MOBILITY 2022

The Twelfth International Conference on Mobile Services, Resources, and Users

ISBN: 978-1-61208-962-1

June 26th –30th, 2022

Porto, Portugal

MOBILITY 2022 Editors

Anders Fongen, Norwegian Defence University College, Norway

MOBILITY 2022

Forward

The Twelfth International Conference on Mobile Services, Resources, and Users (MOBILITY 2022), held between June 26th and June 30th, 2022, continued a series of events dedicated to mobility-at-large, dealing with challenges raised by mobile services and applications considering user, device and service mobility.

Users increasingly rely on devices in different mobile scenarios and situations. "Everything is mobile", and mobility is now ubiquitous. Services are supported in mobile environments, through smart devices and enabling software. While there are well known mobile services, the extension to mobile communities and on-demand mobility requires appropriate mobile radios, middleware and interfacing. Mobility management becomes more complex but is essential for every business. Mobile wireless communications, including vehicular technologies bring new requirements for ad hoc networking, topology control and interface standardization.

We take here the opportunity to warmly thank all the members of the MOBILITY 2022 technical program committee, as well as all the reviewers. The creation of such a high-quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to MOBILITY 2022. We truly believe that, thanks to all these efforts, the final conference program consisted of top-quality contributions. We also thank the members of the MOBILITY 2022 organizing committee for their help in handling the logistics of this event.

We hope that MOBILITY 2022 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of Mobile Services, Resources, and Users.

MOBILITY 2022 Chairs

MOBILITY 2022 Steering Committee

Lounis Adouane, Université de Technologie de Compiègne / Heudisayc, France
Omar Sami Oubbati, University of Laghouat, Algeria

MOBILITY 2022 Publicity Chairs

José Miguel Jiménez, Universitat Politècnica de València (UPV), Spain
Laura Garcia, Universitat Politècnica de València (UPV), Spain

MOBILITY 2022
Committee

MOBILITY 2022 Steering Committee

Lounis Adouane, Université de Technologie de Compiègne / Heudisayc, France
Omar Sami Oubbati, University of Laghouat, Algeria

MOBILITY 2022 Publicity Chairs

José Miguel Jiménez, Universitat Politècnica de València (UPV), Spain
Laura Garcia, Universitat Politècnica de València (UPV), Spain

MOBILITY 2022 Technical Program Committee

Mohamed A.Aboulhassan, Pharos University, Egypt
Osama M.F. Abu-Sharkh, Princess Sumaya University for Technology, Amman, Jordan
Lounis Adouane, Université de Technologie de Compiègne / Heudisayc, France
Vijayan K. Asari, University of Dayton, USA
Mohamad Badra, Zayed University, Dubai, UAE
Matthias Baldauf, OST - Eastern Switzerland University of Applied Sciences, Germany
Chaity Banerjee, University of Central Florida, USA
Leandro Becker, Federal University of Santa Catarina (UFSC), Brazil
Olfa Ben Rhaiem, University of Sfax, Tunisia
Khadija Bouraqia, ENSEM, Morocco
Peter Brída, University of Žilina, Slovakia
Ivana Bridova, University of Zilina, Slovakia
Simeon Calvert, Delft University of Technology, Netherlands
Carlos Carrascosa, Universitat Politècnica de València, Spain
Chao Chen, Purdue University Fort Wayne, USA
Daniel Delahaye, Ecole Nationale de l'Aviation Civile (ENAC), Toulouse France
Anatoli Djanatliev, University of Erlangen-Nuremberg, Germany
Mohand Djeziri, Aix Marseille University, France
Mohamed El Kamili, Higher School of Technology - Hassan II University of Casablanca, Morocco
Ayman El-Saleh, A'Sharqiyah University, Oman
Brahim Elmaroud, Higher Institute of Maritime Studies - Casablanca, Morocco
Javier Fabra, Universidad de Zaragoza, Spain
Hassen Fourati, University Grenoble Alpes, France
Jordi Garcia, CRAAX Lab - UPC BarcelonaTECH, Spain
Gelayol Golcarenarenji, University of the West of Scotland, UK
Javier Ibanez-Guzman, Renault, France
Sergio Ilarri, University of Zaragoza, Spain
Jin-Hwan Jeong, SK telecom, South Korea
Christian Jung, Fraunhofer IESE, Germany
Georgios Kambourakis, University of the Aegean, Greece
Hassan A. Karimi, University of Pittsburgh, USA
Hamzeh Khalili, Fundació i2CAT, Barcelona, Spain

Imran Khan, Insight Center for Data Analytics | University College Cork, Ireland
Hanane Lamaazi, C2PS Center - Khalifa University of Science and Technology, Abu Dhabi, UAE
Francesca Martelli, IIT-CNR, Pisa, Italy
Ignacio Martinez-Alpiste, University of the West of Scotland, UK
Antonio Matencio-Escolar, University of the West of Scotland, UK
Subhasish Mazumdar, New Mexico Tech, USA
Weizhi Meng, Technical University of Denmark, Denmark
Farshad Miramirkhani, Isik University, Turkey
Javad Mohammadi, Carnegie Mellon University, USA
Alireza Morsali, McGill University, Canada
Tathagata Mukherjee, The University of Alabama in Huntsville, USA
Tatsuo Nakajima, Waseda University, Japan
Jose E. Naranjo, University Institute for Automobile Research (INSIA) | Universidad Politecnica de Madrid, Spain
Omar Sami Oubbati, University of Laghouat, Algeria
Luigi Pariota, University of Naples "Federico II", Italy
Hyoshin (John) Park,North Carolina A&T State University, USA
Wuxu Peng, Texas State University, USA
Alejandro Ramirez, Siemens AG, Germany
Ruben Ricart-Sanchez, University of the West of Scotland, UK
Anna Lina Ruscelli, TeCIP Institute - Scuola Superiore Sant'Anna, Pisa, Italy
Mahsa Sadeghi Ghahroudi, Glasgow Caledonian University, UK
Viliam Sarian, Scientific Research Institute for Radio, Russia
Régine Seidowsky, COSYS-GRETTIA | Univ. Gustave Eiffel | IFSTTAR, France
Alireza Shahrabi, Glasgow Caledonian University, Scotland, UK
Haichen Shen, Amazon Web Services, USA
Danny Soroker, IBM T.J. Watson Research Center, USA
Harald Sternberg, HafenCity Universität Hamburg, Germany
Daxin Tian, Beihang University, Beijing, China
Markku Turunen, Tampere University, Finland
Sudip Vhaduri, Fordham University, USA
Dario Vieira, Efrei Paris, France
Ulrich Walder, Graz University of Technology, Austria
Shuliang Wang, Beijing Institute of Technology, China
Rainer Wasinger, University of Applied Sciences Zwickau, Germany
Mudasser F. Wyne, National University, USA
Cong-Cong Xing, Nicholls State University, USA
Renjun Xu, Zhejiang University, China
Hong Yang, Nokia Bell Labs, Murray Hill, USA
Paul Yoo, Birkbeck, University of London, UK
Lisu Yu, Nanchang University, China
Mariusz Zal, Poznan University of Technology, Poland
Jianhua Zhang, Clarkson University, USA
Xiao Zhu, University of Michigan, USA
Wolfgang Zirwas, Nokia Bell Labs, Munich, Germany
Makia Zmitri, CNRS/GIPSA-Lab, France
Kamil Zyla, Lublin University of Technology, Poland

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Population-based Routing in LEO Satellite Networks 1
Anders Fongen and Lars Landmark

Combined Algorithm for Voronoi Diagram Construction in Application to Dynamic Ride Sharing 5
Anton Butenko and Jorge Marx Gomez

Fusion of Distributed Sensor Tuple Spaces and Agents using Broadcast Radio Communication for Mobile Networks 9
Stefan Bosse

SmartHelm: Design and Implementation of Open Source based Behavioral Data Repository 15
Harish Moturu, Johannes Schering, and Jorge Marx Gomez

Population Based Routing in LEO Satellite Networks

Anders Fongen
Norwegian Defence University College (FHS)
Lillehammer, Norway
email: anders@fongen.no

Lars Landmark
Norwegian Defence Research Establishment (FFI)
Kjeller, Norway
email: lars.landmark@ffi.no

***Abstract*—Packet switching in a Low Earth Orbit (LEO) satellite network may calculate the best traffic route through the less populated areas of the planet, in order to avoid relaying through the busiest satellites. A number of ideas for route calculation in a LEO system have been evaluated and the performance results are presented. This paper reports from ongoing research on Space Information Networks (SIN) with the purpose to offer application services in a LEO satellite network to mobile users. The conclusion is that route calculation based on population density gives a moderate, but significant improvement in resource utilization.**

Keywords*—*LEO satellites; space information networks; population density; mobile computing.

I. INTRODUCTION

The term SIN describes an information system located in space [1]. The concept is an evolution of satellite networks as they are known from the 1960s to present day, where satellites evolve from "radio mirrors" with plain wideband transponders, towards networks of interconnected satellites providing connectivity services based on store-and-forwarding of data packets. This evolution represents an *increasing system complexity* in the spacecrafts. It is a reasonable prediction that future LEO systems will offer application services and even an *Application as a Service* (AaaS) platform.

Which advantages can be achieved through the deployment of a SIN? Two main characteristics of the services distinguish a SIN from ordinary Internet services:

1) Global coverage for mobile clients,
2) Very low latency.

The round-trip time through a satellite at 500 km altitude can be as low as 3.3 milliseconds. Low latency will drive new time sensitive cooperative applications like remote surgery, autonomous aircrafts, etc., and is also one key property of 5G mobile networks.

