Towards Recognising Individual Behaviours from Pervasive Mobile Datasets in Urban Spaces

Radoslaw Klimek

Department of Applied Computer Science, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Krakow, Poland; rklimek@agh.edu.pl

Received: 15 February 2019; Accepted: 6 March 2019; Published: 14 March 2019

Abstract: Mobile phone network data, routinely collected by its providers, possess very valuable encoded information about human behaviors. Intensive tourist activities in urban spaces bring smartness via mobile phone fingerprints into the understanding of an urban ecosystem. Due to the diverse processes that govern mobile communication, mining the geolocations of individuals seems to be non-trivial, tedious, and even irregular, which can lead to an incomplete trajectory. Enriching trajectories with infrastructural facilities is another challenge. We provide a unified approach, comprised of both informal and formal elements, to obtain a common framework, which maps pervasive datasets into a collection of individual patterns in urban spaces, to obtain context-enhanced trajectory reconstructions. Through the algorithmization of the approach, we acquire a study that provides new insights on individual and anonymized tourist behaviors. In order to obtain individual behaviors, it is necessary to carry out an arduous extraction process. We propose a multi-agent system architecture and predefined message streams, which are transported on a message-broker platform. We also propose all of the basic algorithms that compose the prototype of the entire multi-agent system. All algorithms were formally analyzed due to termination and time complexity. System evaluation, together with a few basic experiments, was also carried out. The performance evaluation results authenticate system feasibility, credibility, and vitality. Those factors prove its effectiveness and the possibility to build the target system, whilst supporting every urban ecosystem. The system would also strongly influence municipal services to understand urban context and operate more effectively in order to support tourist activities to become safer and more comfortable.

Keywords: mobile phone network; pervasive dataset; urban ecosystem; individual trajectory construction; multi-agent system; algorithmization; tourist movement

1. Introduction

1.1. Problem Statement

Pieces of data, ubiquitously generated during the plain interaction between the mobile phone and the serving telecommunication network, are a rich source of information. These pervasive datasets are recorded and stored in Base Transceiver Stations (BTS), which are basic devices providing wireless communication between mobile phones and a telecommunication network. The general availability of mobile phones, accompanying us in everyday life, provides a great potential towards identifying people activities. (Anonymized) Call Detail Records (CDRs), produced during the above-mentioned interactions, would allow us to estimate the locations of important places, as well as other behavioral aspects of inhabitant/tourist activities, especially if some other open and available technologies are applied to support this. It is worth noting that nothing can replace the common nature of the mobile phone data collected. Likewise, the most intelligent applications installed on your smartphone are...
not enough, since not everyone uses smartphones and not everyone agrees to install a monitoring application. Nowadays, there are more mobile phones than personal computers. Thus, the (ordinary) mobile phone is the most democratic communication means, while leaving traces in the form of CDRs.

Quality is crucial for a successful tourism industry. Thus, the evaluation of effectiveness, in achieving the goals of assumed and expected sustainable tourism development, is fundamental. Tourist trajectories and patterns of behavior might be extracted from mobile phone datasets, and thus replace inefficient and traditional questionnaires, no matter if paper or web-based, within leisure, recreation, and tourism scenarios. Since manual surveys are so expensive to conduct in terms of time and money, automatic surveys, based on the analysis of pervasive datasets from mobile networks, seem to be an excellent alternative, the benefits of which are hard to overestimate.

1.2. Objectives and Contribution

The purpose of this paper is to map population datasets into a collection of individual behaviors. Those behavior models can be successfully used in the analysis and evaluation of tourist traffic intensity within an urban area. The aim of this work is to propose a system, in a prototype version, preceding the intended target system, to monitor tourist city activities, as well as enrich innovative urban ecosystems. Thus, the contribution of the paper is a unified approach, which consists of a multi-agent system, identified mobile data streams, and algorithms for software agents, to solve the important problem of supporting urban ecosystems. It allows us to convert an informal description of a general idea into a group of algorithms. Another contribution is the novel method of mapping filtered pervasive streams of datasets into a collection of individual and anonymized tourist activities, located in any tourist destination. This research paper, which contains the algorithmization of the presented problem is characterized by novelty due to the new approach, methodology used, individual trajectory discoveries, and context-enhanced trace reconstructions. Our approach allows us to rethink previous works with reference to complete mobility data, complex mobile communication, tedious trajectory mining, and moreover, enhanced with infrastructural facilities for each trajectory. It provides novelty due to trajectory uniqueness and movement predictability. This research opens some new directions, especially related to implementation and experiments in particular.

1.3. Paper Organization

The work is composed of several parts. Section 3 presents the basic issues connected with BTSs, from which CDR flows come. Section 4 discusses primary matters related to tourist questionnaires, their specifics, and presents initial ideas that underlie the process of system creation. The proposed system is designed as a multi-agent system, as presented in Section 5. Basic messages flowing through the system are also described. Section 6 includes all of the most important algorithms that create the base for the system to work. It presents the method of data processing, along with methods of synchronization and multithreading. Section 7 consists of information related to formal system evaluation, as well as basic experiments to enable its validation. Section 8 discusses the conclusions drawn and the possible directions of further research works.

2. Related Works

There are many works that have considered behavior recognition in ubiquitous computing and their relevant subsets, which focus primarily on pervasive datasets stored in BTSs as a subject of these research interests. Mobile phone datasets have gained both in importance and popularity (see the review paper by Şahin and Zhen [1]), both for empirical and theoretical research purposes concerning medicine [2,3], engineering, education, and even ethnic segregation [4]. The CDR datasets are the subject of intensive pre-handling and cleaning to obtain big dataset for further studies that are concerned with behaviors and activities [5]. CDRs are intensively analyzed to calculate different factors, coefficient, correlation matrices, etc. [6]. CDRs can be also the subject of intensive processing
using modern graph databases and their query languages; see for example Neo4j [7], for detecting abnormal customer behaviors.

Urban analysis is a very popular subject of analysis; see the paper by Ratti et al. [8]. Most of the work focuses on the entire stream of behaviors rather than considering an individual one. For example, in the work by Reades et al. [9], mobile phone data was analyzed as data that create a holistic and dynamic city system. This allows the building a dynamic and real-time representation that goes city-wide. The work by Isaacman et al. [10] provided a method of identifying inhabitants’ important locations by clustering and regression. Based on some simple rules, algorithms for selecting home and work locations were described, and both individual and group behavior were considered. The work by Calabrese et al. [11] described mobile phones in a real-time urban monitoring system, based on fixed sensors and GPS receivers. These combined approaches allow the preparation of a monitoring platform to visualize vehicular traffic and the movements of pedestrians. Behavioral patterns are discussed in the work by Gonzalez et al. [12].

In the article by Zhao et al. [13], information extracted from CDRs was compared by taking into account the total travel distance, the movement entropy, and the radius of gyration. The authors claimed that CDRs underestimate the total travel distance. In our approach, we make a precise measurement of the phones’ geoposition, without being reliant on being within the BTS area. The article by Järv et al. [14] discussed human activity-travel behavior patterns. CDRs for anonymous mobile phone users allow the building of weekly, monthly, and seasonal routines and variability, as well as examining individual spatial behaviors. The article by Dong et al. [15] is another work concerned with CDRs, with the aim of proposing the traffic semantics concept, and an algorithm, to provide support for transportation planning agencies. One of the aims was to extract data for traffic zone division. The article by Steenbruggen et al. [16] showed the use of mobile phone data to improve urban planning and management, finding CDRs to be a good source of estimating real-time traffic. The article by Becker et al. [17] was concerned with human-mobility patterns for human traveling activity, moving around, to support urban planning, deploying city facilities, and to decrease urban traffic.

The article by Qin et al. [18] considered the tourist movement in the city of Beijing, based on CDR data. Analytics in this area included some selected tourist facilities (for example, the Forbidden City, the Summer Palace, the Olympic Forest Park), and the presence of tourists is confirmed there by hot spots. In other words, the exact trajectory in the means of determining the geolocation trace is not proposed, but the work is solid and comprehensive. The article by Lwin et al. [19] used one-week CDRs and GIS information to calculate an hourly link population and flow directions, which were grouped into two opposite sides. However, these calculations were based on call activities, and not by determining the exact phone trajectory. In an article by Thuillier et al. [20], the weekly patterns of human mobility were given. Individuals were classified into six presence profiles, then individuals were clustered. New indicators for the population were proposed. In the article by Huang and Xiao [21], traffic sensing and estimation, based on log data analysis, was performed. A survey concerning learning algorithms was provided, as well as future works suggested.

Big data analytics is always valuable within the area of tourism, creating opportunities to obtain a large amount of valuable data; see the article by Li et al. [22]. Ferrari et al. [23] provides studied to discover events via recorded human mobility patterns. The articles by Ahas et al. [24,25] referred to tourist movement and constituted a solid observation of the behavior of people, through the use of passive mobile positioning. Visitors were identified via roaming information, generally only foreign visitors. This allows the assessment of the possibilities to enhance tourism statistics. This approach is similar to ours, but the differences include: another method of identifying visitors, real-time visitor geopositioning, dynamic detection of tourist facilities, and combining them with visitor trails.

The article by Karam et al. [26] discussed security issues for cloud computing, providing a high-level abstraction model to help multiple agents to specify agents’ intentions, which enabled agents to optimize the time of interaction with the adopted programming model, showing “desires” and
“wishes”. The article by Al Ridhawi et al. [27] also discussed cloud computing concerning a probabilistic learning technique in the cloud. The problem of optimizing the resource in computing to manage the cloud provider’s resources was discussed in the paper by Al Ayyoub et al. [28]. Although cloud computing is important, in our work, we have only algorithmized our problem, and locating computing in the clouds can be the subject of further research activities, so we will return to it in the future. The article by Baker et al. [29] is another work referring to cloud computing; however, it also proposed a workflow model for an autonomic service composition. This work can be used in further studies, since workflows play an important role in the system design, and obtaining its logical specifications (see [30]), enables formal analysis of behavioral models in a logical style.

The problem of agentification the Internet of Things (IoT) was discussed in some articles; see for example the articles by Maamar et al. [31], and by Kwan et al. [32]. Our approach covers a partially-proposed methodology, containing the definition of an ecosystem, agentification of things, as well as implementing a case study.

Summarizing these works, the dominant observation is that there is a lack of a focus limited only to the individual behaviors within mobile phone datasets. However, the works influence this research by showing a challenging research direction, as well as considering some patterns of behaviors. The comprehensive and surveyed work by Blondel et al. [33] discussed mobile phone datasets analytics and patterns. It showed the many important aspects of social networks, geographical partitioning, and urban planning. The availability of mobile phone datasets builds a potential that could benefit urban ecosystems. However, we are interested in collecting more individual behaviors of a selected group of people, that is visitors, and not only foreign people. Their behavior patterns created during a city visit are specific and unique.

This work follows up on paper [34]. However, in the current work, the changes are rather far-reaching. Firstly, the multi-agent system architecture was fully modified: now it is much more realistic, while better reflecting the specified aims of the system, as well as its tasks are well-divided and dispersed. Even greater changes are related to the proposed algorithms, which were fully re-designed. This resulted from the fact that in the previous version, the algorithms had many simplifications, and as a consequence, they could not be successfully realized. In the current version, the algorithms are much more refined and take into consideration the new architecture of a multi-agent system. The current work differs from the previous work, due to the experiments that render the proposed system.

3. Preliminaries

Systems for mobile communications (e.g., GSM or UMTS) are well established. There are many works that have introduced the world of data communication procedures; see [35].

The most obvious part of the mobile phone network is a base station. A Base Transceiver Station (BTS) is a piece of equipment that enables wireless communication between a user and a network. Currently, cities and regions are covered with a relatively dense network of BTSs; see for example Figure 1. However, outside the cities, networks are less dense. In each case, they gather and store important and interesting information about different types of user activities.

A Call Detail Record (CDR) contains data recorded and produced by telecommunication equipment. CDRs, as collections of information, have a special format [36]. Below is given a sample fragment of a CDR text, decoded from binary format. The first row must contain a header row, which includes the field names:

```
"Call Type", "Call Cause", "Customer Identifier", "Telephone Num Dialled", 
"Call Date", "Call Time", "Duration", "Bytes Transmitted", "Bytes Received", 
"Descrip", "Chargecode", "Time Band", "Salesprice", "Salesprice (pre-bundle)", 
"Extension", "DDI", "Grouping ID", "Call Class", "Carrier", "Recording", 
"VAT", "Country of Origin", "Network", "Tariff code", "Remote Network", 
"APN", "Diverted Number", "Ring time", "RecordID", "Currency"
```
The meaning of these columns is not analyzed here, since they are intuitive, and a very detailed discussion exceeds the scope of this paper. Location information is extracted as part of the interaction data. These location observations, i.e., the moment of the phone’s entry into the area of a station (log in) and the moment they leave that area (log out), are of fundamental importance to the considerations given in the following sections of the paper.

CDRs serve a variety of functions. Mobile phone companies can also shape the form of records, for example introducing new fields, if necessary, to establish the whereabouts of an individual during their stay within the range of a station. Broadly speaking, the format of the CDR varies among providers; some programs also allow CDRs to be configured by the user. For the purpose of this work, the existence, or introduction, of certain fields allowing the identification of people as visitors (see Line 3 and the following in Algorithm 2 and see Line 17 and the following in Algorithm 1), as well as direct information about logging in or logging out of the BTS are assumed.

Information about the phone logging in and logging out, to and from a particular BTS, is extremely important for many reasons. Firstly, we know where the phone is located, since it can only be logged in to one station within the monitored area at a particular moment. Secondly, this base station can initiate certain actions, with the support of neighboring BTSs, which enable the geolocation of the mobile phone (see Line 34 in Algorithm 2), according to the methods described in [11]. It is commonly known that it gives a measurement precision level of around 150 m. The measurements are periodical and enable the building of a phone track, point by point, through the entire period of a phone’s presence within the monitored area.

The ecosystem is a distributed, self-organized, and open system, which gathers knowledge about (selected aspects of) a smart city environment. It constitutes a community of digital devices and their environmental functioning, as a whole (hardware, software, services). This system might be extended to consider other aspects of a smart city, for example urban pollution, fire and emergency systems, water and sanitation, energy, etc., since tourist movement may strongly affect these aspects of city life. Thus, the diagram shown below could be much more elaborate, showing more details, taking many other aspects into account. However, it exceeds the main goals of this work and may be subject to other future research.

Thus, we will also show and build our system in a certain context created by an urban ecosystem; see Figure 2.

Some users (actors) of a smart ecosystem, that is context-aware, are identified: BTSs, other supporting services, emergency services, and public transportation management. A BTS compounds
an infrastructure of mobile telephony, where CDR data are gathered, and partially, the algorithms of our system are also performed. Other supporting services consist of other external services that support our system; see Line 3 in Algorithm 5. An emergency service is an organization that ensures public safety and deals with emergencies if they occur (ambulance service, police, fire brigade, and others). Public transportation management consists of systematic processes, which collect and analyze information about conditions. These are required as inputs for the urban planning processes, in order to support decision-makers for the appropriate strategies. The above actors operate in the context-aware urban system, which consists of the following sample use cases: visitor monitoring (UC1), manage transport (UC2), and manage crisis (UC3). Brief descriptions of the use case features are provided instead of a formal scenario.

![Use case diagram](image-url)

**Figure 2.** A use case diagram for a smart urban ecosystem, or outline of the idea.

Visitor monitoring (UC1) is our system, the prototype of which is presented in this work. The main objective of this system is to understand the behavior of a large group of objects, namely visitors and tourists, staying within the city area, whose behaviors can influence the entire city.

Manage transport (UC2) means supplying chain management for transportation operations within the public area. When tourist activities in selected areas increase, the responses might be comprised of: increasing the frequency of buses/trams, shuttle services, if necessary, activating additional bicycle rental systems, etc.

Manage crisis (UC3) means a process dealing with events that threaten the general public. When tourist activities in selected areas increase, responses might be comprised of: launching/establishing a special emergency call number, increasing the number of open/active/overnight pharmacies in selected areas, increasing the number of hospital emergency rooms, improving security and enforcement of regulations, etc.

4. Tourist Destination Questionnaires

A questionnaire is a form that contains a set of questions usually directed at statistically important tourist activities. A tourist questionnaire is a typical way to gather information, which can be used for managing a context-aware urban ecosystem. A questionnaire for tourist movement in destinations is discussed now, to clarify how smart systems based on recognizing tourist activities work. In the
work of Sirakaya et al. [37], not only current research trends for leisure, recreation, and tourism were surveyed, but there were also numerous questionnaires included.

Lisbon, the capital city of Portugal, as well as its surroundings, are considered and used as an example. Tourists/visitors stay in Lisbon and, probably day by day, visit its monuments and various tourist attractions. However, some tourists during their stay in the city may wish to visit its surroundings, e.g., Fátima (religious reasons) or Cascais (recreational reasons), as well as Sintra, which is known for historical and architectural monuments and is classified as a UNESCO World Heritage Site; see Figure 3. All of these places/sub-destinations, except Fátima, are located in the Grande Lisboa subregion (see http://en.wikipedia.org/wiki/Grande_Lisboa).

Figure 3. Lisbon and close/distance surroundings; see also [34]. (the base map is from Google Maps.)

A sample and common questions for visitors are shown in Table 1. There are also many other tourist questionnaires available, for example [38,39]. These questionnaires are distributed to visitors during their stay at a destination. They refer to many details of the visitors’ trip and stay. Forms are usually designed by tourism organizations for people who are going to spend at least one night at the destination. Questionnaires are conducted anonymously.

One of the main objectives of the questionnaires is to know more about visitor characteristics for marketing purposes, as well as to identify the size of the tourism activity. Other characteristics cover types of visitors (foreign or home, business or leisure, overnight or day trip). They also allow us to identify where visitors, if any, go outside the examined basic destination, and what is the scale of sub-destination visits.

The purpose of this paper is also to provide methods of gathering information about tourist movements automatically, that is to replace manual surveys with a fully-automated process, and then
use this information for a smart urban ecosystem. It should be noted that the typical granulation for a BTS is about 500 m in a city (urban areas) and about 1000 m outside a city. On the other hand, there are some advanced algorithms and models [11] to enable the estimation of a phone position between stations with an accuracy of about 150 m in urban areas. Let us also note that the Home Location Register (HLR) is maintained in mobile networks in order to provide information about subscribers who are registered in a core/local network. The Visitor Location Register (VLR) is the opposite, in that it provides information about network visitors (outside/country or foreign). These two records are important for the approach, since they allow us to find out who is a visitor and who is not. Although there are some exceptions, the probability of correct verification based on VLRs/HLRs is very high. We will use these methods, and will refer to them, in our system; see Line 17 in Algorithm 1. In case of any difficulties or doubts, the billing databases of mobile providers might additionally be examined.

| Questions | Answers |
|-----------|---------|
| 1. When did you come to Lisbon and what day of your stay is it today? | ... |
| 2. Did you come to Lisbon directly from your residence? | Y/N |
| 3. How long are you going to stay in Lisbon? | ... |
| 4. Are you staying in Lisbon? | Y/N |
| 5. In what type of accommodation in Lisbon do you stay, if any? | hotel, hostel, etc. |
| 6. What were your main aims when selecting this destination? | business, culture, religion, etc. |
| 7. What means of transport did you use to get to Lisbon? | train, car, airplane, etc. |
| 8. Do you travel in a group? | Y/N |
| 9. Do you use local guides? | Y/N |
| 10. How many times have you been in Lisbon before? | ... |
| 11. How much money do you spend per person? | ... |
| 12. Which places outside Lisbon do you want to visit during your stay? | ... |
| 13. How do you find selected aspects of your visit (from 1 to 5)? | [aspects to evaluate] |
| 14. How do you find selected services in Lisbon (from 1 to 5)? | [services to evaluate] |
| 15. What are the most attractive places in Lisbon? | ... |
| 16. What sources of information did you consult before arrival? | [options to select] |
| 17. How would you like to spend your time during your next stay in Lisbon | ... |
| 18. Are you going to recommend Lisbon to your friends? | Y/N |
| 19. Are you going to come to Lisbon again? | Y/N |
| 20. Personal information about a respondent | ... |

The analysis of points/questions in Table 1 leads to the following taxonomy based on the information expected to be obtained from the BTS datasets, which constitutes an informally-expressed algorithm:

1. answers that are obviously easy to obtain, e.g., Point 1 or 3;
2. some answers are available through digging deeper, but direct analysis of the BTS data is still required, e.g., Point 2 and the VLR/HLR records;
3. a certain number of answers need a pattern analysis for individuals, e.g., the comparison of the locations during the day and night for Point 4, or less/limited mobility (business) and greater mobility (an active city exploration typical for tourists) for Point 6;
4. some answers require a pattern analysis for a group, if any, of visitors, in other words, a group of objects are examined, to see if they are moving together, e.g., the city exploration with a group of...
mobile phones/visitors for Point 8, or with a local (cf. VLR/HLR records) mobile phone of a local
guide for Point 9;
5. some points need additional (open) technologies to answer questions, e.g., OpenStreetMap
(OSM) (see: http://en.wikipedia.org/wiki/OpenStreetMap) to locate/identify selected objects
like airports or railway stations for Point 7, hotels/hostels for Point 5, museums/churches for
Point 6, or suburban areas (close or distant) for Point 12;
6. there are some answers that require historical data analysis, e.g., a previous presence in a
destination for Point 10;
7. some answers require access to commercial/bank data, e.g., credit/debit cards used in the
destination for Point 11;
8. several answers could be obtained while analyzing, for example, social networks, reservation
systems, or web vendors, e.g., sources of information for Point 16;
9. some answers could be obtained when web forms are sent directly to mobile phones, after the
visit in the destination is over, e.g., sources of information for Points 13–15, 20;
10. some points for which obtaining answers based on BTS datasets are impossible, or problematic,
e.g., Points 17–19;
11. last but not least, there is some information that could be extracted from the BTS data and
that is not usually the subject of any questionnaire (thus, no points in Table 1 are indicated
here), but it could be used to analyze other parameters of tourist activities, e.g., intensity of
call/sms/mms/web transmissions during the entire visit, or in particular places, and through
numerous valuable conclusions that follow.

The above classification is crucial and gives an idea of the foundations for the solutions and
methods proposed in the paper, that is how use information gathered in CDRs is treated as a base for
the pro-active decisions of an urban ecosystem. In other words, the above classification constitutes a
base for methods of building knowledge about tourist activities; see Line 8 in Algorithm 7. However,
this would be a topic for a separate research project. The purpose of this article is to provide a vision of
the prototype system that collects data, which will then be subjected to such in-depth analysis.

5. Multi-Agent System

A multi-agent system and its architecture is proposed in this section. The system is used to
solve the problem of surveying the tourist movement in a destination, in the way described in the
previous section.

The following taxonomy of the agent is proposed:

A – Angel-the-guard agent, an agent created for a new phone that appears within the entire monitored
area, but only when the object is classified as a visitor. From that moment, the agent exists in
a system until the object leaves the area. It stores the entire trace that refers to the particular
visitor. From a data flow point of view, this type of data is collected from agent B, and goes to
agent A, through agent X. Agent B calls out information about different visitors, and agent X
redirects these data to the proper agent A. When the phone leaves the monitored destination,
all information gathered is passed to agent Q, through agent P, and agent A is removed from
the system;

B – BTS agent, this agent is present in every BTS and gathers data related to all visitors monitored
by the system (not all pieces of data gathered in a BTS belong to visitors). Data are sent away to
the particular agent A, through agent X. After a successful sending process, the data are deleted
from the agent B repository;

X – eXchange agent, this agent is only responsible for redirecting and further transferring the messages
received from different B agents to proper A agents. Redirecting is performed after partly
decoding the message, including information about the phone number;

F – Facility agent is an agent that exists in the system permanently, the purpose of which is to
identify and process newly-detected tourist facilities. This kind of event means the geolocation
of an infrastructural object within the monitored area. The basic facilities result from the needs of a particular questionnaire and among them may be: airport, train station, hotel, office building, restaurant, cinema, theater, religious building, graveyard, etc. It is assumed that for identification purposes that different services are used, for example OpenStreetMap, which enables the identification of particular facilities (if there is such a need, tourist facilities can even be manually edited by an administrator for agent F, which may force introducing/deleting particular tourist facilities). Summing up, agent F keeps a list of all analyzed facilities within a monitored area;

P – event Processing agent, the agent analyzing all particular routes collected by agents A and, on the basis of tourist facilities identified by agent F, applying them to the routes gathered by particular A agents. In such a way, obtained routes are enriched with information concerning infrastructural objects, if such objects exist, along the route of a particular visitor. It is carried out by comparing the geolocation of an infrastructural object and the visitor’s route;

Q – Questionnaire agent is an agent existing in a system, the purpose of which is to update a questionnaire, or questionnaires, built in this destination. The questionnaire is updated when an object leaves the entire destination and its agent A is to be removed;

M – Managing agent is an agent existing in a system permanently, the purpose of which is to initiate global system variables, as well as to manage other agents.

Figure 4 presents the basic architecture of the proposed multi-agent system.

Figure 4. Basic architecture of the proposed agent system, or the agent relationship chart showing agents’ constructions (solid lines) and destructions (dashed lines).

The number of A agents in a system is equal to the number of visitors in a given destination. The number of B agents is equal to the number of BTSs within a monitored area. There is only one F agent in a destination, as well as only one P agent. Although only one X agent was presented, it is always possible to increase their number to improve the capacity of the system. A similar remark can concern agent P, as well.

A list of more important variables in a system includes:

- **BTSList**: list of all BTSs that cover the monitored area and enables the gathering of all data necessary to study the behaviors of visitors;
- **facilityList**: list of infrastructural objects within a monitored area. Those objects can be identified by different types of services, and the presence of phones in their surrounding area is registered as
a meaningful event. After completion of all data, namely the whole route of visitors with facilities, an analysis of traces of tourists’ presence, in terms of questionnaire questions, is performed;

- **visitors**: list of mobile phones within a monitored area that were selected for tracing, as they are identified as unknown, i.e., coming from, and registered in, another (far) area. It is worth mentioning that every phone from *visitors* has its agent A;

- **BTSPhoneList**: a variable defined locally in each BTS, which includes a list of phones classified to be monitored as visitors and remaining within range of a certain station. It is important to mention that the sum of all variables belonging to *BTSPhoneList*, thus from all BTSs, is equal to variable *visitors*.

Let us pay attention to sets *visitors* and *BTSPhoneList*, the elements of which are attributed to time, or with a time-assigned value. In practice, this means that for each phone in these sets, we can get the (last) value of the time in which the phone was observed. Thus, we have to refer to the well-known operations on sets, which are union $∪$ and difference $\setminus$. We redefine union $∪$ in this way, that if we have several of the same elements, but differently time-attributed, the element with the largest time attribute is selected for the final set. Difference $\setminus$ remains unchanged, that is certain elements of the set are deleted, regardless of their time attributes. We introduce another difference operation $\setminus_t$, which works just like the classic one, except for situations when an element to be removed has a higher time attribute than the one indicated for deletion; in this case, the element with the higher time attribute is not removed. Let us consider some examples to clarify these informal definitions. Say $t_1, t_2, t_3, t_4, t_5, \ldots$ means sequential time instances that are used to attribute elements of a set. Then, we have: $\{ph_{23_t_1}, ph_{25_t_2}\} ∪ \{ph_{20_t_1}, ph_{23_t_1}, ph_{25_t_2}\} = \{ph_{20_t_1}, ph_{23_t_1}, ph_{23_t_1}, ph_{25_t_2}\}$, $\{ph_{23_t_1}, ph_{25_t_2}\} \setminus \{ph_{25_t_1}\} = \{ph_{23_t_1}\}$, and $\{ph_{23_t_1}, ph_{25_t_2}\} \setminus_t \{ph_{25_t_1}\} = \{ph_{23_t_1}, ph_{25_t_2}\}$.

The structure of the most important messages in the system is shown in Table 2.

### Table 2. Structure of basic messages transferred within the system.

| Message Code | Structure |
|--------------|-----------|
| M2B          | phone + visit = [true,false] |
| B2M          | phone + logType = [login | logout] + visType = [visitor | resident | unknown] + BTSid |
| B2X          | phone + timestamp + geolocation |
| X2A          | timestamp + geolocation |
| A2P          | phone + 1{timestamp + geolocation} |
| P2Q          | phone + 1{timestamp + geolocation + (facility)} |

(timestamp $≡$ date + time, geolocation $≡$ longitude + latitude, facility $≡$ category + name)

(Symbols for the table: “$≡$” is defined as; “$=$” is equal to; “$+$” conjunction; “( )” is optional; “[ ]” the choice of one possibility; “$|$” separator of disjoint choice related to [ ]; “( )” iterator of possibilities with a given minimal number of occurrences. The meaning of particular elements is intuitive, whereby “timestamp” means a time stamp, a certain moment in time, when measurements were performed, “geolocation” is the latitude and longitude of the object, “facility” is an observed object of infrastructure.)

### 6. Methods and Algorithms

Several algorithms for handling the entire system are proposed in this section. They refer to the classification of agents defined in the previous section. (Incidentally, we resigned from defining the agents’ ports to which messages are sent, so as not to over-formalize. This does not result in ambiguity. We also use two well-known and equivalent notations for the substitution operation, that is $v := v \ast z$ and $v := \ast z$, for two arguments and a sample operand.)
Algorithm 1 The M agent operations (M-operations).

**Input:** message : B2M

**Output:** phoneVisit : M2B

1: procedure CHECKING1
2:   procedure LOGInOut
3:     if m.logType = login then
4:       critical visitors2 := \{m.phone\} end;
5:     visitors := \{m.phone\}
6:   else
7:     visitors := visitors \{m.phone\};
8:     critical visitors2 := \{m.phone\} end
9: end if
10: end procedure
11: loop
12:   m := get(message);
13:   if m = nil then delay(d1); continue
14: end if
15: if timeGtEq(m.phone, visitors \ visType2) then
16:   if m.visType = unknown then
17:     m.visType := verifyVisitor(m.phone);
18: end if
19: if m.visType = visitor then
20:   if m.phone \ visitors \\{visType\} then
21:     start A-operation(m.phone);
22:     visitors := \{m.phone\}; logInOut;
23:     send(B, m.BTSid, (m.phone, true));
24:   else
25:     refreshTime(m.phone, visitors);
26:     critical refreshTime(m.phone, visitors2) end; logInOut
27: end if
28: else
29:   visitors := visitors \\{m.phone\};
30:   critical set time to \infty\ for m.phone in visitors end;
31:   send(B, m.BTSid, (m.phone, false))
32: end if
33: else
34:   send(B, m.BTSid, (m.phone, false))
35: end if
36: end loop
37: end procedure
38: procedure CHECKING2
39: loop
40:   for every ph \in visitors2 do
41:     if ph in visitors2 longer than d0 then
42:       critical visitors2 := visitors2 \\{ph\} end;
43:     destruct A-operation(ph)
44: end for
45: end loop
46: end procedure
47: initiate BTSList;
48: visitors := ∅; visitors2 := ∅;
49: start agent F-operations, P-operations, Q-operations, X-operations;
50: start agent all B-operations according to BTSList;
51: start process checking1, checking2;
The entire system is initiated by agent M, whose operations are illustrated as Algorithm 1. The agent starts and initiates important system variables; see Line 48 and the following lines. There is a pre-defined set of BTSs BTSlist that covers the monitored destination. Variable visitors stores all phones that are located within a monitored area and are recognized as tourists, or more generally, visitors. Variable visitors2 consists of phones suspected of possibly leaving their destination, since it was observed that they logged out from a BTS. However, this logging out can be momentary, since a tourist will be immediately, or after some time, taken by another BTS, and in such cases, a phone will be transferred back to variable visitors. In the following lines of the algorithm, all agents are initiated. In the last line, two concurrent internal processes are called out. We have only one critical section in the algorithm since only one variable is protected, and we do not need to differentiate sections, otherwise we should use the following sample syntax: critical <sec_name>: <instructions> end. Thus, “<sec_name>:” is omitted here.

Process checking1 is responsible for control of log in and log out operations of all monitored phones, in all BTSs, that is objects recognized as visitors, to and from particular BTSs. Function timeGtEq in Line 15 checks if the phone (first argument) has a higher or equal time attribute than the element in the set (second argument). If the phone does not belong to the set, then timeGtEq returns true. In this way, we provide the processing of newer information about a phone, if at the same time some information is received. Procedure refreshTime in Line 25 updates the value of the time attribute, if the phone belongs to the set and if it has a lower attribute (we could also write the equivalent code: if (m.phone ∈ visitors) then visitors := ∪{m.phone} end if). The important instruction is in Line 17 because the called procedure enables determining, using VLR/HLR services, if an object is registered outside the monitored area. It will then be recognized as a visitor. In procedure verifyVisitor, other actions can also be taken to enable object classification, for example describing if it is located, informally speaking, a short or a long distance from the monitored area, thus if it is an apparent visitor or, for instance, a close neighbor commuting to work, or a real visitor. Although it is an important issue, it will be omitted now, since it is of a rather technical nature and there is the possibility to carry out a thorough analysis in the future.

The work of agent B, located in each BTS, was presented as Algorithm 2. The algorithm starts from Line 39. Then, three other processes are initiated. Process BTSlogin is responsible for controlling log in and log out operations of objects. In Line 3, an important procedure is called out. It verifies newly-appearing CDR records, which have the values visitor or unknown in the field visType, or the values login or logout in the field logType. Later on, m1 is sent to agent M in order to verify and compare it with variable visitors. Process BTSvisiting enables denoting that phone was recognized as a visitor. Then, this information is introduced to CDRs; see Line 23. This is a result of the requirement to not re-examine records, in the BTS database, which were identified as belonging to a visitor. In other words, once identified, they do not change their status. Process phoneLocation determines the current geolocation of every tourist’s phone that is within range of a BTS; see Line 34. The process of geolocation, where neighboring BTSs are included, is a separate procedure, which works according to the well-known rules [11].

The task of agent X (see Algorithm 3) is to redirect all messages from the B agents and then send them to the proper A agents.

The received message has an encoded phone number, which becomes deleted, and the remaining part of the message is sent to the proper A agent.
Algorithm 2 The B agent operations (B-operations).

Input: CDRecords : B2M; \(\triangleright\) collected in a BTS
Input: m2 : M2B; \(\triangleright\) message from agent M
Output: m1 : B2M; \(\triangleright\) message to agent M
Output: m3 : B2X; \(\triangleright\) message to agent X

1: procedure BTSLOGIN
2: \hspace{0.5em} loop
3: \hspace{1.5em} m1 := get(CDRecords); \(\triangleright\) only new, omitting visType = resident
4: \hspace{1.5em} if (m1 = nil) then delay(d2) \(\triangleright\) d2 established by admin
5: \hspace{1.5em} else
6: \hspace{2.5em} send(M, m1); \(\triangleright\) visType = visitor or visType = unknown
7: \hspace{2.5em} if m1.visType = visitor then
8: \hspace{3.5em} if m1.logType = login then
9: \hspace{4.5em} critical BTSPhoneList := \cup\{m1.phone\} end
10: \hspace{3.5em} else
11: \hspace{4.5em} critical BTSPhoneList := \\{m1.phone\} end
12: \hspace{2.5em} end if
13: \hspace{2em} end if
14: \hspace{1em} end if
15: \hspace{0.5em} end loop
16: end procedure

17: procedure BTSVISITING
18: \hspace{0.5em} loop
19: \hspace{1.5em} m := get(m2); \(\triangleright\) from agent M
20: \hspace{1.5em} if (m = nil) then delay(d3) \(\triangleright\) d3 established by admin
21: \hspace{1.5em} else
22: \hspace{2.5em} if m.visit = true then
23: \hspace{3.5em} denote CDRs for m.phone as visType = visitor;
24: \hspace{3.5em} critical BTSPhoneList := \cup\{m.phone\} end
25: \hspace{2.5em} else
26: \hspace{3.5em} critical BTSPhoneList := \\{m.phone\} end
27: \hspace{2em} end if
28: \hspace{1em} end if
29: \hspace{0.5em} end loop
30: end procedure

31: procedure PHONELOCATION
32: \hspace{0.5em} loop
33: \hspace{1.5em} for each ph in BTSPhoneList do
34: \hspace{2.5em} m3 := geolocation(ph); send(X, m3); \(\triangleright\) to agent X
35: \hspace{2em} end for
36: \hspace{1em} delay(d4) \(\triangleright\) d4 established by admin
37: \hspace{0.5em} end loop
38: end procedure

39: BTSPhoneList := \{\}; \(\triangleright\) visitors in BTS area

40: start process BTSlogin, BTSvisiting, phoneLocation; \(\triangleright\) inner processes
Algorithm 3 The X agent operations (X-operations)

Input: \( m_1 : B2X \); \( m_2 : X2A \)
Output: \( m_2 : X2A \)

1: \( \text{loop} \)
2: \( \text{get}(m_1); \)
3: \( \text{if } m_1 = \text{nil} \text{ then delay}(d5) \)
4: \( \text{else} \)
5: \( m_2.\text{timestamp} := m_1.\text{timestamp}; \)
6: \( m_2.\text{geolocation} := m_1.\text{geolocation}; \)
7: \( \text{send}(A, m_1.\text{phone}, m_2); \)
8: \( \text{end if} \)
9: \( \text{end loop} \)

The process of agent A is visualized as Algorithm 4.

Algorithm 4 The A agent operations (A-operations).

Input: \( \text{trace} : X2A \);
Output: \( \text{entireTrace} : A2P \)

1: \( \text{destructor A} \)
2: \( \text{send}(P, \text{entireTrace}); \)
3: \( \text{end destructor} \)
4: \( \text{entireTrace} := \emptyset; \)
5: \( \text{loop} \)
6: \( \text{get}(\text{trace}); \)
7: \( \text{if } \text{trace} = \text{nil} \text{ then delay}(d6) \)
8: \( \text{else} \)
9: \( \text{add} \text{trace} \text{to} \text{entireTrace} \)
10: \( \text{end if} \)
11: \( \text{end loop} \)

The first instruction is located in Line 4. The main task of the agent is to collect the registered geolocations, which will constitute the entire route of a particular phone. Data arrive from different B agents, via agent X. It is necessary to point out that agent A has its own destructor, whose task is to send all gathered data about the route to agent P, where it becomes subjected to further processing. After using the constructor’s method, the agent itself turns into a singular state, and all resources allocated to it are finally removed by calling a process, in this case agent M; see Line 42 in Algorithm 1.

Agent F (see Algorithm 5) is used to build a list of infrastructural objects that exist within the monitored area.

Algorithm 5 The F agent operations (F-operations).

Input: services like OpenStreetMap
Output: facilityList

1: \( \text{initiate} \text{facilityList}; \)
2: \( \text{loop} \)
3: \( \text{obj} := \text{scan OpenStreetMap for objects}; \)
4: \( \text{update} \text{facilityList} \text{using} \text{obj}; \)
5: \( \text{....} \)
6: \( \text{if} \text{required} \text{facilityList then} \)
7: \( \text{send}(P, \text{facilityList}) \)
8: \( \text{end if} \)
9: \( \text{end loop} \)

Those objects will later be taken into consideration when building questionnaires. The following two basic cases will be analyzed: a phone is located nearby such an object or this object was recorded
directly in the tourist’s route. As previously mentioned, the algorithm, that is agent F, is dealing with only one issue, which is updating and keeping the list of objects within the monitored area valid. It is true that infrastructural objects do not change very often; however, it seems that it is beneficial to have a separate and specialized agent responsible for this type of task.

Enrichment of the tourist’s route of infrastructural objects located along the phone trail is performed by agent P; see Algorithm 6.

Algorithm 6 The P agent operations (P-operations).

Input: 
- trace1 : A&P
- facilityList
Output: 
- trace2 : P&Q

1: loop
2: get(trace1); ▷ from agent A
3: if trace1 = nil then delay(d7) ▷ d7 established by admin
4: else
5: require(F, facilityList); ▷ fresh list from agent F
6: trace2 := enrich(trace1, facilityList); ▷ add facilities
7: send(Q, trace2); ▷ to agent Q
8: end if
9: end loop

In Line 5, the list of infrastructural objects is downloaded. This list is built and updated by agent F. It is supposed that there is a possibility that some changes in the facility list were made after the previous inquiry, so it is necessary to download it once again. In Line 6, there is a direct verification of the route and enrichment of its description by infrastructural objects located along the route or nearby. This problem itself is a separate topic, since it is a procedure that does not follow general rules. For instance, there is the question of how much time does the visitor have to stay nearby in order to record his/her presence, or, equivalently, to omit accidental situations. However, this topic exceeds the main aim of this work.

Agent Q, after gathering all necessary data about the tourist’s route, updates all questionnaires on the basis of tourist’s data; see Algorithm 7.

Algorithm 7 The Q agent operations (Q-operations).

Input: 
- track : P&Q ▷ full tracking with facilities (from agent P)
Output: 
- Que1..n ▷ some questionnaires

1: initialize Qei from i = 1 to n ▷ initialize every Que
2: loop
3: get(track); ▷ d7 established by admin
4: if track = nil then delay(d7)
5: else
6: store track in a global repository; ▷ assume we have a repository
7: for i = 1..n do
8: update(Qei, track); ▷ add new data
9: end for
10: end if
11: end loop

It is assumed that, generally, many different questionnaires can be prepared; however, at least one is necessary. Line 8 shows the updating process for a questionnaire. This type of updating can be the topic of a separate project and research study, where numerous detailed questions connected with the procedure of filling in the questionnaire have to be answered. Some of those questions can be simple, but others are much more complex; see Section 4.
As a form of summary, see Figure 5, which shows basic data and message flows in the system, that is in all algorithms presented above.

![Figure 5. Basic data flows in the system (only single instances of B and A processes are shown).](image)

Let us note that the entire system is initiated with the following basic set of agents: M, Bs, X, F, P, and Q. When no visitor is recognized, then no agent A is created.

The agents presented in this section, as well as their algorithms deal with all important aspects of the subject. When presenting the agents, some attention was paid to a few minor problems. However, in order to avoid getting too deep into unnecessary details, those problems can be solved in the future. Some other minor problems also exist, which were deliberately omitted in order to avoid overcomplicating the content. An example of such a case may be a situation when an object classified as a visitor stayed too long within a monitored area, becoming, in fact, its resident. There may be different reasons for such a situation to occur, and it can be solved by introducing a maximal time limit a person can stay within the monitored area.

7. Evaluation and Experiments

We provide both theoretically- and practically-oriented considerations to show the entire characteristic of the proposed approach.

7.1. Evaluation

Finding time complexity, which signifies the total time required by algorithms to run, is important for any algorithm. Please find the following statements below.

**Algorithm 1 always terminates.** Proof: The algorithm does not contain recursions. All instructions are precisely defined. The infinite loop symbolizes the readiness for constant processing and can be changed into a loop with an exit condition or the break instruction when required. The “for” loop is performed strictly and a prescribed number of times. There is only one critical section that protects the variable \( \text{visitors2} \). The lock mechanism occurs in different and separate lines in process checking1 and only once in process checking2. If requests appear from both processes at the same time, they will be queued. Thus, the algorithm is deadlock free. If there are no data, the reception operation does not stop processing. Then, the random delay time \( (d1) \) is set for the next reception; see Line 13. All data sending operations (see Lines 23, 31, 33, as well as lines 21 and 42) are asynchronous.

**Algorithm 1 has \( \Theta(m) \) complexity, where \( m \) is a total number of monitored phones.** Proof: The algorithm has a dominant loop running through \( m \) phones, observed as visitors in the entire destination. The “for” loop is performed a limited number of times. The other instructions have fixed costs. Finally, the average value is \( \Theta(m) \).

**Algorithm 2 always terminates.** Proof: The algorithm does not contain recursions. All instructions are precisely defined. The infinite loop symbolizes the readiness for constant processing. The “for” loop is performed a prescribed number of times. There is only one critical section for the variable BTSPhoneList. The lock mechanism occurs in processes BTSlogin and BTSvisiting. If requests appear
from both processes at the same time, they will be queued. Thus, the algorithm is deadlock free. A lack of data does not stop processing; see Lines 4 and 20. The sending operations are asynchronous; see Lines 6 and 34. Algorithm 2 has \( \Theta(b) \) complexity, where \( b \) is a number of monitored phones in a BTS. Proof: The algorithm has a dominant loop running through \( b \) phones observed as visitors in a BTS. The same holds for the loop “for”. The other instructions have fixed costs. Finally, the average value is \( \Theta(b) \).

Algorithm 3 always terminates. Proof: the algorithm does not contain recursions. All instructions are precisely defined. The infinite loop symbolizes the readiness for constant processing. A lack of data does not stop the reception operation. The sending operation is asynchronous. Algorithm 3 has \( \Theta(1) \) complexity. Proof: The procedure only transfers data. All instructions have fixed costs.

Algorithm 4 always terminates. Proof: The algorithm does not contain recursions. All instructions are precisely defined. The infinite loop symbolizes the readiness for constant processing. A lack of data does not stop the reception operation. The sending operation is asynchronous. Algorithm 4 has \( \Theta(t) \) complexity, where \( t \) is an average number of elements of which the typical track consists. Proof: The algorithm has a dominant loop running through \( t \) elements of an entire trace of a phone. Finally, the average value is \( \Theta(t) \).

Algorithm 5 always terminates. Proof: the algorithm does not contain recursions. As it previously mentioned, the instructions in Lines 3 and 4 only symbolize concurrent gathering and updating facilityList. In fact, data gathering should be organized as a background processing. The infinite loop symbolizes the readiness for constant process. The sending operation is asynchronous. Algorithm 5 has \( \Theta(o) \) complexity, where \( o \) is a number of facilities registered in a monitored area. Proof: The algorithm processes all facilities \( o \) registered in the monitored area.

Algorithm 6 always terminates. Proof: The algorithm does not contain recursions. All instructions are precisely defined. The infinite loop symbolizes the readiness for constant processing. A lack of data does not stop the reception operation. The operation in Line 5 is a procedure call, but we assume that facilityList is always available. The sending operation is asynchronous. Algorithm 6 has \( \Theta(t) \) complexity, where \( t \) is an average number of elements of which the track consists. Proof: The entire trace, which is browsed, of a phone consists of \( t \) elements.

Algorithm 7 always terminates. Proof: the algorithm does not contain recursions. All instructions are precisely defined. The “for” loop is performed a prescribed number of times. The infinite loop symbolizes the readiness for constant processing. A lack of data does not stop the reception operation. We assume that the update procedure (see Line 8) always terminates; on the other hand, it could be organized as a background process. Algorithm 7 has \( \Theta(m \cdot t) \) complexity, where \( m \) is a number of phones, and \( t \) is the average number of elements in a track, which are required for a questionnaire analysis. Proof: The algorithm has a dominant loop running through \( m \) phones, observed as visitors in the entire destination. We process each questionnaire with an average cost \( t \), that is it depends on the number of elements in a track. Finally, the average value is \( \Theta(m \cdot t) \).

We are also interested in the whole system, understood as a group of cooperating agents and their algorithms. The entire system has \( \Theta(m \cdot t) \) complexity, where \( m \) is a number of phones and \( t \) is the average number of elements of a typical trace. Proof: We use the parenthesis structure to show the calling of algorithms/processes in the entire system:

\[
A1(\ldots, A5, A6, A7, A3, A2 \cdots A2, \ldots, A4 \cdots A4, \ldots)
\]

where letter “A” is related to the particular algorithm and the bottom dots specify other instructions outside the calling point, while the middle dots specify multiple instances. Using the previous statements and proofs, we get the following results. A5, in practice, has fixed costs \( \Theta(1) \), due to the fixed number of facilities. A6 provides \( \Theta(t) \). A7 provides \( \Theta(m \cdot t) \). A3 has fixed costs \( \Theta(1) \). All A2 provide \( \Theta(b) \), due to concurrent performance. All A4 provide \( \Theta(t) \), due to concurrent performance. Thus, we obtain \( A1(\ldots, \Theta(1), \Theta(t), \Theta(m \cdot t), \Theta(1), \Theta(b), \ldots, \Theta(t), \ldots) \). A1 provides \( \Theta(m) \), and \( \Theta(m \cdot t) \) dominates in the parenthesis structure. Finally, the complexity of the proposed system is \( \Theta(m \cdot t) \).
7.2. Experiments

Programming experiments were also carried out, with the aim of verifying the proposed system. They are of a limited range, mainly related to the figure of 10,000 (ten thousand) visitors, or tourists, but also to the fact that the works were carried out with random data, which were also connected to the calculation of geolocation. About 100 different events (arriving at facilities) have been generated for each visitor. The experiments enable the verification of the system’s possibilities and present its working vision and the future target version. Another important role is the illustration of the system’s functions.

Figure 4 presents the architecture of the multi-agent system; see also Table 2. In our experiments, messages were transported using the Kafka [40,41] platform. It is an efficient message broker platform, which enables different types of data streams to be sent. The experiments were performed within a chosen area of the system, that is the busiest one, in relation to BTSs. The experiments were limited to the following three types of messages:

1. messages initiating the creation of agent A;
2. messages connected with a mobile phone logging in and out to and from a BTS;
3. messages connected with a phone location; great attention needs to be paid to a large number of such messages, which results from the fact that every BTS regularly calculates geolocation, separately for each phone logged into a given station, and those messages are later sent to agents A.

Figure 6 presents the topics of the most important elements, that is in the busiest part of the system.

Agent B located in each station can send all types of messages. Kafka, once again, proved itself to be an efficient means of transmission for huge message streams, as was already mentioned in the work [42], Section VI.B, also when the particular system elements were dispersed in a computing cloud ([42], Figure 6), which mirrors the real conditions very well.

Data processing possibilities will also be presented as an example of analysis when it comes to the particular questions in a questionnaire. Generally speaking, data analysis of that kind is a challenging task and should be solved separately. For that reason, only a few simpler, but quite interesting cases will be presented. The way of coding the particular problems is illustrated, as well. First of all, attention needs to be paid to Listing 1 including a fragment of trace, or route, registered in relation to the particular tourist. The limited size of this work enables presenting a listing that only shows parts of such a trace, in reality covering numerous positions together with registered events, but, above all, the routing progress within the city.
Listing 1: Fragment of registered trace for a particular tourist (dots mean leaving positions from the record).

```
phone=3249, 22.12.2018, 19.01, 6.0, 5.0, airport, Balice Airport;
phone=3249, 22.12.2018, 19.02, 6.0, 6.0;
phone=3249, 22.12.2018, 19.03, 7.0, 6.5;
.....
phone=3249, 22.12.2018, 23.03, 9.0, 9.5, hotel, Cracovia Hotel;
phone=3249, 22.12.2018, 23.04, 9.0, 9.5, hotel, Cracovia Hotel;
phone=3249, 22.12.2018, 23.05, 9.0, 9.5, hotel, Cracovia Hotel;
.....
phone=3249, 23.12.2018, 08.01, 9.0, 9.5, hotel, Cracovia Hotel;
.....
phone=3249, 23.12.2018, 10.05, 11.0, 8.5, monument, Wawel Castel;
phone=3249, 23.12.2018, 10.06, 11.0, 8.5, monument, Wawel Castel;
phone=3249, 23.12.2018, 10.07, 11.0, 8.5, monument, Wawel Castel;
.....
phone=3249, 23.12.2018, 14.40, 8.0, 10.5, museum, National Museum;
phone=3249, 23.12.2018, 14.41, 8.0, 10.5, museum, National Museum;
phone=3249, 23.12.2018, 14.42, 8.0, 10.5, museum, National Museum;
.....
phone=3249, 23.12.2018, 18.29, 9.5, 9.0;
phone=3249, 23.12.2018, 18.30, 9.5, 9.5, club, Karlik;
phone=3249, 23.12.2018, 18.31, 9.5, 9.5, club, Karlik;
.....
phone=3249, 24.12.2018, 08.01, 6.0, 5.0, airport, Balice Airport;
```

Listing 2 includes a fragment of code, which presents an analysis of the means of tourist arrival. We assume by default that he/she arrives by car. However, we later investigate a certain number of initial events appearing in the registered visitor’s trace. If another means of transport appears, for instance airport, bus station, or train station, this place is assumed to be the correct one. It is, of course, a disputable issue if we should examine a certain limited number of initial events, or rather concentrate on events within a certain time limit, from a first phone logging in within a monitored area. We have chosen one solution, but other approaches are also possible. Listing 3 includes fragments of code, which present the results of investigation if a visitor visited museums while staying in the city. If a tourist visited two or more museums, this kind of behavior is interpreted as one of the goals of the trip. There are, of course, different ways of analyzing this issue, for example a tourist could arrive to the city and visit only one museum, maybe even to see one particular painting, and such visiting cases also have to be taken into consideration. In such cases, it seems to be a reasonable solution to check the length of their stay in a museum, which can be determined on the basis of the tourist’s trace. It is an open topic, and our aim here is to present the means of data processing, as well as proving its feasibility and reachability. It seems that our goal was fully achieved.
Listing 2: Way of reaching the city.

```java
questionAnswer = "car";
len = length(processedTrace);
if (len > 1000){
    len = 1000;
}
for (int i = 1, len, i++){
    processedT = processedTrace[i];
    if ((processedT.category == "airport") ||
        (processedT.category == "train_station") ||
        (processedT.category == "bus_station")){
        break;
    }
}
if (processedT.category != ""){
    questionAnswer = processedT.category;
}
```

Listing 3: Visiting museums.

```java
monumentsVisited = 0;
questionAnswer = false;
for (int i = 1, length(processedTrace), i++){
    processedT = processedTrace[i];
    if ((processedT.facilityType == "monument") ||
        (processedT.facilityType == "museum")){
        monumentsVisited++
    }
}
questionAnswer = monumentsVisited > 1;
```

Figure 7 presents the results of analysis, of the whole verified population of visitors, when it comes to the way of reaching the city. A question about this aspect of staying in the city is one of the first questions from a questionnaire presented in Section 4, and most probably, it would be present in every other analyzed questionnaire. Figure 8 illustrates the ways that visitors spend their time during their stay in the city. These are two important criteria that are not mutually exclusive. Practically, this means that people who like visiting different museums can also take part in numerous club parties organized in the city.

![Figure 7. The way tourists arrive to the monitored city.](image-url)
In the described experiment, a similar analysis was carried out on another, potentially interesting question. The examination is, in fact, checking gathered data related to each visitor. These data have a form of data stream gathered by agent A and later enriched by agent P; see data stream P2Q in Table 2. Content reasoning can be carried out on the basis of this kind of material. The issue of the precise definition of the investigated events always remains. An example can be a dilemma: should visiting museums be connected with a number of visited places or should another criteria be taken into consideration. It needs to be mentioned that these are problems of a different nature, and our goal was to design an IT system that can gather a huge amount of data for further proper analysis.

![Figure 8. The way of spending time in the city or visiting museums and parties.](image)

Some of the questions related to questionnaires are easy to answer, the others require deep analysis; however, this was already discussed in this work; see informal considerations in Section 4.

8. Conclusions

The paper presents a novel method for mining the individual behaviors of visitors in a destination from pervasive BTS-based datasets. The questionnaire behavior (see Table 1) gives an informal idea of how our system works. The system is authenticated through the introduced architecture of a multi-agent system, through the proposed algorithms, as well as conducted experiments.

The presented system opens up real possibilities of the implementation of proper software, which would be essential in the process of tourist traffic evaluation within the monitored area, understanding its specifics, advantages and disadvantages, supporting municipal services, etc. Some difficulties related to the system were discussed in this work; however, these are rather minor problems of a technical nature. After making proper decisions, they can all be easily solved.

The proposed system enables the gathering of huge amounts of data; however, as previously mentioned, the main issue is to fill in questionnaires on the basis of possessed data collections. This problem can be solved as well, as presented in the work, but it should be treated as a separate project.

Another issue is access to data gathered in BTSs. There may be legal issues related to privacy and the sensitivity of personal data. However, these data can be anonymized, since our main interest is collective behaviors, for example the behaviors of tourists, considered as individuals, visiting a given area, not individual ones carrying information about individual preferences and privacy. Data gathered in BTS networks is the most widespread type of data, due to the prevalence of mobile phones, and cannot be replaced by any other widely-available phone applications. Last, but not least, let us pay attention to the sentence of the U.S. Supreme Court concerning the records of numbers called not being protected by the Constitution, but practically, there is limited protection, delegated to law-making power to acts of lower bodies (see [https://en.wikipedia.org/wiki/Smith_v._Maryland](https://en.wikipedia.org/wiki/Smith_v._Maryland)). The problems presented above illustrate the future work directions very well, which should include more detailed algorithms and other mentioned procedures. It is also worth noting that our approach could also be extended to hybrid data sources, that is combining CDR-data and sensor data; see [42].
Funding: This research received no external funding.

Acknowledgments: I thank my students Tomasz Borowicz and Krzysztof Świder (AGHUniversity of Science and Technology) for their help with the experimental part of this research.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

- BTS: Base Transceiver Station
- CDR: Call Detail Record
- GPS: Global Positioning System
- GIS: Geographic Information Systems
- GSM: Global System for Mobile Communications
- UTMS: Universal Mobile Telecommunications System
- HLR: Home Location Register
- VLR: Visitor Location Register
- OSM: OpenStreetMap

References

1. Şahin, F.; Yan, Z. Mobile Phones in Data Collection: A Systematic Review. *Int. J. Cyber Behav. Psychol. Learn.* 2013, 3, 67–87. [CrossRef]
2. Bengtsson, L.; Lu, X.; Thorson, A.; Garfield, R.; von Schreeb, J. Improved Response to Disasters and Outbreaks by Tracking Population Movements with Mobile Phone Network Data: A Post-Earthquake Geospatial Study in Haiti. *PLoS Med.* 2011, 8, e1001083. [CrossRef] [PubMed]
3. Jones, K.H.; Daniels, H.; Heys, S.; Ford, D.V. Challenges and Potential Opportunities of Mobile Phone Call Detail Records in Health Research: Review. *JMIReHealth uHealth* 2018, 6. [CrossRef] [PubMed]
4. Slim, S.; Ahas, R.; Mooses, V. Are younger age groups less segregated? Measuring ethnic segregation in activity spaces using mobile phone data. *J. Ethn. Migr. Stud.* 2018, 44, 1797–1817. [CrossRef]
5. Jiang, D.; Huo, L.; Song, H. Rethinking Behaviors and Activities of Base Stations in Mobile Cellular Networks Based on Big Data Analysis. *IEEE Trans. Netw. Sci. Eng.* 2018. [CrossRef]
6. Ma, Q.; Wang, W.; Yao, Q.; Zhou, J.; Quo, L. Factor analysis on call detail record. In Proceedings of the 2018 27th Wireless and Optical Communication Conference (WOCC 2018), Hualien, Taiwan, 30 April–1 May 2018; pp. 1–5.
7. Geepalla, E.; Abuhamoud, N.; Abouda, A. Analysis of Call Detail Records for Understanding Users Behavior and Anomaly Detection Using Neo4j. *Adv. Intel1. Syst. Comput.* 2018, 753, 74–83. [CrossRef]
8. Ratti, C.; Frenchman, D.; Pulselli, R.M.; Williams, S. Mobile Landscapes: Using Location Data from Cell Phones for Urban Analysis. *Environ. Plan. B Plan. Des.* 2006, 33, 727–748. [CrossRef]
9. Reades, J.; Calabrese, F.; Sevtsuk, A.; Ratti, C. Cellular Census: Explorations in Urban Data Collection. *IEEE Pervasive Comput.* 2007, 6, 30–38. [CrossRef]
10. Isaacman, S.; Becker, R.; Cáceres, R.; Kobourov, S.; Martonosi, M.; Rowland, J.; Varshavsky, A. Identifying Important Places in People’s Lives from Cellular Network Data. In *Pervasive Computing*; Lecture Notes in Computer Science; Lyons, K., Hightower, J., Huang, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6696, pp. 133–151.
11. Calabrese, F.; Colonna, M.; Lovisolo, P.; Parata, D.; Ratti, C. Real-Time Urban Monitoring Using Cell Phones: A Case Study in Rome. *IEEE Trans. Intell. Transp. Syst.* 2011, 12, 141–151. [CrossRef]
12. Gonzalez, M.C.; Hidalgo, C.A.; Barabasi, A.L. Understanding individual human mobility patterns. *Nature* 2008, 453, 779–782. [CrossRef]
13. Zhao, Z.; Shaw, S.L.; Xu, Y.; Lu, F.; Chen, J.; Yin, L. Understanding the bias of call detail records in human mobility research. *Int. J. Geogr. Inf. Sci.* 2016, 30, 1738–1762. [CrossRef]
14. Järv, O.; Ahas, R.; Witlox, F. Understanding monthly variability in human activity spaces: A twelve-month study using mobile phone call detail records. *Transp. Res. Part C Emerg. Technol.* 2014, 38, 122–135. [CrossRef]
15. Dong, H.; Wu, M.; Ding, X.; Chu, L.; Jia, L.; Qin, Y.; Zhou, X. Traffic zone division based on big data from mobile phone base stations. *Transp. Res. Part C Emerg. Technol.* 2015, 58, 278–291. [CrossRef]

16. Steenbruggen, J.; Tranos, E.; Nijkamp, P. Data from mobile phone operators: A tool for smarter cities? *Telecommun. Policy* 2015, 39, 335–346. [CrossRef]

17. Becker, R.; Cáceres, R.; Hanson, K.; Isaacman, S.; Loh, J.M.; Martonosi, M.; Rowland, J.; Urbanek, S.; Varshavsky, A.; Volinsky, C. Human Mobility Characterization from Cellular Network Data. *Commun. ACM* 2013, 56, 74–82. [CrossRef]

18. Qin, S.; Man, J.; Wang, X.; Li, C.; Dong, H.; Ge, X. Applying Big Data Analytics to Monitor Tourist Flow for the Scenic Area Operation Management. *Discr. Dyn. Nat. Soc.* 2019, 1–11. [CrossRef]

19. Lwin, K.K.; Sekimoto, Y.; Takeuchi, W. Estimation of Hourly Link Population and Flow Directions from Mobile CDR. *ISPRS Int. J. Geo-Inf.* 2018, 7, 449. [CrossRef]

20. Thuillier, E.; Moalic, L.; Lamrous, S.; Caminada, A. Clustering Weekly Patterns of Human Mobility Through Mobile Phone Data. *IEEE Trans. Mobile Comput.* 2018, 17, 817–830. [CrossRef]

21. Huang, J.; Xiao, M. State of the art on road traffic sensing and learning based on mobile user network log data. *Neurocomputing* 2018, 278, 110–118. [CrossRef]

22. Li, J.; Xu, L.; Tang, L.; Wang, S.; Li, L. Big data in tourism research: A literature review. *Tour. Manag.* 2018, 68, 301–323. [CrossRef]

23. Ferrari, L.; Mamei, M.; Colonna, M. Discovering events in the city via mobile network analysis. *J. Ambient Intell. Humaniz. Comput.* 2014, 5, 265–277. [CrossRef]

24. Ahas, R.; Aasa, A.; Roose, A.; Mark, Ü.; Silm, S. Evaluating passive mobile positioning data for tourism surveys: An Estonian case study. *Tour. Manag.* 2008, 29, 469–486. [CrossRef]

25. Al-Ayyoub, M.; Jararweh, Y.; Daraghmeh, M.; Althebyan, Q. Multi-agent based dynamic resource provisioning and monitoring for cloud computing systems infrastructure. *Cluster Comput.* 2015, 18, 919–932. [CrossRef]

26. Al-Ridhawi, I.; Kotb, Y.; Aloqaily, M.; Kantarcì, B. A probabilistic process learning approach for service composition in cloud networks. In Proceedings of the IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, ON, Canada, 30 April–3 May 2017; pp. 1–6. [CrossRef]

27. Al-Ridhawi, I.; Kotb, Y.; Aloqaily, M.; Kantarcì, B. A probabilistic process learning approach for service composition in cloud networks. In Proceedings of the 12th International Conference on Research Challenges in Information Science (RCIS), Nantes, France, 29–31 May 2018; pp. 1–6. [CrossRef]

28. Al-Ayyoub, M.; Jararweh, Y.; Daraghmeh, M.; Althebyan, Q. Multi-agent based dynamic resource provisioning and monitoring for cloud computing systems infrastructure. *Cluster Comput.* 2015, 18, 919–932. [CrossRef]

29. Baker, T.; Rana, O.F.; Calinescu, R.; Tolosana-Calsanz, R.; Bañares, J.Á. Towards Autonomic Cloud Services Engineering via Intention Workflow Model. In *Economics of Grids, Clouds, Systems, and Services*; Altmann, J., Vanmechelen, K., Rana, O.F., Eds.; Springer International Publishing: Cham, Switzerland, 2013; pp. 212–227.

30. Klimek, R. Pattern-based and Composition-driven Automatic Generation of Logical Specifications for Workflow-oriented Software Models. *J. Logical Algebraic Methods Program.* 2019, 104, 201–226. [CrossRef]

31. Maamar, Z.; Faci, N.; Boukadi, K.; Ugljanin, E.; Sellami, M.; Baker, T.; Angarita, R. How to agentify the Internet-of-Things? In Proceedings of the 12th International Conference on Research Challenges in Information Science (RCIS), Nantes, France, 29–31 May 2018; pp. 1–6. [CrossRef]

32. Al-Ridhawi, I.; Kotb, Y.; Aloqaily, M.; Kantarcì, B. A probabilistic process learning approach for service composition in cloud networks. In Proceedings of the 12th International Conference on Research Challenges in Information Science (RCIS), Nantes, France, 29–31 May 2018; pp. 1–6. [CrossRef]

33. Kwan, J.; Gangat, Y.; Payet, D.; Courdier, R. An Agentified Use of the Internet of Things. In Proceedings of the IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Chengdu, China, 15–18 December 2016; pp. 311–316. [CrossRef]

34. Blondel, V.D.; Decuyper, A.; Klings, G. A survey of results on mobile phone datasets analysis. *EPJ Data Sci.* 2015, 4. [CrossRef]

35. Klimek, R. Mapping population and mobile pervasive datasets into individual behaviors for urban ecosystems. In Proceedings of 15th International Conference on Artificial Intelligence and Soft Computing (ICAISC 2016), Zakopane, Poland, 12–16 June 2016; Lecture Notes in Artificial Intelligence; Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M., Eds.; Springer: Cham, Switzerland, 2016; Volume 9692, pp. 683–694. [CrossRef]
36. Federation of Communication Services. *UK Standard for CDRs; Standard CDR Format*, The Federation Of Communication Services Ltd: Beckenham, UK, January 2014.

37. Sirakaya-Turk, E.; Uysal, M.; Hammitt, W.; Vaske, J. (Eds.) *Research Methods for Leisure, Recreation and Tourism*; CABI Publishing: Wallingford, Oxfordshire, UK, 2011.

38. Arillas Business Association. Tourism Questionnaire for Arillas and Surrounding Area. 2015. Available online: http://arillas.de/arillas_questionaire.pdf (accessed on 21 January 2019).

39. Tourism and Cultural Affairs Bureau, City of Sapporo. Questionnaire for Tourists from Foreign Countries. 2015. Available online: http://www.city.sapporo.jp/keizai/kanko/program/documents/h21_eigo.pdf (accessed on 21 January 2019).

40. Apache Software Foundation. Apache Kafka 0.10.2 Documentation, 2018. Available online: http://kafka.apache.org/documentation.html (accessed on 7 January 2019).

41. Apache Software Foundation. Apache ZooKeeper Release 3.4.8. Documentation, 2018. Available online: https://zookeeper.apache.org/doc/r3.4.8/ (accessed on 24 January 2019).

42. Klimek, R. Exploration of Human Activities Using Message Streaming Brokers and Automated Logical Reasoning for Ambient-assisted Services. *IEEE Access* **2018**, *6*, 27127–27155. [CrossRef]

© 2019 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).