On the Universal Generation of Mobility Models

Alberto Medina
Raytheon BBN Technologies
amedina@bbn.com

Gonca Gursun
Computer Science Dept.
Boston University
goncag@cs.bu.edu

Prithwish Basu
Raytheon BBN Technologies
pbasu@bbn.com

Ibrahim Matta
Computer Science Dept.
Boston University
matta@cs.bu.edu

Abstract*—Mobility models have traditionally been tailored to specific application domains such as human, military, or ad hoc transportation scenarios. This tailored approach often renders a mobility model useless when the application domain changes. Furthermore, the failure to adapt the mobility model to accurately match the new domain naturally leads to wrong conclusions about the performance of protocols and applications running atop. In this paper, we propose a mobility modeling framework based on the observation that the mobility characteristics of most mobility-based applications can be captured in terms of a few fundamental factors: (1)** *Targets; (2)** *Obstacles***; (3)** *Dynamic Events***; (4)** *Navigation***; (5)** *Steering behaviors***; , and (6)** *Dynamic Behaviors***. We have designed and implemented a Universal Mobility Modeling Framework (UMMF), which enables the instantiation of a mobility model from a wide** *universe* **of possibilities defined by the aforementioned factors. We describe the mapping from application-domain-specifics to UMMF elements, demonstrating the power and flexibility of our approach by capturing representative mobility models with good accuracy in terms of a large number of topological metrics. We also describe several specific mobility scenarios and their UMMF-based model representations.**

I. INTRODUCTION

Advancing the state-of-the-art of algorithms and protocols for mobility-bound networks (e.g. mobile ad hoc networks) is constrained by limitations of available approaches for characterizing and modeling the node mobility patterns. Consequently, there is an imperative need of mobility models that are representative of the application domain scenarios associated with them. There are two main paths to meeting such a need: (1) the *Model-to-Trace* approach, corresponding to the development of *mathematical* mobility models that attempt to capture the mobility characteristics of certain scenarios; and (2) the *Trace-to-Model* approach, consisting of measuring mobility traces from actual applications, characterizing them, and then designing mobility models derived from such characterizations. Both approaches provide their own set of advantages, and both come with some limitations. The work in this paper focuses on the former approach, but we also discuss how our contributions impact the advancement of the *Trace-to-Model* approach (see Section VII).

There has been a plethora of research efforts focusing on narrow set of issues involved in the *Model-to-Trace* approach [6]. Despite of such efforts, our community has not yet reached a state in which accuracy and representativity of defined mobility models can be assumed soundly and ubiquitously. Our contributions in this work stem from our focus on a challenging *invariant* that can be observed in mobility-related research efforts across the board: there exists, in general, a one-to-one mapping between application domain scenarios (e.g. DTN, MANETs, VANETs, etc.) and the associated mobility models that seek to represent them. Consequently, modeling new application scenarios usually entails either the creation of new models from scratch, or the mapping of such scenarios to general mobility models (e.g. Random Way-point (RWP) [4], Reference Point Group Mobility (RPGM) [9]) that are not representative exactly because of their generality. This makes it difficult to comprehensively assess the performance and correctness of new protocols and algorithms for mobile networks.

In this work, we introduce *UMMF: a Universal Mobility Modeling Framework* based on the observation that the mobility characteristics of most mobility-based applications can be captured in terms of a relatively small set of fundamental factors such as: (1) *Target*; (2) *Obstacles*; (3) *Dynamic Events*; (4) *Navigation*; (5) *Steering behaviors*; , and (6) *Dynamic Behaviors*. In this paper we describe the design of UMMF, and show its effectiveness by *instantiating* several mobility models from the wide *universe* defined by the aforementioned fundamental factors.

The advantages offered by the UMMF framework are manifold: (1) better modeling realism; 2) reproducibility of research results; (3) enabling of basic research on dynamic topologies, MANETs, and other intrinsically mobile application scenarios; and (4) aiding the development of techniques to translate real mobility traces to accurate synthetic mobility models. The contributions of this work fall in all the above categories, and therefore we argue that UMMF will play a fundamental role in advancing the state-of-the-art of mobility-related research and applications.

The rest of this paper is organized as follows. Section II summarizes the state-of-the-art in mobility modeling; Section III discusses the notion of universality underlying UMMF,

The research of Medina and Basu was sponsored in part by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement #W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. Basu also received partial support from the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053. The work of Gursun and Matta was partially supported by NSF CISE/CCF #0820138 and CISE/CSR #0720604.

and describes the design and implementation characteristics of tool implementing UMMF; Section IV describes the anatomy of a UMMF-based mobility model; Section V describes the interfaces between UMMF and external systems; Section VI demonstrates the power and flexibility of our approach for defining representative mobility models by describing the mapping from application-domain-specifics to UMMF elements; Section VII discusses issues related to the *Trace-to-Model* approach; and finally Section VIII concludes the paper with future research directions.

II. RELATED WORK

The imperative need for mobility models for the study of mobility-sensitive protocols and algorithms has propelled significant research efforts in the area of mobility modeling. These efforts trickle down into two general areas: (1) the development of abstract mobility models, which work closer to the notion of *one-size-fits-all* models that could be used across-the-board in different research efforts; and (2) the development of tailor-made mobility models directly aimed at specific application domain scenarios. Works such as [2] and [6], survey the state-of-the-art in mobility modeling and applications research.

Abstract mobility models, such as the classic Random Waypoint (RWP [4]), and its group mobility counterpart, Reference Point Group Mobility (RPGM [9]), have been widely used by researchers who want to evaluate their protocols under node mobility [6]. On one hand, abstract models are attractive due to their simplicity and analytical tractability. On the other hand, these type of models are not representative of the application scenarios in which they are applied, because of their generality.

In contrast, tailor-made models have been developed for specific application domain scenarios. For example, in the area of Disruption Tolerant Networking (DTN), mobility models such as Message Ferrying [22] and Data Mules [20] have been proposed; and the influence on mobility exerted by environmental factors such as obstacles, pathways, and attraction points (i.e. popular places) has been studied [12]. Such models are more representative, but they lack flexibility since they are too contextual.

Recently, several research efforts ([8], [14]) have studied the mobility characteristics of humans in some specific contexts; other efforts ([16], [17]) have studied the mobility modeling problem from the perspective of social networking theory; and [18] proposes the concept of event and role based modeling specifically tailored to the domain of disaster recovery networks. Works like these expand the universe of tailor-made mobility models, which arguably are more representative than abstract models, but again, it is difficult to apply them elsewhere outside of their target scenarios.

