WIPER: An Emergency Response System

Alec Pawling, Tim Schoenharl, Ping Yan, Greg Madey
Dept. of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46656
{apawling,tschoenh,pyan,gmadey}@cse.nd.edu

ABSTRACT

This paper describes the WIPER system, a proof of concept prototype, and progress made on its development to date. WIPER is intended to provide emergency response managers with an integrated system that detects possible emergencies from cellular communication data, attempts to predict the development of emergency situations, and provides tools for evaluating possible courses of action in dealing with emergency situations. We describe algorithms for detecting anomalies in streaming cellular communication network data, the implementation of a simulation system that validates running simulations with new real world data, and a web-based front end to the WIPER system. We also discuss issues relating to the real-time aggregation of data from the cellular service provider and its distribution to components of the WIPER system.

Keywords

Emergency Response System, Data Mining, Agent-Based Simulation, Geographical Information Systems.

INTRODUCTION

The Wireless Phone-Based Emergency Response (WIPER) system is a proof of concept prototype designed to utilize a cell-phone network as a set of sensors for gathering and presenting data to emergency response managers. The system would monitor the network data in real time for anomalous activity, run simulations to predict population movement during a crisis, and provide emergency response managers with a current view of the affected area using GIS tools (Madey et al., 2007, Madey, Szabó, & Barabási, 2006, Schoenharl, Bravo, & Madey, 2006, Schoenharl, Madey, Szabó, & Barabási, 2006). In this paper, we describe the current status of the development of the WIPER system.

Existing software tools, such as EVResponse (Thomas, Andoh-Baidoo, & George, 2005) and COMBINED (Tatomir & Rothkrantz, 2005) provide mechanisms for manually gathering information relating to the current status of a crisis situation. There is a high cost associated with such systems in terms of time and money. Wireless devices and network infrastructure must be purchased to facilitate data collection, personnel must be trained to use the technology, and personnel must be deployed to the affected area to collect the data. In contrast, WIPER provides information about the situation through a pre-existing network, requiring no investment in infrastructure or deployment; however, we gain these advantages at the cost of data flexibility.

To counteract the limitations of the data, the WIPER system is designed to use machine learning methods to generate hypotheses about the causes of anomalies detected in the data. These hypotheses are tested using a dynamic data driven application system of simulations. Dynamic data driven application systems (DDDAS) are characterized by an ability to incorporate new data into running simulations. Research in DDDAS is motivated by a need for greater accuracy in complex simulations, *e.g.* simulations for predicting weather or wildfire propagation (Douglas & Deshmukh, 2000). Once the WIPER system detects an anomaly, it will start a suite of simulations based on the hypotheses generated. These simulations will be validated using new data as it becomes available, allowing simulations that do not reflect the real world situation to be pruned.

In addition to the prediction capabilities, WIPER will provide the ability to view the development of a crisis in real-time, the ability to propose and evaluate responses in near real-time, and the ability to collect and analyze streaming information from a cellular communication network. The WIPER system will analyze dynamic data from the cell phone network in real-time, providing the functionality to detect crises as they emerge. Responding to events from the anomaly detection system, GIS-based simulations of the region will be launched and results collated and

presented to planners. Finally, the web-based console will allow emergency response managers to quickly examine the current state of the environment, see predicted outcomes from the simulations, and evaluate possible courses of action.

OVERVIEW OF THE WIPER SYSTEM

The WIPER system consists of five components, each of which is described briefly below.

- The *Decision Support System* (DSS) is a web-based front end through which emergency response managers interact with the WIPER system.
- The *Detection and Alert System* (DAS) monitors streaming network data for anomalous activity. There are various aspects of the cell-phone network data that may be of interest, including overall usage levels, spatial distribution of the call activity, and the underlying social network.
- The *Simulation and Prediction System* (SPS) receives anomaly alerts from the DAS, produces hypotheses that describe the anomaly, and uses simulations in conjunction with streaming activity data to validate or reject hypotheses.
- The *Historical Data Source* (HIS) is a repository of cell phone network data that resides in secondary storage. This data is used to determine the base-line behavior of the network against which anomalies are detected and to periodically calibrate and update the DAS.
- The *Real-Time Data Source* (RTDS) is designed to receive transaction data directly from a cellular service provider. The RTDS is responsible handling requests for streaming data from the DAS, SPS, and DDS and streaming incoming data to these components in real-time.

