Design and Implementation of An Agent-Based Simulation for Emergency Response and Crisis Management*

Timothy Schoenharl and Greg Madey[1]

Submitted: January 2010; Accepted: January 2011

ABSTRACT

This paper introduces the simulation component of the WIPER system [Schoenharl et al., 2006]. The Wireless Integrated Phone-based Emergency Response (WIPER) system is a Dynamic Data-Driven Application System (DDDAS) that uses a stream of cellular network activity to detect, classify and predict crisis events. The WIPER simulation is essential to classification and prediction tasks, as the simulations model human activity, both in movement and cell phone activity, in an attempt to better understand crisis events. These Agent-Based simulations are parameterized with agent location data from the cellular service provider and agents inhabit a GIS space representative of the urban area. Simulations include models for normal and crisis behaviors in an urban setting. Simulations generate call activity and agent locations, similar to that generated by the observation of cell phone users in an urban setting. A taxonomy of crisis events is presented, which has simplified the development of the simulation. We present an overview of the design and implementation of the WIPER simulation along with validation and verification of the system and an evaluation of runtime characteristics.

1. INTRODUCTION

Fires, riots, traffic jams and terrorist attacks are crisis events that can impact our lives. When possible, prevention is best. However, these events cannot be entirely eliminated, and so it is advantageous to spend time and effort developing techniques to mitigate their impact. Simulations of these events can help us to understand the development of these events, as well as evaluating strategies for dealing with these crises.

*The research on WIPER is supported by an NSF Grant, CISE/CNS-DDDAS, Award # 0540348.
[1]Corresponding author. craig.c.douglas@gmail.com

Crisis events occur without warning on short time scales relative to normal human activities. Simulations designed to model human behavior under crisis scenarios faces several challenges when attempting to model behavior in real time. The DDDAS approach seeks to address that challenge by incorporating real time sensor data into running simulations[?][?]. Dynamic, Data-Driven Application Systems is an approach to developing systems incorporating sensors and simulations where the simulations receive streaming data from the sensors and the sensors receive control information from the simulations.

WIPER works to detect and predict the course of crisis events and advise emergency response planners with up to date information on crisis scenarios. WIPER works by monitoring a stream of cell phone activity data to detect anomalies, then uses Agent-Based Modeling simulations, created from and updated against the streaming data, to provide more thorough understanding of crisis events. Finally, the information regarding potential anomalies, call information and simulation output is presented to users in a web-based console. A thorough introduction to the WIPER system can be found in [Schoenharl et al., 2006].

Agent-Based Simulation is an accepted paradigm for simulating human behavior in realistic environments. The canonical examples of this are in traffic simulation, a relatively mature field, and in simulating the spread of infectious disease through urban areas. Important to both of these approaches is capturing the movements of individuals in the area. Simulating events like traffic jams or outbreaks of Avian Flu is advantageous for several reasons. First, a well developed simulation can provide insight into the spread of the disease or the conditions and behaviors that can lead to traffic jams. Second, computer simulations give planners the ability to evaluate scenarios to deal with the crisis event. Planners using the EpiSims simulation [Barrett et al., 2005] can evaluate strategies for preventing the spread of Smallpox in a large urban area.

2. BACKGROUND

Agent-Based Models are well suited to simulating the behavior of complex systems, such as ecological systems, biological simulations and simulations of human behavior and movement [Railsback et al., 2005, Sheikh-Bahaei et al., 2005, Christley et al., 2007, Robalino and Lempert, 2000, Barrett and et al, 2004, ?]. These simulations cover a vast range of application domains, but there are similarities in the systems that make them amenable to the ABM approach. First, the system is composed of heterogeneous actors, which are called agents when simulated. Next, the desired output of the system is an aggregate over all the agents in the system. This can be some measure of the state of the agents, such as location, emotional state, etc. Finally, the simulation output is generated as a result of the interactions of the agents with themzselves and the environment.

2.1. Traffic Simulations

Several Agent-Based Models of traffic flow currently exist [Barrett and et al., 2004] [Hidas, 2002]. These models are useful in that they can produce useful information over a variety of traffic conditions, from traffic jams to steady state traffic flow, which is a limitation of earlier approaches [Greenberg, 1959, Greenshields et al., 1935]. The output of these simulations can be used for short term prediction of events such as traffic jams and routing around construction areas, or long term analysis for planning and development of infrastructure.

2.2. Epidemiological Simulations

Agent-Based Modeling has been used in spatially explicit epidemiological simulations [Barrett et al., 2005, ?]. In simulations such as EpiSims, modelers can simulate the outbreak of a biological agent such as smallpox in an urban area and examine the effectiveness of various strategies to control the spread of the disease.

2.3. Integrating GIS with Agent-Based Models

Several research groups have explored the integration of GIS data sources with Agent-Based Models [Batty and Jiang, 19! Itami and Gimblett, 2001, Bian, 2004]. The advantages of this approach are that agents can interact with realistic environments, improving the validity of the simulation's predictions, as well as allowing planners and architects to evaluate the effectiveness of designs before they are built.

Using GIS data sources to represent the environment in Agent-Based Models requires rethinking the approach to simulation. Often the environment in an ABM is a grid structure, with agent movement delineated in discrete units. However, movement on a GIS is continuous and simulations must take this into account. Often land-use simulations will use discrete parcels of land in order to avoid issues with modeling arbitrary areas [Waddell, 2002]. In the WIPER simulation we address the issue by simulating agent movements in terms of meters per time step and avoiding discrete representations of space.