The general design framework and a presentation of related problems and research questions have been previously presented in [2]. Beside the planned information services presented in that paper, a packet forwarding service will always be necessary as a baseline service to applications which cannot be designed along the guidelines for a SIN information service.

The chosen approach to the investigation of routing algorithms has included the earth's population distribution as a parameter. Satellites flying over densely populated areas of the earth are assumed to be busy serving requests from surface clients, and should not be burdened by additional packet relay tasks. The position, and thus the population density within the satellite's footprint can be calculated by anyone at any time and taken into account for the route calculations. Links between satellites can be predicted based on their relative positions so a link discovery protocol and subsequent link state distribution is unnecessary. The effect of this approach has been evaluated through simulation in a software model.

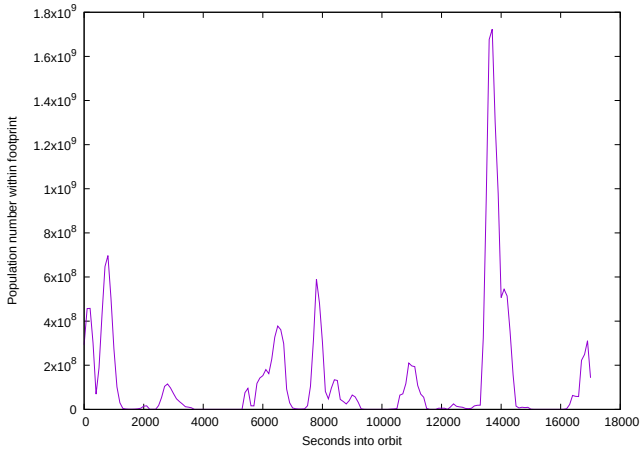

Figure 1. The population number inside the footprint of a satellite during three subsequent orbits.

The contributions of this paper are the results from a novel approach to routing optimization in LEO satellite networks, where the population density serves as a factor in link weight calculations.

The remainder of this paper is organized as follows: In Section II, the design rationale for the use of population data in the route calculations is presented. Section III briefly presents related research, and Sections IV and V describe the software based simulation platform and the satellite constellation chosen for the experiments. Sections VI and VIII present the evaluation of the three first routing methods and the delay-tolerant routing method. Section IX presents conclusions and topics for future research.

II. DESIGN RATIONALE

An important choice in our SIN studies is to include the earth's population density into the traffic analysis and resource planning. In particular for lower altitudes, the satellites will spend large fractions of their time over inhabited areas, mixed with shorter intervals of extremely high density. Figure 1 shows the population number inside the footprint of a satellite

 ISBN: 978-1-61208-962-1

during three subsequent orbits (283 minutes). The orbits cover different great circles of the planet and, therefore, show different results. It is likely that the rate of incoming requests will follow a similar pattern. An appealing idea is to allow idle satellites to offload busy ones, since neighbouring satellites in the network can communicate through high speed inter-satellite links. This approach to resource planning is extending the traditional design of LEO satellite systems.

A comprehensive model of a LEO system has been modeled in software and has allowed a wide range of design ideas to be evaluated for efficiency and resource consumption.

III. Related research

The term SIN has been used to describe networks of satellites and high altitude aircrafts (drones, balloons) with different service levels. Existing satellite networks like Iridium and Starlink [3] offer only communication services, the latter on a very large scale and with high bandwidth. A number of authors have proposed "Cloud Computing in Space" through the addition of larger satellites with sufficient energy and computing resources for taking on these tasks [4][5].

The concept of "Cloud Computing in Space " has strong relations to "Edge computing" (EC), and there are several studies on how to solve resource management problems when EC is applied to a LEO system. Li et al. [6] analyse some of these problems as well as provide a survey on similar studies. European Telecommunications Standards Institute (ETSI) engages in Mobile EC as a service oriented strategy for mobile network, but, like most other efforts, does not deal with the problems of frequent hand-over operations of services with a large state space. Also, none of the mentioned efforts propose the application of population density data as a parameter for their resource planning.

The results presented in this paper will not deal with technical details in the communication technology, but rather view the SIN as a distributed system. The authors are not aware of other efforts to investigate routing mechanisms based on population density.

IV. Software model

Based on the design rationale presented in Section II, a software model, written in Java, is in continuous development for the purpose of testing different hypotheses related to the operation of a SIN. With this model, we have studied link formation, request traffic formation and distribution, cache performance, routing efficiency, state transfer during handover etc.

The software model is equipped with population density data [7] which is used to calculate the population number within reach of the satellite at a given altitude. The colorized backdrop in Figure 2 illustrates the population distribution as seen from 500 km altitude.

Furthermore, the software allows any number of satellites to have common or individual orbital elements, although a simplification has been introduced in the current version to assume that all orbits are circular, not elliptic.

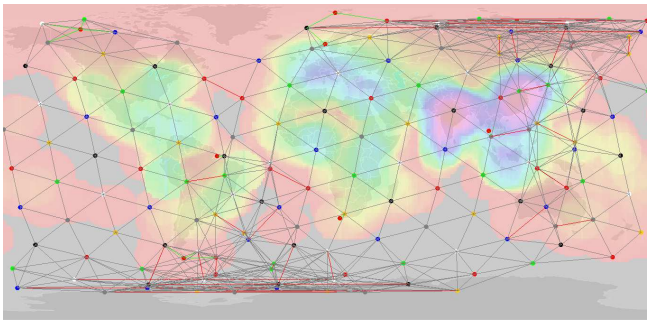

Figure 2. Screenshot from the satellite constellation model

During experimentation with the software model, it has been possible to establish a constellation with a reasonable number of satellites, which is able to form a completely connected grid yet with an altitude low enough to allow for very low latency and inexpensive and lightweight ground terminals.

V. A candidate constellation for study

Satellite networks servicing civilian mobile clients using handheld equipment tend to operate in LEO. E.g., the orbit altitude of Iridium satellites is 781 km, which allows for lightweight ground terminals without the need for antenna deployment. The inclination of the orbit can be made so steep that the polar regions are fully covered, or given a lower angle to spend more time over the densely populated latitudes closer to the equator.

The choice of orbit altitude determines the diameter of the *footprint*, e.g., the circular region of the earth's surface with potential connectivity, and also the maximum communication distance for inter-satellite links and uninterrupted service for ground terminals. Simply stated, a lower orbit altitude reduces the design constraints on the ground terminals and provides higher communication capacity, but increases the cost due to the higher required number of spacecrafts.

Figure 2 shows a screenshot from the software model, containing 150 satellites with an orbit inclination of 75 degrees and an altitude of 500 km. The colors on the backdrop indicate the population density inside the footprint of a satellite in that position (contrary to the local density at that exact position). The population density is considered to be a parameter for the estimation of the request rate received from ground surface clients. Other possible parameters, like regional Internet penetration, time of day and day of week may be taken into account at a later time. Changes in the demographics are assumed to be slow and are not taken into account.

VI. Routing decisions in a LEO network

The purpose of this paper is to evaluate different routing methods. Even though some LEO networks (e.g., Starlink) route traffic through terrestrial networks, the focus will be kept on routing through inter-satellite links. The availability of these links is predictable, as stated in Section I.

Four different algorithms for route calculation have been evaluated:

 ISBN: 978-1-61208-962-1

1) Hot potato: Establish the geographical direction (bearing) to the receiver and choose the link with the direction nearest to this bearing.
2) Unweighted Dijkstra: Use Dijkstra's shortest path algorithm with all links given the same cost.
3) Weighted Dijkstra: Same as above, but with links given weight according to the population density within the receiver's footprint.
4) Delayed weighted Dijkstra: Same as above, but with an introduction of a delay to wait for a better route to occur.

The evaluation of the different algorithms was based on the calculation of the total population number within the footprints of the satellites along the route. The assumption is that a smaller value of this number indicates that mostly idle satellites are involved in the packet forwarding, and consequently an improved resource utilization.

Although the familiar Dijkstra's algorithm is used, this is still a source routing approach where the full route is calculated by the sender and attached to the message.

All four algorithms were tested on the same route going between ground stations in Lillehammer, Norway and Bankok, Thailand.

VII. Evaluation of the route algorithms

This section will present experimental details and results of the four chosen routing algorithms listed in Section VI.

A. Hot potato

One simple routing algorithm is to simply "throw" the packet towards the destination. This is accomplished by selecting a link pointing in the best direction. Since the location of the target and the directly linked neighbors are known, this is just a matter of computing bearings and comparing angles between the resulting vectors.

As indicated on the example shown in Figure 3, an efficient route is selected with a nearly optimum number of hops, but is also prone to looping and the introduction of a time-to-live element in the message was needed to remove them from endless cycles. The packet loss ratio was, therefore, observed to be higher than the other alternatives.

B. Unweighted Dijkstra

The traditional Dijkstra's Shortest Path algorithm was applied to the link collection. Every link was given the same cost, which optimizes the number of hops only, without regard to the population number below.

In contrast to the hot potato method, a link is not selected unless it is a part of the path to the target, so looping does not occur and potential packet loss is detected during path calculation. Less resources are thus wasted on packet loss occurring after several hops.

One resulting path is shown in Figure 4, with the same number of hops as the hot potato method. The focus on the number of hops rather than bearing allows this method to choose other paths still effective, e.g., over the polar regions.

