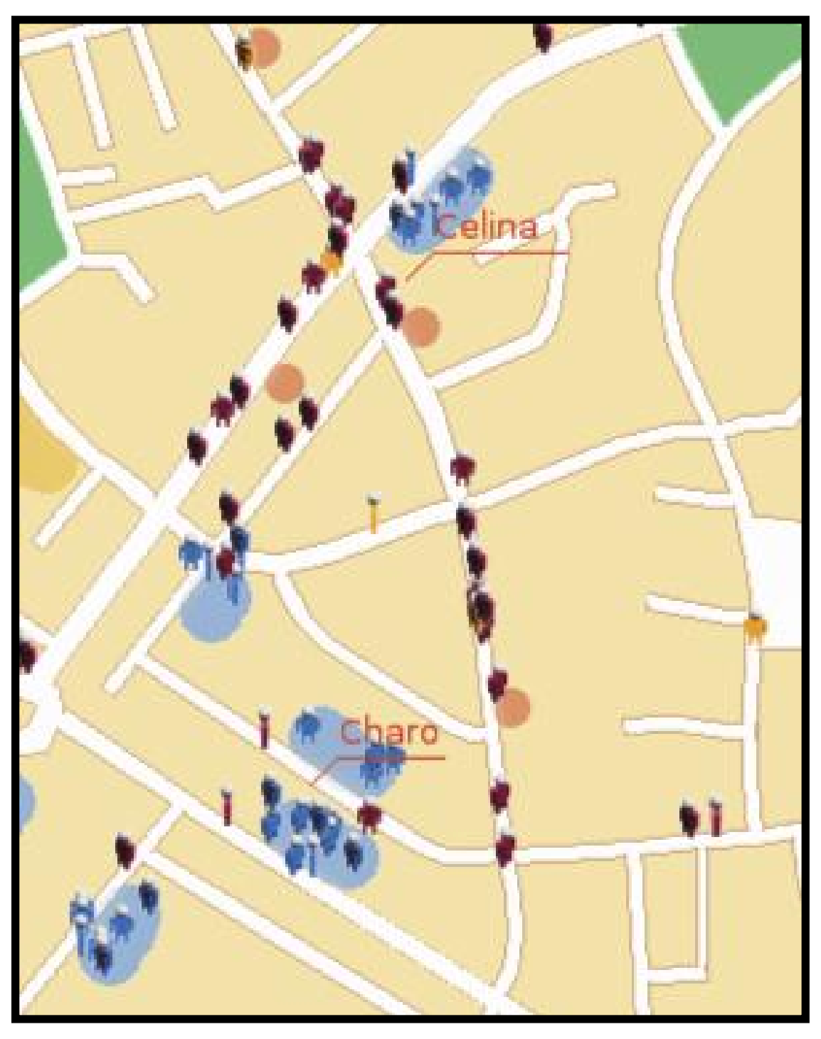


A Generic Large-Scale Simulator for Ubiquitous Computing

Miquel Martin[1], Petteri Nurmi[2]
miquel.martin@netlab.nec.de, petteri.nurmi@cs.helsinki.fi
[1]NEC Network Laboratories, Heidelberg, Germany
[2]Helsinki Institute for Information Technology HIIT, Helsinki, Finland

Introduction

Currently, **testing** the **behavior**, **performance** and **scalability** of mobile, context-aware applications and services is a difficult task. A closely related problem is the evaluation of machine learning algorithms in context-aware settings. Some reasons for the current situation are:

- Context data is difficult to gather and process
- There are no public sensor infrastructures
- Sensor devices are expensive
- Test settings are hard to reproduce
- The required data often comes from complex scenarios and possibly from multiple users.

To ease the situation, we present **a generic simulation tool for ubiquitous computing.** The simulator employs techniques from the field of simulation, which makes our simulator applicable to a wide range of tasks.