3. PROBLEM DESCRIPTION

When a crisis occurs emergency responders will often quickly receive notice through standard emergency reporting channels, such as the 911 system. However, reports from bystanders often contain inaccuracies and individuals often lack the ability to judge the scope of crisis events. In situations like this Emergency Responders need tools that let them quickly assess the type and extent of the crisis. Under ideal circumstances, it would be even better to offer responders the ability to get predictions on the development of crisis scenarios. This allows responders to plan for the evolution and mitigation of such crises.

It is in this context that we present the WIPER simulation. The simulation is designed to model human movement and calling activity under normal circumstances and during crisis events. The simulations are designed for several purposes. First, simulations are created with streaming data from the crisis area and, in conjunction with an online validation approach, attempt to identify the detected crisis event. Second, the simulations can be used for short-term prediction of the evolution of crisis events.

4. TAXONOMY OF CRISIS SCENARIOS

In order to simulate and predict the course of crisis events, it is necessary that we have, a priori, a set of crisis scenarios that we can draw from. To make the process of developing these simulations more straightforward, we have tried to analyze various crisis events and organize the events into a taxonomy. We placed the scenarios into functional groups, focusing on the behaviors and actions of the agents that will elicit these behaviors.

4.1. Crisis Categories

We divide the crisis scenarios into 3 categories based on the principal movement characteristics of the agents. These categories are not meant to be exhaustive but merely descriptive of the events that we seek to simulate. The categories are as follows:

- Flock - Agents move in roughly organized fashion, like a mob.
- Flee - Agents move away from a disturbance.
- Jam - Agents move towards their customary goals, but are constrained, as in a traffic jam.

For the Flock category, agents move as a group, but without explicit leadership in a manner similar to the BOIDS movement model [Reynolds, 1987]. The Flock category is currently composed of one movement model, the mob model. This can be used to simulate scenarios where crowds of people are causing a disturbance, such as the WTO protests that occurred in Seattle in 1999 [Wikipedia, 2007d].

The Flee category is a much broader category and is applicable in a wide range of crisis scenarios. The category consists of models where agents are attempting to move away from some disturbance. The models in this category can be described concisely as flee from point, flee from line (not necessarily a straight line, this can include rivers/coastlines), flee from an area and bounded flee, where the agents get a certain distance away and stop. Some examples of crisis events that fit these scenarios would be people fleeing from a burning building (either a flee/bounded flee from point or flee from area, depending on map resolution), inhabitants fleeing a chemical spill (flee an area) and residents fleeing a tsunami (flee a line).

The final category is Jam, a collection of movements that are constrained. Agents in this category are trying to reach a destination (which may be unique for each agent), but the actions of all the agents together serves to create an event where movement is restricted for the entire system. The canonical example of this type of behavior is a traffic jam. This type of crisis scenario is often not necessarily an emergency event, though it can be, as in the case of the traffic jams on North-South highways in Florida in 2005 during the Hurricane Rita evacuation [Wikipedia, 2007b].

5. DESIGN

The WIPER Agent-Based Simulation has been designed using Design Patterns [Gamma et al., 1995]. The use of Design Patterns is a common approach to the development of object-oriented software and in this regard they can be applied quite well to Agent-Based Models.

All simulation scenarios share important components. The GIS files representing geography, political boundaries, tower information, etc are the same for all scenarios. A centralized simulation model class handles the initialization and setup of the simulation, placing agents onto the map, setting agent movement and activity models and handling the schedule.

5.1. Application of Design Patterns to Simulation

In order to reduce the amount of time and effort spent in writing the code for these simulations, we break out the agents' movement and activity models as Strategy and Singleton Patterns [Gamma et al., 1995]. This is possible because many aspects of agent state remain constant, regardless of the underlying phenomenon that we intend to simulate. We extract and encapsulate agent behaviors related to movement and activity into objects, outside of the agent itself. Agents then retain a pointer to this movement or activity model object. Although this does seem to introduce a semantic disconnect with how we expect an agent to be designed, this approach offers huge benefits in model development and allows researchers to run simulations where it is easy to initialize a population of agents with a few models and tractable to initialize the population with a large number (100 or more) of movement or activity models. More importantly this flexibility is apparent during the running of a simulation, when agents can change their movement or activity model at runtime, easily and without adverse affects on simulation performance, as switching models can be done as easily as changing a pointer.

5.2. Crisis Scenario Components

The crisis scenario taxonomy is used to guide our design of the crisis movement and activity models. A standard movement model has been developed in order to provide a baseline for comparison against the crisis models. The standard model, nominally referred to as "Move and Return" is intended to represent the activity of citizens in a non-crisis scenario. Agents have a "home" location, set at the beginning of the simulation and an alternate location, which we refer to as the "work" location. Agents move from the "home" to the "work" in the morning and return "home" in the evening. In the simulation, the locations of home and work can be set for each individual agent, as well as defining the movement schedule, which should allow the model to be validated against empirical movement studies. Such studies using cell phone calling data are currently underway.

The taxonomy of crisis scenarios provides a framework for the implementation of the movement models. All of the models in a given category share similarities and can be arranged into an inheritance hierarchy accordingly, allowing reuse of related movement code.

6. IMPLEMENTATION

The WIPER simulations are written in Java using the RePast Agent Modeling Framework [North et al., 2006]. The WIPER simulation also uses a number of Java APIs that come bundled with the RePast distribution, including the Colt High Performance Scientific Library [Hoschek, 2004] and GeoTools [geotools, 2006] and OpenMap [Technologies,] for GIS.