There have been research efforts [15] seeking to incorporate techniques born in the domain of AI Game programming [5], such as the use of decomposing the the mobility patterns in a game environment into a set of individual behaviors that can be combined to enable mobile agents to exhibit *pseudo-intelligent* behaviors. There is an overlap between our work and works such as [15]. However, we argue and demonstrate in this paper that such behavioral building blocks are inadequate to fully capture complex mobility scenarios, and other equally important building blocks must be available during the modeling process. Furthermore, our work has a larger scope with respect to providing an environment that enables the use of building blocks to construct complex mobility models, and generate from them mobility traces and dynamic topology statistics to be used across-the-board in mobility-related research efforts. Bai et al also propose a framework for specifying and and generating mobility models with a larger scope than the traditional mobility modeling approaches [3]. However, our methodology and framework is more general as it provides a building-block approach to mobility model generation, and offers a larger scope for generating a bigger *universe of mobility models* and a larger set of target application domain scenarios in which they can be applied.

The above summary of the state-of-the-art in mobility modeling elicits a fundamental issue: capturing representative mobility characteristics implies a one-to-one mapping between target application domain scenarios, and the associated mobility models. Any research effort in an application domain not considered before, will entail the development of a tailor-made mobility model. This issue is at the crux of our motivation for providing a mobility modeling framework like UMMF, which closes or at least significantly narrows the gap between the world of mobility modeling and the research in specific application domain scenarios that need them.

III. UMMF: A UNIVERSAL MOBILITY MODELING FRAMEWORK

UMMF enables the *universal* generation of mobility models. The notion of universality in this context comes from its mathematical definition (i.e. set theory), where a *universe* corresponds to a set containing all elements with certain characteristics. The main observation underlying our work is that the mobility characteristics of most mobility-based applications can be captured in terms of a relatively small number of fundamental factors: (1) *Targets*; (2) *Obstacles*; (3) *Dynamic events*; (4) *Navigation*; (5) *Steering behaviors*; and (6) *Dynamic Behaviors*. UMMF-based models are formed by *composing* a subset of such fundamental mobility building blocks. In this section, we describe the elements of UMMF, and in later sections we demonstrate its use and applicability in the context of several mobility-sensitive application scenarios.

Figure 1 depicts a hierarchical diagram of the elements comprising a UMMF-based mobility model including: (a) a model environment, which encompasses the modeled *geographical plane*, *targets*, *target sets*, *obstacles*, and *dynamic events*; (b) a *navigation graph* enabling the navigation capabilities of mobile agents; (c) a set of *steering behaviors*, which can be applied individually or in combination to capture the notion of physical forces underlying observed mobility patterns with different levels of complexity; and (d) *scripted dynamic behaviors*, enabling the user to influence the execution of mobility models. In addition to these elements, UMMF-based models

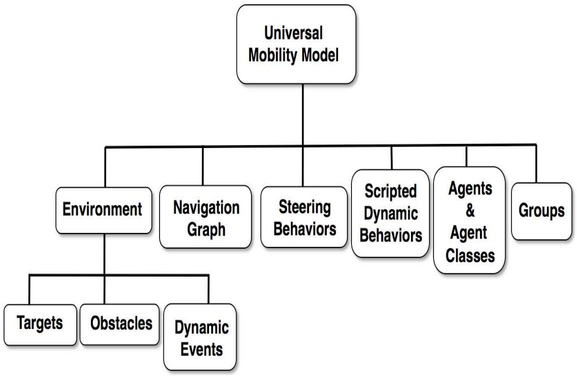


Fig. 1. Hierarchy of Elements Conforming a UMMF-based Mobility Model

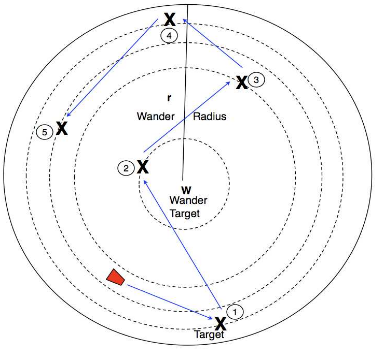


Fig. 2. Example of an Implicit Target Selection process

define *mobile agents* classified into agent classes with specific properties, and group specifications dictating how the defined mobile agents relate to each other. Agents and groups of agents interact not only among themselves but also with the other UMMF building blocks. Below we provide a description of the role played by all UMMF elements.

A. Environment

A UMMF-based model environment involves a simulated geographical area, which is divided into *cells*. Cells play an important role in capturing the semantic characterization of mobility scenarios. The location and dimensions of elements such as targets, obstacles, and dynamic events, are all specified in terms of cell numbers. Furthermore, UMMF environment cells can have different semantic information associated with them (e.g. altitude, threat-level), aimed at increasing model realism.

1) Targets: UMMF Targets are associated with the mobility objectives of agents and groups (i.e. destinations, mission goals). The process of *Target Selection* can be part of the strategic mission goals of the application scenario (*explicit target selection*), or alternatively, targets may be selected implicitly, such as selecting targets in relation to other UMMF elements (i.e. agents, obstacles etc.), or just randomly (e.g. RWP). For example, in RPGM, group members make continuous implicit target selection decisions by following their leader trajectories.

Figure 2 shows the “Wander” target selection scheme provided by UMMF, which captures the scenario of mobile agents wandering around certain geographical location. r corresponds to a *Wander Radius* parameter, the Xs represent targets selected at different time steps, and W represents the center of the Wandering area. An agent employing the Wander target selection process will select random points on the circumference of concentric circles centered at W.

Target selection can also be semantically associated with different attractiveness levels of geographical locations or elements in a given environment. Such level of attraction can be associated with geographical locations, the information associated with the cells surrounding a given location, and also with the properties of individual agents, agent classes, or groups. Note that target selection is a dynamic process. As the simulation evolves, the goals set for mobile agents may change, and such changes will be reflected in changes regarding target selection.

2) Obstacles: UMMF *Obstacles* play the important role of capturing the semantics of environment elements constraining the movement of agents. Mobility constraints can translate into *obstructed routes*, or into *repulsion forces* exerted on agents, causing them to avoid certain areas as they travel towards their targets. The former case is captured in UMMF by means of *navigation graphs*; while the second case is captured by the computation of *repulsive steering forces* (see Section III-C).