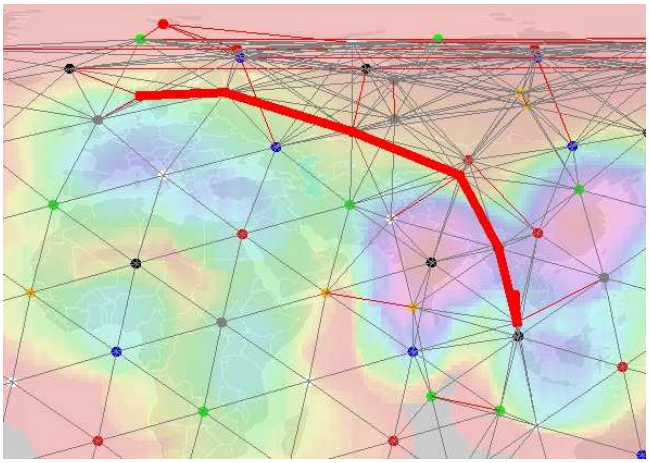

Figure 3. Example result from the hot potato routing algorithm.

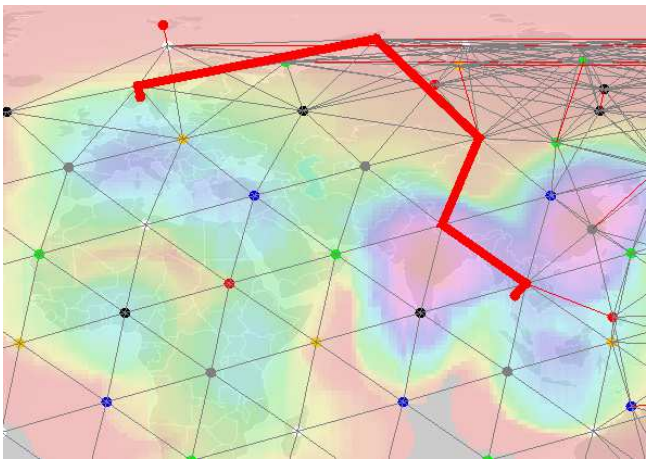

Figure 4. Example result from the unweighted Dijkstra's routing algorithm.

C. Weighted Dijkstra method

The weighted Dijkstra method is similar to the method applied in Section VII-B, but with link costs applied according to the population number inside the footprint of the satellite at the other end of the link. Links with zero population below were given a minimum cost to avoid ridiculously long routes over "free" links.

This method demonstrates the routing principle proposed in this paper; the two former methods are presented as baseline methods for the estimation of the effect of population-based routing calculations. Figure 5 shows one example of a route between Lillehammer and Bankok calculated using this method. It is clearly shown how the route now avoids the densely populated areas, but has a higher number of hops.

D. Comparing the three methods

The efficiency of the three aforementioned routing methods will now be compared. Since the satellite grid is in constant movement, the calculated cost for any route will change over time. For this reason, each of the three routes were calculated repeatedly over 2500 seconds, and the calculated

 ISBN: 978-1-61208-962-1

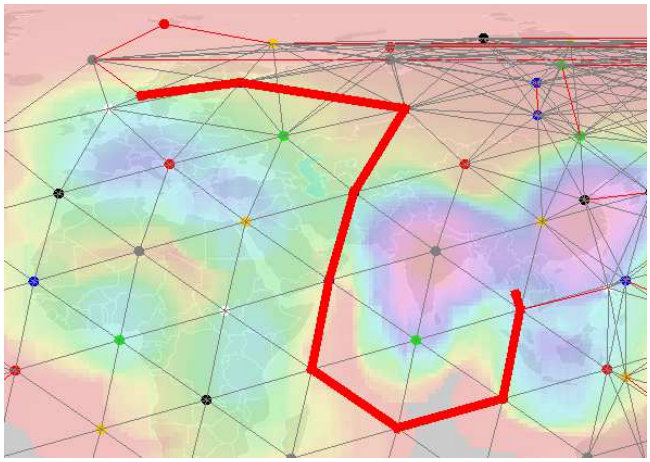

Figure 5. Example result from the weighted Dijkstra's routing algorithm.

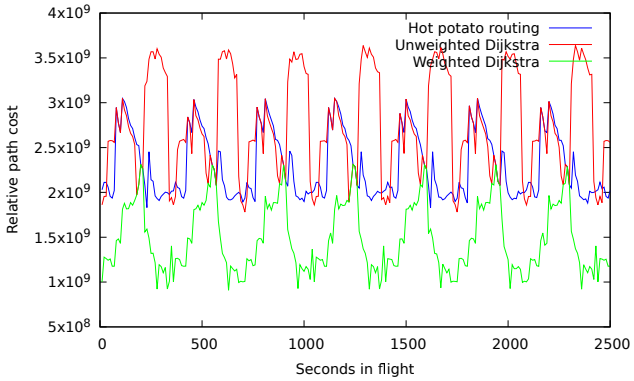

Figure 6. Routing cost for three different routing methods, plotted over time. The numbers on the y-axis should be regarded as relative, since the population dataset does not sum up to the actual earth's population.

costs plotted as shown in Figure 6. The plot offers some interesting observations:

1) The ratio between highest and lowest cost is no more than 3.5
2) The cost variations over time are smaller for the hot potato method than the two methods based on Dijkstra's algorithm
3) All three methods show a strong cyclic behaviour with a period of approximately 350-400 seconds.

This use of Dijkstra's algorithm is a source routing technique which allows many aspects to be taken into account for the routing decisions. Beside regional Internet penetration etc., policy decisions based on the sender's preferences on commercial, security and technical issues may have an impact on the chosen route.

VIII. A PROPOSED DELAY-TOLERANT ROUTING METHOD

The strong cyclic properties of all the three routing methods evaluated in Section VII suggest the design of a routing calculation reserved for delay-tolerant traffic. The average routing cost can be reduced if we allow the packet to wait for an opportunity to send when the cost is relatively low. If successful, and acceptable for the communicating service, this method may even out the communication load of the satellites and in general improve the resource utilization of the system.

TABLE I
RESULTING QUEUING DELAY AND ROUTING COST WHEN DIFFERENT PERCENTILES ARE USED AS SENDING THRESHOLDS

Percentile	Queuing delay (secs)	Routing cost ($\cdot 10^7$)
10	855.92	99.35
20	544.46	105.08
30	91.38	111.11
40	53.79	116.63
50	29.14	119.41
60	12.98	123.83
70	5.78	126.79
80	2.83	127.98
90	0.65	130.05

In order to evaluate this method, the routing cost between the endpoints was sampled every 10 seconds and the different percentiles of the cost distribution were calculated. Messages sent from the transport layer are queued until the calculated routing cost is lower than the required percentile. The average routing cost and queuing delay are measured and presented in Table I.

As expected, a lower percentile will cause a longer queuing delay, which is also apparent in the table. A little more surprising is the relatively small improvement (25%) gained through a much longer queuing delay. The results indicate that this method is not worth the efforts of implementation.

IX. CONCLUSION

Effective routing through a grid of LEO satellites has been the topic of this paper, where the population density on the surface below a satellite determines the link cost. The rationale for doing so was to employ communication resources in idle satellites flying over inhabited areas of the planet. The experiments conducted on a software based simulation platform show clearly the effect of route calculations based on these criteria. Future experiments will combine these experiments during actual service production to study the effect on their response times.

REFERENCES

[1] L. Bai, T. de Cola, Q. Yu, and W. Zhang, "Space information networks," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 8–9, 2019.
[2] A. Fongen, "Application services in space information networks," in *CYBER 2021*. Barcelona, Spain: IARIA, Oct 2021, pp. 113–117.
[3] "Starlink web site," https://www.starlink.com/, [Online; retrieved 03-May-2022].
[4] S. Briatore, N. Garzaniti, and A. Golkar, "Towards the internet for space: Bringing cloud computing to space systems," in *36th International Communications Satellite Systems Conference (ICSSC 2018)*, 2018, pp. 1–5.
[5] S. Cao *et al.*, "Space-based cloud-fog computing architecture and its applications," in *2019 IEEE World Congress on Services (SERVICES)*, vol. 2642-939X, 2019, pp. 166–171.
[6] C. Li, Y. Zhang, R. Xie, X. Hao, and T. Huang, "Integrating edge computing into low earth orbit satellite networks: Architecture and prototype," *IEEE Access*, vol. 9, pp. 39 126–39 137, 2021.
[7] "Gridded population of the world v.4.11," [Online; retrieved 03-May-2022]. [Online]. Available: https://sedac.ciesin.columbia.edu/data/collection/gpw-v4/sets/browse

 ISBN: 978-1-61208-962-1

Combined Algorithm for Voronoi Diagram Construction in Application to Dynamic Ride Sharing

A. Butenko, J. M. Gómez
Department of Computing Science, Carl von Ossietzky University of Oldenburg
Oldenburg, Germany
E-mail: anton.butenko@uol.de
E-mail: jorge.marx.gomez@uni-oldenburg.de

Abstract—A standard Voronoi diagram decomposes a plane into cells with a common closest site. This structure is widely used in computational geometry in application to the nearest neighbor problem. Using Euclidean metric is the most straightforward solution, however, in an urban environment, it may lead to insufficient accuracy that is crucial in applications such as dynamic ride sharing. Deviations in determining the nearest meeting point are especially significant under the presence of obstacles: water reservoirs, railway tracks, highways, industrial zones, as well as hilly terrain. Here, we propose a combined approach for a city Voronoi diagram construction in a generalized metric space. A transportation network is modelled as a weighted graph, so that the route consists of a foot-walking part and the shortest path in the graph. The presented algorithm constructs a continuous Voronoi diagram for a plane using the individual Voronoi cells graph as generator objects. Evaluation for the specific city topography shows that the described algorithm provides more accurate results in comparison with the standard Voronoi diagram.

