutional (Palfrey and Gasser 2012). The networked society (Castells 1996) and the rise of computation have created new possibilities that did not exist before: Geographical Information Systems (GIS) technologies for analysis, design and visualization, connected mobile phones that enable real time monitoring and data collection; virtual reality techniques that allow for real time exploration of future cities; and so on. The increasing sophistication of urban simulation models backed by complexity theories allows for the study of the dynamics of cities. While each of these technologies support planning, integrating them together results in a more comprehensive Planning Support System (PSS), as seen in Figure 1.

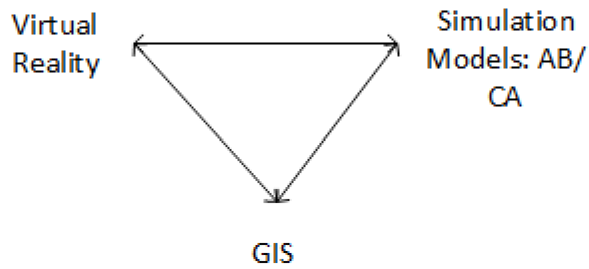

Figure 1: A typical planning support system. Source: Portugali 2012

This integrated system is advocated as the state-of-the-art in PSS (Brail 2006), which typically consists of three component parts (Figure 1): a set of simulation models, usually Agent Based (AB) and/or Cellular Automata (CA), a GIS system, and a set of 2D/3D Virtual Reality (VR) components. The AB/CA models allow for the simulation of future scenarios, GIS systems form the database for such scenarios and VR components are used to see the results. Such systems therefore should enable planners to experience a planning scenario rather than trying to predict (Portugali 1999, Portugali 2012). Inclusion of more complex models allows for detailed explorations. Such structures reformulate the role of simulations: supporting exploration instead of prediction; and facilitating participation instead of providing evidence. Requirements for simulation then become interactivity and immersion (immersion either through the simulation or facilitated through a VR component), and functionality becomes exploration.

Gaming simulation has been used in urban and regional planning. Games provide ways to collectively decide on the problem formulation, system boundaries and on the dynamics of the system that will be addressed. Then, policies can be formulated and tested in this simulated environment (Duke and Geurts 2004, Greenblat and Duke 1981). Lately, games are usually constructed by combining sophisticated computer simulations with interaction and role play (Dearden and Wilson 2011). Simulations provide a realistic context to the policy being tested. In urban planning, there are a number of examples of games being used: in transportation planning, SimCity appears in planning curricula for learning (Raghothama and Meijer 2014, Gaber 2007, Minnery and Searle 2014). Gaming simulation methods supported by integrated planning support systems can potentially provide ways to collectively explore realistic virtual environments and formulate policies and plans. This hybrid of games and computer based simulations form the first step in reconciling the dilemmas mentioned earlier, and form the core of the planning support system as envisioned in this paper.

Parallel and Distributed Simulations is a field of study concerned with using multiple processors and computers to run a simulation with the aim of increasing computational efficiency, and connecting multiple heterogeneous autonomous simulations so they can inter-operate with each other and simulate a more complex scenario than would be possible by a single simulation. One goal of distributed simulations is to build Distributed Virtual Environments (DVEs), which create networked, immersive and interactive environments with high levels of fidelity and aim to provide scalable real time performance for the purposes of training and decision support (Fujimoto 1999, Perumalla 2006).

An analogy can be drawn here between the requirements (in terms of immersion and interactivity) for training and planning. Planning requires similar levels of immersion and fidelity in the simulated world. The concepts and structures within distributed simulations can be used for the construction of future planning support systems. Several standards exist to facilitate such distribution and inter-operation, such as Distributed Interactive Simulations, and High Level Architecture (HLA). The High Level Architecture is an IEEE standard, and despite being widely used in military applications, is not very prevalent in civilian or non-military applications. Implementing and using HLA based distributed simulations requires high degrees of low level technical knowledge, and simulations must be designed and developed to inter-operate. The simulations and models currently used in urban planning are not HLA aware, do not generally inter-operate with other simulations and tools and modifying the simulations to comply with the HLA standard is infeasible. A more feasible solution is to adopt the methods of distributed simulation to achieve simulation integration.