6.1. Activity Models

The activity models define the calling behavior of the agents. There are 3 activity models used in the simulation: the `NullActivity` model, the `AlwaysCall` model and the `DistributionBased` model. All of the `ActivityModel` types are subclasses of `ActivityModel`. `ActivityModel` is designed as a Singleton class (as described in [Gamma et al., 1995]), recognizing that all of the relevant state used to determine an agent's calling activity (Time, Date, Agent location) is maintained in the WIPER agent and in `WiperSimModel`.

WIPER Agents use the `ActivityModel` objects in a Strategy Pattern [Gamma et al., 1995]. This makes it possible to set the agent's behavior type as a parameter and provides fine-grained control over the behavior of individual agents in the simulation. The control over agent behavior allows large scale heterogeneity in agent movement models, an important consideration when attempting to model a large population of human agents.

The `NullActivity` model is a place-holder class. It overrides the abstract methods from `ActivityModel` but does not produce any behavior. This class is intended to be used for testing the Movement models, as it will not alter agent behavior.

The `AlwaysCall` model is another class used for testing. In this class, the `checkCall` method always causes the agent to make a call. This class can be used to test how agents call each other and to test other attributes of calling behavior.

The `DistributionBased` model is the primary call activity model in the simulation. This model requires initialization, it must be given an empirical call distribution as input. When the `checkCall` method is called, the method gets the (simulation) time and date from the `WiperSimModel` and then uses the empirical call distribution to determine the expected call activity at the current interval. A further discussion of the method of generating agent calling activity is given in Section `??.`

6.2. Movement Models

The movement models define the manner in which WIPER agents move on the map. There are 5 different movement models: `NullMovement, RandomMovement, MoveAndReturnMovement, FleeMovement` and `BoundedFleeMovement`. As with `ActivityModel, MovementModel` is an abstract base class and is designed to be a Singleton.

As with `ActivityModel`, WIPER agents use `MovementModels` in the Strategy pattern. In this case, the agent encapsulates the location and any pertinent characteristics (movement speed, location of work or home, etc) and the `MovementModel` accesses these through the agent when calculating the destination.

The `NullMovement` class is a placeholder, implementing the `move` method but without causing the calling agent to actually move. This class is useful when testing `ActivityModels`.

The `RandomMovement` class is used to move agents in a random fashion on the map. When the `move()` method is called, a random direction is chosen and the agent is advanced along that direction. The distance traveled depends on the agent's movement speed and the length of a simulation time step. The agents do not continue traveling along this path, at each time step a new direction is chosen.

The `MoveAndReturnMovement` class defines a "daily routine", where agents travel from a home location to a work location and back. This model works in conjunction with events scheduled in the `WiperSimModel` and with state maintained in the WIPER agents, the home and work locations, information on whether they are traveling to home or work, etc. The WIPER agent maintains the location of its own work and home, and the `MoveAndReturnMovement` uses these to calculate the next position of the agent.

The `FleeMovement` class is an implementation of a crisis movement class. The `move()` method moves each WIPER agent in a straight line away from a disaster location. The location of the crisis is initialized in the `WiperSimModel` at the start of the simulation. In this class, WIPER agents always move away from the crisis and continue moving until the simulation ends. This type of behavior is similar to what is expected in a major disaster scenario, such as the way that people fled from Manhattan on September 11. Due to the relatively short duration of the WIPER simulations, minutes and hours, rather than days, this type of behavior is a good first-order approximation.

The `BoundedFleeMovement` class is a refined version of `FleeMovement`. In this `move` method, each WIPER agent moves directly away from the crisis location until it reaches a threshold, then stops. The crisis location and the flee radius are initialize in `WiperSimModel` at the start of the simulation. This type of crisis behavior is more consistent with what would be seen in a building fire or small-scale crisis, where people flee the crisis until they reach a safe distance.

6.3. The WIPER Simulation Model Class

In the WIPER simulation, the `WiperSimModel` class extends `SimModelImpl` and is responsible for the creation, initialization and management of the simulation. The `SimModelImpl` class is a `RePast` class that partially implements the `SimModel` interface, which is designed to control the schedule, run the simulation and respond to input from the RePast GUI console. In the WIPER simulation, `WiperSimModel` is the largest and most complex class.

When a WIPER simulation is started, `WiperSimModel` receives the initial parameters either from the RePast GUI (for interactive simulations) or from the command line (when used as part of the WIPER Simulation Prediction System or when dispatching simulations to a computational grid). The initial simulation parameters are used to specify which components will be visualized on the graphical display, configure the GIS map with voronoi cells and geographical features, initialize the `WiperAgents` (either from a file or via a string of tower IDs and number of agents at each tower), schedule actions (these can include generating and saving GIS snapshots, logging of calls and agent locations, scheduling each `WiperAgent's step()` method, etc) and handle proper clean up at the end of the simulation (closing output files, etc).

6.4. The WIPER Agent Class

The primary purpose of the WIPER simulation is to model the behavior of cell phone carrying inhabitants of an urban area. The `WiperAgent` class is the implementation of the cell phone agent, encapsulating state information such as current location, home and work locations, current cell tower/voronoi cell. As described above, `WiperAgents` use the Strategy pattern [Gamma et al., 1995] for holding their movement and activity models, which allows unprecedented flexibility across the population of agents. This means that movement or activity models can be assigned individually to agents and that the behavior of an agent can be changed easily by replacing its movement or activity models.

6.5. Centralized Logging Architecture

Agent movements and calling activity are recorded to the WIPER simulation log using a centralized logging architecture. The `DataLogger` class acts as a central entry point to the log, collecting information from agents and sending the results to a file. The `DataLogger` produces two files for each simulation run, an activity file with all call activity and a location file with agent locations. The activity file records agent calling activity in a format identical to the CDR data from our cellular service provider. This compatibility makes it possible to use simulation-generated activity files in place of empirical CDR data in the RTDS component of the WIPER system.

