

携帯端末の現実的な移動モデルを表現可能な ネットワークシミュレータの設計と実装

前田 久美子[†] 佐藤 和基[†] 小西 一樹[†] 内山 彰[†] 廣森 聡仁[†]
山口弘純[†] 安本 慶一[‡] 東野 輝夫[†]

[†] 大阪大学大学院情報科学研究科

[‡] 奈良先端科学技術大学院大学情報科学研究科

本研究では、モバイルネットワークアプリケーションの現実的評価を可能とするネットワークシミュレータ MobiREAL の設計と実装について述べる。MobiREAL シミュレータでは、移動端末を保持する人物などの行動を状態遷移モデルで記述し実行することで、移動端末の現実的行動のシミュレーションを可能とする。MobiREAL シミュレータは移動端末の行動をシミュレートする行動シミュレータ部、行動シミュレータ部と協調してモバイルネットワークをシミュレートするネットワークシミュレータ部、移動端末の行動を視覚的に表示する GUI 部からなる。このシミュレータにより、例えば移動端末の集中により移動速度が落ち、端末密度に偏りが生じるような状況が再現でき、それがデータ配信性能にどの程度の影響を与えるか、など、様々な現実的状況におけるネットワークアプリケーションの正確な評価が可能となる。

Design and Implementation of a Network Simulator with Realistic Mobility Models of Mobile Nodes

Kumiko Maeda[†], Kazuki Satou[†], Kazuki Konishi[†], Akira Uchiyama[†], Akihito Hiromori[†],
Hirozumi Yamaguchi[†], Keiichi Yasumoto[‡] and Teruo Higashino[†]

[†] Graduate School of Info. Sci. & Tech., Osaka Univ., Japan

[‡] Graduate School of Info. Sci., Nara Institute of Sci. & Tech., Japan

In this paper, we describe the design and implementation of a new network simulator called MobiREAL. It allows us to describe practical mobility models and to evaluate mobile network applications under the given practical mobility. MobiREAL simulator is composed of a graphical animator, a behavior simulator and a network simulator. The graphical animator shows the behavior of mobile terminals visually based on simulation results. For describing realistic mobility models, we introduce a state transition model in the behavior simulator. The network simulator runs in cooperation with the behavior simulator. By MobiREAL simulator, for example, we can simulate how users of a navigation application change their behavior as a result of using the application.

1 まえがき

近年の計算機システムのユビキタス化や無線技術の普及に伴い、移動通信端末（以下、移動端末またはノード）が無線 LAN などを通じて通信するモバイルネットワークの研究が盛んに行われてきている。モバイルネットワークプロトコルの設計や実装の際には計算機によるネットワークシミュレーションが不可欠であるが、従来のネットワークシミュレータでは、移動端末の移動は完全なランダム移動やランダムな目的地選定に基づく移動などに限定しているものが多い。例えば、ns-2[1]では random mobility, random waypoint mobility [2] に従った動作のトレースを出力する独立し

たプログラムを提供している程度である。商用シミュレータ QualNet[3] は group mobility, random waypoint mobility, trace mobility に基づく動きのみを提供し、GloMoSim[4] でもそれらに加え random drunken mobility, ECRV mobility, reference point group mobility などのサポートにとどまっている。また、商用シミュレータ OPNET[5] は拡張モジュールとして提供する Wireless Module において、3次元空間における地上や衛星での移動ノードの動きをユーザ定義可能である。しかし、ネットワークアプリケーションから得られる情報や自身の環境情報（視覚情報など）により移動ノードが行動を変化させる状況のシミュレーションは考慮

されていない。

しかし、実際のアプリケーションにおいては、移動端末を保持する人間の行動は、アプリケーションから得られる情報や、視覚情報、心理状況などに左右される。例えば、イベント会場の来場者が携帯端末を保持しているもとの、それら携帯端末間でのマルチホップアドホックネットワークを介して、ナビゲーション情報を配信するアプリケーションの性能評価を行う場合を考える。そのアプリケーションでのナビゲーション情報（例えば時間限定のイベントの情報）に従い、移動端末がそのイベントの開催場所に集中したとすればそれに伴いネットワークトポロジも変化し、端末密度が不均一なネットワークとなる可能性もある。その結果、端末密度がおおよそ均一なランダム移動を仮定して設計されたアプリケーションでは、端末密度が疎である場所への情報到達率が減少したり、逆に端末密度が密である場所では不必要なパケットが生じるなどの問題が発生する可能性もあるが、既存のシミュレータではこのような問題を発見することは容易ではない。