Keywords- Voronoi diagram; dynamic ride sharing.

I. INTRODUCTION

Ride sharing applications are aimed at connecting drivers and passengers in an optimal way. What this optimal way means depends a lot on the specific mobility solution philosophy and its target audience. Nevertheless, most of isthem face such optimization problem as the nearest neighbor search: identifying the point from a set of points which is the closest to a given point according to some measure. The mobility application Instaride [2] developed for the spontaneous shared trips is driven by an instant matching algorithm. It connects drivers and passengers in real time based on the user's mobile device positioning (satellite navigation data, triangulation in mobile network) [1]. In order to minimize the driver's efforts and his route detour, the finite set of preselected fixed points is used for passengers' pick-up and drop-off (named meeting points, in general). Preselection of the meeting points is determined by the environmental conditions and is based on criteria such as parking opportunity, presence of pedestrian zones and easily recognizable landmarks. Such an approach leads to the problem of finding the nearest meeting point for users based on their real-time positions.

The paper structure is the following. The Introduction explains the problem's origin. In Section 2, we describe the concept of the presented approach and introduce the terms and notation. Sections 3 and 4 describe two parts of the algorithm: discrete and continuous. In Section 5, the algorithm steps are given in detail. Section 6 presents the algorithm efficiency evaluation for the specific city topography. Section 7 concludes our work.

II. VORONOI DIAGRAM IN A GENERALIZED METRIC SPACE

One of the most effective ways to solve problems related to the nearest neighbor search is to use the Voronoi diagram. We introduce the following notation here: L_p is a metric space with the corresponding function $\rho : L \times L \to \mathbb{R}^+ \bigcup \{0\}$ that satisfies metric axioms. Then, $S_r(x) = \{z : \rho(x,z) = r\}$ is the metric sphere with radius $r \in \mathbb{R}^+$ and $B(x,y) = \{z : \rho(x,z) = \rho(y,z)\}$ is the bisector of x and y $(x, y, z \in L_\rho)$. It splits L_p into the half-spaces $D(x,y) = \{z : \rho(x,z) < \rho(y,z)\}$ and, lying on the other bisector side, we have $D(y,x) = \{z : \rho(y,z) < \rho(x,z)\}$. For a given finite set of seeds $S = \{s_1, ..., s_k\} \in L_\rho$, the Voronoi cell related to s_i is expressed as

$$VR(s_i, S) = \bigcap_{i \neq j} D(s_i, s_j)$$

and the Voronoi diagram of S:

$$V(S) = \bigcup_{i \neq j} \overline{VR}(s_i, S) \cap \overline{VR}(s_j, S),$$

with the horizontal line denoting closure.

Being the most straightforward solution, a Voronoi diagram based on the Euclidean distance provides tolerable approximation in the urban environment if the points are located quite far apart within the uniform transportation network. In other cases, the results are significantly worse: for short distances, in areas with irregular topography, under the presence of obstacles, or in application to suburbs stretched along the roads forming axon-like structures. The use of other metric functions may improve the accuracy; however, another problem arises: in some cases, the bisector dimension may be more than 1 (this is true even for the Manhattan distance $\rho(x,y) = \sum(|x_1 - x_2| + |y_1 - y_2|)$).

In a number of works, the graph represents the streets network. The discrete network Voronoi diagram is then constructed while the metric used is the link between nodes (e.g. Yomono [5]). However, such models do not allow

 ISBN: 978-1-61208-962-1

shortcuts, which are often used by pedestrians to shorten the paths. Aichholzer et al. [3] consider a plane with Manhattan distance and isothetic transportation network. There are also several works that use the generalized concept of Voronoi region (*needle*) proposed by Bae and Chwa [6].

The approach presented in this work is aimed at being applicable for the non-orthogonal street structure with curvilinear street segments. At the same time, as the ride sharing is spontaneous, we strive to avoid excessive model complexity; only walking to/from meeting points is assumed. The same diagram is used by the driver and the passenger. In addition, being flexible to the possibility of using the available network bandwidth data, the model should also work with the minimum information of this kind. Thus, we believe, the task of developing an optimal method for constructing a Voronoi diagram for a similar class of problems is to find a balance between complexity, accuracy and flexibility in using available data as the latter may vary a lot in different regions.

The main idea of the approach presented below is to construct a discrete Voronoi diagram on a graph and then transform the obtained cells into the seeds or generator objects for the continuous Voronoi diagram on the plane. The latter represents the partition of the plane with a transportation network into proximity regions for the set of the given meeting points.

III. VORONOI DIAGRAM ON THE GRAPH

We consider the area of interest as domain $\Omega \subset \mathbb{R}^2$ containing the city transportation network, providing fixed routes. This network is modelled as a weighted graph $G(V,E)$, where $E=\{e_i\}$ is the set of edges, representing roads and streets. $V=\{v_k\}$ are the graph vertices, corresponding to the intersections and the deadlocks. Non-negative edge weights $w(e_i)$ determine some proximity measure between the vertices connected by the edge e_i. Depending on data availability, it can be, for e.g., edge length or edge travel time. The latter depends on the segment's capacity, inclination, or traffic. Setting $\rho_G(v_i,v_j)$ in an ordinary way as the length of the shortest path between v_i and v_j, one can consider V_{ρ_G} as a metric space. Without additional constraints, it is true for the undirected graph as ρ_G always satisfies the symmetry axiom unlike the directed graph case. Assuming the only way to move (car-driving or foot-walking), we can build a Voronoi diagram graph for $V(G,E)$ with respect to the meeting points $S=\{s_1,...,s_k\}\subset V$ (Figure 1).

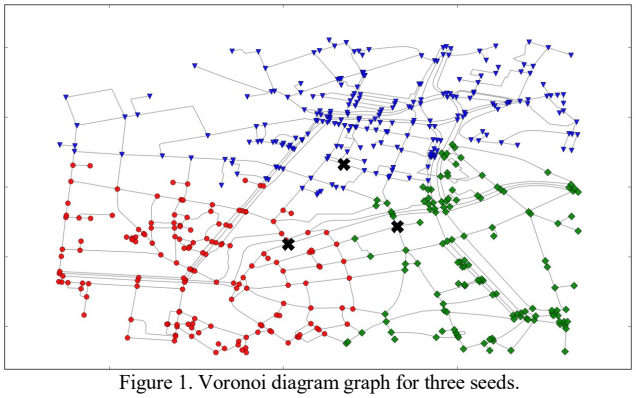

Figure 1. Voronoi diagram graph for three seeds.

The Voronoi diagram brakes up the set of vertices into the direct sum of the Voronoi cells $V=V_1\oplus...\oplus V_k$, where $V_i=VR(s_i,S)$. Let $E_i(s_i)$ be a set of edges connecting vertices within V_i. Then, $E=E_1\oplus...\oplus E_k\oplus C$, where C is the set of «border» edges whose vertices belong to the different cells.

IV. PLANAR VORONOI DIAGRAM

Graph G can be considered not only as a topological structure, but its vertices and edges determine geometrical objects: points and lines within Ω. Hence, each edges subset E_m determines the lines set $E'_m\in\mathbb{R}^2$. Let us consider $E'=\{E'_m\}$, $m=1..k$ as seeds for a planar Voronoi diagram in Ω with the Euclidean distance. Such metric function choice is based on the following assumption. The motion between transportation lines is carried out along a straight line in the direction to the nearest network segment. Practically, the construction of $V(E')$ is based on the search for the metric spheres intersection for each pair (E'_m,E'_n). With the sphere radius variation, the points of these intersections form bisectors $B(E'_m,E'_n)$ and corresponding half-planes $D(E'_m,E'_n)$. Thus, each Voronoi cell $VR(E'_m,E)$ can be computed as an intersection of all half-planes containing E'_m. This process is described in detail in the next section.

V. COMBINED ALGORITHM

The computational algorithm can be performed with the following steps:

1. The city transportation network representation as a graph $G(V,E)$ is obtained from the OpenStreetMap project geodata [6].
2. The set of meeting points $S=\{s_1,...,s_k\}$ is added to the graph vertices V;
3. Construct the Voronoi diagram $V(G)$ graph. If the graph order lets it, brute-force can be used: $\forall v\in V$ find the distance on the graph $\rho_G(v,s_i)\ \forall i$ using the Dijkstra algorithm. If s_j satisfies $\rho_G(v,s_j)=\min_i \rho_G(v,s_i)$ then $v\in VR(s_j)$. Using a more optimal algorithm with the trees can improve the computational performance, if necessary.
4. From the constructed Voronoi cells from the previous step, V_m we get E_m — the subsets of edges that belong to the cell.

 ISBN: 978-1-61208-962-1

5. Each set E_m is transformed into E'_m — the set of the planar lines with their coordinates.
6. Construct the Voronoi diagram graph $V(E')$ using $\{E'_m\}$ as seeds and the Euclidean distance ρ_2 as metric functions:
   6.1. $\forall(E'_m, E'_n)$ in order to find the bisector:
   $$B(E'_m, E'_n) = \bigcup_{r\in(0;\infty)} S_r(E'_m) \cap S_r(E'_m)$$
   interpolate points obtained from this formula for a finite number of radii $r_{k+1} = r_k + \Delta r, \quad r \in [r_{\min}; r_{\max}]$. Here, $r_{\min} = \frac{1}{2}\rho_2(E'_m, E'_n)$ and $r_{\max}$ is the minimum radius that satisfies the condition $S_r(E'_m) \cap S_r(E'_n) \notin \Omega$: the spheres intersect outside of the domain. In the calculations below, a Δr of 10 meters is used;
   6.2. Determine the corresponding half-plane $D(E'_m, E'_n)$ that contains E'_m;
   6.3. Repeat steps 6.1-6.2 for all $m \neq n$. The intersections of the obtained half-planes form the Voronoi cell for the seed E'_m: $VR(E'_m) = \bigcap_{m\neq n} B(E'_m, E'_n) \cap \Omega$;
7. Making a reverse substitution $E'_m \to E_m \to s_m$, gets $VR(s_m, S)$ as the cells of the combined continuous Voronoi diagram based on ρ_2 and ρ_G.