3) Dynamic Events: UMMF enables the modeling of two types of general dynamic events. One corresponds to events taking place at specified or pseudo-randomly chosen times; and the other corresponds to *dynamic obstacles*, which emerge in time like dynamic events and effectively obstruct or invalidate plane areas. Furthermore, UMMF allows the specification of scripted dynamic behavior which can be associated with dynamic events and dynamic obstacles (see Section V).

B. Navigation Graphs

UMMF implements the path planning aspects of a mobility model by using *Navigation Graphs (NavGraphs)*. A *NavGraph* is a graph with nodes representing geographical locations, and edges representing the adjacencies between them. Different types of *NavGraphs* can be created, all aimed at defining and/or constraining the paths through the environment which could be followed by agents to reach their target destinations.

The current UMMF implementation uses *tile-based NavGraphs*, which are constructed by dividing the modeled plane into cells, and assigning a *NavGraph* node to each one. Higher or lower resolution *NavGraphs* can be defined; furthermore *NavGraphs* could also be laid down following different rules (e.g. *Points-of-Visibility, Expanded Geometry [5]*. Notice that a mobile agent is not restricted to move only through the nodes of a NavGraph; instead, they use the NavGraph structure to plan paths between their current locations and the implicit/explicit destinations they choose. The bottom line regarding *UMMF NavGraphs* is that they offer a flexible mechanism to enable and control the locomotion of mobile agents withing the modeled plane.

C. Steering Behaviors

UMMF employs the notion of Steering behaviors to capture the forces that may be underlying some of the observed motion patterns of mobile agents. In an obstacle-free environment, agents can move in the Euclidean direction to their destinations. When environmental constraints such as obstacles exist, agents need to exhibit some level of intelligent behavior and adjust their paths accordingly. UMMF uses Steering Behaviors acting as *attraction* and *repulsion* forces, effectively enabling agents to react to the relationship between themselves and other agents, and the environment.

From a physical perspective, forces can be exerted on agents either individually or in combination. Individual forces corresponds to those such as gravity pulling an object down, a wind force pushing a sailboat forward, and so on. These individual forces can be combined to produce different effects. For example, a gravitational force can be combined with the force exerted by a table, causing an object to remain stationary. Using Newton's laws, a steering force $\vec{S}$ is converted into an agent's acceleration $\vec{a}$, velocity $\vec{v}$, and position $\vec{x}$ after each time step T:

$$\vec{a} = \vec{S}/m; \qquad \vec{v} = \vec{v_0} + \vec{a}T; \qquad \vec{x} = \vec{x_0} + \vec{v}T,$$

where m, $\vec{v_0}$, $\vec{x_0}$ are the mass, initial velocity and the initial position of the mobile agent respectively.

Table I summarizes the steering forces used in our framework. These forces can be categorized into: 1) *Individual Behaviors*; and 2) *Group Behaviors*. Individual behaviors involve forces such as *Seek*, *Arrive*, and *Flee*, which cause agents to react individually to environmental factors or their target selection process; and group behaviors involve forces such as *Pursuit*, *Interpose*, *Evade*, *Obstacle avoidance*, and others, causing agents to behave in accordance to their relation to other agents, and to the relation between them and the environment.

Note that group behaviors can be defined by a combination of individual behaviors that can together cause a number of mobile agents to behave as if in a group. For example, the individual forces of *Separation*, *Cohesion* and *Alignment*, can be combined in a group of nodes to achieve *Flocking* behavior among them. Figure 3 depicts three examples of individual forces: (1) *Seek*, which results in a force vector obtained by adding the current force vector for a given node, with a *desired* vector pointing directly to the target destination; (2) *Pursuit*, which involves an *evader* agent and a *pursuer* agent and results in a force vector obtained by adding the current force vector of the pursuer with a force vector pointing to a predicted location for the evader; and (3) *Hide*, which also involves a pursuer and an evader, but in this case the latter's steering force results from selecting a hiding place behind any obstacle that may be interposed between itself and the pursuer. Figure 4 shows an example of the combination of individual steering forces at a group of nodes to achieve a more complex *emergent* group behavior known as *flocking*. Note that the composition of the Pursuit, Separation, Cohesion, and Alignment steering behaviors generates *correlated mobility models*. Figure 5 also illustrates a similar scenario.

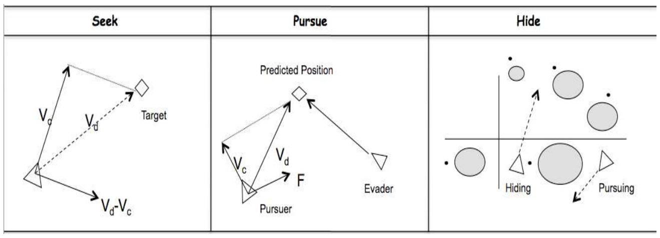


Fig. 3. Individual Steering Forces

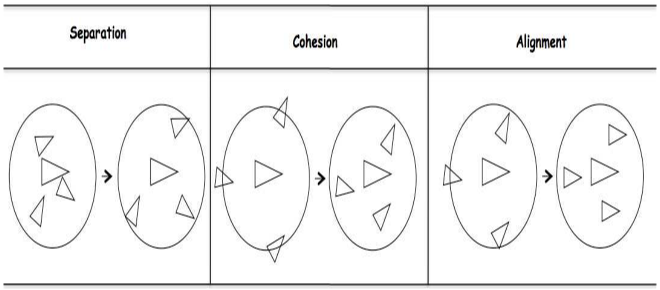


Fig. 4. Combination of Individual Steering Forces to achieve flocking

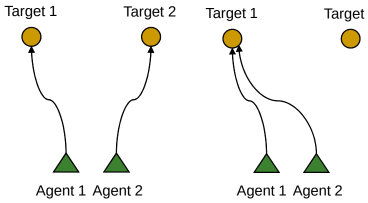


Fig. 5. Target Selection and correlated mobility: (a) attractive force exerted by each target on its corresponding agent is greater than the inter-agent *attractive* steering force; (b) attractive force exerted by target 2 on agent 2 is lower than the inter-agent attractive steering force, hence agent 2 *pursues* agent 1.