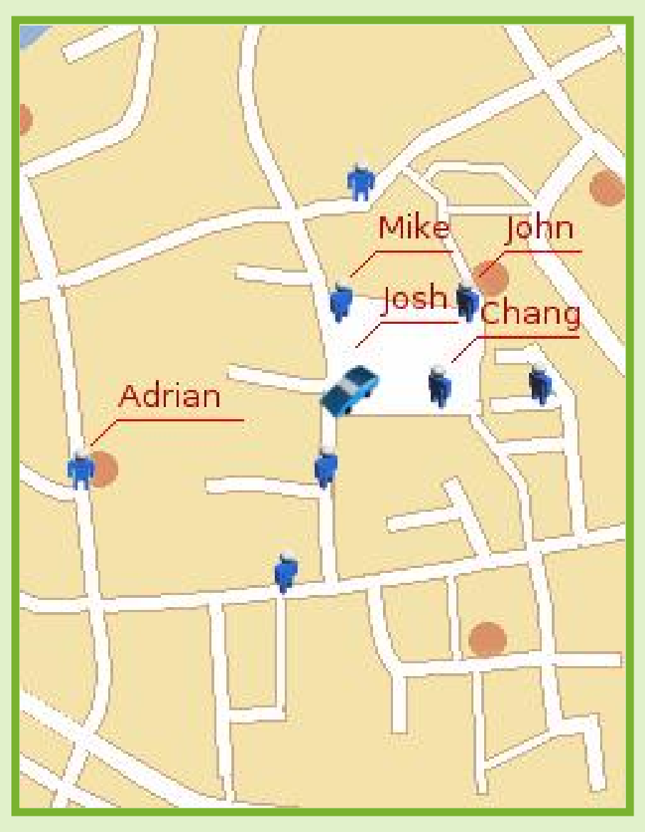
- **How do we simulate the behavior and context of independent agents? What about group behavior?**
- **How do we test the performance and scalability of applications and services?**
- **How can we train and test machine learning algorithms before real user tests?**

Usage example: Prototyping

An important application area for our simulation tool is **application prototyping**. As an example we consider **MobiCar**, a mobile car sharing application.

The goal of the application is to find from requests of users a suitable allocation, i.e., a set of people, whose requests match as closely as possible and who could be willing to share a car.

Before testing MobiCar with users, the used allocation technique should be robust and work relatively well. By using the group simulator, we can test earlier prototypes of the application by **simulating large populations** in a controllable and repeatable environment.



The figure on the left shows how the results of the allocation look like in the group simulator. In addition to easing testing, the visualization eases evaluating the goodness of the allocation.

Simulation Model

The simulation model is split into three parts, each of which is responsible for managing one of three information sources: scenario, individual's behavior and environment.

World Model

The world model is responsible for **modeling the simulation environment** and it defines the places of interest and how global events (e.g., public holidays) are generated.

The simulated environment is specified using two image files: a background image for visualization and an alpha map that defines the areas where agents can walk.

Places can be permanent (e.g. homes) or temporary (e.g. meeting places) and have properties, changeable by either an agent or global event model. Places can be instantiated using a bitmap when based on a template.

Agent Model

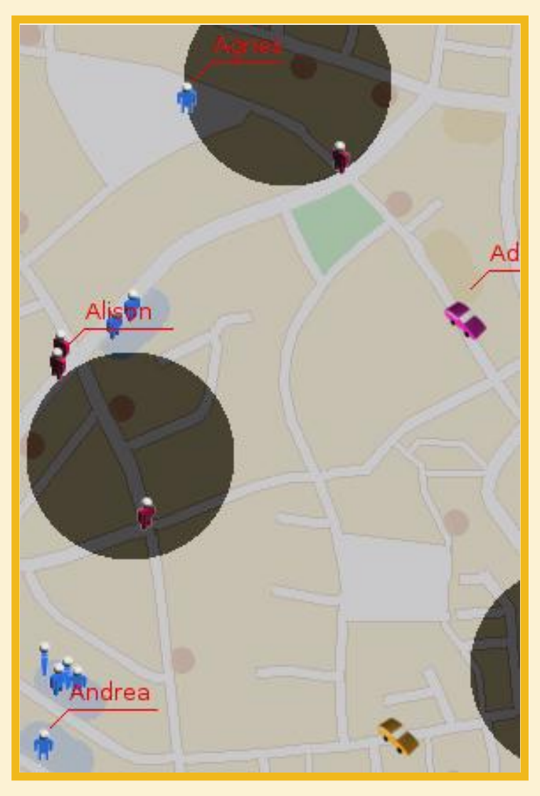
The agent model is the decision logic of individuals, and it is responsible for **making decisions on what to do** and for **updating the state of an agent**. The agent model is specified as a state machine where context triggers state changes. Also, stochastic state machines can be used.

