Doktoranden-Kolloquium

25. Januar 2012

Evaluierung von Bedrohungen und Schutzmechanismen für die Sicherheit und Privatsphäre mobiler Geräte unter Zuhilfenahme von Simulation

M. Sc. Benjamin Henne

DC Sec

TU Clausthal

TECHNISCHE UNIVERSITÄT CAROLO-WILHELMINA ZU BRAUNSCHWEIG

Leibniz Universität Hannover

Problem Statement

- Neue Technologien
  - Smartphones für die Masse
  - Breitband-Internet für mobile Geräte
  - Gerät-zu-Gerät-Kommunikation
    - Bluetooth, Wi-Fi, NFC
- und neue Dienste
  - unzählige (ungeprüfte) Apps
  - Context- und Location-based Services (Qype, Latitude, Foursquare, …)
  - (mobile) Social Media/Networks (Facebook, Google+, …)
  - (mobile) Content-Sharing (Picasa, PicPlz, Instagram, …)

Problem Statement (2)

- Nutzer
  - Anzahl steigt stetig und rapide
  - technisch und in Puncto IT-Sicherheit nicht aufgeklärt
  - unvorsichtig und unbedacht mit privaten Daten

⇒ Vielzahl an alten neuen (alte Probleme in neuer Umgebung) und neuen „mobilen" Sicherheitsproblemen

! Bedrohung von IT-Sicherheit und digitaler Privatsphäre

Problem Statement (3)

- Lösungen und Schutzmaßnahmen
  - ☑ in der Forschung
  - ☐ Maßnahmen verständlich für Endnutzer (Usable Security) – nein
  - ☐ in der Praxis umgesetzt – nein

- Sowohl in der Forschung (Sicherheit sowie Nutzbarkeit)
  als auch in der Umsetzung existieren viele Lücken und Baustellen

- IT-Sicherheit und Erhaltung der digitalen Privatsphäre
  muss gewährleistet und nutzbar sein

Mission und Ziele

- Untersuchung von Bedrohungen der Sicherheit und Privatsphäre auf modernen mobilen Geräten mit Breitband-Internet
  - Lokalisieren von Bedrohungen
  - Analysieren von Bedrohungen
    - Ver-/Ausbreitung (Malware)
  - Evaluieren von Lösungen
    - Malware-Gegenmaßnahmen prüfen
    - Algorithmen mit Ortsbezug analysieren
      - Beispielsweise Location-Cloaking-Verfahren
  - Visualisieren / greifbar machen
    - Sachverhalte Laien verständlich machen

Probleme und Einschränkungen bisheriger Ansätze

- Echtwelt-Daten
  - anonymisierte Bewegungsprofile z. B. von Handy-Netzbetreibern
    - in USA möglich[1], in Deutschland (rechtlich) schwierig
    - Triangulation durch Funkmasten zu ungenau
    - bei Abrechnungsdatensätzen: Position nur bei Anruf/Kurznachricht
    - keine Interaktion (mit Auswirkungen) evaluierbar

[1] http://www.sciencemag.org/content/327/5968/1018.full

- Datenerhebung durch Feldstudien
  - Personen rekrutieren und Geräte beschaffen kostet viel Zeit und Geld
    - nur sehr kleine Studien möglich
  - benötigte sensitive Daten wollen Teilnehmer nicht erheben (Privatsphäre)
    - kommt einem Einbruch in die Privatsphäre gleich
      - Bewegungsprofile, Verbreitung privater Daten
  - benötigte Interaktion kann/soll von Teilnehmern nicht erprobt werden
    - (1) Verbreitung von mobiler Malware (2) in großem Stil