6.6. GIS

Various GIS data sources are used in the WIPER simulation: roads, political boundaries, cellular tower locations, voronoi cells. A voronoi diagram is a tiling on a surface around a set of points $p \in P$ where each point p_i occupies a unique cell and all points within that cell are closer to p_i than to any other $p \in P$. The voronoi cells in the WIPER simulation are a tiling of the map made from the cellular tower locations. These cells provide a good first order approximation of the coverage area around each tower, allowing the simulation to read in agent activity at a tower and translate this to approximate agent locations on a map. Political boundaries, such as postal codes, city limits, etc are useful when attempting to paramterize agents based on census data or other demographics linked to location.

Certain GIS data representing the location and extent of information like chemical dispersion, weather patterns, etc can be added to the simulation. These data can added to the realism of the simulation and potentially will increase the usefulness of simulation output.

7. VALIDATION AND VERIFICATION

Model validation is a necessary step in the development of a simulation. In order to demonstrate the validity of the simulation as a whole, several steps are taken to validate the underlying theoretical model and to verify the implementation. In this section we describe the verification and validation approaches used on the WIPER simulation.

7.1. Face Validation

The most common validation approach is face validation. In this step, a domain expert examines the theoretical model and it's underlying assumptions and determines whether this model is a reasonable representation of the intended phenomenon. For this simulation we have conducted face validation on the conceptual model ourselves, which is a common practice in the field. We have examined the assumptions for the simulation and the design decisions that have been made in order to make the simulation process tractable. Here we briefly present the design decisions.

For the simulation, we choose to model human behavior in 1 minute increments. The choice of time step was dictated by our need to generate calling activity at no more than 1 minute intervals. Agent movement simulations may benefit from smaller time steps, but there is a tradeoff between time resolution and simulation runtime. For the purposes of agent movement simulation, 1 minute intervals provides adequate accuracy to model agent behavior. Agents must move on a simulated represention of the world. We chose to implement this using a GIS, which provides an accurate model of the world. We chose to model cell tower coverage areas using Voronoi cells. This is an accepted technique for representing coverage areas, as stated in [Meguerdichian et al., 2001].

7.2. Input-Output Correlation on Empirical Data

The WIPER simulation is designed to be a component in the WIPER emergency response system. The intended purpose of the simulation is to be used in a DDDAS system where the simulations are updated with and validated against streaming data from a cell phone network. With this in mind, the WIPER simulation was designed to generate simulation output similar to empirical data captured from a cellular service provider and furnished to the WIPER group. Here we present a detailed examination of the simulation output in terms of cell phone activity, comparing it to the empirical data taken from the CDR data. Call Data Record (CDR) data is the data used by cellular service providers for billing purposes. CDR records contain transaction records for the initiation and termination of calls, SMS messages and other services.

In order to perform these tests, we started by running 10 simulations, all with identical input parameters but varying the random seed for each simulation. We later increased this to 100 simulations. We parameterized the simulations with the number of active users for the simulated area, as taken from the empirical CDR data. We set the movement model to null, which causes the cell phone user agents to remain at fixed locations throughout the simulation. We set the activity model to be a Distribution-based model, as described in Section 6.1. The Distribution-based model is designed to generate call activity in a way that is similar to that found in the empirical CDR data.

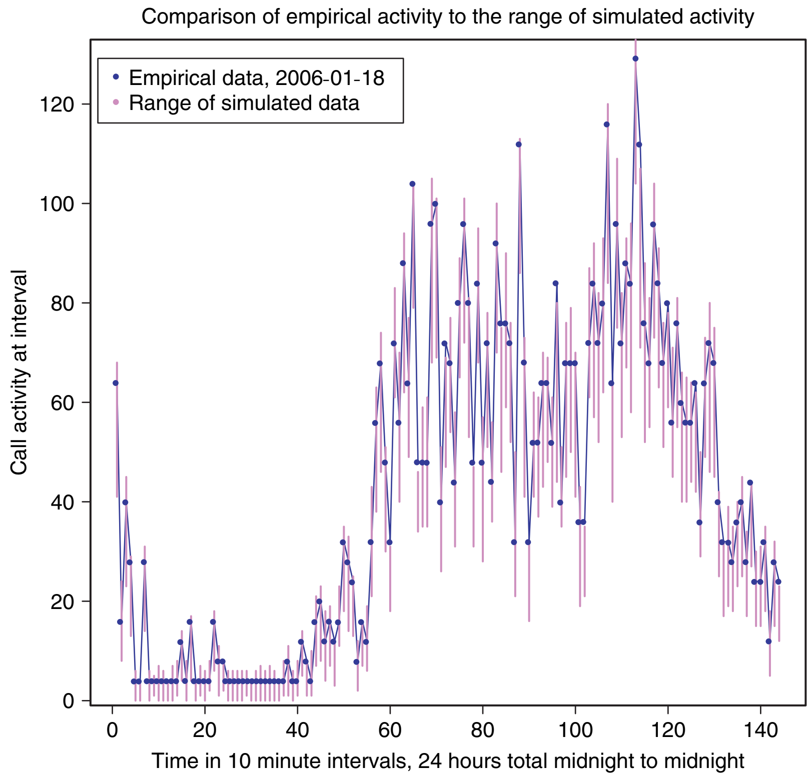

Figure 1. Plot of empirical call activity over the range of simulated call activity, demonstrating that the simulation generates activity without bias.