The simulation designer must define the agent model for each agent. We support archetypes, which are stereotype agent models. Examples of archetypes include selfish and cooperative agents in ad hoc networks.

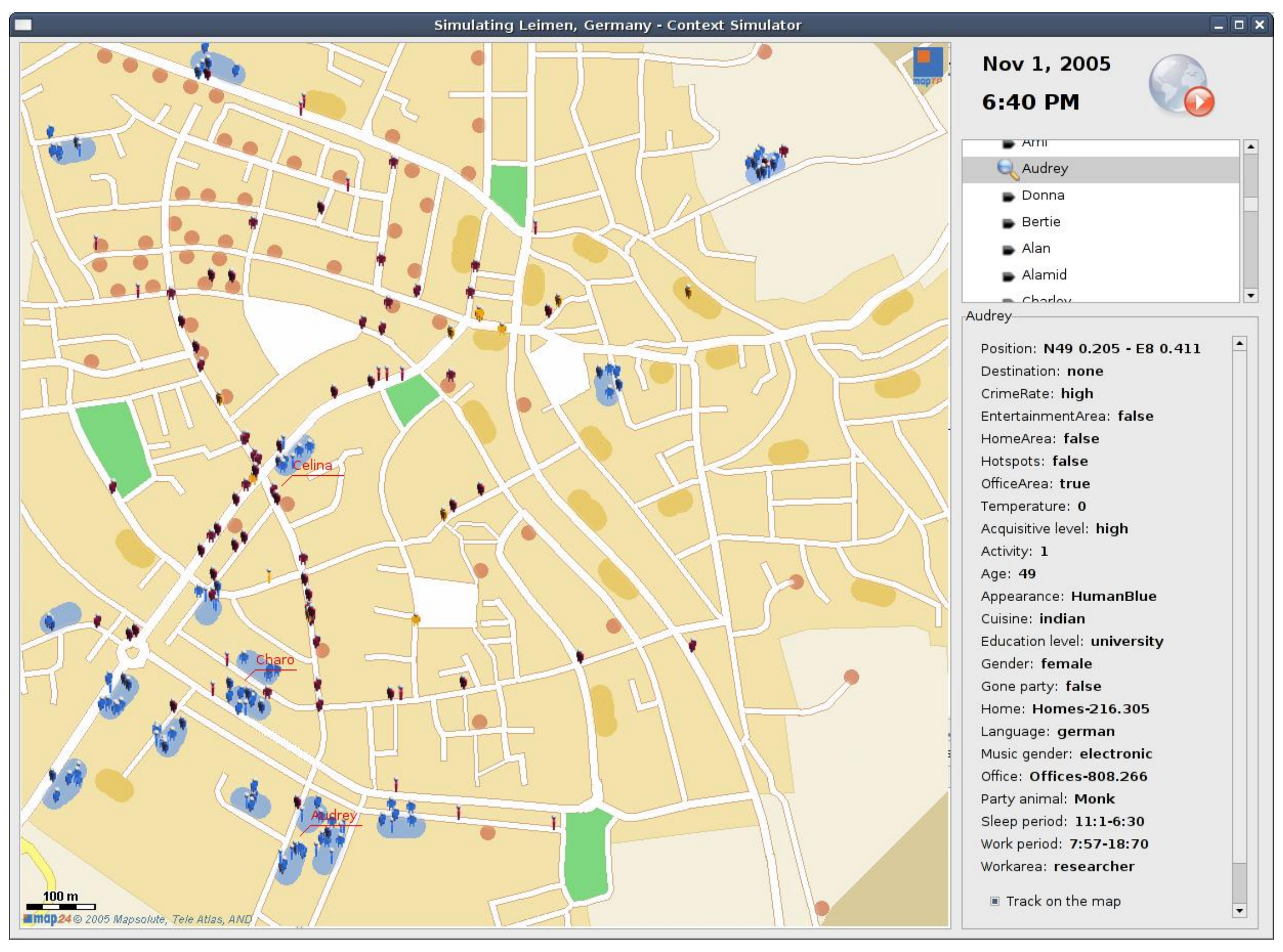
Context Model

The context model specifies the **context variables** that are **part of the simulation** (e.g. temperature or hotspot coverage), how their values are distributed over the environment and simulated over time. At any time, the context of each agent is defined by its personal context (e.g. activity), and the environment context provided by the context model.

The variables in the context model are specified using an overlay image and calibration metadata that defines a mapping between the color intensity in the overlay and the output value.



Top: Temperature overlay , where light areas are colder. Bottom: black areas denote hotspot coverage



Simulating Leimen, Germany - Context Simulator

Nov 1, 2005
6:40 PM

Ami
Audrey
Donna
Bertie
Alan
Alamid
Charley

Audrey
Position: **N49 0.205 - E8 0.411**
Destination: **none**
CrimeRate: **high**
EntertainmentArea: **false**
HomeArea: **false**
Hotspots: **false**
OfficeArea: **true**
Temperature: **0**
Acquisitive level: **high**
Activity: **1**
Age: **49**
Appearance: **HumanBlue**
Cuisine: **indian**
Education level: **university**
Gender: **female**
Gone party: **false**
Home: **Homes-216.305**
Language: **german**
Music gender: **electronic**
Office: **Offices-808.266**
Party animal: **Monk**
Sleep period: **11:1-6:30**
Work period: **7:57-18:70**
Workarea: **researcher**
▣ Track on the map

How do you simulate your own scenario?