そこで本研究では、モバイルネットワークアプリケーションのより現実的かつ詳細な評価を可能とするために、各移動端末が、アプリケーションや自分の周辺環境から得られる情報を元にその行動を変化させる様子を記述可能であり、その記述に基づくシミュレートが可能なネットワークシミュレータ MobiREAL の設計を行う。

MobiREAL シミュレータは、移動端末の位置情報、出現及び消滅に関する情報、及びアプリケーションに対するユーザ入力など、移動端末を保持する人間の行動に関する部分をシミュレートする行動シミュレータ、行動シミュレータとインタラクティブに動作し、ネットワーク及びアプリケーションをシミュレートするネットワークシミュレータ、行動シミュレータとネットワークシミュレータが出力したシミュレーション結果を視覚的に再現する GUI の 3 つのコンポーネントにより構成される。

MobiREAL シミュレータにより、以下のような状況を表現可能となる。例えば前述のイベントナビゲーションアプリケーションにおいて、人気のあるイベント会場の前に人が集中することで移動速度が落ちる状況や、互いの衝突回避により人が自然に左側通行で対面通行する状況、また、アプリケーションからの情報により、特定のイベント会場

に多数の人が移動し、人の分布に偏りがでる状況などが表現できる。それらはネットワークトポロジやその流動性に影響を与えるため、ルーティングプロトコルの堅牢性、再送制御の妥当性など様々なメトリックを実際のアプリケーションに即して評価できる。その結果に基づき、アプリケーションにおける情報再送のタイミングや情報送信間隔などのパラメータの設定及びアプリケーションの再設計を行うことができる。

本稿では、まず 2 章で MobiREAL シミュレータの全体構成について説明する。3 章では行動シミュレータ、4 章ではネットワークシミュレータ、5 章では GUI についてそれぞれ述べる。

2 全体設計

MobiREAL シミュレータはネットワークシミュレータ部、行動シミュレータ部、GUI 部の 3 つのコンポーネントにより構成される (図 1)。

MobiREAL シミュレータを利用するアプリケーション開発者は、行動シミュレータに対し、各移動端末の行動を状態遷移図で指定する（これを行動モデルと呼ぶ）。各状態遷移の遷移条件として、シミュレーション評価対象とするネットワークアプリケーションから得られる情報、自身の端末の周辺に存在する障害物や他の移動端末などの情報（外部情報）に関する条件が指定できる。また、各遷移図はいくつかの内部パラメータ（経由地や目的地、特定の地点に対する興味度など）を保持し、その内部パラメータも遷移条件に利用できる。また、遷移の実行によりそれらのパラメータ値を変更することもできる。この遷移図に基づき、行動シミュレータは各時間ごとの移動端末の位置と速度ベクトルを計算する。また、障害物情報を含むシミュレーションフィールドの情報（地図情報）と、移動端末の初期配置なども指定する。これらの情報はネットワークシミュレータにも渡される。

なお、多数の移動端末に対し、内部パラメータを移動端末ごとに指定することは非効率的であるため、これらの内部パラメータを（一定の分布に基づき）ランダムに生成して複数の移動端末に対し一括指定することもできる（これを行動シナリオと呼ぶ）。