At each time step t, a steering force vector $\vec{F_R}(t)$ is computed for each agent. Each steering force, $\vec{F_X}$, has a weight w_X associated with it, which is specified as an input parameter to UMMF. These weights can be either constants or variable in time depending on the mobility model semantics.

$$\vec{F_R}(t) = \sum w_X(t)\vec{F_X}(t) \tag{1}$$

As depicted in Equation 1, $\vec{F_R}(t)$ is obtained by multiplying all steering forces by their weights and adding them up. The weights of steering forces not playing a role in a given context will be set to zero, effectively disabling their influence in the above sum.

a) Combining building blocks: Steering and Target Selection: The power and flexibility provided by UMMF results from the capability of combining (or composing) its building block elements (e.g. target selection, steering forces, etc.) to produce complex mobility scenarios. An example of this is provided by the specification of a *Wandering* steering behavior in UMMF.

Wandering behavior is modeled in UMMF by specifying for each node a *Wander Target*, which can be changed dynamically, and a *Wander Radius*. These parameters are defined in a Steering Behavior class, at the core of UMMF functionality,

TABLE I
STEERING BEHAVIORS

Type	Behavior	Steering Force Description
Individual Behaviors	Seek	*Attraction force* that draws an agent to a particular target.
	Flee	*Repelling force* that causes an agent to move away from a given geographical location.
	Arrive	*Attraction force* that enables agents to halt their movement upon reaching a target.
	Pursuit	*Attraction force* that is employed in cases where a mobile agent is expected to intersect another agent or any moving element.
	Hide	*Repulsion force* that causes an agent to position itself so that an obstacle is located always between itself and the line of sight of another agent/enemy.
	Evade	*Repulsion force* that enables agents to move towards the opposite direction of an expected intersection with another node.
	Wander	*Attraction force* that causes an agent to behave as a random walker.
	Obstacle and Wall Avoidance	*Repulsion forces* that enable agents to avoid (1) dynamic obstacles as they are encountered on path traversals, and (2) walls.
Group Behaviors	Alignment	*Attraction force* that keeps an agent aligned with respect to others in its group.
	Separation	*Repulsion force* that separates an agent from the others in its group.
	Cohesion	*Attraction force* that causes an agent to move towards the center of mass of its neighborhood.
	Flocking	Combination of *separation*, *alignment*, and *cohesion*.

which is a data member of the agent class. At each time step, a *wandering* agent picks a random circle centered at the current *Wander Target*, by choosing a random radius within a range defined by the *Wander Radius* parameter, and a random angle. Then the agent selects as its target the corresponding point on the circumference of the selected random concentric circle. Finally, an *Arrive* steering force is exerted on the agent, propelling the agent towards its target.

D. Agents, Agent Classes, and Groups

Different types of agents can be defined in UMMF. Unlike most mobility models, UMMF models do not force all agents to move according to the same rules. This is an important flexibility that promotes realism since real-world scenarios are intrinsically heterogeneous in this respect. In UMMF, the specification of mobile agents is primarily based on the notion of groups. Any UMMF model involves the placement and mobility evolution of a set of agent groups. Mobility scenarios entailing a set of nodes operating without group constraints (i.e. RWP), are captured in UMMF by having a single group with no leader.

E. Dynamic Behavior Specification

UMMF enables the modeling of dynamic events during a simulation. For instance, a “bomb explosion”, which destroys a part of the simulation area in a military scenario can be modeled as a dynamic event in UMMF. Dynamic events may cause the alteration of the rules governing the movement of agents, invalidate sub-graphs of the navigation graph, and change the properties of the terrain in the surrounding areas.

IV. EXECUTION OF A UNIVERSAL MODEL

The execution flow of a UMMF-based mobility model is divided into three main stages: (1) Node placement and interconnection; (2) Topology evolution; and (3) Generation and channeling of mobility-related traces. In this Section we elaborate these execution aspects.

A. Event-driven Mobility Simulations

At the core of UMMF there is an event-driven simulation engine carrying out the dynamic evolution of mobility models as a long sequence of events. Each event has an associated event handler with it, and the processing of a given event can result in the generation of one or more additional events to be triggered subsequently.

For example, capturing snapshots of a modeled mobile topology is achieved by defining a set of *Snapshots* events and scheduling their occurrence times at the beginning of the simulation. When each of these events is triggered, the current state of the simulation is dumped into the associated output files, including a set of computed snapshot topology statistics.

B. Node Placement

In the current UMMF implementation nodes are initially placed randomly across the model environment/plane. Extending UMMF to include different node placement strategies is straightforward; for example, nodes could be distributed according to a skewed distribution (e.g. heavy-tail), in a mesh, etc.

C. Network Connectivity

In a network with intrinsic mobility, the connectivity between nodes must be constantly recomputed. In UMMF, nodes are placed and interconnected at the beginning of a model execution, and thereafter connectivity is recalculated at parametrized fixed intervals (e.g. one second). Currently, the UMMF implementation does not include sophisticated *signal propagation models* that could take obstacles and environmental conditions into consideration in order to determine the existence (or the lack thereof) of links between agents. Instead, the establishment of links is based on a simple line-of-sight approach. However, the mobility traces output by UMMF, which contain location information at the defined *snapshot granularity*, can be used in simulation environments that contain advanced propagation models. Part of our ongoing work is focused on expanding this dimension of the UMMF to incorporate realistic propagation models.

D. Topology Evolution

Once nodes are placed on the environment area, and their initial connectivity has been established, UMMF enters a period of *topological evolution*. At the beginning of this evolution period, UMMF performs three tasks: (1) scheduling the set of initial movements of all agents. For example, in a group-mobility scenario, these "initial movement" events will include the selection of a first target for the group leaders, and the associated path planning for them to get to those destinations (i.e. navigation graph operations); (2) scheduling the set of periodic and dynamic events. Currently, the set of periodic events corresponds to what we call *snapshot events*; aimed at taking a snapshot of the current state of the network, including the computation of a set of topological metrics (Section V-C), and updating the output data provided for the elements interfacing UMMF (see Section V. The set of dynamic events corresponds to those defined in the XML configuration file; and (3) initializing the visualization and output data interfaces, which will be used as the model simulation evolves. Following is a description of the handling of periodic and dynamic events.