Figure 1 shows an architectural overview of the WIPER system. The RTDS and HIS will provide the bridge from the service provider and the WIPER system. The figure shows the flow of streaming data from the service provider through the RTDS, possibly by way of the HIS for development and training, and to the remaining components. Requests for streaming data from the RTDS occur via SOAP messages. SOAP messages are also used by the DAS to inform the SPS of potential anomalies in the streaming data.

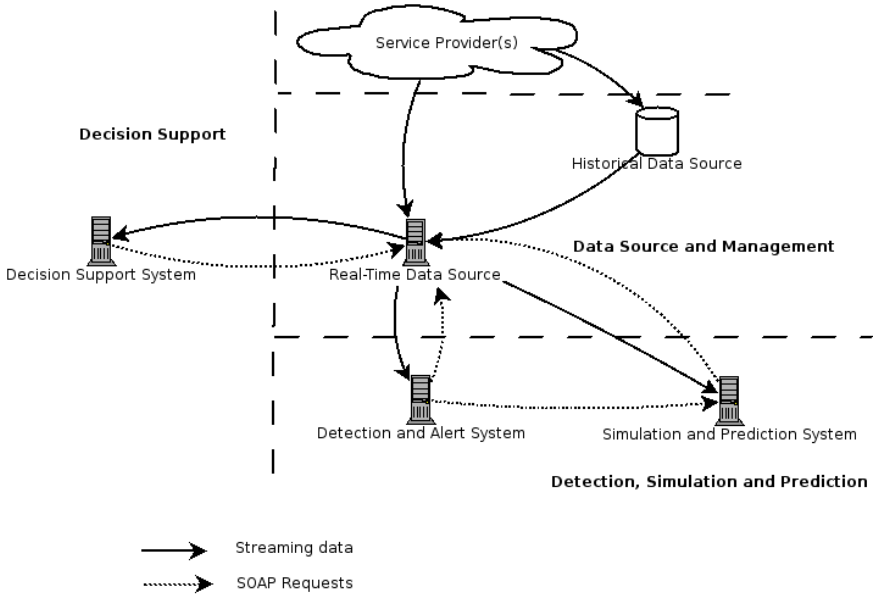

Figure 1: WIPER system architecture.

THE DETECTION AND ALERT SYSTEM

The detection and alert system is designed to examine the streaming data from the cellular service provider for anomalous activity on three axes: call volume, spatial call activity distribution, and the underlying social network. In this section we describe our work relating to the first two axes.

Markov Modulated Poisson Processes

The most basic indicator of anomalous behavior in a cell phone network is an increase or a decrease in call activity. This type of anomaly can be detected by monitoring a time series consisting of the number of calls made in disjoint time intervals of a fixed size, *e.g.* the number of calls made every 5 minutes. A Poisson process, which models the number of random events that occur during a sequence of time intervals, can be used to model the baseline behavior of such a time series: the number of events per time interval follows a Poisson distribution with an expected value of λ, the rate parameter for the process. In this model, the probability of N events occurring in a time step is:

$$P(N;\lambda) = \frac{e^{-\lambda}\lambda^{N}}{N!} \tag{1}$$

for N=0,1,… (Mitzenmacher & Upfal, 2005).

The standard Poisson process is too simple to model many real-world phenomena since the rate of many natural processes varies over time. In the case of human activities, there are daily and weekly cycles. Ihler, Hutchins, & Smyth (2006, 2007) show how a Markov modulated Poisson process, a method that combines Poisson processes and hidden Markov models, can be used to detect anomalies in count data.

A Markov modulated Poisson process uses a rate function that takes into account the overall average rate, the day effect, and the time of day effect. The overall average, , is simply the average rate over all time intervals. The day effect, $\delta_{d(t)}, d(t) \in \{1,\ldots 7\}$, is the average rate over all time intervals for each day of the week, normalized such that the average day effect is 1, *i.e.* $\sum \delta_{d(t)} = 7$. The time of day effect, $\eta_{d(t),h(t)}, h(t) \in \{1,\ldots D\}$, is the average rate for each time interval for each day of the week. The time of day effect for each of the 7 days of the week is normalized such that average time of day effect for each day is 1, *i.e.* $\forall d(t), \sum \eta_{d(t),h(t)} = D$.