Probleme und Einschränkungen bisheriger Ansätze

- existierende (Netzwerk-, MANET-, Agenten-, …) Simulationen
  - Bewegung und Aktionen/Kommunikation getrennt
  - Interaktion eingeschränkt, keine Rückkopplung von Aktionen auf Bewegung
  - Geo-Modell meist nur Raster oder 2D-Fläche
  - wenn Straßenkarten, dann meist proprietäre/kostenpflichte Formate
  - fehlende Skalierbarkeit
  - nicht realitätsnah genug (Karte, POI, …)
- mathematische Modelle, digitale Epidemiologie
  - fehlende Parameter, Personen werden vereinheitlicht
    - z. B. Susceptible-Infectious-Modell: keine räumliche Ausdehnung
  - keine (komplexe) Kommunikation/Interaktion/Infektionsroutine möglich

Unser Ansatz

- Maßgeschneiderte Agenten-Simulation

  Mobile Security & Privacy Simulator (MoSP Sim)

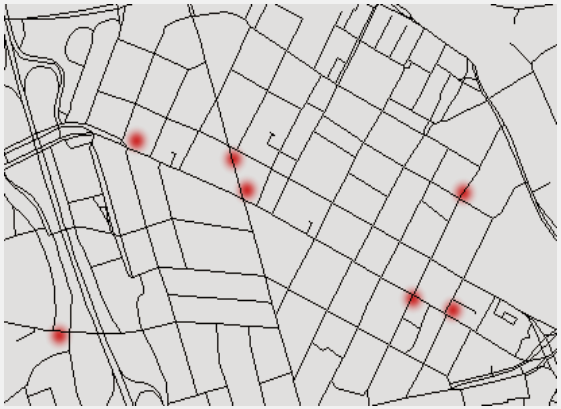
  - Untersuchung mobiler digitaler Privatsphäre & IT-Sicherheit
    - Modellierung von Bedrohungen
    - Modellierung von Lösungen
    - Analysieren ohne in der Realität Probleme zu schaffen (geltendes Recht)
      - Einbruch in die Privatsphäre
      - Malware in Umlauf bringen
  - Greifbarmachen von Problemen und Lösungen
    - Analyse
    - Visualisierung

Mobile Security & Privacy Simulator

- ✓ volle Kontrolle durch Definition von Simulationen in Python
  - Python ist verständlicher als jede XML-Konfiguration
  - MoSP ist Framework und Baukasten
    - einfach zu nutzen
  - basierend auf SimPy
    - „an object-oriented, process-based discrete-event simulation language for Python“
- ✓ Abstraktion
  - „Funkreichweite in Metern“ statt Modellierung von Funk-Signal und –Medium
  - bisher Akku-Verbrauch je nach Nutzung linear angenähert statt Kurve
  - bisher kein ISO/OSI-Stack integriert – abstrakte Kommunikation unter Agenten Kommunikation zu externen Entitäten über Client-Bibliotheken (z. B. HTTP)

Mobile Security & Privacy Simulator

- ✓ geforderte realitätsnahe Welt
  - Geo-Modell: OpenStreetMap-Karten
    - Straßen
    - POI als Ziele
      - » Automatische Anbindung an Straßennetz über MoSP Geo-Tool
    - Bewegung auf Flächen wie Parks *(Future Work)*
  - Positionen (Zenti-)/Dezimeter-genau
  - freie Bewegung in jede Richtung (kein Raster)



```
<node id='42'>
 <tag k='amenity' v='cafe'>
 <tag k='name' v='Notre Dame'>
 <tag k="addr:street"
      v="Limmerstraße"/>
</node>
```