The framework is implemented by integrating a Commercial or Off The Shelf (COTS) microscopic traffic simulator, a microscopic public transport simulator and a pedestrian simulator, using methods from distributed simulations. The results of the integrated simulation is animated and visualized in real time in a gaming engine, which also controls the clocks of all simulations and provides interactivity to the simulations. The gaming engine automatically generates realistic virtual cities through open data sources. The simulation federation also leverages public data streams to enhance realism. APIs (Application Programming Interface) of the simulations enable changing the simulation state at run time, through the gaming engine, facilitating exploration of multiple choices and policies over a single run.

Synchronization methods are employed between the simulations, and between the federation and the gaming engine. The integration mechanisms are generalized, and can potentially be used across any spatio-temporal simulation. Contrary to popular methods of distributing simulations over space or time (Perumalla 2006), the integrated simulation is distributed thematically, i.e. the traffic simulation, pedestrian simulation and associated middlewares run on different machines, over the same space and time.

In the following sections, we describe the architecture and some details of the implementation of the technical framework, followed by its evaluation in the context of two use cases in the cities of Paris and Stockholm. We describe some future work and discuss the implications for simulations in planning.

3 ARCHITECTURE

The architecture has three main components, as represented in figure 2. The first is a gaming and visualization component, which automatically generates a 3D geography of interest from OpenStreetMaps (OSM). The second component is a federation of simulations, which interact with each other through a synchronization API. The result from the integrated federation is sent in real time to the gaming layer, where they are animated. Through the control API, the gaming layer assumes control of the federation, and people can interact with the federation through the gaming layer. Users can control the clock, speed up, slow down, and change parameters and so on. A buffer of the federation results allows people to also go backwards and forwards in time through the federation results. The buffer is stored in the database, as are other data streams which provide realism to the simulations.

All the components are distributed and communicate over the network using TCP/IP or UDP. The gaming component is implemented using Unity3D, a commercial gaming engine. The Data/Control API, the Time Synchronization API, access control to the database and other services such as logic for procedural modeling and so on are implemented using Microsoft's Web Communication Framework (WCF), implemented in C#. The traffic simulation used is Simulation of Urban MObility (SUMO) (Krajzewicz et al. 2012), implemented in C++, and open source microscopic traffic and network simulator. PostgreSQL contains the GIS data, and MongoDB stores unstructured real time data, such as public transport information from the cities (Raghothama et al. 2015).

The following sections describe each component, and their interactions in detail.

4 IMPLEMENTATION

4.1 City Generation

The virtual world represents a real city, and is generated within the Unity3D game engine using data from OpenStreetMaps. OSM publishes geometries and geographies for urban features, such as road networks (defined as inter-connected lines with latitude and longitude points), buildings (defined as polygons, with latitude and longitude for every point in the polygon), traffic signals, footways and so on. One-ways, pedestrian footways, highway classifications and so on are also included. XML or binary data from OSM is parsed and stored in a PostGreSQL database.