We present the call activity output of the first round of simulations in Figure 1. This figure shows output of all of the simulations plotted on top of the empirical data, which is in blue. In this figure the empirical data is the CDR data for the region for one day. This plot demonstrates that the simulated data falls in a range around the empirical data.

A common test to determine if two data sets come from the same generating distribution is the Kolmogorov- Smirnov test. This test is performed pairwise on the output of each simulation against the empirical data. In Table 1, we present the results of the K-S test on the initial round of 10 simulations, including the D value (the K-S test statistic) and whether the result indicates acceptance at the $\alpha = 0.10$, 0.05 and 0.01 levels. As shown in the Table, output

Table 1. Examination of ks test statistics for comparing simulated call activity to empirical data from ten replications of the simulation. Given are the d test statistic and a notation as to whether the test for that simulation is accepted for $\alpha = 0.10$, $\alpha = 0.05$ AND $\alpha = 0.01$.

Simulation	D Statistic	Accept at $\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
1	0.903	YES	YES	YES
2	0.1042	NO	YES	YES
3	0.1319	NO	NO	YES
4	0.1319	NO	NO	YES
5	0.0903	YES	YES	YES
6	0.1181	NO	NO	YES
7	0.1111	NO	YES	YES
8	0.1042	NO	YES	YES
9	0.1250	NO	NO	YES
10	0.0764	YES	YES	YES

from the 10 simulations all pass at the $\alpha = 0.01$ level of significance, 6 of 10 pass at the $\alpha = 0.05$ level of significance and 3 of 10 pass at the $\alpha = 0.10$ level of significance. It should be noted that the D-values are more significant as the value approaches 0.

According to Banks, goodness of fit tests, such as the K-S test, are sensitive to sample size and have a tendency to reject candidate distributions for large sample sizes [Banks et al., 2005]. Our simulations generate a sample of size 144, as the data is aggregated in 10-minute intervals over 24 hours. In Figure 2 we present a look at a histogram of the K-S test statistic on the output of 100 simulations. This was done to visualize how well the simulations fit the empirical data over a larger range of simulation runs. The histogram demonstrates that an appreciable number of the simulations rank above the $\alpha = 0.10$ level of significance and the vast majority rank above the $\alpha = 0.01$ level of significance. This result, on a much larger set of simulation output, demonstrates significant evidence that the WIPER simulation generates call activity data similar to that seen in the empirical data.

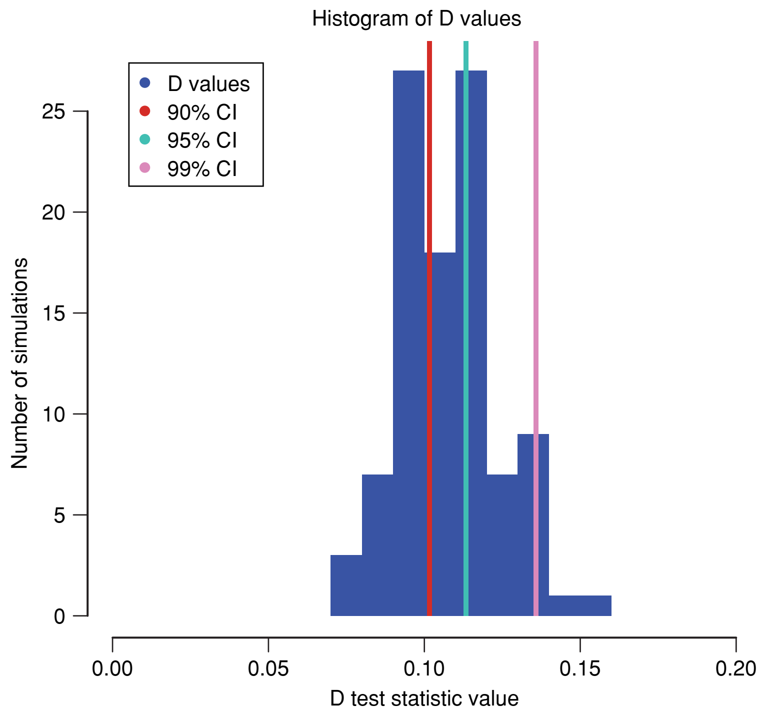

Figure 2. Histogram of the kolmogorov-smirnov test statistics for 100 runs of the simulation. The acceptance values for $\alpha = 0.10$, $\alpha = 0.05$ AND $\alpha = 0.01$ are plotted as vertical bars. lower test statistic values indicate higher confidence.

8. CONTRIBUTIONS AND RESULTS

In this section we present a thorough examination of the characteristics of the WIPER simulation, including runtime performance and scalability. Output of the simulation, including screen shots of the visual components, is also provided.

8.1. GUI Display and Visual Components

A sample screen shot of the WIPER simulation is shown in Figure 3. In this image, the WIPER simulation is being run in GUI mode, using the standard RePast toolbar to control the simulation and adjust input parameters. The map is an OpenMap display showing agent locations and the voronoi cell boundaries, with the voronoi cells colored by the number of agents within each cell's boundaries. The call activity window shows a running display of the call activity of all agents in the simulation, plotted against the empirical call activity for this time and day.