また、アプリケーション開発者は、通常のネットワークシミュレータ同様、評価対象とするアプリケーションをネットワークシミュレータ上に実装

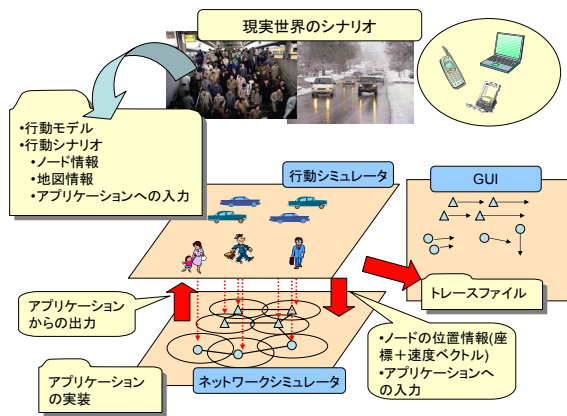


図 1: MobiREAL シミュレータ全体設計図

する。

このもとで、ネットワークシミュレータと行動シミュレータはそれぞれ独立にシミュレーション時間 t 毎のシミュレーションを行い、そのシミュレーション結果を交換した後、次の t シミュレーション時間のシミュレーションをそれぞれ独立に行う。これを繰り返しながらシミュレーションを進行させる。これについては、次節で詳しく述べる。シミュレーション結果はトレースファイルとして出力され、GUI に渡すことで視覚的にシミュレーション結果を再現することができる。GUI では、行動シミュレータから渡されたトレースファイルを入力として、移動端末の移動や情報伝達の様子を可視化することができる。GUI では、拡大・縮小、ステップ実行などにより、シミュレート結果をより詳細に検討することができる。

2.1 行動シミュレータとネットワークシミュレータの相互連携

行動シミュレータとネットワークシミュレータは連携しながら相互に情報交換を行い、シミュレーションを実行する (図 2)。情報交換は任意に指定可能なシミュレーション時間 t ごとに実行される。

具体的には以下のように実行される。シミュレーション開始時には、障害物の配置とともに、シミュレーション時刻 0 における各ノードの座標と速度ベクトル、アプリケーションへの入力が行動シミュレータからネットワークシミュレータへ渡される。その後、各シミュレーション時刻 $n * t$ ($n = 0, 1, \dots$) において、行動シミュレータは各移動端末ごと、直前の t シミュレーション時間 $[(n-1) * t, n * t]$ の行動シミュレーション中において得られた、(a) ネット

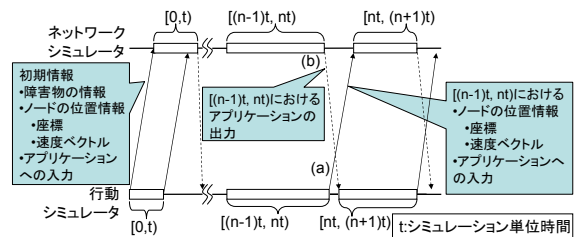


図 2: 行動シミュレータとネットワークシミュレータの相互連携

ワークアプリケーションへの入力 (ユーザ入力), (b) 時刻 $n * t$ における座標及び速度ベクトル, をネットワークシミュレータに渡す (図 2 の (a) の矢印). 同様に, シミュレーション時刻 $n * t$ において, ネットワークシミュレータは移動端末ごとに, $[(n-1) * t, n * t]$ のネットワークシミュレーション中において得られたネットワークアプリケーションからの出力を行動シミュレータに渡す (図 2 の (b) の矢印). このように, 行動シミュレータとネットワークシミュレータは互いに独立にシミュレーション時間 $[(n-1) * t, n * t]$ のシミュレーションを行い, その結果をシミュレーション時刻 $n * t$ に交換し, それを用いてシミュレーション時間 $[n * t, (n+1) * t]$ のシミュレーションを行う。

なお, シミュレーション時間 t の間は, ネットワークシミュレータは受け取った速度ベクトルを用いて行動シミュレータとは独立にノードの位置計算を行う。ただし, この間に行動シミュレータでノードの速度ベクトルを変更している可能性もあるため, t 時間毎の情報交換において, 正確な現在位置座標を行動シミュレータから適宜受け取り位置を補正する。 t の大きさの決定は, アプリケーションに求められるシミュレーション精度とシミュレーションに要する時間とのトレードオフに基づきシミュレータ使用者が設定する。

3 行動シミュレータ

行動シミュレータでは, 現実の行動を記述するための言語の提案とその処理系の開発を目指す。行動シミュレータに対し, 移動端末の行動は行動シナリオと行動モデルにより与えられる。行動シナリオは個々のノードの出現確率や目的地の決定などを一括して指定する。また, 行動モデルは個々の移動端末の行動を記述した状態遷移図 (変数付きの有限状態機械) である。