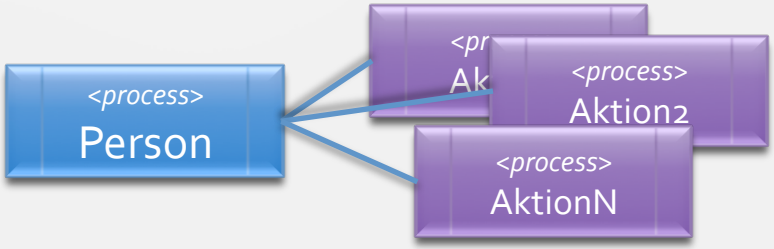
Mobile Security & Privacy Simulator

- ✓ Bewegung
  - auf dem Straßennetz
    - zufällige Bewegung (zur nächsten zufälligen Straßenecke)
    - geplante/geroutete Bewegung (über kürzesten Weg)
    - alternative nicht-kürzeste Routen *(Future Work)*
    - Anhalten
    - Straßen haben Dimension (laufen nebeneinander möglich)
      - » Verbesserung der Genauigkeit von „Wer ist in meiner Nähe?"
    - besondere Aktionen am Kartenrand
  - Verlassen von Straßen
    - Pausieren, beispielsweise an einem POI, wie im Park
    - Betreten von Gebäuden, wie einem Café
      - » Gebäude als einfaches Modell
      - » Gebäude als eigenständige Simulation *(in Bearbeitung)*
    - Laufen auf Flächen (OSM-Attribut *area=yes*) (Future Work)

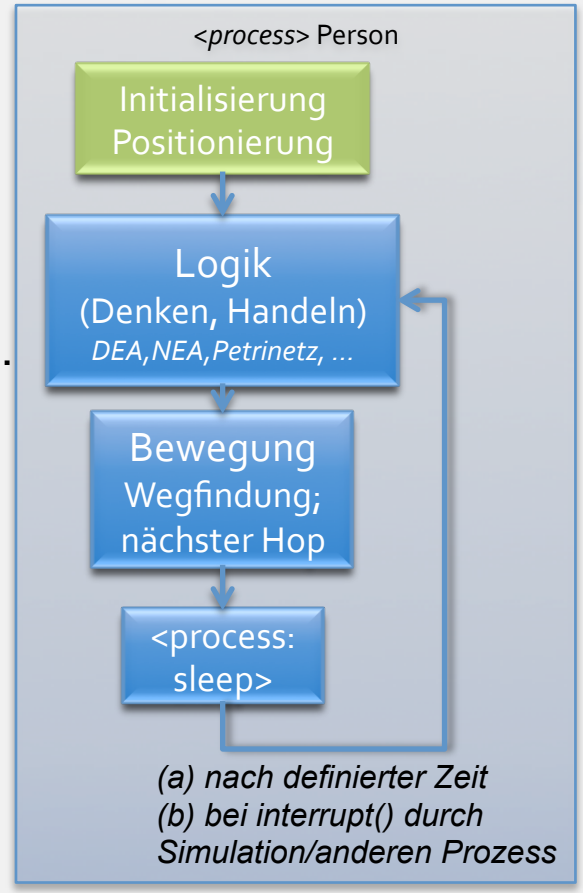
Mobile Security & Privacy Simulator

- ✓ Aktionen und Interaktion
  - i. in jeder Person-ProcessExecutionMethod-Iteration
    - Denken, Handeln
  - ii. in unabhängigen Aktion-Prozessen
    - eigenständige Aktion (Infekt/Telefon)
  - iii. Aufruf einer Person-Methode durch andere Person

ii.



i.

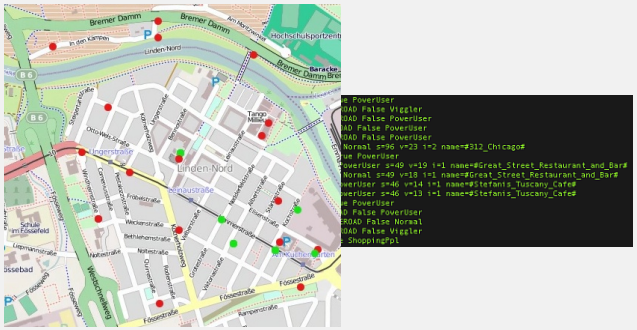


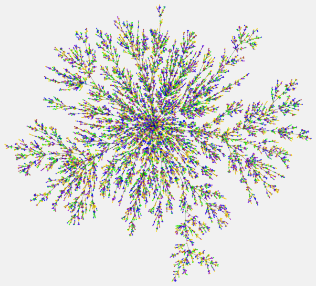
iii. `person7.get_near(1).call(delay=1).infect()`

