Experimental construction of a meeting model for smart office environments

Daniel Minder Pedro José Marrón Andreas Lachenmann Kurt Rothermel
Universität Stuttgart, IPVS, Germany
{minder|marron|lachenmann|rothermel}@informatik.uni-stuttgart.de

ABSTRACT

The simulation of mobile networks requires mobility models that reproduce the movement of nodes in a realistic way. Although many such models exist, they are not well suited for model office scenarios in which movements are mostly caused by people meeting in known locations to discuss some issues. We present first steps towards a meeting model for office environments which is based on the real world movement data gathered after performing a one-week sensor network deployment in our department. We describe details about the processing of the acquired data, the construction, and the execution of the movement model. Finally, we discuss some of the lessons learned throughout the experiment.

1. INTRODUCTION

Many research laboratories are experimenting with sensor networks, but only a few of them have been deployed in real settings. Most such networks are not easily accessible and are deployed on islands [7], vineyards [1], or even on zebras [4]. Errors may arise due to environmental factors which are not observed by the scientist and are therefore unexplainable and irreproducible. For this reason, the department of Distributed Systems at the University of Stuttgart decided to build an in-house Smart Environment for our office using wireless sensor network technology, which is a controlled environment where experiments can be performed.

Before deploying a network, simulations are used to test the application. The quality of the simulations in mobile environments is dependent on the mobility model. However, we have found that existing mobility models do not provide realistic movements for our application. Therefore, we set up an experiment to record the movement and meeting patterns of employees in our department in order to derive information about the duration and composition of meetings. This data was used in the creation of a meeting-based movement model, which is the focus of this document.

The rest of the paper is organized as follows: In the next section we describe 'Sense-R-Us', our Smart Environment application. Section 3 presents the experiment, the processing of the data, and the design of the group meeting model. Some experiences we have gained during the experiment are also described here. Finally, section 4 concludes this paper.

2. SENSE-R-US

The 'Sense-R-Us' application aims at building a Smart Environment in our department using Mica2 motes from Crossbow Technology Inc. We decided to use Mica2 motes since

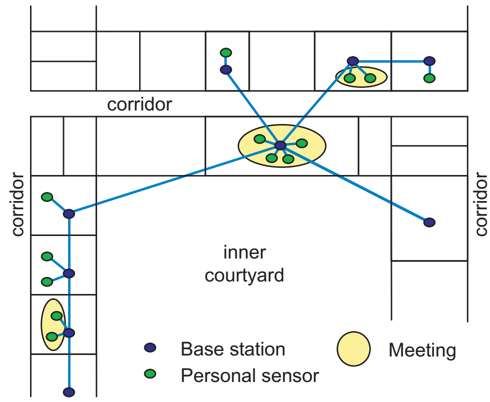

Figure 1: Floor plan with deployed Sense-R-Us

they are readily available and easy to deploy. Additionally they have a low power consumption, are small enough to be carried around, and run TinyOS, so that a powerful development platform is available for testing. The application is divided into mote and PC components. Using a PC-based GUI, the position and status ('isInMeeting', e.g.) of persons can be queried. The location estimation does not have to be very accurate, so that room information is sufficient for our purposes. Other queries the system is expected to answer are "What is the temperature in room X?", "Which rooms are available?", or "How many meetings has person P already had today?".

Two different types of sensor nodes are used in Sense-R-Us: static 'base stations' and mobile 'personal sensors'. Base stations are installed in all office rooms and known locations, e.g., meeting room, lab, and break area. 'Personal sensors' are carried around by the employees. Figure 1 shows the floor plan of our department. The office rooms are situated along three corridors. In the middle of the department, there is an inner courtyard. Since we share the floor with another department, not all of our office rooms are adjoining.

Base stations send out location beacons with their room id all the time. They are plugged to the power outlet and, thus, have no energy constraints. Some of the base stations are connected to the serial port of a nearby PC and — using a small Java application — to the wired LAN. Messages from the GUI running on any PC in the department are routed via LAN to the base station PCs and via radio from base station to base station following a tree-based algorithm.