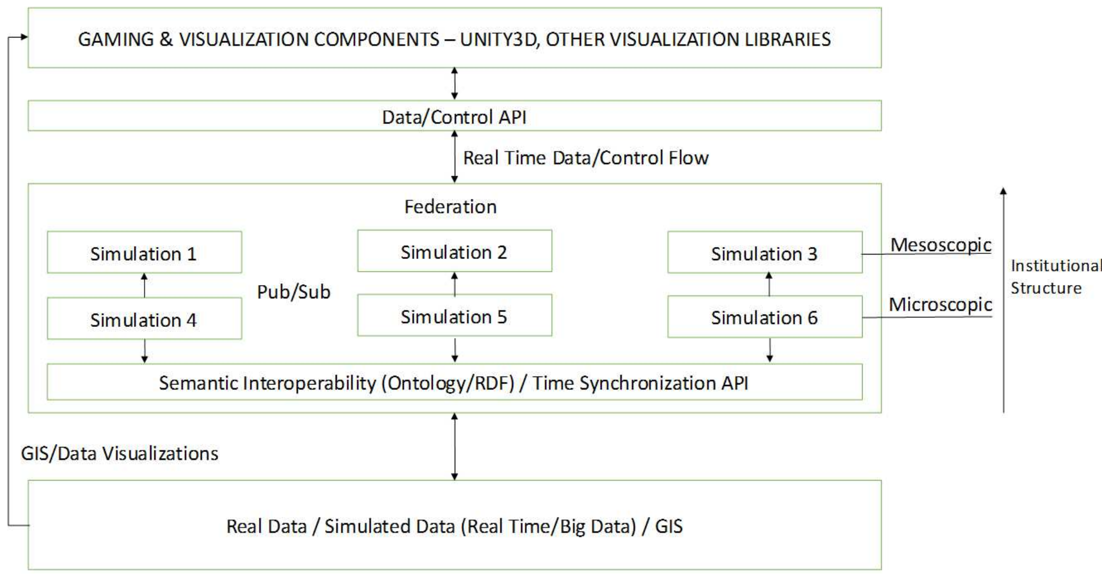

Figure 2: Conceptual Architecture

The Unity engine then draws the different elements of the city from this database. Roads are drawn with sequences of flat rectangles, setting the width according to highway classifications available in the OSM data, with textures applied accordingly. For buildings, a mesh was extruded to resemble the polygon, textured and height was applied either based on data or randomly. Other structures such as parks were drawn as a flat textured rectangle.

The geographical co-ordinates were translated to the Unity system and the map was set from the origin point (0,0,0) of the virtual world. All the graphic elements drawn into the virtual environment preserved their meta-information in the database.

4.2 SUMO Traffic Simulator

SUMO is an open source, highly portable, microscopic and continuous road traffic simulation able to handle large road networks. The traffic simulation is prepared by generating the road network from the OSM data. The geography of the simulation should match the geography of the 3D city. While either geography can be larger or smaller than the other, simulation results out of the scope of the 3D city appears to be driving on a flat space with no features around, rendering it unrealistic. The simulation files are created using OSM data and travel demand data where available. SUMO results are sent over a network to the Unity game engine to be animated within the 3D city.

Figure 3 shows a diagram of the flow of the interaction between the Unity engine and SUMO, each step implementing a functionality that form the final communication chain.

The Unity engine assumes control over the SUMO simulation using the TraCI interface (Traffic Control Interface), an interface which enables remote communication with SUMO over TCP/IP (Wegener et al. 2008). Unity connects as a client with SUMO running as a TraCI server, and once connected, Unity can control the simulation clock. The Unity engine implements a listener waiting for SUMO results, in the form of Floating Car Data (FCD, data containing a location for every vehicle, for every time stamp) to be sent from SUMO over another TCP connection to collate the results of the simulation. TCP is chosen for this over UDP for data consistency, and the FCD is packaged as XML (eXtended Markup Language). SUMO generates FCD for every vehicle in the simulation for every timestep, and packages that into an XML object, and sends it over the TCP channel to the listener. Parallel data streams are chosen for computing efficiency; limitations of the TraCI interface render inefficient the use of only one channel for communication. A

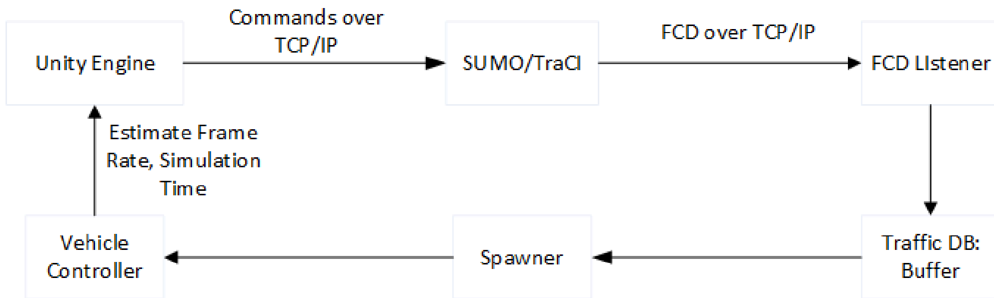

Figure 3: Components for SUMO-Unity Communication

buffer stores the results of all vehicles being simulated per timestep. This buffer allows the user to scroll backwards through the simulation.

A spawner reads incoming data from the buffer and creates vehicles for each new vehicle and removes vehicles no longer in the scene. A vehicle controller updates positions for each car for each timestep, and stops updating vehicles no longer within the camera frustum, improving performance.

Conservative synchronization (Perumalla 2006) is employed in the communication between Unity and SUMO. The results being animated within Unity must be the latest results from the simulation, i.e. the animation and simulation should proceed in lockstep. Despite Unity being in control of the SUMO clock (Unity can request the next timestep whenever necessary), this is hard to achieve. If SUMO proceeds ahead of the animation, then the people interacting with the simulation will react to outdated data. If Unity completes the latest timestep animation before the next timestep computes and arrives, it will not be possible to get good frame rates required for smooth animation.