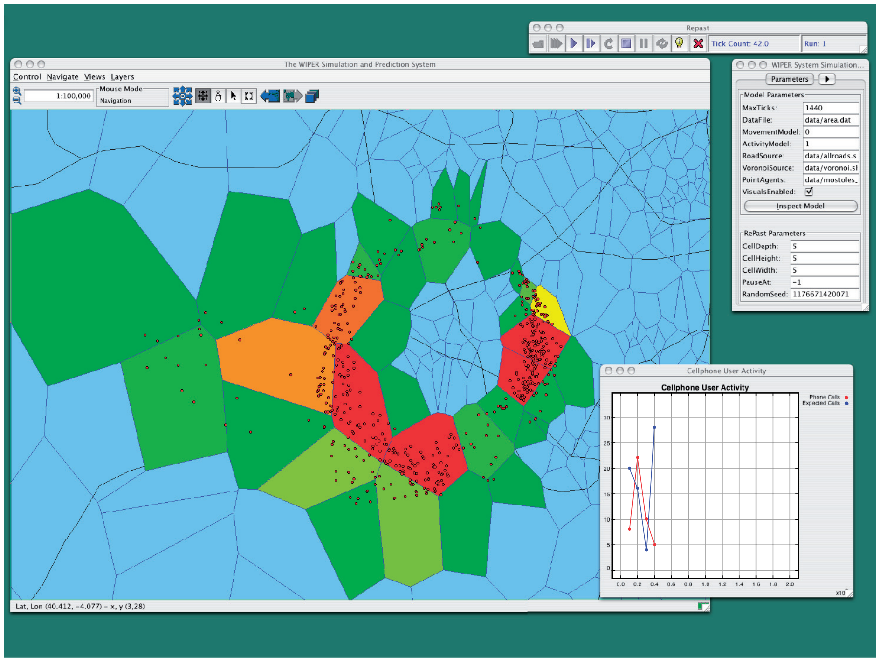

Figure 3. Screen shot of the wiper simulation. This simulation is a flee movement model with distribution-based activity model. Agents are represented as red dots with the voronoi cells colored by the number of agents.

When the WIPER simulation is run interactively, a researcher can observe via the GUI display the state of the system and the recent agent behaviors. In order to improve runtime performance, the plotting of agent locations can be disabled. The overall agent locations can still be determined through the coloring of the voronoi cells but with a lower degree of accuracy.

8.2. Runtime Performance

The runtime performance of the WIPER simulation in response to changing levels of graphical output is presented in Table 2. All simulations for this analysis are started with 500 agents, using the `MoveAndReturnMovement` and `DistributionBased` models. The simulation runs for one simulated day, 1440 time steps, with each time step representing one minute. The simulation can be run without any graphical display (the default when running simulations on a computational grid), with a GIS display that shows information such as cell tower locations, the

Table 2. Cumulative runtime for 20 runs of the simulation with varying levels of graphical output. GIS - GIS display, showing tower locations and voronoi cells colored by number of contained agents. Agent locations - the location of all agents are added to the GIS display. Snapshots - every 10 time steps the simulation makes a snapshot of the GIS display.

GIS	AGENT LOCATIONS	SNAPSHOTS	TIME
NO	NO	NO	240.34s
YES	NO	NO	354.70s
YES	YES	NO	363.44s
YES	YES	YES	35200.29s

surrounding Voronoi cell for a tower and agent locations and finally the simulation can generate snapshots of the GIS display, which can be displayed on the graphical console of the WIPER system or compiled together to form a movie.

The results shown in Table 2 are the total cumulative running time for 20 runs of the simulation. The results clearly demonstrate that there is a cost associated with running the graphical display and generating snapshots of the GIS. It is interesting to note that the penalty for visualizing the agents on the GIS display is negligible, compared with the cost of the GIS. Generating the snapshots only once every 10 time steps requires an approximately 2 orders of magnitude cost.

The scaling of the simulation with regards to number of agents, area fixed, is shown in Figure 4. In the simulations, all `WiperAgents` are started with `MoveAndReturnMovement` and `DistributionBased` models and are initialized into the same geographical area. For each level of agent population size, 20 replications with unique random seeds were run. The agent population varied from 10 to 1,000,000 in the following manner: 1,000 – 10,000 by increments of 1,000, 10,000–100,000 by increments of 10,000 and 100,000–1,000,000 by increments of 100,000.

8.3. Design Contributions

The development of an Agent-Based Modeling simulation, as with software in general, is an iterative and ongoing process. A simulation is never truly “finished”, as there is always room for improvement in some area: speed of

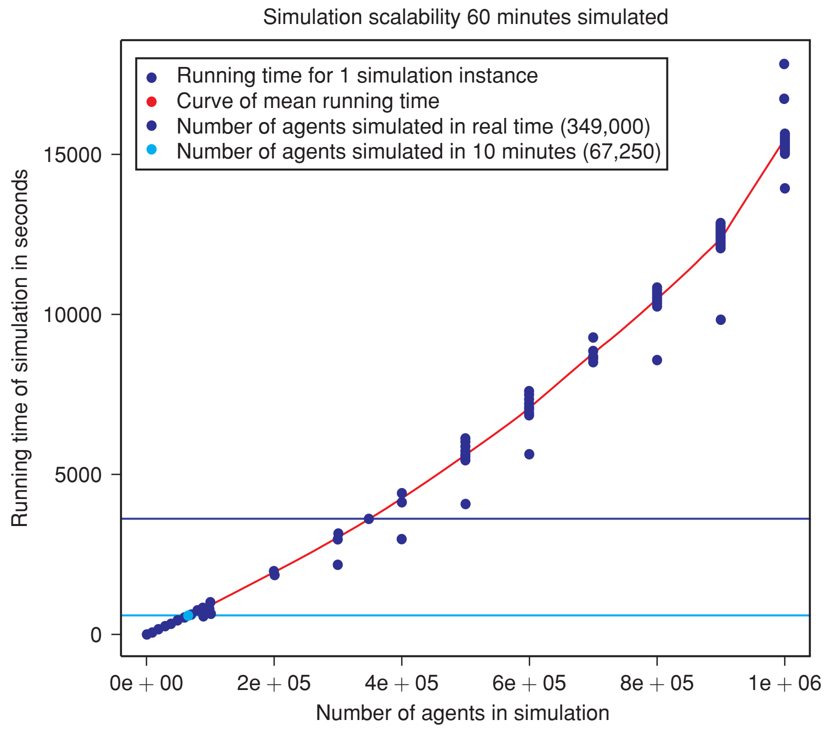

Figure 4. Simulation scalability with respect to number of agents, 20 replications at each level. In this instance we run simulations for 60 minutes of simulated time. In this graph we examine agent simulations out to a population size of 1,000,000 agents.