The overall average establishes the baseline rate. The day effect and time of day effect indicate the relationship between the overall average and the expected rate for the day or time of day. The product of the three terms gives a weekly cycle that approximates the real world data. So, the rate function for a Markov modulated Poisson process is

$$\lambda(t) = \lambda_0 \delta_{d(t)} \eta_{d(t),h(t)} \tag{2}$$

To illustrate the components of the rate function, we compute the overall average rate, the day effect, and the time of day effect from one month of real cell phone data. Figure 2 shows each component of the rate function along with one week of empirical data.

The use of a Markov modulated Poisson process to identify anomalies in the cellular transaction data is described in detail in Yan, Schoenharl, Pawling, and Madey (2007).

An Online Hybrid Clustering Algorithm

In this section, we consider the problem of detecting anomalies in the spatial distribution of the call volume. The coverage area is divided into spatially disjoint areas: Voronoi cells centered at the cell towers. We describe an online hybrid clustering algorithm for detecting anomalies in a multivariate time series in which each item consists of the vector of call volumes for each cell tower.

The clustering problem is defined as follows: let a data set D consists of a set of data items $\{\vec{d}_1, \vec{d}_2, \ldots\}$ such that each data item is a vector of measurements, $\vec{d}_i = \langle d_{i,1}, d_{i,2}, \ldots, d_{i,n} \rangle$. The goal of clustering is to group similar data items together. For our purposes, the distance between similar items is small. Clustering provides a convenient way for finding anomalous data items: anomalies are the data items that are far from all other data items. These may be data items that belong to no cluster, or they may be the data items that belong to small clusters.

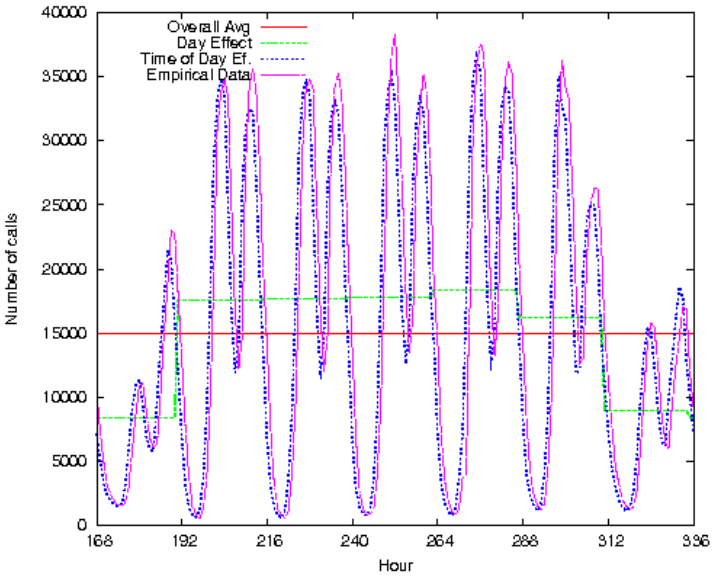

Figure 2: The overall average rate (λ_0), day effect combined with the overall average ($\lambda_0 \delta_{d(t)}$), and time of day effect combined with the overall average and the day effect ($\lambda_0 \delta_{d(t)} \eta_{d(t),h(t)}$) along with one week of empirical count data from the cellular communication network.

Traditional clustering algorithms can be divided into three types: partitional, hierarchical, and incremental. Partitional algorithm, such as *k*-means and expectation maximization, divide the data into some number, often predefined, of disjoint subsets. These algorithms often start with a random set of clusters and iterate until some stopping condition is met. As a result, partitional algorithms tend to converge at local minima. Hierarchical algorithms divide the data into a nested set of partitions and are useful for discovering taxonomies in data. They may either take a top-down approach in which an initial data cluster containing all of the data items is iteratively split until each data item is in its own cluster, or a bottom-up approach in which clusters initially consisting of only a single element are iteratively merged until all of the data items belong to a single cluster. Often, hierarchical algorithms must compute the distance between each pair of data items in the data set, and, therefore, tend to be computationally expensive. Incremental algorithms consider each data example once, immediately deciding either to place it in an existing cluster or to create a new cluster. These algorithms tend to be fast, but the result often depends on the order of the data items (Jain, Murty, & Flynn, 1999).