Key variables for synchronization are the frame rates, the number of vehicles in the current timestep in the simulation, and the time required to animate the vehicles (spawn new vehicles or update old vehicles from their old position to the new one.) The next timestep has to be requested, and should be parsed and made available for the spawner and vehicle controller by the time finish with the current timestep. Given that there is no control over how long SUMO takes to compute one timestep (this depends on the number of vehicles and the total number of timesteps to simulate), all optimizations and control has to be done on the Unity side. The time delay between requests for the next timestep is decided by continuously estimating the delay for the next timestep's arrival from previous delays.

4.3 SiPS Pedestrian Simulator

SiPS (Simulation of Pedestrian Students) is a proprietary simulator for modeling social and decision-making behaviors for pedestrians. Navigation is achieved through the Unity navigation mesh (navmesh), which provides realistic steering for virtual characters. During city generation, the navmesh is generated automatically on foot ways. This navigation system also includes collision avoidance and dynamic re-pathing, which allows pedestrians to avoid dynamic obstacles (such as vehicles) and other agents of the system with automatic re-pathing in real-time.

The design of the decision-making process for pedestrians is done through a custom built Finite State Machine (See Figure 4 for an example). This tool, widely used in games for representing artificial behavior and artificial intelligence, allows the representation of different states of thinking of an agent (pedestrian) in an easy and highly understandable way (Cass 2002). This makes it easy to understand and change the decision logic of the agents, both for developers and especially for experts who may not knowledgeable about programming.

Pedestrian agents have knowledge about the world, can look up information through interfaces to open data APIs and have properties based on the 3D object that represents them in the geography. These properties can be leveraged to provide social behaviors, for example proximity can be used to pass messages

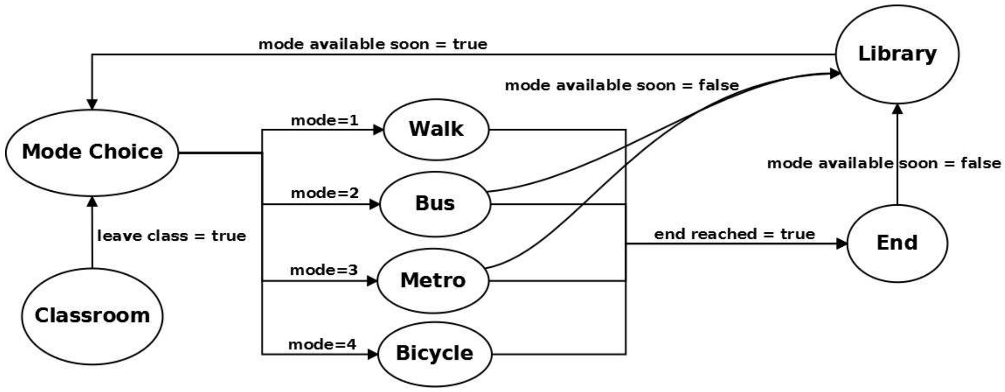

Figure 4: Finite state machine that implements the decision-making for pedestrians.

between agents. Users can also change the FSM to implement their own pedestrian models, providing a sandbox for experimentation. Similar to the SUMO animation, a spawner generates the pedestrians.

4.4 Interaction between Simulations

Integrating both pedestrian and SUMO simulations vastly expands the scope of the environment. The integration is achieved in Unity, by running both simulations at the same time, enabling communication between them and ensuring their respective agents can interact. The integration is achieved by making pedestrians and vehicles aware of each other, and to allow their respective simulations to take corrective measures.

Pedestrians become aware of vehicles through the Unity navigation mesh. Pedestrians prioritize the foot ways, but when they move onto the road this awareness is activated. This awareness is implemented with a sphere centered in the pedestrian (See Figure 5). The radius of the sphere is set to 20 m (in scale), assuming that this is the average distance at which drivers will start stopping when seeing pedestrians on the road, but this radius is also parameterized and can be changed by the user.

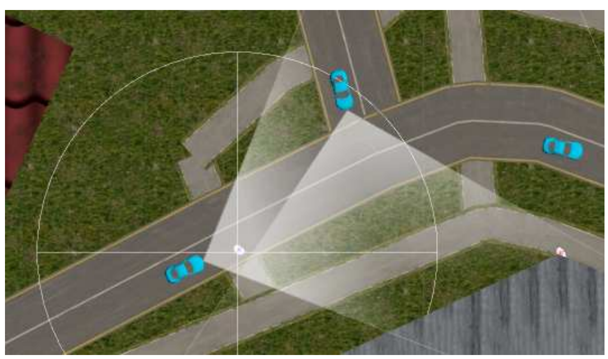

Figure 5: Screenshot of the simulation that shows a pedestrian crossing. The circle represents the area of awareness: the vehicles inside the circle and going towards the pedestrian (in this figure, the vehicle on the left) will be forced to brake.