VI. EVALUATION

The accuracy of the two types of Voronoi diagrams was compared for the central part of Oldenburg, Germany. Without claiming to be a full-fledged test, such comparison illustrates the potential prospects of the above-presented approach. For a given area with 1 km radius and 15 meeting points, we construct the standard Voronoi diagram based on the Euclidean Distance (DE) and the combined diagram in a way described above (DC) (see Figure 2).

Comparing the two corresponding types of Voronoi cells C_1 and C_2, their area-weighted average difference can be calculated as:

$$\Delta S = \frac{1}{S(\Omega)} \sum_i \frac{S(C_1^i \triangle C_2^i)}{S(C_1^i \cup C_2^i)} \cdot (S(C_1^i) + S(C_2^i)) \cdot 0.5$$

For the considered example, $\Delta S \approx 0.31$. Also, for 1000 random locations uniformly distributed within the domain, we determine the nearest meeting point in three ways: a) from DE; b) from DC; c) by computing the routes to all the meeting points with the Openrouteservice engine (ORS) [4] and detecting the meeting point corresponding to the minimum route length. The results are the following:

- The nearest meeting points obtained from DE and DC are equal - 792 locations.
- otherwise - 208 locations.

For latter 208 locations, we determine the nearest meeting point with ORS:

- the meeting points obtained from ORS and DC are equal - 165 locations.
- the meeting points obtained from ORS and DE are equal - 41 locations.
- otherwise – 2 locations.

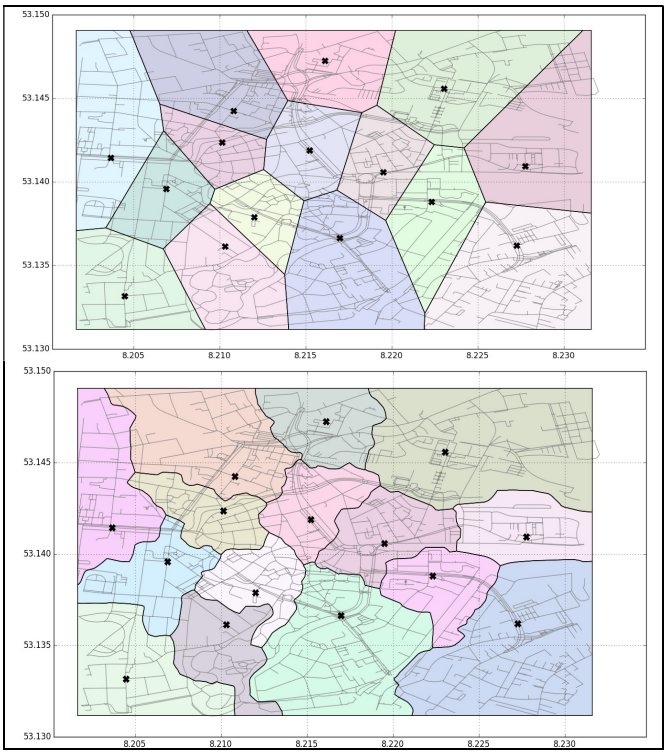

Figure 2. Voronoi diagram for Oldenburg city centre: a) DE; b) DC.

VII. CONCLUSION

There are several steps to be performed next in the context of this work. First, a full algorithm evaluation must be performed in a large area using a number of different criteria. Second, its computational complexity must be evaluated and, probably, reduced. Third, the potential of using additional bandwidth data must be analyzed.

Nevertheless, at this stage, one can conclude that, despite the number of simplifications, the described algorithm provides more accurate results in comparison with a standard Voronoi diagram. At the same time, complex topography features processing requires further study since they are probably the main reason for the remaining imprecision. These include multi-level road crossings, tunnels, elongated geometric objects and natural obstacles.

REFERENCES

[1] M. Eilers et al., “An instant matching algorithm in the context of ride-hailing applications, using isochrones and social scoring”, In:, (Hrsg.), Informatic 2021. Gesellschaft für Informatik, Bonn, pp. 103-114, 2021, doi: 10.18420/informatik2021-007.

[2] Instaride, https://instaride.webflow.io, retrieved: May 2022.

[3] O. Aichholzer, F. Aurenhammer, and B. Palop, “Quickest Paths, Straight Skeletons, and the City Voronoi Diagram”, Discrete and Computational Geometry, 31. pp. 17-35, 2004, doi:10.1007/s00454-003-2947-0.

[4] Openrouteservice, https://www.openstreetmap.org, retrieved: May 2022.

[5] H. Yomono, “The Voronoi diagram on a network”, Technical report, Nippon Systems Co, Tokyo, 1991.

[6] S. W. Bae and K.-Y. Chwa, “Voronoi Diagrams with a Transportation Network on the Euclidean Plane,” Int. J. Comput. Geometry and Appl., vol. 16, pp. 101-112, 2004, doi:10.1007/978-3-540-30551-4_11.

 ISBN: 978-1-61208-962-1

Fusion of Distributed Sensor Tuple Spaces and Agents using Broadcast Radio Communication for Mobile Networks

Stefan Bosse
Department of Mathematics & Computer Science
University of Bremen
28359 Bremen, Germany
e-mail: sbosse@uni-bremen.de

***Abstract*— Short-time and short-range device-to-device and device-to-service communication in ad-hoc mobile networks is a challenge. Although Internet access is widely available, there are places that are not covered by wireless IP networks, or IP networks are not suitable for ad-hoc short-time and short-range communication with spatial context. In this work, devices communicate in a connectionless and ad-hoc way by Bluetooth broadcast messaging available in any smartphone and in most embedded computers. Bi-directional connectionless communication is established via parallel advertisement and scanning modes by exchanging data tuples. The communication is performed via a tuple space service on each node. Tuple space access is performed by simple event-based agents. Mobile devices can act as tuple carriers that can carry tuples between different locations. Additionally, UDP-based Intranet communication can be used to access tuple spaces on a larger spatial range. The Bluetooth Low Energy Tuple Space (BeeTS) service enables opportunistic, ad-hoc and loosely coupled device communication with a spatial context.**

Keywords- Distributed Data Processing; Tuple Spaces; Sensor Networks; Internet of Things; Multi-agent Systems.

I. INTRODUCTION

Ubiquitous and pervasive computing introduces a big amount of visible and non-visible low-resource and mobile devices with an exponentially increasing number of deployed embedded systems. Commonly used authenticated and user-centred communication is not always suitable anymore. Most Internet-of-Things (IoT) devices and smart sensors are still connected via the Internet using IP communication and that are accessed by a server that collects sensor information periodically or event-based. Although Internet access is widely available, there are places that are not covered and WLAN and mobile cell communication requires a descent amount of power not always available. Additionally, the residential time of mobile devices can be below one minute, not suitable for ad-hoc connection-based and negotiated communication. Finally, the spatial context (the environment in which the sensor or devices are situated) is not considered (or lost) by Internet connectivity. Social contact tracing and social interaction commonly takes place only in a small spatial area. In this work, smart devices communicate in a connectionless and ad-hoc way by using low-energy Bluetooth available in any Smartphone and in most embedded computers, e.g., the Raspberry PI devices. Bi-directional connectionless communication is established via the advertisement and scanning modes. The communication nodes can exchange small data or functional tuples using a tuple space service. Mobile devices act as tuple carriers that can carry tuples between different locations. Additionally, UDP-based Intranet communication can be used to connect tuple spaces.

Tuple spaces are widely used for data storage for multi-processing in distributed and parallel systems. The Bluetooth Low Energy Tuple Space (BeeTS) service is a lazy distributed tuple space server and client. BeeTS uses Bluetooth and UDP broadcasting for tuple space interaction and tuple exchange. BeeTS supports tuples with an arity up to 4. A tuple space provides a spatial context, i.e., tuple space access (by mobile devices) is limited to nearby devices. Distributed tuple spaces can be connected by node routers using IP communication if available. The router composes global space sets by tuple exchange and replication. The router is customised by function code rules. These rules can be changed at run-time and the code can use Machine Learning algorithms to optimally distribute tuples.

The novelty of this work is two-fold. Firstly, a ubiquitous radio broadcast medium is used for low-distance communication in ad-hoc mobile networks combined with a unified tuple space paradigm. Secondly, the tuple space communication is performed by simple reactive event-based agents programmed in JavaScript that can be sent to a node via the tuple space operations, too. A proof-of-concept use-case featuring smart interactive building control is presented. Beside the communication technology, basic security aspects are addressed, too.

II. RELATED WORK

Originally, Bluetooth was introduced as a short-range wireless communication technology used for linking peripherals (like ear phones) to smartphones. Mid-range and connectionless Bluetooth communication is used in several crowd sensing and crowd interaction use-cases, e.g., used for attendance tracking [1] and most prominently for pandemic contact tracing [2]-[4]. These use-cases basically perform unidirectional producer-consumer communication using advertisement packet broadcasts. In this work, asynchronous bi-directional communication between nodes of a set that are within the receiving range is addressed.