Personal sensors receive the location beacons and determine their position by selecting the base station with the highest signal strength. They also send beacons which are received by other personal sensors to update their neighborhood list. This list and microphone data is used to detect the occurence of meetings. This detection runs periodically and not just when a query is to be answered since it may happen that nobody is speaking when a 'isInMeeting'-query is received. Messages received by the personal sensors, like queries from the PC-based GUI, are flooded to other personal sensors to ensure that all of these mobile nodes are reached.

Mobility Models for Simulation

During the development process of Sense-R-Us, the application was tested using TOSSIM, enhanced by a simple graph-based mobility model. Soon, we observed that the resulting movements are far from reality due to the following reasons:

Entity mobility models [2] such as Random-Waypoint regard all nodes as independent. Movements in office environments have such an independent component, for example going to the printer or to the coffee maker, but many movements are caused by meetings with colleagues, either in their office rooms or in separate conference rooms. With entity mobility models, these meetings only happen occasionally. [2] describes also group mobility models which focus on moving an entire group but not on the formation process of a group.

Secondly, the number of people involved in a meeting varies widely. There are a lot of two-party meetings, but also group meetings of 3 or more persons, and meetings in which all employees of the department are involved. Just as these meetings are planned in reality so that all persons have time to attend the meetings, it is necessary to plan them in simulation. Normal mobility models do not support such planning.

Additionally, the structure of organizations has to be reflected in the simulation. A department has a head and is divided into several groups which in turn have a group leader. Meetings between people of the same group happen more often than meetings between different groups. It is also obvious that the head of the department meets more often with the group leaders than with the other members of a group. In [3], a social mobility model is presented, but it allows the definition of pairwise relationships only. The model of [6] allows the definition of a complete "Interaction Matrix" between all people. But this matrix is not suitable to define organizational structures and meetings in these structures.

Finally, the duration of meetings varies widely and is additionally dependent on the number of people in a meeting and the composition of the meeting concerning organizational structure.

Although we had a feeling for the nature of meetings, we did not have a quantitative specification for group size, group composition, and duration of meetings. These were gained by performing the experiment explained below.

3. EXPERIMENTAL EVALUATION

The goal of the experiment was to record the locations and meetings of several persons. It should give answers to the question "who meets with whom at which location at what time for how long?".

3.1 Setup

We started by placing 10 base stations in the rooms depicted in figure 1. Although similar in topology, the difference to Sense-R-Us is that the base stations do not work as gateways to the wired LAN, but send beacons with their room ID every 10 seconds. 11 employees were involved in the experiment, carrying around personal sensors. These sensors send ID beacons every 10 seconds and receive ID beacons from base stations and other personal sensors. They do not have query processing capabilities, but save the received beacons with their signal strength and a timestamp to the serial flash. The Matchbox file system was used to provide an easy way to store the data during the experiment and to retrieve the data when the experiment was over. To be able to correlate the meeting data of different sensors, a simple time synchronization mechanism based on a master time sensor was used.

The experiment started on a Tuesday afternoon and ran one week until the next Tuesday afternoon. The personal sensors were handed out and collected in a department meeting. Afterwards, all logger data was read out from the personal sensors. Since the experiment period was short, all persons participated in the experiment without interruption.

3.2 Data Processing

Logger data from the serial flash was processed in several steps to extract the relevant information. The four processing steps to clean the data, i.e., to detect and remove false meetings, are described in the following paragraphs.

In step 1, we computed the time a node (both base stations and personal sensors) was heard from a single personal sensor. To avoid counting beacons which were recorded accidentally as meetings when two persons passed each other in the corridor, beacons had to be heard for at least 30 seconds in order to be considered part of a meeting. On the other hand, due to possible transmission errors it was necessary to tolerate missing beacons: the beacons of a node had to be missing for at least 60 seconds to be regarded as lost. The result of this processing step for one day is shown in figure 2. The bars (grey and black) indicate the intervals during which mote 11 has seen the base stations (Base 1–10) or other personal sensors (Mote 1–10).

Analysis of the intervals for base stations 1 and 2 reveals that most of the time both base stations were heard when the personal sensor was located in the room of base 1. Base 2 was actually the base station of the neighbor room. Therefore, in step 2 the strongest base station had to be selected. The intervals of other base stations that overlap with the interval of the strongest node were deleted or shortened. The results are shown as black bars in the lower part of figure 2 — the intervals of the personal sensors were not changed in this step. The grey bars indicate the intervals that were removed during this step. Base 1 is always stronger than base 2 and base 3 almost always than base 4. Therefore, all bars of base 2 have been deleted which is correct since the employee was not in the room of base 2 that day. Also all the bars of base 4 between 207351s and 213021s have been deleted since the employee actually was in the room

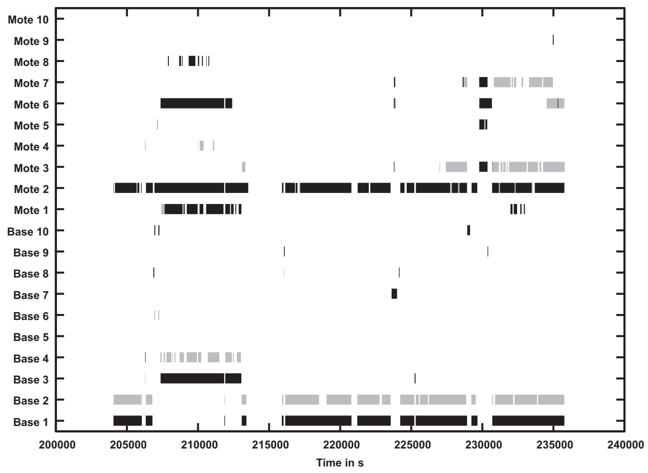

Figure 2: Results after processing step 1 (all bars) and step 4 (black bars only) for personal sensor 11

of base 3 at that time. This step was performed for each personal sensor separately. After this step, the position for each personal sensor at any point of time is unambiguous.

In a meeting of two persons, person A has to hear person B and vice versa. Likewise, based on the logger data, a meeting is only a meeting when mote A hears mote B and at the same time mote B hears mote A. Therefore, in step 3 such mutual meetings were looked for. To perform this step, it was necessary to combine the data of all personal sensors.

Step 4 is similar to step 2 for personal sensors. Since a personal sensor can receive beacons from a base station in the neighbor room, it can receive beacons from a personal sensor in the neighbor room, too. For each meeting of motes A and B, the positions of motes A and B during the meeting interval are determined. The parts of the meeting interval with contradictory positions are deleted. Without this step, a meeting of two persons would be detected although they are in different rooms. The results of steps 3 and 4 are shown as black bars in the upper part of figure 2 — the intervals of the base stations were not changed during these steps. The grey bars in the upper part were removed during this step. For example, the bars for motes 3 and 7 between 230808s and 234698s have been deleted which is consistent with reality since both employees had a meeting in the neighbor room, but not with the employee carrying mote 11.

It is not possible to proof the correctness of the described processing steps. Thereto, all the employees involved in the experiment would have had to record their movements manually which is a high effort reducing normal working time. Moreover, it is very likely that also the manual records are incomplete since sometimes the employee forgets to write the movement down. As indicated in the above paragraphs, we have checked the data processing with samples. For some meetings, we know for sure that they have or have not happened. At least for these meetings, the data processing works correctly, and it is very likely that it does for other meetings as well.

Based on consistent two-pair meetings, it is now possible to extract group meeting information. In a group meeting, all

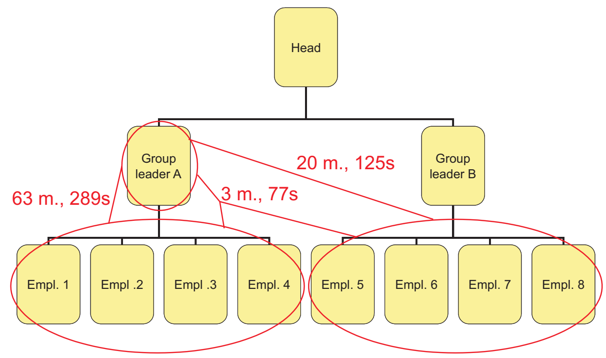

Figure 3: Meetings in organizational structure

involved persons hear each other. This can be detected by constructing a graph of all mutual meetings at a time and finding all cliques (fully connected subgraphs) in this graph. When a node joins (or leaves) an existing meeting, a new meeting starts with one more node (or one less node).

3.3 Model Construction

Using the data gathered during the experiment, we have created a model that contains the following parts: the composition of a meeting, its frequency, its duration, and its location. We show how these parts are identified from the experiment and how they are used in the model.

3.3.1 Composition

In companies, apart from personal relations, most meetings are due to work, and some groups cooperate more closely than other groups. These groups are defined by the organizational structure of the company. Figure 3 depicts the structure of the employees involved in the experiment. Employees 1–4 form one group, employees 5–8 another one. Group leader A and B and the head of the department are treated as separate 'one-man' groups.

For all meetings we have analyzed how many persons from which group took part in the meeting. As result, in the model 20 different compositions of meetings were defined.

3.3.2 Frequency

Depending on the composition of a meeting, its frequency varies. Figure 3 shows as an example the number of meetings observed during the experiment (abbreviated by 'm.') of group leader A with his own group (which means 'at least one person of group A'), with group B, and with people from both groups. As expected, most of the meetings happen within his own group. Meetings with the other group or with both groups also happen, but are less frequent. In the model, every group combination can have its own frequency to be used in the simulation. The frequencies are given in absolute numbers relative to a common time period during which the indicated number of meetings are expected to happen.

3.3.3 Duration

Just as the frequency depends on the composition, the duration also varies from meeting to meeting. The average duration of meetings of group leader A is annotated in figure 3. For example, meetings with his own group are longer than meetings with group B or both groups together.

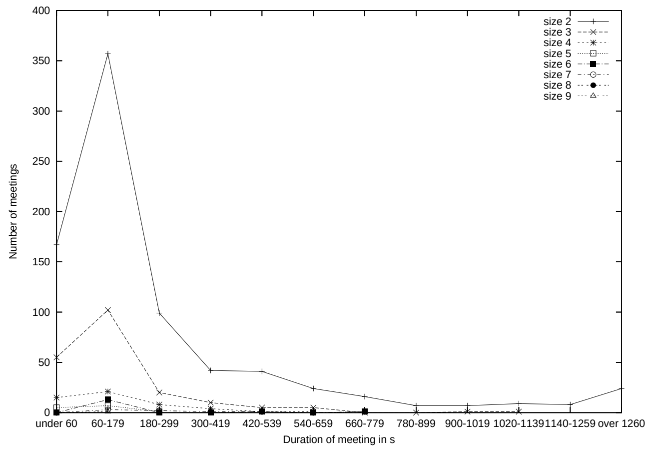

Figure 4: Histogram of duration of group meetings

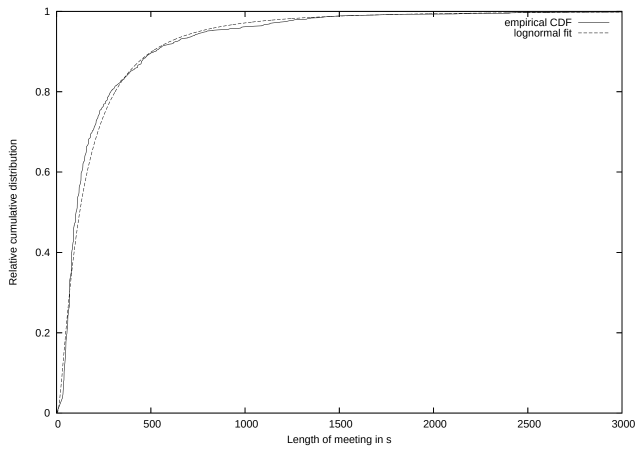

Figure 5: Empirical and calculated distribution

The duration of meetings is neither equally nor normally distributed, therefore the mere use of an average value would be misleading. In figure 4, the meetings are broken down into meeting size and duration of meeting. All meeting sizes show similar duration characteristics: Most meetings take between 1 and 3 minutes, but 19% of all meetings are longer than 5 minutes. The histogram of the duration has a high spike and a long tail to the right. The log-normal distribution provides such a density function. The average value and standard deviation can be calculated from the measured durations and are used to calibrate the distribution function.

When simulating the model, a duration has to be assigned to each meeting. Therefore, a normal distributed random number is calculated and converted to a lognormal distributed random number using the formula $random\_lognormal = e^{\sigma' random\_normal + \mu'}$ with $\sigma' = ln(\sigma)$ and $\mu' = ln(\mu)$. The result is directly used as the meeting duration. For each of the 20 combinations of meetings used, μ and σ are defined as derived from the experiment.

In figure 5, the empirical distribution of all meetings and a lognormal distribution function with $\mu' = 4.8086$ and $\sigma' = 1.1048$ are shown. The good fit is also indicated by a root mean square error of 0.7328.

3.3.4 Location

The location of a meeting depends both on the size of the meeting and on the composition. For example, meetings in which more than a certain number of persons are involved usually take place in the conference room or are informal meetings in the coffee corner. Smaller meetings with the head of the department mostly happen in his room. Usually, two-party meetings happen in the room of one of the participating persons. We do not show a detailed evaluation and the realization in the model due to space restrictions.

3.3.5 Model Execution

To schedule the meetings, the simulation time is divided into smaller sections. The length of a section is chosen so that a meeting with the same characteristics cannot start twice in a section. The meetings that happen in each section are selected randomly, but based on the frequency of the meeting composition. For each meeting, free candidates in each group participating in the meeting are selected. If there are no free candidates in each group, the meeting is postponed to the next section. Otherwise, a starting time and random persons of each group are selected for the meeting. The duration is calculated using the lognormal distributed random number described above.

Then, a location is assigned to each person for all free (i.e., 'non-meeting') intervals. Thereby, the own office has a higher probability. This assignment reflects the individual behavior of persons when they do not have a meeting. This step is necessary before assigning locations to the meetings because a meeting can only be scheduled to the office of one of the participants if the colleagues sharing the office are not there that time. Finally, the location of the meetings are set to one of the rooms of the participants, to one of the common rooms, or to an unknown location. In the last case, the movement algorithm will move the participants to a location outside of the department where no one else is meeting already. In all other cases, the algorithm will move them to the location the meeting is taking place.

3.3.6 Model Evaluation and Adaptation

When setting the simulation time to the time period during which our measurements were taken, the model generates exactly the same number of meetings for each composition. For compositions with lower frequencies, the medium duration deviates from μ given in the model due to the lognormal distribution, but when considering all meetings together, the distribution measured in the experiment is approximated.

The model is general and can be adapted to other, bigger scenarios. The values given above reflect our measurements with 11 employees. When applying to other organizations, the meetings compositions have to be derived theoretically or measured by an experiment. For each composition, the frequency and duration parameters have to be figured out as well. Finally, the office rooms of the persons and meeting locations have to be defined.

3.4 Discussion

During the analysis, we discovered several points which directly influenced the accuracy of our data.

Signal strength was sometimes inaccurate. The signal of a base station placed right behind the wall in the neighbor room was sometimes stronger than the signal of a base station at the opposite corner of the same room. Thus, a personal sensor was assigned to the wrong room. Moreover, the actual meeting this person might have had has not been detected, and phantom meetings might have been added. A possible solution includes a good placement of base stations and their calibration. A base station should be installed in the center of an office room, if possible, but not at walls to adjacent rooms. The transmitting power of each installed base station should be adjusted during a calibration phase. For several positions in the office room, the minimal power is determined which is necessary to receive a personal sensor. The maximum of the minima can be used as new transmitting power of the base station.

Base stations were not installed in every room. If an employee stayed in a neighbor room without a base station, he was located in the room with the base station. Walls attenuate the signal, but as described in the last paragraph, this effect is too inaccurate to serve as a distinguishing feature. The calibration described in the last paragraph could alleviate this problem if the best solution — the installation of base stations in every room — is not feasible.

Not all employees took part in the experiment. Only two of four groups of our department were equipped with personal sensors. But certainly there have been meetings between employees with sensors and employees without sensors. These meetings have not been recorded. But since every employee of the two groups carried a sensor, we believe that the meetings between them have been detected quite accurately. Generally, the short time frame of the experiment and the small number of participants do not limit the idea of the model, but reduces only the accuracy of the calibration parameters.

The granularity of the beacon interval (10 seconds) can lead to shorter meeting intervals of almost 20 seconds. If two sensor nodes meet just after the nodes have sent the beacons the meeting interval will start at the next two beacons. If the two sensors part just before the nodes send the next beacon, the meeting interval will end at the last two beacons. This effect has a greater influence on shorter meetings, but can be ignored for long meetings. Shortening the beacon interval would result in a higher energy consumption and, therefore, a shorter lifetime of the personal sensors. In the experiment it would have been necessary to read out the logger data several times during the experiment since otherwise the data flash would have run out of space.

According to the recorded data, one employee was in his own room only 5.3% of his working time which did not correspond to reality. We discovered that he had bent the antenna to make the mote fit into a cellular phone case. This way, the mote was neither able to receive nor to send any beacons. Clear instructions for the users of new devices help to avoid such problems. A 90 degree rotation of the antenna, so that it can be folded parallel to the long side of the mote, would make the Mica2 mote more practical.

Several of the processing steps we have presented in section 3.2 have been improved or developed during the data processing phase since the problems were unknown in advance. Therefore, it would have been almost impossible to develop an application with full in-network processing of the beacon packets from scratch. We believe that it is always necessary for unknown environments to deploy a test network. Analysis of the acquired sample data gives hints how to improve the quality of data. The application has to be changed accordingly and deployed again. An alternative is a system like TinyCubus [5] that allows for the replacement of the data processing component with an improved one.

4. CONCLUSION

In this paper we have described an experiment to gather real world movement data of employees in an office environment. We have shown the necessary steps to clear up the data and to extract the relevant information. These were used to build a meeting driven movement model for office scenarios. Finally, some problems and other issues we encountered during the experiment and the evaluation were discussed.

We have also presented a Smart Environment application based on Mica2 motes. The sensor network is hybrid, consisting of static and mobile nodes and multiple gateways to the wired network.

Currently, the application is about to be finished and will be deployed soon. Then, we will be able to assess how realistic our model is simulating the movements in our department.

5. REFERENCES

[1] J. Burrell, T. Brooke, and R. Beckwith. Vineyard computing: Sensor networks in agricultural production. *IEEE Pervasive Computing*, 3(1):38–45, 2004.

[2] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[3] K. Herrmann. Modeling the sociological aspects of mobility in ad hoc networks. In *Proc. of the 6th ACM intl. workshop on Modeling analysis and simulation of wireless and mobile systems*, pages 128–129, 2003.

[4] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet. In *Proc. of the 10th Intl. Conf. on Architectural Support for Programming Languages and Oper. Systems*, pages 96–107, 2002.

[5] P. J. Marrón, A. Lachenmann, D. Minder, J. Hähner, R. Sauter, and K. Rothermel. TinyCubus: A Flexible and Adaptive Framework for Sensor Networks. In *Proc. of the 2nd Europ. Workshop on Wireless Sensor Networks*, pages 278–289, 2005.

[6] M. Musolesi, S. Hailes, and C. Mascolo. An ad hoc mobility model founded on social network theory. In *Proc. of the 7th ACM intl. symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 20–24, 2004.

[7] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin. Habitat monitoring with sensor networks. *Comm. of the ACM*, 47(6):34–40, 2004.