When the pedestrian crosses a road, the sphere will inform all vehicles around. Then, the vehicles will calculate if they are heading towards the pedestrians and if they need to stop. To do so, two angles of +45 and -45 degrees are drawn from the front vector of the vehicle and they check if the pedestrians are inside any of them. The vehicles that detect pedestrians inside that angle will brake immediately and stop in the next timestep of the simulation.

An additional approach in which vehicles were casting for pedestrians was also implemented, but it turned out to be very computationally expensive, since cars were continuously checking for pedestrians, resulting in a drop in the frame rate. With our implementation, pedestrians only check for vehicles when they are crossing the road, without a need for checking in cases when they are walking in foot ways.

As described above, pedestrians that send messages to vehicles force them to stop. This braking behavior was implemented on the SUMO side. A braking method sends a TraCI command to SUMO to stop the vehicle. SUMO will subsequently react and will adapt the simulation accordingly. The time the vehicle will remain stopped is determined by a customizable parameter which specifies the number of seconds (timesteps) that the vehicle will stay standby until it checks for pedestrians again. In our design we set that parameter to 3 seconds, since that is the minimum time a pedestrian needs to cross a road in our simulation. However, if there are pedestrians still on the road, the vehicle will remain stopped, so the driver will be forced to wait until there are no more pedestrians in the way.

The same synchronization method as used for Unity-SUMO communication is used in this integrated scenario as well. Given that the pedestrian simulation has no concept of time, all simulations and animation can be synchronized to a real clock, and Unity and SUMO can be synchronized to a barrier (which is the latest time step).

5 USE CASES

The framework was evaluated in the context of two use cases, one each in the cities of Stockholm and Paris. The problem for the city of Stockholm is to change the scheduling of university classes to reduce congestion in traffic and public transport. Three major universities in the city of Stockholm are located close to each other, and leads to a lot of congestion. The challenge is to provide appropriate information to students so that they leave classes at times when congestion is less likely to occur.

The 3D geography for the case study application is a small area around the technical university. SUMO simulates traffic and pedestrian agents function as students. The user can make choices on delaying public transport, sending information to the agents, canceling transport choices, changing vehicle densities and so on for the simulation, apart from the options of scrolling backwards and forwards, speeding up or slowing down, pausing and restarting the simulation. Pedestrians and cars react to each other, allowing the user to experience how traffic reacts based on changes.

The problem for the city of Paris is to manage crowds and security during large events in two stadiums. The police commanders want to design the physical area around the stadiums, such as placing road barriers, cameras, police personnel, vehicles, etc. around the stadiums to ensure that the crowd behaves accordingly and to ensure appropriate response in case of a crisis.

The 3D geography for the use case is an area around the stadiums. The crowd is simulated in a proprietary simulation engine developed by a French defense company. The artificial intelligence (AI) model for the pedestrians is very complex, allowing for a wide range of behaviors. There are different types of pedestrian models, such as regular people, very important persons (VIPs), police, terrorists, criminals and so on. Different behaviors are activated by simple tweaking of agent parameters. Traffic is simulated in SUMO, and the same interaction model and interface is used, except that the traffic in this case is minimal since a lot of roads will be blocked. The simulation is controlled through a TCP connection, while data are streamed from the simulation through a User Datagram Protocol (UDP) stream.

The players have the option of completely configuring the physical space around the stadium. They can block/unblock roads, move or introduce cameras, add objects such as ticket barriers, change the operational status of different objects such as automated teller machines (ATMs), change the schedules of VIP arrivals, deploy additional forces, change pedestrian parameters and so on. Once again, there is a wide design space available for the players. Different cameras provide different views, and the views available to each player can even be filtered based on model subtype; for example, some players can see only pedestrians, some can see only VIPs, and so on.

6 FUTURE WORK

The simulations integrated are currently all microscopic. The parameters in these simulations can be related to and integrated with Systems Dynamics models, which present an overview of the system from a macroscopic perspective. This will enable transitions for the users across scales, moving to greater levels of detail when necessary, for example, for the investigation of behaviors of individuals.