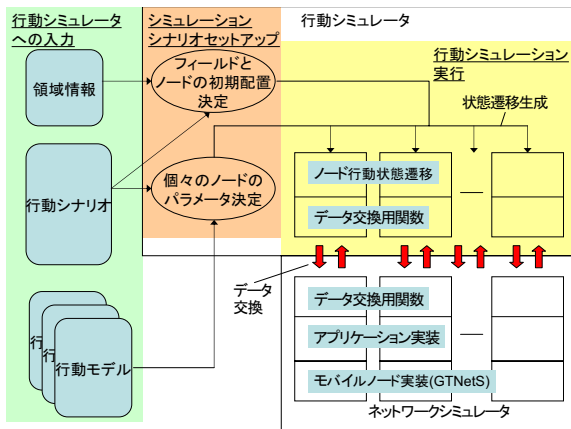


図 3: 行動シミュレータ全体図

3.1 行動シナリオ

歩行者全体を考えた場合、どのような状況においてもある一定の歩行流(多数の歩行者の流れ)が発生することが多い。例えば駅の改札口では、電車の到着前には改札口へ向かう歩行者が多いため、改札口へ向かう歩行流が発生するが、電車の到着後には改札口を出る歩行者が多いため、改札口から離れる歩行流が発生する。また商業空間では店によって客の出入りする人数は異なるため、店によって付近の歩行流も異なる。このような歩行流を制御するためのパラメータを行動シナリオとして与えることで、マクロの視点においても現実に近いシミュレーションを行うことができる。図3に行動シミュレータの全体図を示す。図3の行動シナリオとして、ノードの出現確率、どの目的地にどの程度の割合のノードが向かうかを表す値、個人の行動に影響を与える情報活用度や興味度といったパラメータの基本値などを与える。これらの値を基本に、以下で述べる行動モデルの内部パラメータを決定する。

3.2 行動モデル

行動モデルは、アプリケーションとの入出力インタフェース、視覚情報の入力インタフェースを入出力ポートとし、目的地リストなどを内部変数として保持する拡張有限状態機械である。シミュレータ使用者は、いくつかの拡張有限状態機械と、それにしたがって行動するノードの集合を指定する。これと行動シナリオで指定された基本値を基に、内部変数の初期値が与えられたインスタンスがノードごとに生成される。

図4に歩行者の典型的行動を表す行動モデルを

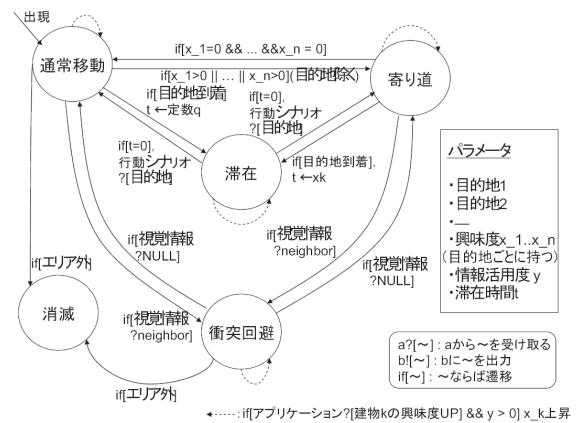


図 4: 典型的な歩行者の行動モデルの例

示す。一般に、人が行動するにはある目的地へ向かって移動する。そこで、その目的地を優先順位付きのパラメータとして持たせる。この際、ネットワークアプリケーションからの情報がある場合には、その内容に従って優先順位を変更したり、新規に目的地を持たせたりすることにより、行動を変化させる。例えば、イベントの宣伝をネットワークアプリケーションにより行いたい場合には、ある端末(例えば固定端末)がアプリケーションへの入力として、イベントの開催時刻、開催場所を与える。アプリケーションを介して、この情報を受け取った他のノードは情報を活用するかどうかを情報活用度に応じて決定し、活用するならばこの開催時刻に合わせて開催場所へと向かうようになる。

また、目的地として交差点などの経由地点を指定することにより、曲がり角などを含む複雑な経路を通る歩行を実現することができる。

3.3 シミュレーション領域

シミュレーション領域は図3の領域情報により生成する。領域は長方形を想定し、各辺の長さ[m]を指定する。建造物などのオブジェクトは座標集合で指定する。例えば、長方形のオブジェクトの場合には4点の座標を指定し、円形の場合には多角形に抽象化し、その点集合を指定する。オブジェクトには単なる障害物、進入可能な建物などの種類も指定する。