For online detection of anomalies in telecommunication data, none of these traditional clustering algorithms are appropriate. Partitional algorithms, typically require *a priori* knowledge of the number of clusters in the data. To complicate matters, the appropriate number of clusters may change over time as the underlying process changes. Hierarchical methods are too computationally expensive. Incremental methods tend to be too inflexible. Therefore, we use a hybrid clustering algorithm, which combines two clustering algorithms to take advantage of the strengths of both, while minimizing the drawbacks (Cheu, Keongg, & Zhou, 2004, Chipman & Tibshirani, 2006, Surdeanu, Turmo, & Ageno, 2005).

We combine an incremental method, the leader algorithm (Hartigan, 1975), with a partitional method, the *k*-means algorithm. The leader algorithm is straightforward:

- For each new data item:
    - Find the nearest cluster.
    - If the distance between the new data item and the nearest cluster is less than a user defined threshold, add the data item to the cluster.
    - Otherwise, create a new cluster that is defined solely by the new data item.

This algorithm is appealing because it is efficient and simple; however, it is too inflexible for the dynamic nature of the data. Each cluster has a fixed location and a fixed size, established by the first data item assigned to the cluster

and the user defined threshold, respectively.

To counteract these drawbacks, we use the *k*-means algorithm to establish clusters with enough data items to provide meaningful mean and standard deviation values for each feature. The mean describes the location of the cluster and the standard deviation describes the size. Once the clusters are established, each new data item is either added to an existing cluster, in which case the feature means and standard deviations are updated, or the data items is flagged as anomalous. The threshold depends on the size of the cluster, and is therefore a function of the feature standard deviations. See Pawling, Chawla, & Madey (2007) for a detailed description of the online hybrid clustering algorithm.

THE SIMULATION AND PREDICTION SYSTEM

Once an anomaly is detected, the simulation and prediction system generates hypotheses describing the crisis based on the current state of the cellular communication network. Using these hypotheses and a pre-defined set of crisis scenarios, the SPS starts a suite of simulations in an attempt to predict the evolution of the situation.

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 nor entirely realistic; rather they provide a reasonable starting point for this proof of concept prototype. The categories are as follows:

- *Flock*: 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 (Burgess & Pearlstein, 1999).
- *Flee*: Agents move away from a disturbance. This 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 and 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).
- *Jam*: Agents move towards their customary goals, but are constrained, as in a traffic jam. 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 highways in Texas in 2005 during the Hurricane Rita evacuation (Harden & Moreno, 2005).

As the simulations proceed, they are periodically validated and verified using new data from the telecommunication network. We use the Kolmogorov-Smirnov (KS) test to compare the output of each simulation with empirical data. According to Banks, Carson, Nelson, & Nicol (2005), goodness of fit tests, such as the KS test, are sensitive to sample size and have a tendency to reject candidate distributions for large sample sizes, so we are careful to limit the size of samples generated by the simulations.

Our approach for addressing the challenges of creating and updating agent based simulations in a DDDAS is to aggregate sensor data to a larger level, to the cell-tower level in the case of WIPER, and to introduce random variation in the data. A naive approach would suggest that it is important to maintain as much data as possible about each individual in the system. Cell phone data could potentially yield location of every individual within a few meters, though our current data set does not have this level of resolution. Canonical model development practice suggests that such an approach would be counter-productive, leading to naive realism, unduly adding complexity to the model without a corresponding increase in model accuracy or usefulness (Banks et al., 2005, Grimm et al., 2005).

Figure 3 shows a graphical comparison of two approaches to revising simulations with streaming data: updating and reparameterizing. We define updating to refer to the process of restarting simulations with approximate information on agent locations (and other parameters). In the figure, this corresponds to receiving information on the number of agents in a Voronoi cell, but without specific location information on each agent. We define re-parameterizing as the

process of maintaining a 1 to 1 correspondence between human beings in the real world and agents in the simulation. As information streams in about the corresponding referent in the real world, we modify the state of each agent to reflect it's current location. Note that, while the simulations have a precise location for each agent, the real world data approximates an individual's location using the Voronoi lattice. When updating or reparameterizing a simulation, the agents are positioned randomly within the appropriate Voronoi cell.

Schoenharl (2007) describes the simulation and prediction system in detail.