In [5], multi-hop networks are established with Bluetooth Low Energy (BLE) using a new multi-hop Generic Attribute Profile (GATT) layer service and connection-based node group connections (a group is a set of nodes within a specific spatial diameter). A connection protocol starting with advertisement, negotiation (requiring authorisation credentials),

 ISBN: 978-1-61208-962-1

and final set-up of a bi-directional communication channel is time consuming and can require up to one second.

Connection-based channel communication is mostly not suitable to transfer small amounts of data, especially, from mobile devices with high spatial fluctuation. Instead, the advertisement mode is attractive to transfer small amount of data via broadcasting. Small sensor nodes and economy advertisement tracking use this kind of communication. Current deployment of smartphones for social contact tracing relies mostly on this method, too. But transferring data payload via advertisement packets has some drawback, mostly the issue of a low reception probability, discussed [6]. If the number of devices sending advertisement packets in the receiving range of a device increases, the probability p_i to receive at least one advertisement packet from a sequence of identical packets decreases significantly. But nonetheless, [6] concluded that BLE broadcasting is still suitable in most situations.

The generative tuple space paradigm is well suited for ad-hoc mobile networks [7], especially if this paradigm is coupled with the agent paradigm [8]. The concept of fusion of mobile devices with a set of heterogeneous sensors providing a sensor service that can be accessed via the Internet was introduced by [9].

III. Formalisation of Distributed Tuple Spaces

A tuple space is basically a data base containing tuples. A tuple is a poly-typed ordered set of data values. The number of data values specifies the arity of the tuple, i.e., $tu=\langle v_1,v_2,..,v_k\rangle$, $|tu|=k$. Each tuple space TS can be divided into independent sub-spaces $TS= ts_1 \cup ts_2 \cup .. \cup ts_n$, with ts_i holding only tuples of arity i. The data type of each element of a tuple can be arbitrary, i.e., any scalar value (float, integer, Boolean), or composed values, i.e., arrays or structures. In this work, the data elements are limited to scalar values, more specifically, float32, int16, and short strings (or data bytes).

Tuple space communication is generative, i.e., the lifetime of a tuple can be longer than the lifetime (or presence) of the producer process. There are producer and consumer processes. A producer process uses the *out*(⟨*tuple*⟩) operation to store a tuple in the space. The consumer process uses the input operation *inp*(⟨*pattern*⟩) to remove a tuple. A tuple is found by pattern matching. A pattern is a tuple with actual and formal parameters (wild cards). Any matching tuple is returned. The input operation destroys the tuple atomically, i.e., one input request consumes at most one tuple. In distributed asynchronous systems, this is difficult and expensive to be achieved. There is a read operation *rd*(⟨*pattern*⟩) that returns only a copy of the matching tuple. Tuples can be deleted by using the remove operation *rm*(⟨*tuple*|*pattern*⟩). Tuples can have an unlimited lifetime, but practically the lifetime is limited either by the tuple space service (removing old tuples) or by the *out* operation providing a lifetime t, i.e., $tu(t)=\langle v_1,v_2,..,v_k,t\rangle$, $|tu|=k$. The determination of the tuple lifetime is difficult in advance and depends on the application and the producer-consumer interaction time scale, but upper boundaries can be defined.

The read and input operations are typically synchronous, i.e., as long as there is no matching tuple, the requesting process is blocked. This operational semantic requires in distributed systems reliable and synchronous bi-directional communication that is not available in this work. For this reason, the read operation is just a request that can be fulfilled within a time interval $[t_0, t_1]$, or not (time-out).

IV. Unreliable Broadcast Communication

It is assumed that there is a broadcast medium B, e.g., using radio waves, which can reach a number of nodes $N_B=\{n_i\}_{i=1}^{k}$ defining a receive area/range coverage $C(B,N)(t)$ that changes with time t. B can send broadcast messages m to all listening nodes reachable by B. The set of nodes within B can vary on time and spatial scale. Furthermore, it is assumed that there is a probability $p_i(m,r_{i,j},[t_0,t_1]) \in [0,1]$ that a message m is received by a node i sent by node j in distance r within a time interval $[t_0,t_1]$. These two assumptions are fundamental for the proposed distributed tuple spaces.

It is assumed that single packets that can be sent over B are strictly limited by a small number of bits, e.g., 200-300 Bits. This requires a compact and optimized message format, discussed in the next sub-section.

Messages

There are seven different message types:

- *OUT* operation stores a tuple in all tuple spaces receiving this message;
- *RD* and *INP* operations that request tuples from all receiving tuple spaces;
- *TEST* operation checks for the existence of a tuple or set of tuples;
- *TUPLE* is either an initial message sending this tuple to all receiving nodes without storing the tuple in the respective tuple space, or a reply to a tuple request;
- *IAMHERE* and *WHERE* messages are used for node search.

The message format is rather simple and contains a header and the tuple data payload. A sequence number stored in the header is required to detect the reception of multiple copies of the same message, a prerequisite for deployment of the Bluetooth device back-end that sends a message multiple times, explained in Sec. 4.2. The signature byte specifies the following tuple data pay-load. Depending on the back-end communication device and the supported packet format, the number of pay-load bytes can be very small. The signature field specifies the type of each tuple element with a tuple limit of four elements. For Bluetooth advertisement packages, there is $N_{BLE}=32$, for the UDP back-end it is at most $N_{UDP}\geq 512$. The message header and the data payload are encoded in an BLE advertisement packet using one device local name attribute (ASCII85 encoded) and seven 16 Bit service UUID attribute fields. The header and payload are stored in a linear format using UDP messaging,

In contrast to typical tuple space services, the tuple operations are not atomic here. They can be executed at any given time point t in the near future or never, and the set of reachable tuple spaces that execute the request is not bound and can be zero. There is no assumption that neither a message arrives at a specific node nor that the request is pro-

 ISBN: 978-1-61208-962-1

cessed successfully. There are filter rules processed by agents that can prevent tuple operation execution, too. That means, the *INP* operation is only a suggestion to all receiving tuple spaces to remove a matching tuple. All operations pose a probabilistic behaviour, i.e., there is a probability ≥ 0 that a message is processed. The encoding of tuples is done automatically. Before a tuple is encoded and packed, a signature is derived; numbers are classified either in integer 16 Bit or float 32 Bit values depending on the actual value.

Bluetooth LE Back-end

Bluetooth is a standard radio communication technology connecting master with slave devices (peripherals) over short distances in the range of 10-300 cm. The data exchange is performed over negotiated connections. A connection is negotiated using the advertisement and scanning mode of Bluetooth devices. A peripheral device advertises its service by sending out a short advertise message with preliminary information about the devices. A Bluetooth host (master) device, typically a smartphone, will receive these advertisement messages if it is in scanning mode. One limiting factor of communication by mobile devices is power consumption, which is addressed by BLE devices that can adapt to various communication situations with different power levels. Devices can access remote tuple spaces of nearby neighbouring nodes (typical in the range of 1-10m) by using BLE broadcasting (called ble-ts).

A device in advertisement mode will send out periodically advertisement messages that contain a small payload depending on the advertisement message class, shown in Fig. 1. In this work, the payload is limited to 32 Bytes. There are 40 RF channels in BLE, each separated by 2 MHz (centre-to-centre). Three of these channels are called the primary advertising channels (labelled 37, 38, and 39), while the remaining 37 channels are called the secondary advertisement channels (they are also the ones used for data transfer during a connection) [10]. The primary channels are switched randomly in periods. On the other hand, the scanning device has to switch the (primary) receiving channels randomly, too. There is a probability p that an advertisement packet is received if both the scanner and the advertiser are switched on the same channel and if there is no other sending within the receiving range creating collision (invalidation of the message).

The bandwidth and latency is limited by the advertisement time $t_{ad}=t_e-t_s$. The maximal number of independent messages that can be sent per second is $1/t_{adv}$. Assuming a channel switching time of $t_{sw} \approx 100$ms (a typical default value), a switching dead time $t_{de} \approx 2$ms, then $t_{ad} \geq 3t_{sw}$, and typically t_{adv}=500ms. The likelihood $p(N_r \geq 1)$ that a receiver b receives a message from a (i.e., at least one advertisement packet was received) depends on the distance $r_{a,b}$, the send power P_t, the antenna gains G_t and G_r, the channel switching times T_{sw} of a and b, the receiver and sender dead times t_{de} , the total active advertisement time t_{ad}, and the packet send frequency $1/t_{sn}$.

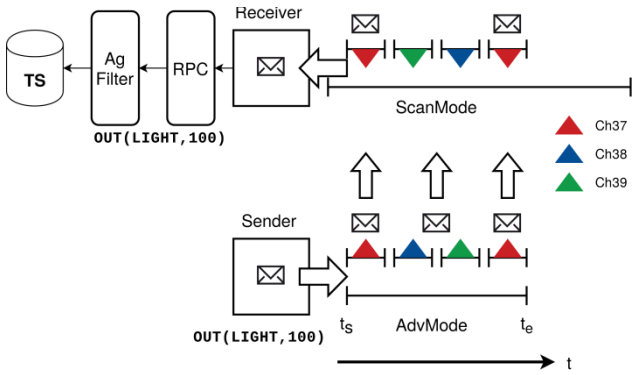

Figure 1. Principle BLE-based broadcast communication using advertisement packets. The sender and receiver switch their radio channels randomly and periodically. A packet m containing a tuple space request is sent multiple times in t ∈ [t_s,t_e].