Cities generate and publish a lot of data on the public domain, which form rich resources for exploration. These data streams can all be integrated into the virtual world, and annotated to the physical objects. For example, public transport delay information can be associated with bus and metro stops. The heterogeneity of the data streams needs to be managed somehow, through a semantic data manager since it can augment both the simulations and the visualizations.

Given that multiple heterogeneous simulations are integrated, the scenarios for the simulations need to be managed. The question is whether the scenario is defined for the federation as a whole or for each simulation separately. Further, a scenario description for the user interaction (game play) stemming from the problem description needs to be defined. User interactions and simulations logs can all feed into scenario descriptions for the future. A scenario manager to manage different scenarios and scenario descriptions needs to be built.

Validation is a critical issue. The entire federation needs to be validated, as well as the virtual environment. Again, the question of whether the entire framework needs to be validated as a whole, which is a hard task, or whether partial and sub-validations can be done remains. The agents have properties stemming from their own models and physical properties derived from the 3D object, and both these sets of properties need to be validated. The validation and fidelity of the representation depends on the function of the architecture. For some components, such as the traffic simulation, the fidelity can be very high but for some components such as the pedestrian sandboxes (where users can create models) validation will be hard to achieve. Also given that users interact continuously with the virtual world, their interactions and the effects on the simulation will need to be tested and validated (Balci 2003).

7 CONCLUSION

The paper describes the implementation of a distributed, integrated and interactive virtual world capable of supporting methods and processes for planning support. Through the use of integrated simulations, procedural modeling and real data streams, the virtual world creates realistic environments for exploration.

The simulations all run on different machines over a network, and animated in real time within the gaming engine. The gaming engine assigns more behaviors to the agents, such as physical attributes of space and collision detection etc. Every agent (whether from the pedestrian simulation or SUMO) has behaviors that are derived both from the game engine and from the simulation, and these augmented behaviors are leveraged for the integration. These behaviors (or other behaviors based on 3D properties) can be created and added to the objects in the gaming engine, enabling further interactions in this physical space. For example, collision detection and proximity is leveraged in the pedestrian simulation for message passing. This interaction is based on methods from distributed simulation, but also represents a general way of achieving interaction among simulations which were not designed of developed to interact with other simulation.

The simulations are based on real data, such as Origin Destination matrices for trip generations, public transport data for delay information and schedules, OSM data for the geography and so on. This enhances the realism of the simulation federation.

Integrating simulations expands the scope of the simulated environment. Integrating also means that each individual aspect of the system need not be modeled again, but only the higher order logic of their interaction needs to be implemented. For example, when using SUMO and the French pedestrian simulation, only the logic of their interaction needed to be implemented.

Moving away from attempting to make accurate predictions and forecasts, to accurately representing dynamics to support experimentation, participation and exploration is a new approach to simulations in planning. The need for more accurate models remains, but given that comprehensive representations of systems is becoming hard to achieve, perhaps simulations should aim to represent the dynamics from different perspectives and approaches.

Simulations can function as sandboxes and only build enough structures to allow users to create their own models, from their perspectives. Simulations can provide interactivity and allow for the exploration of systems, where users can play with the system and augment the models with their knowledge. Simulations can provide context to these sandboxes, functioning as background representations of the system while the user changes the system and some models to create something new.

These conceptualizations for simulations stem mostly from the complexity of systems which is hard to capture in models, and the need for participation and multiple perspectives in planning. They posit new requirements for simulations, such as flexibility and the ability to easily add or change the models within, the need for interaction and control, the abilities to interface with other systems, such as visualization systems and data streams and so on. The standards for simulation inter-operability do not suffice, at least within the urban planning domain since there is already a vast body of work already done that does not conform to these standards. Further, these standards need to be extended to cater to these new requirements, primarily flexibility. However, in the short term, the rich body of work within simulation and distributed simulations can be leveraged to use current simulations in urban planning for these newer functions.

This can be achieved by leveraging COTS (Commercial or Off-The-Shelf) simulations and technologies, and other proprietary technologies such as the Unity gaming engine, which vastly reduce the implementation time. Simulations were not built with flexibility and interactivity as main requirements, and a large number of optimizations were necessary, to ensure performance but keeping consistency in mind. Choosing an XML parser over an XML deserializer, adopting the right synchronization methods, not rendering or animating or processing objects outside the camera frustum are examples.