3.4 現実的な行動モデルのためのライブラリ群

MobiREALでは、より現実的な行動を容易に記述するためにグループ行動や、衝突回避処理を行うライブラリも設計している。

3.4.1 グループ行動

街中では、人は1人で歩いているだけでなく、友人等と一緒に歩いている。そこで、グループとしての挙動を表現できるように、グループ行動の要素を導入しライブラリとして利用できる。グループ行動を実現するために、文献 [2] で紹介されている、グループ行動における reference point を導入する。reference point 自体は、個々のノードのように移動するが、その reference point を持つグループ中のノードが、reference point の動きに従って行動することで、グループの行動を統一化できる。このグループ行動のライブラリは例えば、図 4 の状態”通常移動”、”寄り道”において速度ベクトルを決定する際に用いることができる。

3.4.2 衝突回避

現実には人と人がすれ違うときにはぶつからないように歩くのが自然である。その行動を行動シミュレータで実現するために文献 [6] で紹介されているような衝突回避処理を導入する。衝突を避ける対象者が図 4 中の neighbor に相当する。これは自分の移動ベクトルから決定した視野範囲内に存在するノードである。この衝突回避処理では、自分と neighbor の位置、速度ベクトルと人体円半径などによって、そのままの速度ベクトルで進んだときに衝突する領域を求め、その領域には進まないように、速度ベクトルを変化させることができる。また壁などの障害物に対しても、障害物を動かないノードの集合とみなすことで同様の処理を用いて衝突を回避することができる。この衝突回避のライブラリは図 4 の状態”衝突回避”において速度ベクトルを決定する際に用いることができる。

4 ネットワークシミュレータ

MobiREAL におけるネットワークシミュレータでは、行動モデルシミュレーションによる人々の動きに従い、ノードが持つ通信端末により形成されるネットワークトポロジを変化させ、また、そのもとで端末間のネットワーク通信を実現する。以下、使用するネットワークシミュレータの概要、ネットワークシミュレータと行動シミュレータの連携について詳しく説明する。

4.1 GTNetS

MobiREAL では、ネットワークシミュレータとして、米 Georgia Institute of Technology で開発

された GTNetS[7] を使用する。GTNetS は、既存のネットワークシミュレータのスケラビリティにおける問題点を克服し、より高速に大規模なネットワークのシミュレーションを行うという思想で設計されており、C++ による実装、並列実行等の特長を持つ。

GTNetS は、ネットワークシミュレーションを構成する各部品を C++ 言語により実装し、ライブラリとして提供している。これらのライブラリを利用しシミュレーションシナリオを記述することで、GTNetS が提供するネットワークシミュレーション環境を利用することができる。また、必要に応じてクラスを作成することで、シミュレータの拡張を容易に行うことができる。

4.2 GTNetS を利用したシミュレータ部の実現

MobiREAL では、行動シミュレータとのインタラクション部分を独自のクラスとして実装し GTNetS に組み込むことで、行動シミュレータとネットワークシミュレータの連携を実現する。また、同様に、ノードの移動等を扱うクラスも実装して GTNetS に組み込み、行動シミュレータから受けとるノードの速度情報にもとづきノードの位置を更新する。

インタラクション部分では、行動シミュレータとのデータ交換、及び入力データに対する処理を行う。行動シミュレータからインタラクション部分への入力、ノードの発生、消滅、移動等ノードに関する入力と、アプリケーションへの入力の 2 種類に分けられる。アプリケーションへの入力は、そのまま実装されているネットワークアプリケーションへの入力となる。以下、インタラクション部分の実装、入力処理について説明する。

4.2.1 行動シミュレータとのインタラクション

GTNetS は discrete-event タイプのネットワークシミュレータとして実装されている。パケットの送受信等のシミュレーションイベントは、その実行時刻と共にイベントキューに挿入される。その後、イベントキューに挿入されたイベントは、シミュレーション時間順に取り出され処理される。これらイベントキューへのイベント挿入、処理の繰り返しによりシミュレーションが構成される。MobiREAL では、行動シミュレータとのデータ交換と、行動シミュレータから渡されるノード情報、アプリケーション情報に対する処理を行うために、インタラク