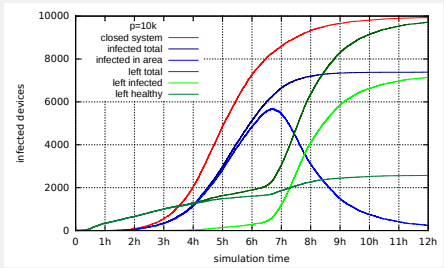
Person7 sucht Personen in der Nähe von 1m
und ruft deren infect()-Methode auf.

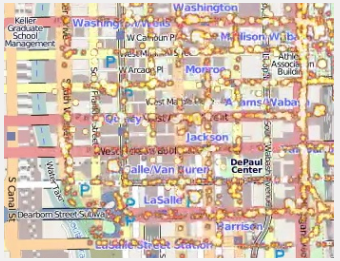
Mobile Security & Privacy Simulator

- ✓ Visualisierung/Analyse
  - Monitor-Framework für Ausgaben verschiedener Art
    - Log-Ausgaben
      - » Analyse
      - » Weiterverarbeitung (gnuplot etc.)
    - Live-Visualisierung
      - » OpenGL-unterstütze Live-Visualisierung von bis ca. 5000 Objekten (Personen, verschiedene geometrischer Primitiva)
    - Nachträgliche Visualisierung
      - » Bildgenerierung aus Log-Daten z. B. für Videoerstellung mit mehr Objekten Beispiel: Heatmap-Video später im Vortrag









Hands-On: einfache Zombie-Infektion

```
class SimpleZombie(Person):
    def __init__(self, *args, **kwargs):
        super(SimpleZombie, self).__init__(*args, **kwargs)
        self.p_infected = False
        if kwargs.get('infected'):
            self.infect()

    next_target = movement.person_next_target_random

    def infect(self):
        if self.p_infected == False:
            self.p_infected = True
            self.p_color_rgba = (1.0, 0.1, 0.1, 1.0)
            self.p_speed = self.p_speed / 2
            start_action(self.infect_other)

    @action(1, start=False)
    def infect_other(self):
        if self.p_infected == True:
            self.get_near(1).call(delay=1).infect()
```

initialisiere Person: infiziere, falls als infiziert gestartet

setze Bewegung auf zufällig

infiziere :=
markiere dich selbst als infiziert,
halbiere deine Geschwindigkeit,
(ändere Farbe in Visualisierung,)
starte Aktion „infiziere_andere"

Aktion (jeden Tick, kein Autostart)
falls infiziert, rufe infiziere für alle,
im Umkreis von einem Meter auf

```
s = Simulation(geo=osm.OSMModel('data/BS-HdNt.osm'), rel_speed=60)
m = s.add_monitor(SocketPlayerMonitor, 2)
s.add_persons(SimpleZombie, 49, monitor=m)
s.add_persons(SimpleZombie, 1, monitor=m, args={'infected':True, 'speed':0.7})
s.run(until=1000, real_time=True, monitor=True)
```

Simulation, Karte,
49 gesunde, 1 Zombie,
verbinde zur Visualisierung
simuliere 1000 Sekunden