Positioning simulations as facilitators for exploration can increase trust. In evaluations with stakeholders for the Paris and Stockholm use cases, we realized stakeholders are sensitive to what is hidden in the model, hidden biases and limitations, and so on. Modeling or designing a city necessarily involves multiple communities of practice, and while the virtual world can create open dialogue, it is constrained to a large extent by what the underlying model provides. Enhancing dialogue, providing open access to data and information requires that these models not be black boxes, but should be open reflecting the open system it represents.

The technological capabilities developed, and the framework enables the development of new methodologies, which can overcome limitations in current methods. We believe that simulations can function effectively at facilitating exploration, supporting design and planning and supporting participation. These requirements form a new paradigm for simulations, supporting new methods for planning.

ACKNOWLEDGMENTS

The research described in this paper was conducted while the authors were affiliated with the Department of Transport Science, School of Architecture and Built Environment, KTH, Sweden. The authors would like to acknowledge EIT ICT Labs and TRENoP who jointly funded this research effort. The authors would also like to acknowledge Mohammed Azhari and Miguel Ramos Carretero for the development, Sagar Arlekar and Michael van den Berg for their insights on the programming effort that went into this.

REFERENCES

Alfasi, N., and J. Portugali. 2007, July. "Planning rules for a self-planned city". *Planning Theory* 6 (2): 164–182.

Balci, O. 2003. "Verification, Validation, and Certification of Modeling and Simulation Applications: Verification, Validation, and Certification of Modeling and Simulation Applications". In *Proceedings of the 35th Conference on Winter Simulation: Driving Innovation*, WSC '03, 150–158. New Orleans, Louisiana: Winter Simulation Conference.

Batty, M. 2007. *Cities and Complexity: Understanding Cities with Cellular Automata, Agent-Based Models, and Fractals*. The MIT Press.

Batty, M. 2015. "Models again: their role in planning and prediction". *Environment and Planning B: Planning and Design* 42 (2): 191–194.

Batty, M., and P. M. Torrens. 2005. "Modelling and prediction in a complex world". *Futures* 37 (7): 745–766.

Brail, R. K. 2006. "Planning Support Systems Evolving: When the Rubber Hits the Road". In *Complex Artificial Environments*, edited by J. Portugali, 307–317. Springer Berlin Heidelberg.

Bueren, E. v. 2009. *Greening Governance: An Evolutionary Approach to Policy Making for a Sustainable Built Environment*. IOS Press.

Byrne, D. 2003, November. "Complexity Theory and Planning Theory: A Necessary Encounter". *Planning Theory* 2 (3): 171–178.

Cass, S. 2002, December. "Mind games [computer game AI]". *IEEE Spectrum* 39 (12): 40–44.

Castells, M. 1996. *The Information Age: Economy, Society, and Culture. Volume I: The Rise of the Network Society*. Blackwell.

Davidoff, P. 1965, November. "Advocacy and Pluralism in Planning". *Journal of the American Institute of Planners* 31 (4): 331–338.

Dearden, J., and A. Wilson. 2011, July. "Using Participatory Computer Simulation to Explore the Process of Urban Evolution". *Transactions in GIS* 15 (3): 273–289.

Devisch, O. 2012. *The metaverse as lab to experiment with problems of organized complexity.* Ashgate Publishing.

Duke, R. D., and J. Geurts. 2004. *Policy Games for Strategic Management*. Rozenberg Publishers.

Fujimoto, R. M. 1999. "Exploiting Temporal Uncertainty in Parallel and Distributed Simulations". In *Proceedings of the Thirteenth Workshop on Parallel and Distributed Simulation*, PADS '99, 46–53. Washington, DC, USA: IEEE Computer Society.

Gaber, J. 2007, December. "Simulating Planning SimCity as a Pedagogical Tool". *Journal of Planning Education and Research* 27 (2): 113–121.

Greenblat, C. S., and R. D. Duke. 1981. *Principles and practices of gaming-simulation*. Sage Publications Beverly Hills, CA.

Grüne-Yanoff, T., and P. Weirich. 2010, February. "The Philosophy and Epistemology of Simulation: A Review". *Simulation & Gaming* 41 (1): 20–50.

Hajer, M. A. 2003, May. *Deliberative Policy Analysis: Understanding Governance in the Network Society*. Cambridge University Press.

Harvey, D. 1973, September. *Social justice and the city*. Johns Hopkins University Press.

Healey, P. 1997, January. *Collaborative Planning: Shaping Places in Fragmented Societies*. UBC Press.