execution, ease of use, memory footprint, variety of crisis scenarios, accuracy/validity of model results, etc. With this in mind, the WIPER simulation has been designed to be straightforward to maintain and extend. The simulation is designed using conventional software engineering techniques such as Design Patterns [Gamma et al., 1995]. Components of the simulation are tested and verified using an extensive and comprehensive Unit Testing suite. The simulation has been developed using an Agile approach, with the design of the simulation revised several times using the concept of Refactoring as described by Fowler [Fowler, 1999]. Although these characteristics of the WIPER simulation are difficult to measure quantitatively it is clear that they certainly contribute to the extensibility and maintainability of the simulation.

Table 3. Average running time and standard deviation for simulations with 10-10,000 agents. Times are in seconds. Simulations display low standard deviation that scales appropriately with runtime.

Number of Agents	Mean Running Time	Std. Dev.
10	5.6172	0.14025334
20	6.4272	0.09688997
⋮	⋮	⋮
90	11.7548	0.14338526
100	12.4444	0.15564597
200	20.0124	0.19751118
⋮	⋮	⋮
900	73.4960	1.88104138
1000	80.6840	0.86200348
2000	154.2744	1.94242563
⋮	⋮	⋮
9000	667.6448	8.62510054
10000	738.6588	10.71719814

The WIPER simulation is part of a Dynamic, Data-Driven Application System and, as such, there are special design criteria that need to be met. First, simulations are intended to simulation short time periods, between 10 minutes to 2 hours. Thus the WIPER system is designed for scalability and short run times, and has been shown to display excellent runtime performance. Second, simulations must be created from streaming data. In order to address this we have made it possible to create simulations with a reasonable representation of agent locations, as would be available in a real-world situation. Agent locations are known to the cell tower level and agents are randomly distributed within the tower area. Third, simulations are updated and validated against the streaming data. The updating and online validation process are beyond the scope of this paper, but they require that agents are able to use interchangeable models that can be validated via a ranking and selection process against online data. Thus the simulation has been designed with interchangeable movement and activity models.

Developer and user documentation is provided in the form of annotated JavaDoc pages, with extensive comments from the developer and references to relevant classes in the various APIs used by the simulation, e.g. RePast, OpenMap, etc. Additional documentation for end users is provided on a research wiki and will be made available upon request.

9. SUMMARY

In this paper we have presented the WIPER simulation, an Agent-Based Model for simulating the movement and calling behavior of cell phone users. The simulation has a calling activity model based on the activity observed in empirical data taken from cellular service provider records. The activity generated by the simulation is indistinguishable from the empirical data according to statistical tests. The movement models demonstrate various behaviors that are important when attempting to simulate crisis behavior. Also, as the simulation is designed to be a part of the WIPER system for emergency response, where simulations are updated with streaming data, runtime performance is an important development goal. We have demonstrated through a scalability exploration the runtime characteristics of the simulation, showing that it displays adequate performance for use in the time-critical WIPER system.

10. FUTURE WORK

The current cell phone activity models represent usage under normal scenarios. We would like to study the existing CDR data and examine behavior of people during crisis events. We will use this information to guide the creation of crisis activity models and will use our existing validation framework to evaluate this approach.

In order to better model human behavior and the spread of information as it relates to crisis events, it may be useful to examine creating a social network structure on the agents. The could allow us to study information diffusion across a population, as well as examining the interplay between physical location and distance in a social network.

REFERENCES

1. [Balci, 1998] Balci, O. (1998). *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, chapter Verification, Validation, and Testing. John Wiley & Sons, New York, NY.
2. [Bankes, 2002] Bankes, S. C. (2002). Agent-based modeling: A revolution? *Proceedings of the National Academy of Sciences*, 99:7199–7200.