1) Periodic Events: UMMF's XML configuration file defines a *Snapshot parameter*, which dictates the periodicity in which snapshots of the dynamic topology will be taken. By default this parameter is set to one second. Upon the occurrence of a snapshot event, a *snapshot-handler* function is invoked to perform the following tasks: (1) Update the positions of all nodes; (2) check which nodes arrived to their targets, and set them to perform their next task as dictated by the underlying model (e.g. new target selection and new path finding computation); (3) update the topology configuration (i.e. connectivity); and (4) compute a set of topological statistics associated with the newly formed topology.

Notice that updating the state of the topology, involves computing the current values of all the Steering Forces acting upon the agents in the network. For example, nodes that have approached existing obstacles or geographical locations which are associated with repelling forces will have their steering forces updated to influence their movement and cause them to avoid such obstacles and/or geographical areas.

2) Handling Dynamic Events: UMMF Dynamic events behave similar to their periodic counterpart (i.e. topology snapshots), but they can take place at arbitrary times during a model execution, and that they may have associated with them specific scripted behaviors to be invoked by the function that handles them (Dynamic Event Handler). UMMF uses Lua [10] to allow users to script the execution of certain UMMF functions (e.g. *Change Target*, *Change Target Set*, *Change Steering Behavior class*, etc.). The current UMMF implementation exposes a basic set of such functions to the user, and we envision that this dimension of the framework will evolve significantly and quickly as the tool is employed in a variety of application domain scenarios.

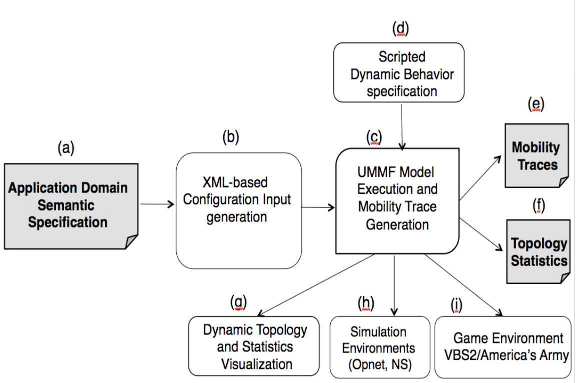


Fig. 6. High-Level UMMF Flow Diagram.

V. UMMF INTERFACES

A fundamental aspect of a framework such as the UMMF is the flexibility it should offer regarding the interfacing of its core functionality with external systems and applications. UMMF provides such flexibility by enabling interfaces with: (1) a dynamic visualization tool (e.g., a custom-developed tool called *VizTools*) for model analysis and topological visualizations; (2) a simulation environment (OPNET [11], NS [19] etc.) to enable the application of UMMF-based models for studying and evaluating network protocols for mobility-based application domain scenarios; (3) data analysis tools, for enabling the study of the fundamental properties that govern the formation and evolution of the generated dynamic topologies; and (4) a dynamic scripting environment (i.e. *Lua* [10]), enabling the direct influence of the modeler on the execution of a UMMF-based model. Below we describe the process flow of UMMF-based models, and provide a description of the aforementioned interfaces.

A. High-level UMMF Flow Diagram

The application of UMMF to modeling scenarios entails a set of stages as depicted in Figure 6: (a) a *semantic characterization* of the mobility patterns associated to the given application domain scenario, such as the objectives of the application (e.g. seek target, perform certain task, return to base, etc.), the characteristics of the mobile agents (e.g. five groups, three agent classes), and the characteristics of the environment where the given application develops (e.g. geographical contour map, static and dynamic obstacles); (b) a *XML-based configuration* file mapping the semantic characterization of the model to a combination of UMMF components; (c) a model execution, in which UMMF's generation engine takes the XML model configuration and generates a set of time-series data for the associated dynamic topology. This box implements all the UMMF elements described in Section III; (d) a *scripted specification* of dynamic behaviors provided by the user (i.e. modeler) based on a *Lua environment* [10]; and a set of UMMF interfaces providing data in the context of (e) *mobility traces*; (f) time-series of topology statistics; (g) *dynamic topology visualization*; (h) *simulation environments*; and (i) *game and other virtual reality environments*. The following subsections provide a brief description of the main UMMF interface components.

B. Mobility Traces

UMMF outputs a sequence of topological snapshots capturing timestamped position information for all agents and other model entities. Furthermore, the output also provides time series data associated with a set of topology statistics. This dimension of UMMF will evolve rapidly in the near future, causing the output contents and characteristics to expand and improve as the framework is applied in a wide variety of application domain scenarios.

C. Topology Statistics

It is important to understand the fundamental properties of mobility-related application scenarios and to develop accurate models for representing them because mobility has a significant impact on communication parameters such as path length, delay, jitter, etc; in isolation, mobility would be of no interest to developers of protocols and algorithms for the related environments. The behavior of these communication parameters is in turn determined by the properties of the dynamic topologies associated with the given scenarios. Therefore, it is the goal of UMMF to produce data that aids in determining the properties of the generated dynamic topologies such that they can be correlated to communication parameters. UMMF computes a set of topology statistics every time a *Snapshot event* takes place. The finer the snapshot granularity, the higher the resolution of the gathered statistics. Two general types of statistics are considered. One set corresponds to metrics directly related to the topological snapshots, such as network diameter, number of connected components, average path length, and the like. The other set corresponds to properties associated directly with the mobility characteristics for the underlying scenarios such as network stability, link duration, inter-contact times, etc.

D. VizTools: Dynamic Visualization

UMMF interfaces with an *in-house-developed visualization tool* called *VizTools*, which provides seamless and quick visualization capabilities. For the sake of space we do not elaborate on VizTools and its integration with UMMF; it suffices to say that the visualization of dynamic topologies, significantly enhances the modeling process, both in terms of the user experience and in terms of the accuracy of the developed models.

E. Lua-Scripted Dynamic Behaviors

Lua [10] is a lightweight, imperative and functional scripting language with extensible semantics. Lua is used in a wide range of application domains such as embedded systems, mobile devices, web servers, and game environments. We incorporated Lua's capabilities into the design of UMMF in order to enable the scripting of dynamic behaviors, which can be specified by users and be associated with the occurrence of dynamic events. When the dynamic event is triggered, the associated Lua script gets executed. UMMF exploits Lua's well-defined application programming interface (API), which allows the invocation of exposed UMMF functionality from the the Lua environment.