Beispiel-Simulation Security: Mobile-Evil-Twin-Attack

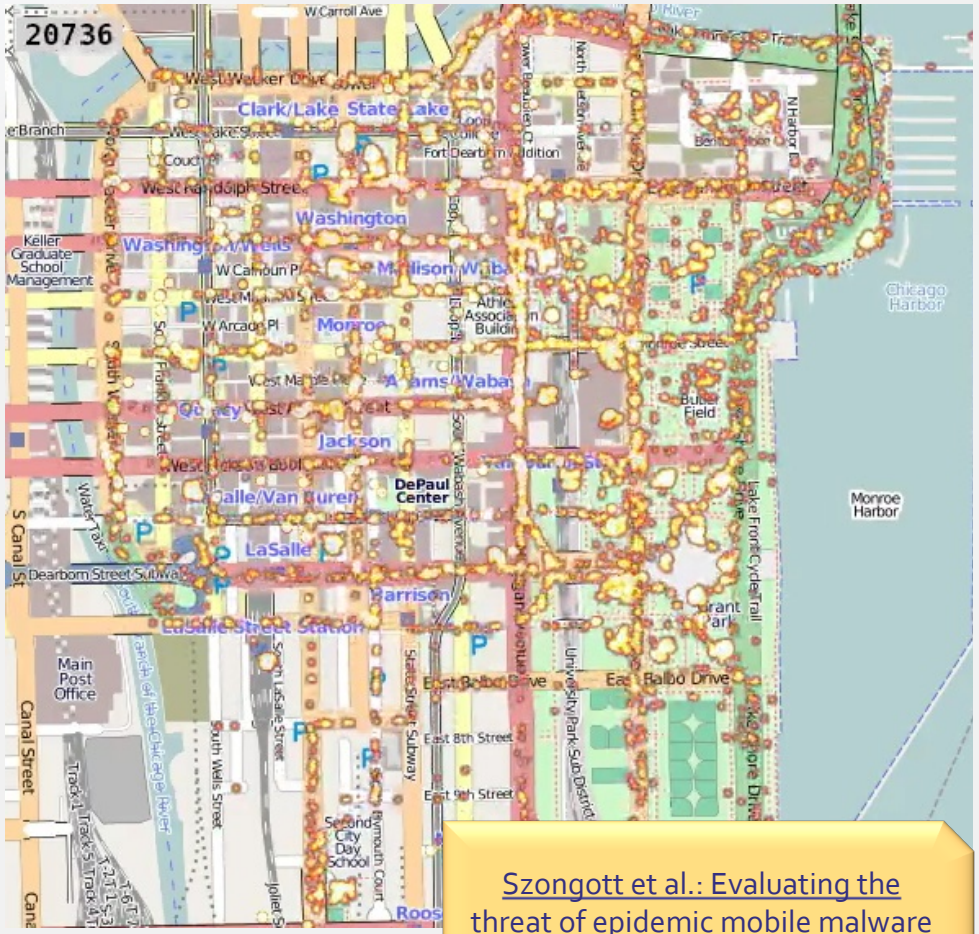
- WLAN-basierter iPhone-Exploit
  - Infektionsrisiko und Verbreitung von realem Exploit beurteilen, Location-based Countermeasures testen
    - einfache Agentensteuerung via Zustandsautomat mit zufälligen Übergängen (Zustände: Bummeln, Café besuchen, in den Park setzen, …)
    - verschiedene Benutzergruppen mit verschiedenem Verhalten
    - Bewegungsziele zufällig oder POI aus Geo-Daten
    - Aktionen an Zielpunkten: Pausieren, Kaffe trinken, …
    - Betreten von definierten Gebäuden (Cafés)
      - Im Gebäude: mathematisches Modell für Infektion
    - Auf der Straße räumliche und zeitliche Modellierung des Exploits
      - WLAN-Verbindung zu bösartigem Gerät, in Reichweite bleiben (15 Meter, 15 Sekunden), mit Infektion wird Client zum Wirt bis sein Akku leer

Szongott et al.: Evaluating the threat of epidemic mobile malware (WiMob 2012)

Beispiel-Simulation Security: Mobile-Evil-Twin-Attack (2)

- WLAN-basierter iPhone-Exploit
  - MET-Epidemie: *Heatmap der infektiösen Bereiche der simulierten Fläche*

nachträglich gerendertes Video



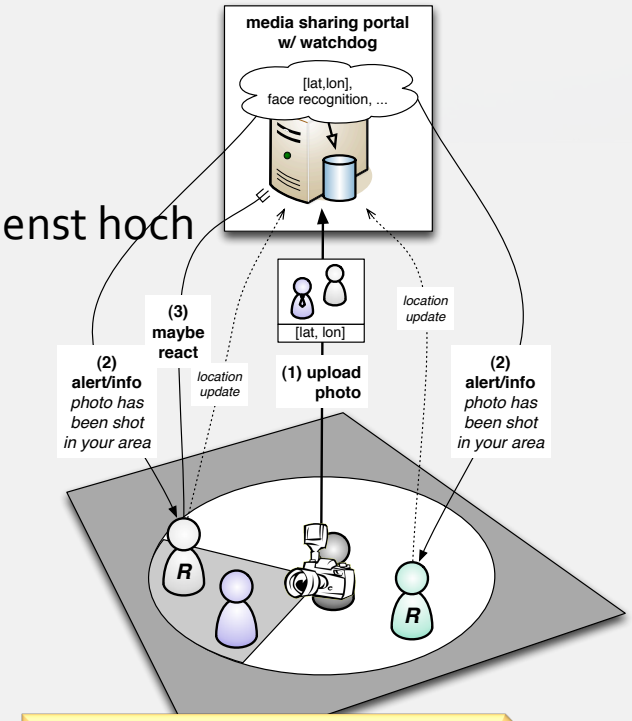
Szongott et al.: Evaluating the threat of epidemic mobile malware (WiMob 2012)

Beispiel Simulation Privacy: SnapMe Foto-Watchdog

- Evaluierung eines Privacy-Mechanismus in einem Foto-Sharing-Portal
  - Dienst-Effektivität und Anwendbarkeit prüfen

Interaktion von Simulation und Echtwelt-Dienst

1. Personen schießen Fotos
2. Personen laden Fotos mit Geo-Referenz zu Dienst hoch
   - Foto enthält Referenzen auf abgebildete Personen
3. Prototyp-Dienst auf einem LAMP-Server verarbeitet Foto
   - statische Koordinatenprüfung und LBS
4. Dienst benachrichtigt Personen falls potentiell betroffen



Henne et al.: SnapMe if you can: Privacy Threats of other Peoples' Geo-tagged Media and What We can do about it (WiSec2013)

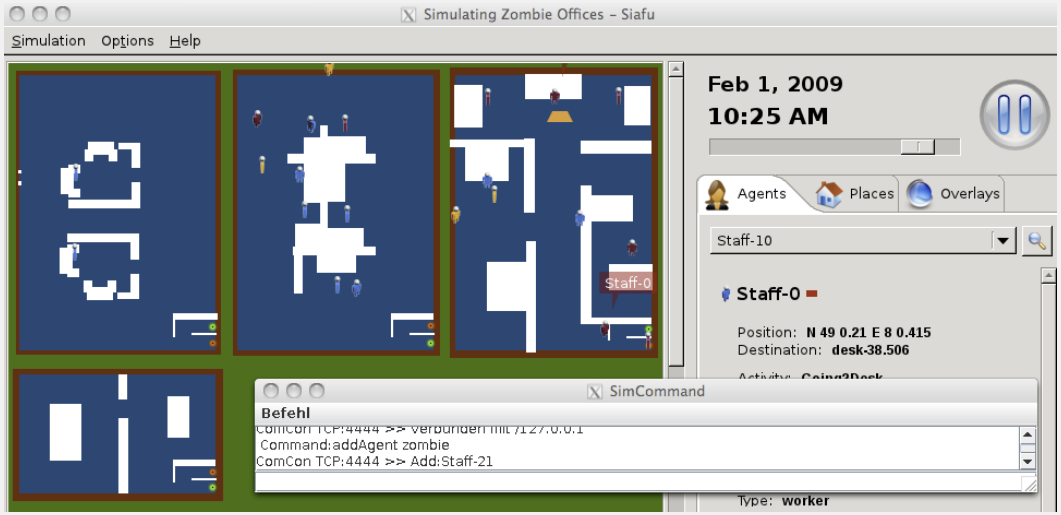
Verbindung mehrerer Simulationen

- Partitionieren der Straßenkarte mit Übergängen am Kartenrand *(später)*
- Simulation von Unterszenarien fern der Straße
  Simulation innerhalb von Gebäuden
  - bisher: mathematisches Modell
  - in Arbeit: eigenständige Simulationen

- Verbindung von Simulationen über Unix/Network-Sockets
  - Austausch von Agenten
  - Synchronisierung der Zeit-Schritte
  - Abwägung von Lastverteilung und Synchronisierungsoverhead
  - *Evaluierungen folgen*

MoSP Indoor-Simulation

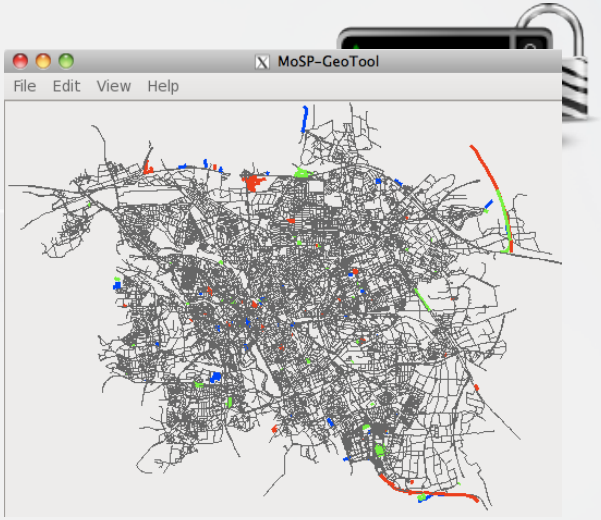
- eigenständige Simulation innerhalb von Gebäuden *(in Bearbeitung)*
- Anpassung des *Siafu Context Simulators*[1] (Java)
  - Stockwerke, Treppen
  - Meter statt Pixel, m/s statt Pixel pro Schritt
  - Float-Koordinaten; Beibehaltung 8er-Moore-Nachbarschaft für Navigation
  - Agenten hinzufügen und entlassen über TCP-Verbindung



Henne et al.: Coupled multi-agent simulations for mobile security & privacy research (DEST2012)

[1] http://siafusimulator.sourceforge.net

MoSP Geo-Tool



- Vorverarbeitung von OpenStreetMap-Karten
  - Qualität der OSM-Karten meist sehr gut
  - jedoch zu Teilen mit Fallstricken versehen
    1. Partitionierung des Karten-Graphen
       - keine Bewegung und kein Routing zwischen Teilgraphen
    2. POI in Karten verzeichnet, doch kein Zugang vom Straßennetz
    3. Kartengraph teils zu genau – nicht notwendige Komplexität des Graphen
       - nicht notwendige Speicherkomplexität bei vorberechnetem Routing
  - MoSP Geo-Tool
    1. Verbindet Partitionen, kann kleinste Partitionen eliminieren
    2. Verbindet ausgewählte POI mit dem Straßennetz
       - direkt oder intelligent (z. B. POI in Gebäuden über deren kartierte Eingänge)
    3. Generalisiert die Karte nach Nutzervorgaben

Future Work

- MoSP Simulator
  - Personenmodellierung verbessern; Baukastenteile dazu erstellen
  - alternative Wegfindung mit Orientierung am Menschen
  - Laufen auf offenen Plätzen
- MoSP Indoor-Simulator
  - Erste Version des Indoor-Simulator
  - Verbindung von Simulationen
- Security & Privacy Research
  - Folgearbeiten zur MET, Foto-Watchdog
  - andere Ansätze evaluieren mittels Simulator
  - Verbindung Simulation und reales mobiles Gerät
  - Usable Mobile Security, Metadaten schützen statt löschen, …

- Mobile Security & Privacy Simulator
  - https://bhenne.github.io/MoSP/ (docs)
  - https://github.com/bhenne/MoSP (code)

by B. Henne, C. Szongott, P. Tute, F. Ludwig.