Healey, P. 2006, May. "Relational complexity and the imaginative power of strategic spatial planning". *European Planning Studies* 14 (4): 525–546.

Hughes, J., and S. Sadler. 2013, January. *Non-Plan: Essays on Freedom, Participation and Change in Modern Architecture and Urbanism*. Routledge.

Innes, J. E. 1996, December. "Planning Through Consensus Building: A New View of the Comprehensive Planning Ideal". *Journal of the American Planning Association* 62 (4): 460–472.

Innes, J. E., and D. E. Booher. 2010, January. *Planning with Complexity: An Introduction to Collaborative Rationality for Public Policy*. Routledge.

Jacobs, J. 1961. *The death and life of great American cities*. Random House LLC.

Krajzewicz, D., J. Erdmann, M. Behrisch, and L. Bieker. 2012, December. "Recent Development and Applications of SUMO - Simulation of Urban MObility". *International Journal On Advances in Systems and Measurements* 5 (3&4): 128–138.

Lee, D. B. 1994, March. "Retrospective on Large-Scale Urban Models". *Journal of the American Planning Association* 60 (1): 35–40.

Lindblom, C. E. 1959, April. "The Science of "Muddling Through"". *Public Administration Review* 19 (2): 79–88.

Minnery, J., and G. Searle. 2014, January. "Toying with the City? Using the Computer Game SimCity4 in Planning Education". *Planning Practice and Research* 29 (1): 41–55.

Palfrey, J. G., and U. Gasser. 2012. *Interop: The Promise and Perils of Highly Interconnected Systems*. Basic Books.

Perumalla, K. S. 2006. "Parallel and Distributed Simulation: Traditional Techniques and Recent Advances". In *Proceedings of the 38th Conference on Winter Simulation*, WSC '06, 84–95. Monterey, California: Winter Simulation Conference.

Portugali, J. 1999, November. *Self-Organization and the City*. Springer Science & Business Media.

Portugali, J. 2012. "Complexity Theories of Cities: Implications to Urban Planning". In *Complexity Theories of Cities Have Come of Age*, edited by J. Portugali, H. Meyer, E. Stolk, and E. Tan, 221–244. Springer Berlin Heidelberg.

Raghothama, J., M. Azhari, M. R. Carretero, and S. Meijer. 2015. "Architectures for Distributed, Interactive and Integrated Traffic Simulations". In *Forthcoming*.

Raghothama, J., and S. Meijer. 2014. "A Review of Gaming Simulation in Transportation". In *Frontiers in Gaming Simulation*, edited by S. Meijer and R. Smeds, Volume 8264 of *Lecture Notes in Computer Science*, 237–244. Springer International Publishing.

Roo, G. d., and W. S. Rauws. 2012. "Positioning Planning in the World of Order, Chaos and Complexity: On Perspectives, Behaviour and Interventions in a Non-linear Environment". In *Complexity Theories of Cities Have Come of Age*, edited by J. Portugali, H. Meyer, E. Stolk, and E. Tan, 207–220. Springer Berlin Heidelberg.

Salingaros, N. A. 2000, October. "Complexity and Urban Coherence". *Journal of Urban Design* 5 (3): 291–316.

Wee, B. v. 2015, May. "Toward a new generation of land use transport interaction models: a viewpoint". *Journal of Transport and Land Use* 0 (0).

Wegener, A., M. Pirkowski, M. Raya, H. Hellbrck, S. Fischer, and J.-P. Hubaux. 2008. "TraCI: An Interface for Coupling Road Traffic and Network Simulators". In *Proceedings of the 11th Communications and Networking Simulation Symposium*, CNS '08, 155–163. New York, NY, USA: ACM.

AUTHOR BIOGRAPHIES

JAYANTH RAGHOTHAMA is a PhD student at KTH Royal Institute of Technology, Sweden. His research interests focus on the intersection between simulation, gaming and design in complex systems. His email address is jayanthr@kth.se

SEBASTIAAN MIEJER is Professor of Health Care Logistics at KTH Royal Institute of Technology and part-time associated with TU Delft. He leads the GAming and Participatory Simulation lab (GaPSlabs), focusing on games, simulations and visualizations for design of complex systems in society. Sebas is currently coordinator for several EU projects, chairman of ISAGA, and frequent reviewer for a series of journals and conferences, especially when gaming and simulation methods are involved. His email address is smeijer@kth.se