F. XML-based Configuration Files

The specification of UMMF models is done through XML-based configuration files whose contents are constrained by a *UMMF XML Schema*. The UMMF XML Schema controls the syntax and grammar specification of UMMF configuration files, helping modelers to provide proper syntax and ordering in their configuration files. UMMF provides a validating XML parser which validates its input against the UMMF XML Schema.

VI. APPLICATION DOMAIN SCENARIOS

In order to appease the imperative need for mobility models to study the correctness and performance of mobility-dependent protocols and algorithms, many mobility models have been proposed in the literature (see Section II). In this Section we elaborate on the flexibility of UMMF for capturing both, "generic" models such as RWP and RPGM, as well as models that are tailor-made for specific application domain scenarios. The cases described correspond to a small subset of the *universe* of models that can be represented and generated using UMMF. Nevertheless, this set nicely illustrates the capabilities offered by our framework.

A. Random Waypoint Model (RWP)

In the RWP model, each agent randomly selects a geographical point as its target, and moves towards it at a constant velocity, randomly selected from a given range $[V_{min}, V_{max}]$. Upon arriving to its target location, each agent pauses for a specified period of time (i.e. pause time). This process is repeated by each agent until the end of the simulation. In UMMF, RWP is implemented by specifying a random target selection process, and agents are "steered" by exerting on them an *Arrive* steering force, causing them to move along straight lines towards their targets. In order to assess the congruence between the original and the UMMF implementation of RWP, we evaluate the following scenario: 50 agents; a 500×500 simulation area; a velocity range of $[2, 10]$; a transmission range for each agent of 50; and a simulation time of 20000 time units (e.g. seconds). Table II compares both implementations with respect to the first four moments of several topological and network properties. The presented results correspond to the average of 20 individual runs, and they include the *Mean* and *Variance* of each metric, as well as the *Skewness* and *Kurtosis* statistics in order to assess the asymmetry and peakedness of the distributions of the metrics.

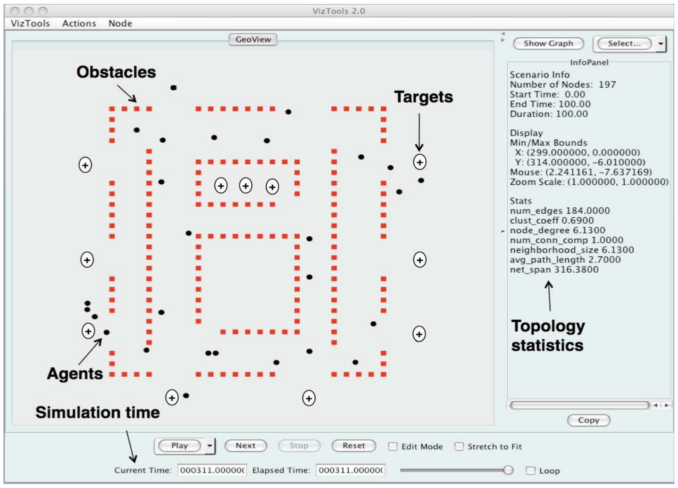
The bottom line delivered by the results in Table II is that both implementations are indeed congruent.

B. Reference Point Group Mobility (RPGM) Model

In RPGM, agents move in groups; two types of agents are defined: group leaders and group members. Group leaders move around the simulated area according to the RWP model, and they provide a reference point for group members. The scoping area for each group corresponds to a circle defined by

TABLE II
RANDOM WAY POINT MODEL MOMENT TABLE

	Mean		Variance		Skewness		Kurtosis	
	RWP	UMMF	RWP	UMMF	RWP	UMMF	RWP	UMMF
Node Degree	2.07	2.06	0.13	0.13	0.57	0.59	3.74	3.84
Path Length	2.74	2.73	0.84	0.84	0.98	0.99	3.89	3.90
Clust. Coeff.	0.71	0.71	0.0028	0.0028	-0.16	-0.15	2.98	3.01
Contact Times	11.56	11.52	1.07	1.07	0.41	0.41	3.36	3.37
Inter-Contact Times	263.65	264.28	903.91	895.45	0.56	0.56	3.53	3.60
Node Speeds	4.98	4.95	0.01	0.01	-0.05	0.00	2.66	3.08
Neighborhood Size	2.07	2.06	0.13	0.13	0.57	0.59	3.73	3.84
Conn. Components	16.79	16.90	9.42	9.51	0.07	0.07	2.97	2.95
Link Breakages	4.38	4.37	4.86	4.87	0.50	0.50	3.33	3.35



```
Scenario Info
Number of Nodes:  197
Start Time:  0.00
End Time: 100.00
Duration: 100.00

Display
Min/Max Bounds
  X: (299.000000, 0.000000)
  Y: (314.000000, -6.010000)
Mouse: (2.241161, -7.637169)
Zoom Scale: (1.000000, 1.000000)

Stats
num_edges 184.0000
clust_coeff 0.6900
node_degree 6.1300
num_conn_comp 1.0000
neighborhood_size 6.1300
avg_path_length 2.7000
net_span 316.3800
```

```
Current Time: 000311.00000(  Elapsed Time: 000311.00000(
```

Fig. 7. VizTools snapshot view of Manhattan model execution

the location of the leader as its center, and a diameter defined by a *Group Span* parameter. Group members move randomly within this defined scope.

In UMMF, RPGM is implemented by specifying a random target selection process for the group leaders, and exerting an *Arrive* force to make them move towards their targets. Group members do not require a target selection process; instead their movement is defined by exerting on them a *Pursuit* force with the group leader as the pursued entity. Given an evader and its speed, the *Pursuit* force acts as an *attraction* force by predicting the geographical point where the pursuer agent can catch the evader agent; once that location is determined, a *Seek* steering force is applied on the pursuer agent. The *Pursuit* force is an example of a relatively more complex steering behavior, combining implicit target selection with a *Seek* steering force.

Table III presents performance results similar to those presented for RWP, corresponding to average values for 20 independent experiments, 10 groups, 5 agents per group, including the leader, a transmission range of 50, a 500×500 simulation area, a fixed agent speed of 5, and a simulation time of 20000 seconds. These results show that both implementations of RPGM are indeed congruent with respect to the analyzed metrics.