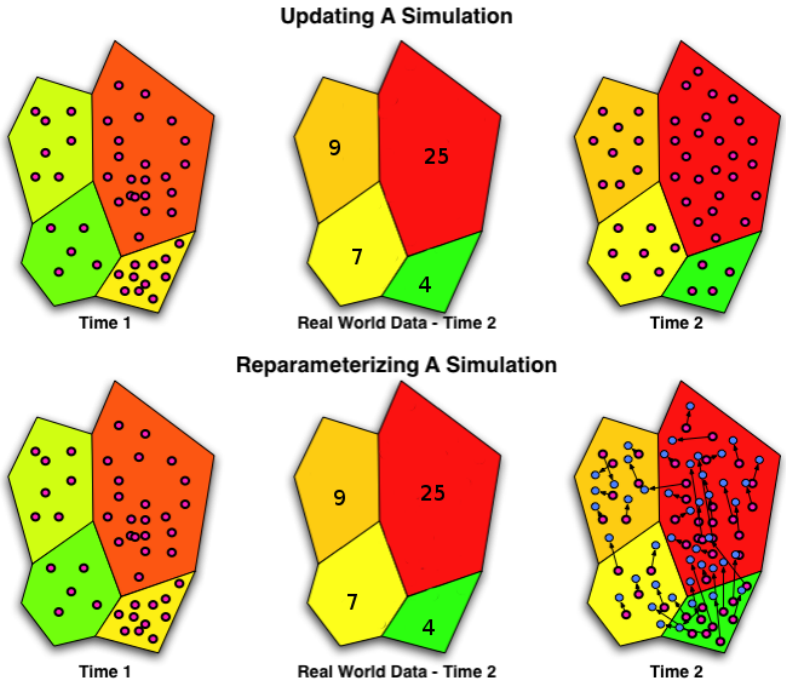

Figure 3: A graphical comparison of updating and reparameterizing simulations from streaming data. When reparameterizing a simulation, agent locations and parameters are changed to conform to the streaming data. Updating a simulation causes larger-scale properties, such as numbers of agents in a cell, to be reset. Note that while the simulation agents have specific locations, the real world data consists only of count data for each Voronoi cell. Whether updating or reparameterizing the simulation, the agents are distributed randomly within their respective cells.

THE DECISION SUPPORT SYSTEM

The decision support system (DSS) is the web-based front end through which emergency response managers interact with the WIPER system. Figure 4 shows a screen-shot of the web-based console. The DSS displays the state of the SPS simulations, the real-time status of the telecommunication network, and anomaly alerts generated by the DAS. Emergency response managers will be able to specify and evaluate mitigation plans, using agent-based simulations, through the web interface. Emergency response managers will be able monitor crisis areas using satellite maps and GIS images overlaid with activity data, as well as view the raw data from the telecommunication network.

THE REAL-TIME DATA SOURCE

The real-time data source (RTDS) forms the bridge between the cellular service provider and the WIPER system. The RTDS will receive raw transaction data from the service provider and aggregate and filter the data as needed by the components of the WIPER system. The RTDS will stream the processed data to the appropriate components.

The real-time data source must be a real-time system, meaning that in addition to the functionality constraints, the system must conform to temporal constraints, providing guarantees on the maximum time required to complete specific tasks (Sha et al., 2004). We want to distribute data to each component as quickly as possible; however, we also want all streams to be reasonably synchronized so that no component gets too far ahead or behind of the others.