ションイベントを作成し、シミュレーション時間 t ごとに処理する。

まずシミュレーション開始前に、インタラクシオンイベントを時刻 t にスケジューリングする。このスケジューリングにより、 $[0, t)$ のシミュレーションを行った後、時刻 t でインタラクシオンイベントが処理され、行動シミュレータとのデータの受け渡し、受け取ったデータに対する処理が行われる。インタラクシオンイベントの最後には、次のインタラクシオンイベントを時刻 $2t$ にスケジューリングし、次の行動シミュレータとのデータ受け渡しに備える。このような一連の処理を、行動シミュレータによってシミュレーション終了の情報が渡されるまで繰り返す。

4.2.2 ノードの発生、消滅に対する拡張

GTNetS は、ネットワークトポロジの変化に乏しい有線ネットワークを主な対象としているため、ネットワーク通信を行うノードを、シミュレーション開始前に全て生成しておき、シミュレーション中に新規ノードの追加、既存ノードの削除は行わない。一方、無線ネットワークにおいては、ノードの移動だけでなく、ノードの発生や消滅も頻繁に発生するため、これらのノードの変化に対し GTNetS で適切に対処する必要がある。現時点では、あらかじめノードを十分な数だけ生成しておき、行動シミュレータのノードとネットワークシミュレータのノードとのマッピングを行うノード管理クラスを導入し、行動シミュレータからのノード追加、削除情報に対して、あらかじめ生成しておいたノードをネットワークトポロジに追加、削除することで、ノードの発生、消滅に対処している。しかし、この方法では、ネットワークシミュレータ上に多数の未使用ノードが存在することになり、計算機資源を余分に消費してしまう欠点があるため、上記の解決法に代わる根本的な対策を行う予定である。

4.2.3 ノードの移動に対する拡張

GTNetS ではノード移動を計算、管理するモビリティクラスが用意されており、各ノードの移動は、ノードに割り当てられたモビリティクラスのインスタンスに依存する。

MobiREAL では、行動シミュレータから受け取ったノードの移動情報（速度ベクトル）をネットワークシミュレータに反映させるため、速度と変化時刻の組を蓄積出来るようなモビリティクラスを作成

し、GTNetS に組み込んでいる。行動シミュレータからノードの速度ベクトルを受け取った時点では、ノード位置を更新せずに、蓄積情報の更新を行う。無線計算等でノードの現在位置が必要になった場合のみ、ノード位置と速度、蓄積情報によって位置計算が行われる。ただし、このように速度ベクトルのみを受け取り行動シミュレータと連携する方法では、ベクトルの精度や、位置計算が原因で誤差が生じる可能性がある。そこで、定期的にノードの位置情報を受け取り、誤差を修正する。位置情報を受け取った場合は、過去に速度ベクトルが必要なくなるため、蓄積情報の初期化を行う。

4.3 今後の課題

より現実世界に近いシミュレーションを行うためには、行動シミュレータから渡されるフィールド情報を有効利用すべきであると考えられる。例えば、無線電波を通さない、または弱めるような建物の存在を考慮した場合のネットワークシミュレーションと、建物の存在を考慮しない場合のシミュレーションでは、アプリケーションによる情報の伝達状況は大きく異なる可能性がある。詳細なシミュレーションの実現のためには、GTNetS の物理層、データリンク層、MAC 層に該当する部分の修正、拡張、及び、無線計算時に用いる無線伝播モデルの拡張を行い、現実世界と同じように無線伝播を計算する必要がある。

また、スケーラビリティも検討すべき項目の一つである。スケーラビリティ向上の手段として、ネットワークシミュレーションの並列化が考えられる。GTNetS の並列化機能では、シミュレーションするノード群を複数の計算機で分担することで、計算機一台あたりの仕事量を減らし、実行時間の減少を実現している。しかしながら、現時点では無線環境のシミュレーションの並列化は行えないため、該当部分の拡張を行う必要がある。

他には、ネットワークシミュレーションの精度を下げることで、計算時間を短縮するという方法が考えられる。具体的には、位置の近い複数ノードを単一のノードとして扱うノードの抽象化や、複雑な無線伝播計算を簡略化するといった無線計算の抽象化が挙げられる。これらを実現するために、効率的な抽象度の決定及びシミュレータでの実装方法の調査が現在の課題である。

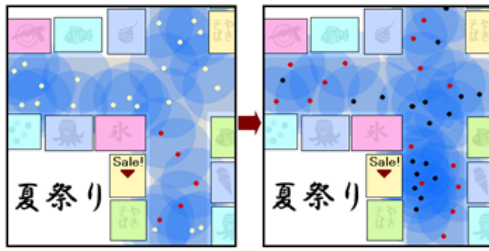


図 5: GUI のイメージ画像

5 GUI

GUI では携帯端末の動きやネットワークの可視化を行い、わかりやすい形でシミュレーション結果を提示し、その理解を視覚的に手助けすることを目的としている。

5.1 GUI の機能

Windows 上で動作する MobiREAL の GUI には処理速度、描画能力の観点から DirectX9 を採用している。現時点での GUI は俯瞰視点での人の行動の可視化を実装しており、シミュレーションフィールドを表す地図画像の上で、人を表す小さい円と、その人が持つ無線端末の電波が届く範囲を表す大きい円が、その人の行動に従って移動する (図 5)。地図画像の上には、ビルなどの電波の伝達を妨害する障害物を表示することができ、無線伝達範囲を表す大きい円とあわせて、おおまかにではあるが各ノード間における通信の可否を確認することができる。

GUI は行動シミュレータとネットワークシミュレータから出力されたシミュレーション結果トレースファイルを入力とし、またそのファイルで地図画像に任意のビットマップ画像を指定することが可能である。GUI には拡大、縮小機能 (1cm/pixel ~ 100m/pixel)、再生速度の変更機能 (実時間比で 0~256 倍)、ステップ実行機能など一通りの機能が備わっている。

5.2 今後の展開

GUI ではノードの状況の違いを一目で見分けられるよう、人を表す円を色分けする機能を実装中である。例えばネットワークアプリケーションでブロードキャスト中のある特定の情報に着目し、その情報が伝わっているか否かで色分けを行えば、情報が伝達していく様子を視覚的に把握することが可能である。またその情報が伝わる事により行動が変化した人を色分けすることで、その情報の伝

達が人の行動にどう影響を及ぼすかを見ることもできる。

また MobiREAL では将来的に何千、何万ノードレベルでの大規模なシミュレーションを目標としており、GUI 上で相当数のノードを同時に表示しても見やすく、わかりやすいようにするため、GUI における表現上の抽象化も検討している。

これらに加え、ユーザからの入力、例えば地図情報やシナリオの入力、ノードの初期配置の入力に対し、GUI による補助を行うことも検討している。

6 まとめ

本稿では現実世界の人物行動などを記述でき、それに基づいたモバイルアドホックネットワークアプリケーションの評価を可能とするネットワークシミュレータ MobiREAL の設計について述べた。MobiREAL シミュレータは人物行動に関するシミュレートを行う行動シミュレータと、ネットワークに関するシミュレートを行うネットワークシミュレータがインタラクティブに動作することにより、ネットワークアプリケーションによる人物の行動への影響、及びその結果生じるネットワークトポロジの変化を再現でき、より現実的なシミュレートを行うことが可能となっている。今後は引き続き MobiREAL シミュレータの開発を進め、その実用性の評価を行っていく予定である。

参考文献

- [1] ns-2, <http://www.isi.edu/nsnam/>
- [2] Tracy, T., Jeff, B. and Vanessa D. : "A Survey of Mobility Models for Ad Hoc Network Research", *Wireless Comm. & Mobile Computing(WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends, and Applications*, vol. 2, no. 5, pp. 483 - 502 (2002).
- [3] QualNet, <http://www.scalable-networks.com/>
- [4] Zeng, X., Bagrodia, R. and Gerla M. : "GloMoSim: A Library for the Parallel Simulation of Large-scale Wireless Networks", *Proc. of ACM Parallel and Distributed Simulation (PADS'98)*, pp. 154 - 161 (1998).
- [5] OPNET, <http://www.opnet.com/>
- [6] 岡田公孝, 和田剛, 高橋幸雄 : "個人行動をベースにした歩行モデルと歩行流シミュレーション", 日本オペレーションズ・リサーチ学会, 春季研究発表会 (2003).
- [7] Riley, G. F.: "The Georgia Tech Network Simulator", *Proc. of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research*, pp. 5 - 12 (2003).