C. Manhattan Model

The *Manhattan mobility model* was proposed for the study of *Vehicular Ad hoc networks (VANETS)*, seeking to capture the movement of vehicles/agents within an urban area. A simple implementation of this model in UMMF involves defining: (1) a set of obstacles that constraint the plane area effectively providing the semantics of streets in an urban area; (2) a set of targets which effectively provide the semantics of intersection points; (3) a set of agents which will move around the plane, repeatedly selecting random intersection points and moving towards them. Figure 7 shows a VizTools view of this scenario modeled in UMMF.

D. Disruption-Tolerant Networking (DTN)

DTNs are networks designed to be resilient to disruptive connectivity. Some DTN examples [21] include: (1) urban settings involving vehicles meeting opportunistically, and performing data transactions that enable connectivity between isolated geographical regions; (2) rural environments conformed by a set of *village-like* areas which may have internal connectivity but are isolated among themselves and from the rest of the world; (3) networks of sensors which may be static but are used to gather statistics regarding the movement of other entities, such as animals in wildlife settings; and (4) networks of autonomous robots distributed in a given environment assigned with the task of relaying data between otherwise disconnected areas. The semantics of this type of environments can be effectively mapped to UMMF models. Below we provide a description of a few examples of such mappings.

1) Zebra Mobility: In this model nodes correspond to Zebras moving around in a field. The environment encompasses a set of *grazing areas* and set of *water ponds*. Zebras move independently around the environment following alternating mobility patterns, corresponding to *roaming*, *grazing* and *drinking*.

In UMMF this model is implemented as follows: two target sets are defined; one corresponding to *grazing areas*, and the other corresponding to the set of *water ponds*. Three Steering Behaviors are defined: *roaming*, *grazing*, *drinking*. When in *roaming steering behavior*, Zebras pick random targets and move at higher speeds; while in *eating/drinking* steering behavior, Zebras “Wander” around specified targets. A set of dynamic events are defined. Each dynamic event is associated with a Lua script, containing instructions to change target sets and/or steering behaviors. Zebras start in a grazing mode, and change state as the dynamic events are triggered.

TABLE III

REFERENCE POINT GROUP MOBILITY MODEL MOMENT TABLE

	Mean		Variance		Skewness		Kurtosis	
	RPGM	UMMF	RPGM	UMMF	RPGM	UMMF	RPGM	UMMF
Node Degree	5.92	5.87	1.85	1.74	1.15	1.09	5.62	5.22
Path Length	1.20	1.22	0.06	0.06	2.10	2.06	9.59	9.45
Clust. Coeff.	0.97	0.97	0	0	-1.04	-0.96	4.12	3.94
Distance from Leader	5.06	5.64	0	0.59	-0.65	-0.02	3.11	1.51
Contact Times	12.55	12.47	1.66	1.63	0.56	0.46	3.63	3.16
Inter-Contact Times	280.74	286.74	981.23	1033.70	0.68	0.66	3.75	3.90
Diameter	426.99	431.44	3700	3800	0.09	0.13	3.02	3.03
Neighborhood Size	5.92	5.87	1.85	1.74	1.15	1.09	5.62	5.22
Conn. Components	7.66	7.61	1.67	1.69	-0.35	-0.34	2.90	2.91
Link Breakages	6.60	5.51	64.57	44.05	1.69	1.66	6.62	6.48

2) Message Ferries: This model was introduced in [22] seeking to provide a mobility model in the domain of *Store-Carry-and-Forward* scenarios, where nodes relay data to other nodes as they move around, storing messages until they can be delivered. Two types of nodes are defined: (1) regular nodes, and (2) message ferries. Regular nodes can be static or mobile; message ferries are intended to visit regular nodes according to some routing specification with the purpose of getting and delivering data messages from and to them. This mobility scenario has a wide variety of applications, and research for specific domains involves the design and analysis of route layouts for the message ferries.

This model can be easily captured in UMMF. A simple mapping would be as follows: Two types of agent classes are defined, one for regular agents and the other for message ferries. The routes to be followed by message ferries are set by laying down different target sets (one for each route), which are then followed in round-robin by the message ferries. Two *Steering behavior* elements are defined; one specifies the behavior of the regular agents with zero speed (static agents), and no associated target selection process; the other specifies the behavior of the message ferries, with an *Arrive* Steering force, and a round-robin target selection process to pick targets sequentially from a given target set. Further refinement of this model could involve incorporating different types of communications (i.e. node initiated vs. ferry initiated), defining dynamic events and their associated Lua scripts, which trigger communication transactions by changing the behavior of nodes and/or ferries, such as setting the behavior of regular nodes as *Pursuers* of message ferries or viceversa.

E. Military Mobility Scenarios

In a military context, access to accurate mobility models is of paramount importance for assessing the effects of mobility on the performance of communication exchanges between *Network-centric Warfare* technologies, during military operations. The military spends significant amounts of resources in performing live exercises that seek to evaluate specific technologies in actual, but constrained, environments. The semantics of such military exercises can be effectively mapped to UMMF's building blocks, and the use of the associated models in simulated studies would be extremely valuable.

We modeled in UMMF, a military exercise scenario carried out in Lakehurst, New Jersey. The scenario consisted of a set of military convoys carrying out the mission of leaving their base and traversing a series of scattered checkpoints. During the exercise, a series of simulated bombs are detonated, effectively disabling some of the checkpoints, and causing the agents forming the convoys to scatter and regroup, and subsequently adapt their routing path to avoid disabled checkpoints. Such a scenario was implemented in UMMF very easily by: (a) a set of targets capturing the semantics of the military checkpoints; (b) several groups associated with the military convoys; (c) round-robin target selection for group leaders; (d) exertion of an *Arrive* force on group leaders to push them towards each target; (d) the exertion of a *Pursue* force on member nodes to cause them to follow their leader; (e) the exertion of *Separation*, *Cohesion*, and *Alignment* forces to all nodes, in order to keep the unity of the convoy at all times (e.g. group span and regrouping); (f) a series of dynamic events capturing the semantics of the detonating bombs. These dynamic events do not have a Lua script associated with them; using Lua scripts the semantics of the model could be expanded to have, for example, mobile agents being injured and/or killed by the occurrence of a bomb event in their vicinity; and finally, (g) the exertion of a *Wall-avoidance* force on all nodes to cause them to be repelled by the areas were the bombs detonated.