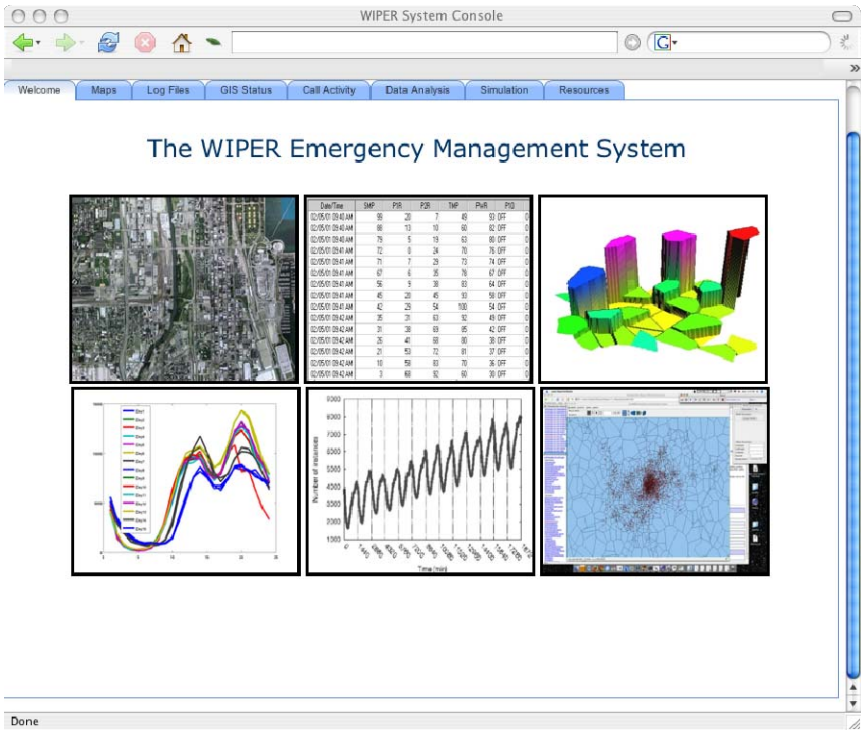

Figure 4: The DSS web-based console of the working WIPER prototype. The console provides easy, standards-compliant access to all of the components of the WIPER system, allowing emergency planners access to the real-time data, both overall activity and spatially aggregated, simulation output and information on system status. The components of the system seen here are (clockwise, beginning in the upper left corner): Satellite map of the affected area, raw data from cellular service provider, 3D activity intensity map, 2D plot of city-scale network activity, historical trend of activity and 2D visualization of the city simulation.

The main challenge we face in developing the RTDS is the lack of control over the service provider's network, meaning that we cannot make use of real-time network protocols in transmitting data from the towers to the RTDS, and we can make no assumptions about the order in which transactions are received by the RTDS. Instead, we will empirically and adaptively estimate the time required for transaction information to traverse the network and arrive at the RTDS. Using this value, we will determine when each item of the aggregated data streams is sent to the WIPER components. With this dynamic estimate, we will attempt to guarantee that most, ideally all, data items for an aggregation interval have arrived before sending the data while preventing the delay from becoming too large.

The RTDS is driven by an aperiodic process: the arrival of transaction items; therefore, the standard periodic task model for real-time systems (Liu & Layland, 1973) is not suitable. Instead we use the rate-based execution model, which defines the temporal requirements for a task by four parameters: (1) a time interval, y, (2) the maximum expected number of transaction arrivals, x, in a time interval y, (3) the maximum desired time, d, required to process each transaction, and (4) the maximum execution time, e, to process each transaction. When the rate at which transaction records arrive is less than y, the transactions will be processed within the desired time, d; however, when the transaction rate exceeds y, the additional transactions are delayed (Jeffay & Goddard, 1999). A system using the periodic task model will fail if the transaction rate becomes too large (Pawling, 2007); however, the rate-based execution model is robust to overload at the cost of delayed execution.

SUMMARY

In this paper, we have described the WIPER system and implementations of several of its components. The WIPER system relies on concepts from a variety of areas in computer science, including online data mining for detecting anomalies in data from a cellular communication network, dynamic data driven application systems for the online validation of predictive simulation, web services for communication among the WIPER system components, and real-time systems for timely and synchronized aggregation and distribution of transaction data from the cellular service provider.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No. CNS-0540348.

REFERENCES

1. Banks, J., Carson, J., Nelson, B., & Nicol, D. (2005). *Discrete-event system simulation* (Third ed.). Upper Saddle River, NJ: Prentice Hall.
2. Burgess, J., & Pearlstein, S. (1999, December 1,). *Protests delay WTO opening.* Washington Post.
3. Cheu, E. Y., Keongg, C., & Zhou, Z. (2004). On the two-level hybrid clustering algorithm. In *International conference on artificial intelligence in science and technology* (pp. 138–142).
4. Chipman, H., & Tibshirani, R. (2006). Hybrid hierarchical clustering with applications to microarray data. *Biostatistics, 7*(2), 286–301.
5. Douglas, C., & Deshmukh, A. (2000, March). *Dynamic data driven application systems.* http://www.dddas.org/NSFworkshop2000.html.
6. Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W. M., Railsback, S. F., Thulke, H.-H., Weiner, J., Wiegand, T., & DeAngelis, D. L. (2005, November). Pattern-oriented modeling of agent-based complex systems: Lessons from ecology. *Science, 310*(5750), 987-991.
7. Harden, B., & Moreno, S. (2005, September 23,). *Thousands fleeing Rita jam roads from coast.* Washington Post.
8. Hartigan, J. A. (1975). *Clustering algorithms*. New York, NY, USA: John Wiley & Sons.
9. Ihler, A., Hutchins, J., & Smyth, P. (2006). Adaptive event detection with time-varying poisson processes. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining.* New York, NY, USA: ACM.
10. Ihler, A., Hutchins, J., & Smyth, P. (2007, December) Learning to detect events with Markov-modulated Poisson processes. *ACM Transactions on Knowledge Discovery from Data, 1*(3), 13.
11. Jain, A. K., Murty, M. N., & Flynn, P. J. (1999, September). Data clustering: A review. *ACM Computing Surveys, 31*(3), 264–323.
12. Jeffay, K., & Goddard, S. (1999, December). A theory of rate-based execution. In *Proceedings of the 20th IEEE real-time systems symposium* (pp. 304–314).
13. Liu, C. L., & Layland, J. W. (1973). Scheduling algorithms for multiporgramming in a hard-real-time environment. *Journal of the ACM, 20*(1), 40–61.
14. Madey, G. (2008). *WIPER: The integrated wireless phone-based emergency response system.* http://www.nd.edu/~dddas.
15. Madey, G. R., Barabási, A.-L., Chawla, N. V., Gonzalez, M., Hachen, D., Lantz, B., Pawling, A., Schoenharl, T., Szabó, G., Wang, P., & Yan, P. (2007). Enhanced situational awareness: Application of DDDAS concepts to emergency and disaster management. In Y. Shi, G. D. van Albada, J. Dongarra, & P. M. A. Sloot (Eds.), *Proceedings of the international conference on computational science* (Vol. 4487, pp. 1090–1097). Berlin, Germany: Springer.
16. Madey, G. R., Szabó, G., & Barabási, A.-L. (2006). WIPER: The integrated wireless phone based emergency response system. In V. N. Alexandrov, G. D. val Albada, P. M. A. Sloot, & J. Dongarra (Eds.), *Proceedings of the international conference on computational science* (Vol. 3993, pp. 417–424). Berlin, Germany: Springer-Verlag.
17. Mitzenmacher, M., & Upfal, E. (2005). *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge, United Kingdom: Cambridge University Press.
18. Pawling, A. (2007, October). *A system for detecting anomalies in data streams for emergency response applications.* Dissertation Proposal.
19. Pawling, A., Chawla, N. V., & Madey, G. (2007, December). Anomaly detection in a mobile communication network. *Computational & Mathematical Organization Theory*, *13*(4), 407–422.
20. Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on computer graphics and interactive techniques* (pp. 25–34). New York, NY, USA: ACM Press.
21. Schoenharl, T. (2007). *Updating and validating simulations in a dynamic data driven application system*. Unpublished doctoral dissertation, University of Notre Dame.
22. Schoenharl, T., Bravo, R., & Madey, G. (2006). WIPER: Leveraging the cell phone network for emergency response. *International Journal of Intelligent Control and Systems*, *11*(4), 209–216.

23. Schoenharl, T., Madey, G., Szabó, G., & Barabási, A.-L. (2006). WIPER: A multi-agent system for emergency response. In B. van de Walle & M. Turoff (Eds.), *Proceedings of the 3rd international information systems for crisis response and management conference.*
24. Sha, L., Abdelzaher, T., Arzén, K.-E., Cervin, A., Baker, T., Burns, A., Buttazzo, G., Caccamo, M., Lehoczky, J., & Mok, A. K. (2004). Real time scheduling theory: A historical perspective. *Real-Time Systems*, *28*, 101–155.
25. Surdeanu, M., Turmo, J., & Ageno, A. (2005). A hybrid unsupervised approach for document clustering. In *Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining.*
26. Tatomir, B., & Rothkrantz, L. (2005, April). Crisis management using mobile ad-hoc wireless networks. In *Proceedings of the second international ISCRAM conference.*
27. Thomas, M., Andoh-Baidoo, F., & George, S. (2005). EVResponse - moving beyond traditional emergency response notification. In *Proceedings of the eleventh Americas conference on information systems.*
28. Yan, P., Schoenharl, T., Pawling, A., & Madey, G. (2007, October). *Anomaly detection in the WIPER system using a Markov modulated Poisson process.* Working Paper.