1) Create a map for your scenario (i.e. background and its alpha map)
2) Define the places: either individually or using templates
3) Create overlays for your context variables. Define how they are distributed and simulated
4) Model your agents: either create a new agent model, or use the archetypes
5) If necessary, define the global events in the world model
6) Run the simulator and either visualize the context, or extract it as a CSV data

Related work

We highlight the following work:

- Simulation of sensor measurements in smart spaces, [Huebscher,2004]
- Context simulation in virtual reality environments, e.g., QuakeSim [Bylund,2002] and UBIWISE [Barton,2003]
- Multi-agent simulation for middleware testing, [Sanmugalingam,2002]

The advantages of our simulator are its **generic design**, the **usage of well established simulation techniques**, and the **quick modeling and deployment of new scenarios**.

References

[Barton, 2003] – J.J. Barton, V. Vijayaraghavan; UBIWISE, A Simulator for Ubiquitous Computing Systems; Technical Report HPL-2003-93; HP-Labs, Palo Alto

[Bylund, 2002] – M. Bylund, F. Espinoza; Testing and Demonstrating Services with Quake III Arena, Communications of the ACM (CACM); 45(1):46 – 48; Jan, 2002.

[Huebscher, 2004] – M.C Huebscher, J.A. McCann; Simulation Model for Self-Adaptive Applications in Pervasive Computing; Proceedings of the 15th Intl. Workshop on Database and Expert Systems (DEXA), 694 – 698

[Sanmugalingam, 2002] – K. Sanmugalingam, G. Coulouris; A Generic Location Event Simulator; In Proceedings of the International Conference on Ubiquitous Computing (UbiComp), 308 – 315, 2002.

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY