# GROUP 4 - ASSIGNMENT 2

**Robert Brünig, Lorenzo Donini, Aurora Terzani**

# MOBILITY MODEL FOR THE FMI BUILDING USING THE ONE SIMULATOR

## Introduction

Being connected even when being away from home or office is becoming ever more important. But planning the infrastructure for such connected mobility (e.g. WLAN) can be difficult.
For estimating the requirements of the infrastructure simulations of the networks' clients and their movements can be a very powerful tool. In this work the Informatics & Mathematics building of the Technical University of Munich is used as an example for such a simulation. We aim to create a model that realistically captures the movement of people within and in the surroundings of the faculty building.
We will capture and analyse data from the simulation to determine the movement characteristics of the users. Further work could be done to simulate the networking environment and deriving ways of enhancing the network quality for the mobile clients within the building.
This report will be structured as detailed here: a brief description of the the scenario to model, followed by an analysis of mobility models in existing literature (1), with regard to the models and approaches best suited for this scenario; then the implementation of our custom model will be presented, as well as simulation results and observations.

## 1. Description of the simulation area

The faculty building is located in the south of the research centre campus of Garching, near Munich. There is a metro station in the north east of the building reachable through a five minute walk. Many of the students travel to the faculty using the metro. Aside from the metro, there are two large parking lots in the east and in the west respectively of the building. The bounds of our simulation area will cover the building itself and a reasonable space around it to account for people coming from the metro and the two parking lots. Those will be important for having an accurate representation of the inbound student flow and hence the students inside the building.
On the inside, the faculty building is separated into several areas in which students show different movement behaviours. Directly on entry students enter the main hall, where different activities such as studying, social interaction and eating are carried out. There are 10 "fingers" connected to the main hall where many of the lecture and seminar rooms are located. Also there are several large lecture halls and a library directly attached to the main hall.

# 2.Analysis of the mobility in the scenario

## Analysis of mobility

The FMI building can be entered from different points. There is one on the north-eastern corner of the building, which acts as the main link to the rest of the campus: from here people can reach the subway station (UBahn), the canteen, other food courts and especially other classrooms and faculties. Other enter/exit points are dislocated both on the eastern and western part of the building for an easier access to the parking areas.
The building is structured on multiple floors and people can move between them using either elevators or stairs. The building also offers two slides inside the main hall, that can be used to descend from the last floor.
All movements that we could notice during a preliminary observation phase, proved to be related to the different kind of activities carried out inside the building, also depending on the people involved in these activities. We observed that professors and students typically move towards lecture rooms or laboratories according to their schedule (e.g. based on a timetable). The main difference between students and professors appeared to be based on the kind of social interactions with peers: outside lecture rooms students typically interact with other students while professors interact with other professors. This is due to professors being employees at the university, while students have more free time during the day and tend to aggregate and spend more time together. The primary meeting point is the main hall; here people can speak, study or eat, either together or on their own.
We encountered that most of the activities carried out by students, besides attending lecture, take place out inside the main hall, adding a certain degree of variety in a person's actions, compared to the somewhat fixed duration of lectures.
Students also stay for a variable time inside the library, which is located in the south-western corner of the building. Professors on the the other hand, typically stay in their offices when they are not inside a lecture room.
Moreover, there are some points of interests for all people, such as the cafeteria land the restrooms. In these places, however, the time spent is very small if compared to other activities mentioned above.
Given this overview on the kind of mobility that we can observe in the faculty, we are going now to survey existing approaches to mobility modelling in order to evaluate which one would fit us best.

## Suitable known models

Depending on the rules that govern the behaviour of the considered agents, we can distinguish two classes of models: trace-based approaches and synthetic models.
The first class sets the distribution according to some statistics extracted from the considered traces and for this reason it is suitable only for the environment from which these statistics have been collected. If we consider, for example, the most popular lectures or the hours in which the main hall is most populated, we could draft statistics and base our model on these.
On the other hand, since synthetic models are based on social attitude and regular schedules, it is more feasible to extrapolate these informations from simple observations rather than from computed statistics, for which we would need a huge amount of data.
For this purpose we collected some real-life data and guidelines that we would need to follow for designing an effective model. Since our goal was to analyse the micro mobility within a building, we considered movements on a minute-timely basis. For complexity reasons, we did not consider collisions between nodes or with tiny objects inside the building, but rather considered them as potential random social interactions.

We will now split our analysis into three parts, each of which is dedicated to a specific family of models, taking different approaches to modelling human mobility; our goal is to underline the properties of each model that we think fit best into our scenario.

## Exploring location preference

The basic idea of this model is that the probability of returning to a specific location is correlated with the frequency it is visited in the first place. Considering for example the lectures with a big number of students or the lunch hours when the probability that a big number of people will go to the canteen or stay in main hall, we could store a set of preferred places for each of the users and decide the next destination they will move towards by choosing randomly one of them out of the given set.
Following the *ORBIT*, each node selects a subset of points generated at the beginning of the simulation and moves between them. We could develop our model based on this idea and, as the *Time-Variant Community (TVC)* does, we could also describe the movements of the nodes using a Markov chain, where the probabilities of transitions between places are driven by the realistic distribution of a node returning to the same place. In *TVC*, movements are parted into time slots, during which different reference locations in the simulation area are associated with the nodes. Another feature of this method, that we could use for modelling our scenario, is that in each time slot a node can move either freely in the whole map or only inside a restricted area: this takes in account both a random movement and the coming back to a preferred place (previously visited).
Therefore it is possible to use these two kinds of movement in order to simplify our model and try to simulate a behaviour as close to reality as possible.
We will be focusing on a daily model, but if we wanted to extend our analysis to a weekly routine, we could probably exploit the characteristics of the *SLAW* model to capture regular back comings to the same location, since subsets of our points of interest remain fixed for each user. However, the jumps between these slots follow the least-action principle: it could be useful for a simple model but not as realistic as we want our model to be.
Moreover *SLAW* is complex and depends on many parameters. For this reason, the *SMOOTH* model, having the same realism as *SLAW* but with simpler parameters, was introduced. Considering the available data and statistics, it would be possible to pursue this model creating communities with different level of popularity so that nodes visit communities based on their popularity level. In this model, however, nodes usually tend to visit communities close to them rather than moving about randomly: this is not true for our scenario because the preferred locations were chosen based on the needs of users and the activities they can carry out, wherever these are located.

## Modelling Agenda

We noticed that the waypoints in our scenario are mainly chosen according to the activities of each user. For this reason the agenda model class probably fits our scenario better, since it focuses on reproducing realistic temporal patterns of movements by explicitly including repeating daily activities in human schedules.
Scaling down from the a huge scenario (e.g. a city) to the view of a single building, we can follow an *“Agenda driven”* model (presented by Zheng et al.) that is based on a real collected data set in which the activities are strictly related to a location and to the time the user spends there. The model also includes a motion generator that profits from Dijkstra’s algorithm to find the shortest path between two locations (i.e. activities). This model takes into account the heterogeneity of the population. In general is not applicable to a different scenario from the one from which it is derived but this isn't a problem for our analysis.

The *Working Day Movement* model combines as many different movement characteristics into one model as possible with a scalable approach. In our case we need to fit different kinds of movements in only one resulting model, hence we can neglect the use of sub-models. The idea we can extrapolate from this model is that each node is initially assigned some preferred locations that work as a starting point for the mobility in the different sub-models. Also different social patterns are introduced in the model and it is thus possible to limit a node group's movement to a specific area (like a district). Although this model was model was built for a high-complexity scenario like a city, it still offers many interesting hints for a deeper analysis of our scenario.

**Social Graphs**
These kind of models utilise graphs in order to describe social ties: the nodes represent individuals and weighted edges represent the social connections between them. The basic idea of the *CMM (Community-based model)* is that a movement is influenced by the other users with whom social ties are shared (i.e. social attraction). The *HCMM (Home-Cell Community based model)* extends this idea by adding spatial attraction. Depending on which aspect of mobility we want to consider, this model is a good starting point for focusing on the social relationships between nodes. In general, the strength of a social bond between users changes over time and thus nodes could be modelled with a time-varying graph.
With these basic ideas in mind, further models can be suggested. More specifically, one that takes into account the time that a node spends in isolation, implementing the individual walk social patterns, and one that considers the choice of a common destination for the next movement, implementing group trips. In our scenario the modelling of the individual and group study could be developed starting from these concepts.
Our scenario really encourages social relationships, since activities are often related to the role of a person in the faculty and the people with whom a node could have ties. Nevertheless, after implementing our mobility model we noticed that these social aggregations could emerge spontaneously with a high enough number of nodes, giving us time to focus on different aspects of mobility (see section 4 for further considerations).

The majority of the models mentioned above are scaled to work on a city level; for this particular assignment we had to scale the model to a lower level, in our case a single building.
Because each of the mentioned approaches is designed to reproduce only a subset of patterns and movements, in order to construct our model in a more realistic way, we decided to combine different techniques into a custom model. To do this we extrapolated from each of the analysed mobility models the most suitable features for our aim.

## 3. Model implementation

Our aim was to build a mobility model specifically for the main building of the faculty of Mathematics and Informatics in Garching.
As mentioned before already, it is very difficult to capture all the aspects of mobility in a scenario as complex as this. We chose to focus on some of these aspects and analyse the results of the simulations, neglect hence some other aspects. In particular, our model was not based on statistics obtained through data mining, but rather on simple observations and realistic node behaviours.

We have decided to base our solution on the main activities that a node can carry out in the building. For simplicity, we deliberately ignored some minor activities that we felt were not important for the purpose of the simulation.
The activities we focused on are related to a specific locations and can potentially be accomplished by different kinds of nodes. We observed that there are three main categories of people visiting the building daily: professors, students and others (all the ones that don't fit in the previous groups, like administrative staff or visitors). Out of these macro groups, we decided to only focus on the students, as they represent the majority of the people at the campus.
Referring directly to the solution architecture, we used ***TumCharacter*** as an abstraction for every possible actor inside the solution. The ***TumStudentMovement*** represents the concrete movement model for each student, used directly by the ONE Simulator to generate actual movements. However, considering how many of the implemented activities are common to both students and professors, it would be possible to extend the current solution as to include the latter without having to change much logic inside the code.
The main activities, which our model is based upon, are: *travel, enter, exit, lecture, eat, bathroom, group study, individual study, library and simple social interactions.*
As in a Markov's chain, every activity represents a state: a node can switch freely between activities during a simulation and, therefore, behave dynamically. This usually happens on a timely basis, i.e. after a specific activity has ended, another one can be started.
Before explaining the state-based behaviour, it is important to stress that our custom model is still map-based: we used a map of the faculty that points out the borders of the building, evidences the fingers as well as other particular areas of this faculty. By looking in more detail at the map, we decided to split it into subareas: we separated the main hall from the fingers and considered the library and the the big lecture room (Hörsaal 1) as separate entities as well. The main purpose for this was to allow more fine-grained behaviours and prevent people from moving in and out of all possible areas while doing random activities. The ***FmiBuilding*** class takes care of most of these functionalities.
While, of course, it is still possible for any node to cross the borders between the mentioned subareas, these movements can be restricted and handled in any state. For example, when generating random movements inside the main hall, only coordinates inside the boundaries of the main hall will be considered. In case the coordinates are generated randomly, these will be computed anew until a valid coordinate has been found.
Below we will describe our solution, focusing on the activities and states implemented in our custom model.

## State generation

The core components of our solution are the ***TumStudentMovement*** model and the ***StateGenerator***, which allows nodes to dynamically switch between states as time passes. All states offer a simple ***IState*** interface, exploited by the *TumStudentMovement* in order to delegate the generation of paths and waiting times. This particular architecture allows us to potentially add any number of states and activities to the solution in the future and simply plug them into the *StateGenerator*, thus creating more complex node behaviours and routines.
Since a routine is made up of a sequence of randomly chosen activities in our model, we will now analyse the states that make up these possible routines. We want to underline the fact that we wanted our simulations to be highly customisable, therefore each state was designed to correspond to a specific activity; also, each activity is typically composed of a

movement period, followed by a wait time in which the node is actually staying in one place and carrying out a task, whichever this may be.

## Travel

In the first place we focused on making the nodes enter/exit the simulation area. Since each state allows a user to move and subsequently perform carry out an activity, we came up with this state, which is a state of inactivity of the user before entering the simulation area. At spawn time each node decides when it will reach the campus: before that exact point in time a node will be waiting in a travelling state. After exiting the travelling state, a node will finally enter the simulation area.

## Enter

How the arrival time is decided will be covered later, but regardless of that, a node can enter the simulation area from three different points: the UBahn station and two separate parking lots. To make the simulation seem more realistic, nodes coming from the UBahn station will enter the simulation area in periodic waves, determined by the timetable of the subway (default: every 10 minutes). Node that arrive to the FMI building by car can, instead, enter the area at any time.
Each entry point is connected to a specific entrance of the building on the map: this was done by using a direct sequence of coordinates for each entry point, that a node will automatically follow when entering the building.
The outskirts of the FMI building are not considered for the scope of this assignment, although a more detailed model could consider the activities in the outer areas as well.
Once a node has entered the simulation area, it can enter any other state, depending on specific factors.

## Main Hall

The ***MainHallState*** was built as a more complex state, since different activities can be carried out by students inside the main hall of the FMI building. Given that most of these activities share some common logic, we decided to build a shared state for these. Inside the main hall, a node can roam freely, usually picking a random waypoint for each started activity.
The activities than can be carried out in this particular state are:
- Individual study,
- Group study,
- Social interaction,
- Eat.

For all of the mentioned, a node picks a random destination, moves there using a direct path and then waits for a random time (a lower bound is provided). More specifically, we could call it shortest path movement, but since the main hall is a giant open space with rectangular-like boundaries, the destination is always reachable via a straight direct path. We did not implement or handle any obstacles inside the main hall.
For group study and social interaction a more complex structure was considered, since these states require nodes to physically interact with each other, but the movement and waiting logic remains unchanged. The social interactions will be covered in a later subsection.
When a node doesn't have any scheduled tasks (see lectures), it will choose one of these actions at random, based on dynamic probabilities. For example, during the lunch period (default 11:00 - 15:00) a node will have a high chance of entering this state in order to eat.

In our solution nodes are not allowed to eat multiple times, but there are no limitations to how many times in a row a node may enter the main hall state to perform other activities. A node could potentially linger inside the main hall for several hours, switching between the social interaction and individual study activities. We felt that this should be a key point of our model, since we observed that many people actually spend quite some time inside the main hall, not only in between lecture, but also if they don't have any upcoming lectures.
The priority of an activity can change dynamically over time and the choice of the next state is always responsibility of the *StateGenerator*, which can check some state variables stored inside the *TumStudentMovement* in order to make decisions.

## Bathroom

To make the model as realistic as possible we have introduced this state. Although this state could be handled inside the MainHallState, we decided it would be best to keep it separately, if at some point we wanted to implement a queue for each bathroom.
Given the coordinates of a bathroom in the scenario, the node will follow the shortest path to reach it and spend there the necessary time (default 3 minutes). When deciding the next state of a node, the StateGenerator checks the last time in which a node entered this state to avoid frequent entrance and unrealistic patterns. Hence, it will be much more likely for a node to enter this state after some time has passed (as in hours), rather than entirely at random.

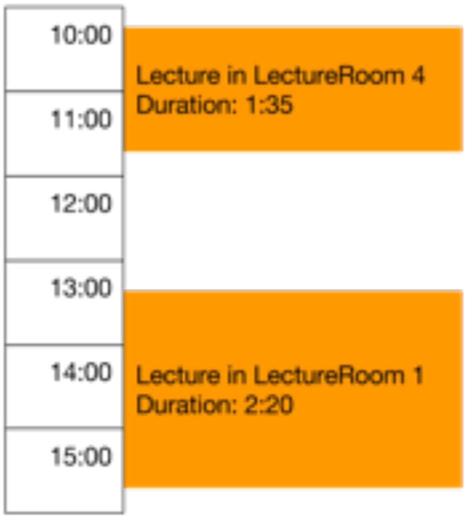
## Lecture

This was considered to be the main state during the daily routine of a student. It indicates a period of time in which a student stays inside a classroom in order to attend a lecture.
There are many classrooms in the building, each one with a different capacity, able to host a specific number of nodes and with a different popularity coefficient, based on the size of classroom. Given the position and the capacity of each class, a ***LectureRoom*** works as a generator of lectures: in each room a sequence of non-overlapping lectures will take place. This creation process takes place before the simulation actually starts. Each lecture has a specific time slot assigned to it, in which nodes can occupy that classroom and attend the lecture. The time slot can vary between 1 and 3 hours in the current implementation, but we also kept into account possible delays and variable start/end times of a lecture. The reason for this choice was to prevent people from exiting different rooms simultaneously and reaching the main hall all the same time.
Nodes decide their daily schedule based on which lecture they will attend during the day. At creation time each character will register to several lectures: these are randomly chosen between all lectures that start within a certain time slot and also depend on the popularity of a lecture. If a lecture room is filled to capacity for a specific time slot, no further nodes will attend a lecture there in that period, but register for another lecture instead.
Since each node already knows its daily schedule at the beginning of a simulation, it will act accordingly:

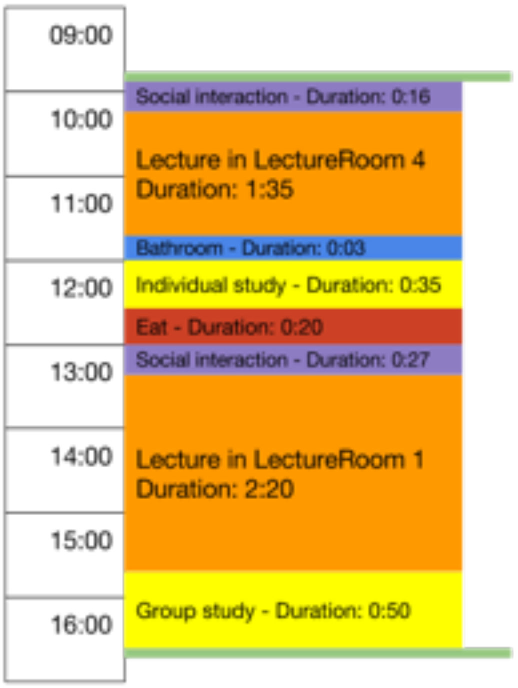
- a character will enter the simulation area before his/her first lecture starts, with a variable offset;
- a character will involve himself in other activities, such as individual study or eating, only in the free period between lectures, as it happens in real life;
- the time remaining before the next scheduled lecture starts is always taken into account for the purpose of choosing activities, providing an upper bound of time that a character can spend inside the main hall/library;

- once a character has no further lectures, he/she will usually leave the simulation area. The character might also decide to stay at the faculty and carry out other activities, since this is a random choice based on a certain probability.

Below is shown a possible routine, in which it is visible that all activities apart from lectures are chosen randomly and could be entirely different from person to person, even if the attended lectures during a day were the same.



Generated Lecture Timetable


09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
Social interaction - Duration: 0:16
Lecture in LectureRoom 4
Duration: 1:35
Bathroom - Duration: 0:03
Individual study - Duration: 0:35
Eat - Duration: 0:20
Social interaction - Duration: 0:27
Lecture in LectureRoom 1
Duration: 2:20
Group study - Duration: 0:50

Final routine: lectures + random activities

Lecture represent, therefore, the basic structure of a daily schedule, around which all other activities grow, eventually differentiating routines to a certain extent. Since we offer only a limited number of activities, the possible combinations are not limitless, but still grant a good degree of heterogeneity in this kind of simulation.

## Library

One of the main locations in the campus is the library that is situated in the south-western corner of the building. Aside from studying inside the main hall, students tend to spend a considerable amount of time studying individually inside the library. We implemented this state apart from the MainHallState for location purposes: while for normal studying nodes pick a random waypoint, in order to study inside the library nodes will have to follow a fixed path in order to reach the destination. As for other states, the amount of time a node stays inside this state is arbitrary, with a lower bound that can be configured.

## Exit

When the time that a node has to spend inside the faculty expires, given the last location of that node, the same path that the node followed when entering the simulation area will be followed in reverse direction. Each character will hence exit the simulation using the same means of transportation he used to come to the faculty.

### Social interaction

For this assignment we did not implement social ties known a priori, but designed a basic social pool used for interactions. This holds true for both simple social interactions and group study. The main difference between these two cases is the size of a social group and the minimum amount of time a node can spend in it (no lower bound for simple interactions).

The ***SocialPool*** class allows nodes to spontaneously create a social group whenever they enter a social-based state. Once a group has been created, other nodes that enter the same state will try to join an existing state (eventually creating a new one in case no social group existed yet). Upon joining a social group, nodes will converge on a location decided by the creator of the group. The idea behind this approach is the meeting point: two or more nodes decide upon a location where they will meet each other and will then get there, at different moments in time and then spend some time together. A priority queue like mechanism guarantees that nodes will first converge to social groups with fewer nodes, rather than groups with many people in it already.

### Configuration

The whole solution was designed considering the principles of customisation and extensibility, in order to facilitate the testing and the addition of new states to the model.

For this purpose we created a ***TumModelSettings*** class, which loads configuration settings and provides a unique access point to all other classes. Whenever a class requires a specific parameter (e.g. the maximum amount of lecture a character can attend) this can be looked up directly in the mentioned singleton class. All settings and parameters are stored inside the ***tum_settings.txt*** file, which is needed in order to load the correct model for the simulation.

Some of the basic parameters used for our tests included a day of 24 hours, in which the first lectures started officially in the time slot starting at 8 A.M. and the last lectures finished at 18 P.M. Single lectures last approximately 45 minutes per each time slot (e.g. 90 minutes for a 2-hour lecture and so on) and the popularity of a lecture goes up accordingly to the capacity of the room that hosts it:

- rooms with a capacity higher than 200 will attract people with a 45% probability;
- rooms with a capacity between 40 and 200 will attract people with a 35% probability;
- all other rooms (capacity < 40) attract people with a probability of 20%.

The simulations were performed with an amount of nodes varying between 300 and 2000. All other significant parameters can be looked up directly inside the configuration file.

## 4.Results of the simulation

Due to the complexity of the solution and the limited time it was not possible to test many different configurations, therefore we sticked to the configurations we felt were the most realistic in the analysed scenario.

Using a different amount of nodes in multiple simulations allowed us to look at macroscopic and microscopic behaviours, putting in evidence different aspects of the mobility.

For starters, the social approach proved to be somewhat limited in simulations with a limited number of nodes, since nodes would wait for a long time on other nodes to join them, possibly leading to a behaviour identical to the one seen for individual study,

defeating therefore the purpose of the activity in the first place. Furthermore we noticed that with a high number of nodes, making nodes converge on a specific point (i.e. where people are meeting) didn't not add too much detail to the simulation on a macroscopic level; not making nodes converge at all showed us that social aggregations naturally emerged, since characters tended to create small groups in which some nodes were closer to each other, without the need to make them actually converge on a specific point.
Nodes move as intended and the flow of people entering/exiting the lecture rooms reflects the behaviour we could observe each day inside the faculty. Since we didn't take the Garching Mensa into account, nodes of course will not leave the building before heading home, eating inside the FMI building instead.
Movements on a microscopic level can be appreciated as well: when tracking the schedule of a single node (either by launching the simulation with only one node or creating logs) it was possible to see that with a good degree of randomness, nodes follow coherent patterns and make realistic decisions.
We also generated reports for node densities and flight length of nodes. The first analysis showed us that, on average, all nodes were around 395 units away from the centre of the simulation area: this is due to the nodes spending quite a fair amount of time inside classrooms. The second analysis showed us that the average flight length of a node was around 84 units, which means that nodes typically don't move across the whole building, but rather cover short distances, due to the activities carried out inside the main hall.

## Conclusion

The suggested solution represents a synthetic mobility model based on key observations of the scenario. We decided to cover only specific aspects of the behaviour of nodes, leaving out other details that we felt were not decisive for the purpose of the assignment. We mainly focused on modelling lectures and the seemingly random movement of people inside the main hall of the FMI building, allowing the nodes to take arbitrary decisions throughout the simulation, based on dynamic probabilities.
The architecture of the solution was designed to aid in extending the functionalities of our model by implementing new states or tweaking existing ones; this gives the possibility to reuse the code and hence fine-tune the scenario to be even more realistic in the future.
In light of what was implemented, we are overall satisfied with the results, as they reflect in great part what was our initial goal.

## References

1. Karamshuk D., Boldrini C., Conti M., and Passarella A., Human Mobility Models for Opportunistic Networks, *in IEEE Communications Magazine*, December 2011