UDP Back-end

In addition to the BLE broadcast communication, nodes that are connected to a local IP network can exchange tuple requests via UDP broadcast messages (called udp-ts). Although UDP messaging is not reliable and there is no acknowledgement of a packet delivery, the transmission probability for wired and switched connections is nearly 1 and mostly independent of the underlying network and the network load. In contrast, the BLE-based communication has a transmission probability of about 0.1-0.5 with a strong relationship to the radio communication load in the near neighbourhood of the stations. Wireless communication is different. Initially, the UDP back-end sends a message only one time. Therefore, a much higher bandwidth and lower latency can be achieved. However, experiments showed that some WLAN access points do not broadcast a received broadcast packet again to all connected client nodes. In dead, wireless UDP broadcasting is transformed in radio peer-to-peer multicasting. The probability p of a wireless connected device to receive UDP broadcast messages sent by other wireless connected nodes can be below 0.3. Therefore, and optionally, the sending of UDP broadcast packets can be repeated like in the BLE back-end. The delta time is chosen randomly within a time interval [1,10] ms. Nodes connected in a wired way via LAN do not show this packet dropping. All nodes connected in a wired way to the WLAN access points receive all radio broadcasted UDP messages.

V. SECURITY

On one hand, strong security in a non-negotiated and connectionless broadcast medium with message packet sizes below 64 Bytes is nearly impossible. On the other hand, this communication architecture and software framework should be deployed in public buildings and city environments controlling critical infrastructure, e.g., controlling ambient light intensity by user demands and user mobility. Any device can send tuple requests without any prior authorisation or authentication (smartphones do this continuously). Today, billions of people using pandemic contact tracing Apps are exactly doing this and are filling the air with broadcast advertisement messages (although, not all contact tracing applications rely on this methodology). So, the reception of broadcast mes-

 ISBN: 978-1-61208-962-1

sages cannot be prevented, and the major layer of security must be handled by the filter agents. Any time a message arrives from a sender there is a unique MAC identifier that is annotated to each message. One approach is a list of authorised devices that are handled in groups by different agents. But the MAC identifiers must be transferred to all devices in the group in advance, which is not suitable for mobile ad-hoc networks.

A second approach uses symmetric two-way encryption with a private and a public key pair. The messages are encrypted using the private key (only known by trusted devices and users) and decrypted in the receiving device. But, due to the hard data size constraints, only Format-Preserving Encryption (FPE) can be applied. Security and encryption is not addressed in this work. BeeTS implement a simple FPE algorithm that is able to encrypt and decrypt short data messages without compromising communication bandwidth or latency. The FPE algorithms can use any alphabet domain capable to encrypt both ASCII and binary data. It uses the aes-256-cbc algorithm. The encryption and decryption each require only about 0.05 µs/Byte on a typical desktop computer. Each tuple space can be protected with its own protection key and processing encrypted messages. An encryptor can be created on the fly (and used by agents, too). The encryptor is integrated in the BLE and UDP *rpc* modules before sending and after receiving a raw message. The encryption maps each data byte to an encrypted data byte independently using look-up tables. This kind of encryption is fast with low computational overhead, but is not safe against brute-force attacks. The mapping tables are created by using a user defined secret key.

VI. BeeTS

A. Heterogeneous Networks

The principal network architecture combining Bluetooth and UDP-IP broadcast communication technologies is shown in Fig. 2. Tuple messages can either be sent via Bluetooth advertisement (based on [6]) or via single UDP packets within a local IP network. BeeTS is programmed entirely in JavaScript [11] and can be executed by node.js with a bluetooth socket modules for BLE access, the *noble* module for the central BLE part, and *bleno* for the peripheral part [12]. Note that BeeTS uses the peripheral and central (master) mode simultaneously (advertising and scanning), requiring a Bluetooth device with version ≥ 4.0. BeeTS is basically a small library module written in JavaScript.

Smartphones act as mobile devices and provide both a rich set of sensors and BLE connectivity. Each communication back-end can receive tuple requests. If there is a listener installed for tuples (with pattern matching), incoming tuples (*TUPLE* message) can be consumed by the listener or not. Otherwise, incoming tuples are stored in the local tuple space.

There are agents acting as a bridge between the communication back-end and the tuple space. They can filter incoming messages and decide to reply immediately, to access the tuple space, or to discard the message. Agents are functional code that listen to incoming tuple requests. There can be more than one agent. Communication between agents is established via the tuple space, too.

A broadcast message sending via BLE enables the advertisement mode of the device for a specific time interval $[t_s,t_e]$. The duration of the time interval Δt determines the receiving probability, the collision probability (if more than one station is sending), the number of advertisement packets that contain the message *m*, and the number of different messages that can be sent per second. The interval time Δt must at least $3 \times t_{sw}$, with t_{sw} as the average channel switching time of the sender (and receiver). It is assumed that the sender and the receiver have the same switching time, typically 100µs. It is important to note that channel switching introduces small dead time intervals (about 1-10ms). A suitable value for Δt is about 500ms.

Each physical communication interface (BLE/UDP) is attached to its own tuple space, i.e., there are two distributed space sets connected via BLE and UDP, respectively. This division is grounded in the spatial context of tuple spaces. Using BLE communication only nearby nodes can insert or remove tuples, whereas UDP communication enables tuple exchange over short and large distances, too. Tuple exchange between BLE and UDP tuple spaces is provided by a customisable router, shown in Fig. 2. Application-specific routing rules (functional code) provide transfers based on patterns and content of tuples. The rule set is dynamic and can be changed at run-time. The router extends the visibility and scope of tuples based on adaptive code. The code can use Machine Learning, e.g., reinforcement learning, to improve tuple space distribution. The routers connect local spaces and compose organised global spaces.

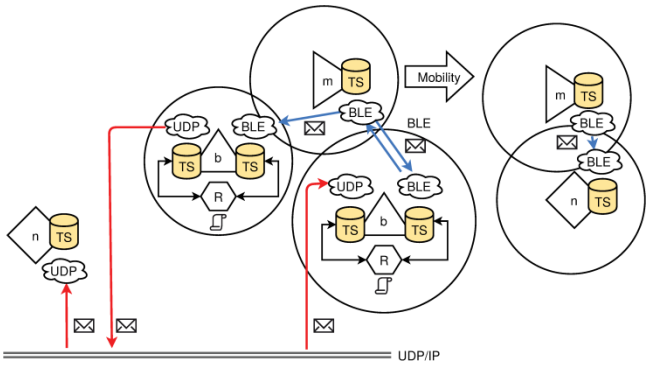

Figure 2. The hybrid network architecture using BLE and UDP-IP broadcast communication; n: stationary node, b: Stationary beacon, m: Mobile node, TS: Tuple space, R: Rule-based router

B. Programming API

Each time a message is received, it is passed to the Remote Procedure Call (RPC) layer. Among the message data, the sender MAC ID, a time stamp, and the signal strength is added to the message. The programming API is rather simple. A tuple space is created for each communication back-end. For each device back-end there is the same set of operations that can be applied on the tuple space: *out* (persistent), *notify* (not persistent), *rd* (preserving), *inp* (destructive). All these operations create broadcast messages. To access the

local tuple space directly, there is a mirror operational set accessed by the *host* attribute.

VII. AGENTS

The BeeTS framework basically provides a communication platform using radio communication like Bluetooth or WLAN. The communication bandwidth of various devices can be significantly limited (e.g., in the case of Bluetooth advertisement mode that can be 2 messages/second only). One main feature of BeeTS nodes is the capability to execute event-based reactive agents programmed in JavaScript that perform, e.g., filtering of incoming tuple space requests, shown in Fig. 3. Agents can act as a bridge between different local tuple spaces, i.e., between ble-ts and udp-ts.

An agent is functional data consisting of private body variables (including functional values) and event handlers that are activated by incoming messages, sensors (if the host platform provides them), periodically, or only one time. An event handler rule consists of an event expression that activates the rule and the hander function that is called on activation. An event rule activation expression can select three different event classes (*ts*, *sensor*, *timer*) with an optional conditional expression. Conditional expressions can access event source variables (actual sensor value, previous sensor value, tuple elements, time). Tuple listeners can consume tuples depending on the return value of the handler function. Not consumed tuples are stores in the local tuple space (if delivered by an *out* operation).

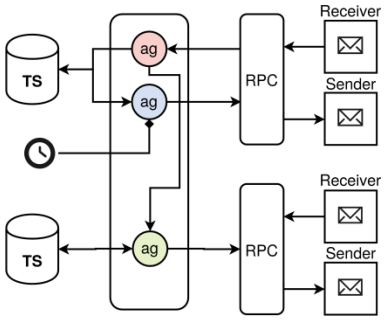

Figure 3. Agents create a bridge between tuple spaces with multiple communication devices (Red: Message event agent, Blue: Timer agent, Green: Router agent).

Agent event handler functions are application specific and can be loaded remotely at run-time via a service API using active tuples. Agents can access the tuple spaces and communication interfaces programmatically as well as sending HTTP(S) requests to external services. The operations that have to wait for a reply always operate asynchronously with a callback function either called on a reply or on a timeout with empty data. As discussed in the next section, the tuple messages that can be sent via BLE have a strict size limit below 40 Bytes. Additionally, only a few independent tuple messages can be sent per second (typically 2/s). This requires progressive and tight scheduling of tuple messaging. This is a typical constraint solving problem that is performed by the multi-agent system, too. The agents have to satisfy the quality of service, e.g., a distributed human-interactive light managements in buildings (shown in the use-case section).

Agents are executed in sand-boxed containers. Since JavaScript objects and function can be serialised to text and deserialised at any time agent snapshots containing actual agent data can be sent with active tuples to other nodes, i.e., tuples that contain code. Typically, encryption is used to secure agent migration.