3. [Banks et al., 2005] Banks, J., Carson, J., Nelson, B., and Nicol, D. (2005). *Discrete-Event System Simulation.* Prentice Hall, Upper Saddle River, NJ, Third edition.
4. [Barrett and et al., 2004] Barrett, C. L. and et al (2004). Transportation analysis simulation system. Technical report, Los Alamos National Laboratory.
5. [Barrett et al., 2005] Barrett, C. L., Eubank, S. G., and Smith, J. P. (2005). If smallpox strikes portland ... *Scientific American.*
6. [Batty and Jiang, 1999] Batty, M. and Jiang, B. (1999). Multi-agent simulation: New approaches to exploring space-time dynamics within gis. *Graphical Information Systems Research - UK* (*GISRUK*) *1999*.
7. [Bian, 2004] Bian, L. (2004). A conceptual framework for an individual-based spatially explicit epidemiological model. *Environment and Planning B*, 31: 381–395.
8. [Bruzzone et al., 2003] Bruzzone, L., Molfino, R., and Zoppi, M. (2003). A discrete event simulation package for modular and adaptive assembly plants. In Hamza, M., editor, *Proceedings of Modelling, Identification, and Control (MIC* 2003*).*
9. [Christley et al., 2007] Christley, S., Alber, M. S., and Newman, S. A. (2007). Patterns of mesenchymal condensation in a multiscale, discrete stochastic model. *PLoS Computational Biology*, 3(e76).
10. [Fowler, 1999] Fowler, M. (1999). *Refactoring: improving the design of existing code.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
11. [Gamma et al., 1995] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
12. [geotools, 2006] geotools (2006). The Geotools project. `http://geotools.codehaus.org.`
13. [Greenberg, 1959] Greenberg, H. (1959). An analysis of traffic flow. *Operations Research,* 7(1): 79–85.
14. [Greenshields et al., 1935] Greenshields, B. D., Bibbins, J. R., Channing, W. S., and Miller, H. H. (1935). A study of traffic capacity. *Highway Research Board Proceedings,* 14: 448–477.
15. [Grimm et al., 2005] Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W. M., Railsback, S. F., Thulke, H.-H., Weiner, J., Wiegand, T., and DeAngelis, D. L. (2005). Pattern-oriented modeling of agent-based complex systems: Lessons from ecology. *Science,* 310(5750): 987–991.
16. [Hidas, 2002] Hidas, P. (2002). Modelling lane changing and merging in microscopic traffic simulation. *Transportation Research Part C: Emerging Technologies*, 10: 351–371.

17. [Hoschek, 2004] Hoschek, W. (2004). Colt high performance scientific library. `http://dsd.lbl.gov/˜hoschek/colt/index.html.` [Online; accessed 28-March-2007].
18. [Itami and Gimblett, 2001] Itami, R. M. and Gimblett, H. R. (2001). Intelligent recreation agents in a virtual gis world. *Complexity International*, 8.
19. [Kennedy, 2006] Kennedy, R. C. (2006). Verification and validation of agent-based and equation-based simulations and bioinformatics computing: identifying transposable elements in the aedes aegypti genome. Master's thesis, University of Notre Dame.
20. [Meguerdichian et al., 2001] Meguerdichian, S., Koushanfar, F., Potkonjak, M., and Srivastava, M. B. (2001). Coverage problems in wireless ad-hoc sensor networks. In *INFOCOM*, pages 1380–1387.
21. [Neumann, 1966] Neumann, J. V. (1966). *Theory of Self-Reproducing Automata.* University of Illinois Press, Champaign, IL, USA.
22. [North et al., 2006] North, M., Collier, N., and Vos, J. (2006). Experiences creating three implementations of the Repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation*, 16:1–25.
23. [Railsback et al., 2005] Railsback, S. F., Harvey, B. C., Hayse, J., and LaGory, K. (2005). Tests of theory for diel variation in salmonid feeding activity and habitat use. *Ecology,* 86: 947–959.
24. [Reynolds, 1987] Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *SIG-GRAPH'87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, New York, NY, USA. ACM Press.
25. [Robalino and Lempert, 2000] Robalino, D. A. and Lempert, R. J. (2000). Carrots and sticks for new technology: Abating greenhouse gas emissions in a heterogeneous and uncertain world. *Integrated Assessment*, 1(1):1–19.
26. [Schoenharl et al., 2006] Schoenharl, T., Bravo, R., and Madey, G. (2006). WIPER: Leveraging the cell phone network for emergency response. *International Journal of Intelligent Control and Systems*, 11(4).
27. [Sheikh-Bahaei et al., 2005] Sheikh-Bahaei, S., Ropella, G. E. P., and Hunt, C. A. (2005). In silico hepatocyte: Agent-based modeling of the biliary excretion of drugs. In Yilmaz, L., editor, *Proceedings of Agent-Directed Simulation Symposium (ADS 2006).*
28. [Technologies,] Technologies, B. Openmap: Open systems mapping technology. http://openmap.bbn.com/.
29. [Turing, 1952] Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society (B)*, 237(631): 37–72.

30. [Waddell, 2002] Waddell, P. (2002). UrbanSim: Modeling urban development for land use, transportation and environmental planning. *Journal of the American Planning Association*, 68(3): 297–314.
31. [Wikipedia, 2006] Wikipedia (2006). Voronoi diagram—Wikipedia, the free encyclopedia.`http://en.wikipedia.org/w/index.php?title=Voronoi_diagram&oldid=47842110.` [Online; accessed 25-April-2006].
32. [Wikipedia, 2007a] Wikipedia (2007a). Box plot—wikipedia, the free encyclopedia. [Online; accessed 15-May-2007].
33. [Wikipedia, 2007b] Wikipedia (2007b). Hurricane rita—wikipedia, the free encyclopedia. [Online; accessed 27-March-2007].
34. [Wikipedia, 2007c] Wikipedia (2007c). Kolmogorov-smirnov test—wikipedia, the free encyclopedia. [Online; accessed 8-April-2007].
35. [Wikipedia, 2007d] Wikipedia (2007d). WTO ministerial conference of 1999 protest activity—wikipedia, the free encyclopedia. [Online; accessed 27-March-2007].
36. [Xiang et al., 2005] Xiang, X., Kennedy, R., Madey, G., and Cabaniss, S. (2005). Verification and validation of agent-based scientific simulation models. In Yilmaz, L., editor, *Proceedings of the 2005 Agent-Directed Simulation Symposium*, volume 37, pages 47–55. The Society for Modeling and Simulation International.
37. [Zeigler and Vahie, 1993] Zeigler, B. P. and Vahie, S. (1993). Devs formalism and methodology: unity of conception/diversity of application. In *WSC* '93: *Proceedings of the 25th conference on Winter simulation,* pages 573–579, New York, NY, USA. ACM Press.