F. Human Mobility and Social Networks

A very active area of research is centered around studying the topological and mobility features of *social networks*. The work in [8] studied data sets capturing the mobility patterns of a very large number of anonymized mobile phone users. They observed that, in contrast to the random trajectories predicted by the prevailing *Levy Flight* and *Random Walk* models, human trajectories show a high degree of temporal and spatial regularity, with each individual being characterized by a time independent characteristic length scale and a significant probability to return to a few highly frequented locations, showing that despite the diversity of their travel history, humans follow simple reproducible patterns.

These results can be translated into a UMMF-based mobility model conformed by agent groups associated with *communities*; each individual agent can be in one of two states, *local* or *roaming* state. When in local state, an agent moves according to *Wander Steering Behavior* within its own community area. While in *Roaming* state, agents can move also in Wander state within the expanded space constraints of the whole simulation area, or they can simply move according to a RWP model. Furthermore, dynamic events can be defined to capture the semantics of agents transitioning between their two states. Every time such a dynamic event is triggered, an associated Lua script will be executed causing agents to change their Steering behaviors and target selection mechanisms.

VII. TRACE-TO-MODEL MAPPING

The ultimate goal behind mobility modeling research is devising models that capture the fundamental characteristics for different application domain scenarios. There are two main approaches to achieve this goal. The focus of this work has been on the *Model-to-Trace* approach, designing and implementing a modeling framework, which enables the definition and implementation of representative mobility models, from which mobility traces can be obtained. Alternatively, a *Trace-to-Model* seeks to derive mobility models from *measured* mobility traces from different application domain scenarios.

The *Trace-to-Model* approach is a very important area of research, as it seeks understanding and insights into the mobility behavior of actual mobile agents in a given scenario. While progress as been made [13], the scope of this approach is limited nowadays by the fact that there are not enough mobility traces available, which are required to develop and test techniques to extract mobility properties from them. While the availability of mobility traces in specific scenarios has been increasing in recent times [1], [7], they are still very limited in scope and number. Therefore, a framework such as UMMF can play an important role in providing extensive synthetic mobility traces associated with a wide variety of scenarios, which can be used to develop the techniques that would eventually enable the derivation of mobility models from real mobility traces.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper we have presented the design and implementation of a *Universal Mobility Modeling Framework (UMMF)*, which is based on the observation that most mobility scenarios, simple or complex, can be effectively decomposed into a sound set of *mobility building blocks*, namely, target selection, obstacles, dynamic events, navigation graphs, steering behaviors, and dynamic behaviors. UMMF implements support for the aforementioned set of mobility building blocks, and the mechanisms to combine them, thus enabling researchers to flexibly define complex mobility models for the study of mobility-sensitive algorithms and protocols. We demonstrated the capabilities of UMMF by describing the mapping of a variety of existing models and scenarios into the framework.

Many fertile areas of research are enabled by UMMF. We emphasize the following important future research avenues: (a) expanding the set of geographical data associated with environment cells (e.g. contour maps, meteorological conditions); (b) inclusion of realistic signal propagation models capturing interactions with terrains and obstacles; and (c) expansion of UMMF interfaces for the generation of richer mobility traces, simulation environments, and game/virtual environments; (d) automatic parameter generation, and parameter range calibration; (e) expanding UMMF interfaces to simulation environments such as *NS* and *OPNET*; (f) integration of UMMF with game and virtual environments, in order to combine the mobility realism of UMMF with the realism offered by these environments; and finally, (g) pursuing UMMF-enabled basic research in the context of dynamic topologies, such as the searching for invariant topological characteristics, as well as answering a plethora of research questions in the context of performance and correctness of mobility-sensitive protocols and algorithms.

REFERENCES

[1] U. Amherst. Dieselnet. http://prisms.cs.umass.edu/dome/umassdieselnet.

[2] F. Bai and A. Helmy. Chapter 1 a survey of mobility models in wireless adhoc networks.

[3] F. Bai, N. Sadagopan, and A. Helmy. Important: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks, 2003.

[4] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. *in the proceedings of MobiCom*, pages 85–97, 1998.

[5] M. Buckland. *Programming Game AI by Example*. Wordware Publishing, 2005.

[6] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC) Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2:483–502, 2002.

[7] Dartmouth. Crawdad project. http://crawdad.cs.dartmouth.edu.

[8] M. C. Gonzalez, C. A. H. R., and A.-L. Barabási. Understanding individual human mobility patterns. *CoRR*, abs/0806.1256, 2008.

[9] X. Hong, T. Kwon, M. Gerla, D. Gu, and G. Pei. A group mobility model for ad hoc networks. *Proceedings of MSWiM*, 1999.

[10] R. Ierusalimschy. *Programming in Lua*. Lua.org, 2006.

[11] O. Inc. Opnet simulator. http://www.opnet.com/.

[12] A. P. Jardosh, E. M. Belding-royer, K. C. Almeroth, and S. Suri. Real-world environment models for mobile network evaluation. *IEEE Journal on Selected Areas in Communications*, 23:622–632, 2005.

[13] M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. In *INFOCOM*, 2006.

[14] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong. Slaw: A new mobility model for human walks. In *INFOCOM*, pages 855–863, 2009.

[15] F. Legendre, V. Borrel, M. de Amorin, and S. Fdida. Modeling mobility with behavioral rules: The case of incident and emergency situations. *AINTEC 2006, LNCS*, pages 186–205, 2006.

[16] M. Musolesi, S. Hailes, and C. Mascolo. An ad hoc mobility model founded on social network theory. In *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 20–24, New York, NY, USA, 2004. ACM.

[17] M. Musolesi and C. Mascolo. A community based mobility model for ad hoc network research. In *REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 31–38, New York, NY, USA, 2006. ACM.

[18] S. C. Nelson, A. F. Harris, III, and R. Kravets. Event-driven, role-based mobility in disaster recovery networks. In *CHANTS '07: Proceedings of the second ACM workshop on Challenged networks*, pages 27–34, New York, NY, USA, 2007. ACM.

[19] NS. Ns simulator. http://www.isi.edu/nsnam/ns/.

[20] A. S. Pentland, R. Fletcher, and A. Hasson. Daknet: Rethinking connectivity in developing nations. *IEEE Computer*, 37(1):78–83, Jan. 2004.

[21] A. Petz, J. Enderle, and C. Julien. A framework for evaluating dtn mobility models. In *SCENES Conference Proceedings*, 2009.

[22] W. Zhao and M. H. Ammar. Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks. In *FTDCS '03: Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems*, page 308, Washington, DC, USA, 2003. IEEE Computer Society.