VIII. DISTRIBUTED SMART BUILDING CONTROL

This use-case deploys three different node classes implementing a distributed building light control system:

- Stationary beacons (Raspberry 3) equipped with BLE and WLAN connectivity and supporting ble- and udp-connected tuple spaces in both test and production deployment;
- Mobile devices (battery powered RP Zero stacked with a smartphone for testing, stand-alone smartphone in production systems) supporting ble- and udp-connected tuple spaces in test and ble-connected tuple spaces only in production environments;
- A central monitoring and light control service supporting UDP-connected tuple spaces.

The network architecture and experimental set-up is shown in Fig. 4. Each node deploys at least one event-based agent that implements necessary node operations like interaction with mobile devices or users, and tuple filtering and bridging. Beacons consume and aggregate mainly sensor data from mobile (sensorised) devices like smartphones and forward micro-surveys from the central server to mobile devices. But beacons can initiate and manage micro-surveys, too. To minimise the number of sent tuples via the BLE device, the mobile nodes monitor the user behaviour by analysing the accelerometer and gyroscope sensors. Updates of light sensor tuples are only sent if either the light conditions changes or the mobile device was moved in space. For rapid prototyping, smartphones are using generic Web browser loading an application page from the locally attached Raspberry PI zero bundled with the smartphone. All sensor data is sent to the embedded computer that executes the mobile application logic and that performed the BLE communication.

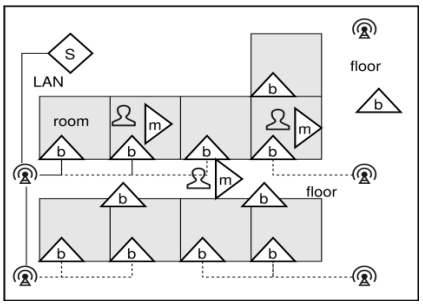

Figure 4. Experimental Set-up; s: Central server, b: beacon (Raspberry PI 3), m: Mobile device; Central server and beacons communicate via WLAN or LAN.

Mobile devices use their light sensor in conjunction with accelerometer and gyroscope sensors to estimate the ambient light conditions and the user mobility by classifying the user activity in rest, smartphone use, and movement phases.

 ISBN: 978-1-61208-962-1

The measured light sensor data is processed by a sensor agent that tries to estimate if the smartphone is currently exposed to external light or if it is stored in a box. If external light is detected, sensor light tuples ⟨"SENSOR", "LIGHT", value, time⟩ are sent via BLE. Nearby beacons distributed in the building about every 10-20m (and one per room/floor) collect these tuples and send aggregated sensor light values to the central server via UDP-connected tuple spaces.

Among sensor tuples, there are micro-survey tuples that are sent from a beacon (initially delivered by the central server via the UDP tuple space) to mobile devices. If a device supports HMI (e.g., a smartphone), a short question is posted to a chat dialogue platform embedded either. The user can answer the question and the answer is passed back to the beacon (or any other beacon due to movement). The beacons collect the micro-survey replies and forward them to the central server.

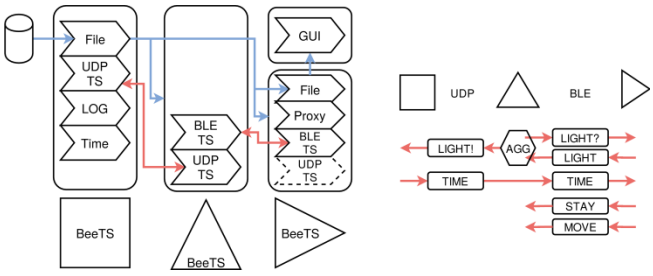

Figure 5. (Left) Communication and service layer s (Right) Tuple flows.

The loss of tuple messages either due to out-of-reachability of beacons, or by collision, or by uncorrelated sender-receiver channel pairing, does not compromise the quality-of-service of the light control in the single rooms and floors. In average, 80% of the sent tuples were received and processed by the beacons. Mobile devices sent about 1 tuple message per second with an average minimal radio range availability of about 10 seconds (due to movement of the users, if any).

Using the light sensors, the mobility assessment using the accelerometer and gyroscope sensors of the mobile devices, and the performed micro-surveys providing user feed-back (satisfaction assessment), the illumination conditions could be optimised with respect to the user demands and energy consumption of about 30% without negative user feed-back and dissatisfaction.

For the evaluation of the loss rate of BLE tuple space communication, a partial set-up was chosen with four beacons at four different spatial positions and four mobile devices here all at the same position. The results of the measured average reception rate *R* (loss is 100-*R*) show an average loss of 10% that can be achieved within a radius of about 2m, 30% in 5m radius. Some nodes can communicate over larger distances up to 10m. The tuple message sending time interval has no significant impact on the loss rate within time interval [500s, 2000s] and with this (small) set-up. When the number of nodes within the radio range will be increased, the loss rate will increase, too. Broadcasting from LAN to WLAN did not work reliable (loss up to 50%), and even unicast UDP communication was not reliable via WLAN. This is a limitation of WLAN communication, although like BLE it is a broadcast medium, all device-AP communication is established as a peer-to-peer connection. A server has to simulate broadcasting by sending N:1 unicast messages.

IX. CONCLUSION

In this work, distributed tuple spaces were used to exchange data between devices providing a spatial context. Smart devices access the tuple spaces by tuple message communication using event-based agents. The communication is connectionless and ad-hoc by BLE broadcasting available in any Smartphone and in most embedded computers. Bi-directional connectionless communication is established via the advertisement and scanning modes by transferring encoded tuple messages. BeeTS enables opportunistic, ad-hoc and loosely coupled device communication with a spatial context. Multiple independent tuple spaces can be serviced on one network node. A use-case study showed the suitability of the broadcast communication for distributed ad-hoc networks preserving a spatial context lacking in other approaches. The spatially averaged loss rate was below 20%.

REFERENCES

[1] C. Maguire, “Attendance Tracking using Bluetooth Low Energy-Enabled Smartphones”, University of Dublin, https://www.scss.tcd.ie/publications/theses/diss/2018/TCD-SCSS-DISSERTATION-2018-015.pdf, 2018

[2] M. Cunche, A. Boutet, C. Castelluccia, C. Lauradoux, and V. Roca, “On using Bluetooth-Low-Energy for contact tracing”, Report, Inria Grenoble Rhône-Alpes; INSA de Lyon. 2020. hal-02878346v5

[3] L Reichert, S. Brack, and B. Scheuermann, “A Survey of Automatic Contact Tracing Approaches Using Bluetooth Low Energy”, Cryptology ePrint Archive, 2020.

[4] J. Li and X. Guo, “COVID-19 contact-tracing apps: A survey on the global deployment and challenges”, arXiv:2005.03599, 2020.

[5] B. Skočir, G. Papa, and A. Biasizzo, “Multi-hop communication in Bluetooth Low Energy ad-hoc wireless sensor networks”, Journal of Microelectronics, Electronic Components and Materials vol. 48, no. 2, 2018.

[6] M. Nikodem and M. Bawiec, “Experimental Evaluation of Advertisement-Based Bluetooth Low Energy Communication”, Sensors, vol. 20, no. 107, 2020.

[7] N. Davies, A. Friday, S. P. Wade, and G. S Blair, “L2imbo: A distributed systems platform for mobile computing”, Mobile Networks and Applications, vol. 3, no. 2, pp 143-156, 1998.

[8] S. Bosse, “Unified Distributed Sensor and Environmental Information Processing with Multi-Agent Systems: Models, Platforms, and Technological Aspects”, ISBN 9783746752228, Epubli, 2018.

[9] A. Bröring, A. Remke, and D. Lasnia, “SenseBox – A Generic Sensor Platform for the Web of Things”, EAI MobiQuitous, 2012.

[10] https://www.novelbits.io/bluetooth-low-energy-advertisements-part-1, on-line, [accessed June 2022]

[11] https://github/bsLab/beets, on-line, [accessed June 2022]

[12] https://github.com/noble/bleno, on-line, [accessed June 2022]

 ISBN: 978-1-61208-962-1

SmartHelm: Design and Implementation of Open Source based Behavioral Data Repository

Harish Moturu, Johannes Schering, Jorge Marx Gómez
Very Large Business Applications (VLBA),
University of Oldenburg,
Oldenburg, Germany.
emails: {harish.moturu, johannes.schering, jorge.marx.gomez}@uni-oldenburg.de

***Abstract*—Data is the driving force in the current world of digitalization for efficient decision making. Data availability plays a key role in almost every research field for improving data understanding and knowledge acquisition. There are many ways in which data can be made available to end users. In some cases, the user may be required to pay for using the data: similarly, there are some other data resources available as open source. In the case of sensitive data such as business related, personal and bio-physical data, there are certain challenges in making these kind of data publicly available. This paper addresses exclusively how an open data portal enables users to freely access bio-physical data that is developed as part of the SmartHelm research project. The open data portal *SmartHelm Behavioral Data Repository* constitutes of two key components: (1) a data management platform that organizes and structures the raw data into a well defined structural form and (2) a RestAPI server that has the functionality of providing the data in the desired format to the end user. The first prototypical version of the data portal constitutes of data obtained from experiments conducted in the project. These data sources are ElectroEncephaloGraphy (EEG), eye tracking and geographical location/ Global Positioning System (GPS) data. The main advantage of this data repository is that it facilitates the users to access all kinds of data without any special limitations such as user account, subscriptions etc. This bio-physical and mobility data will not only help other researchers working in this field, but also it will help municipalities to improve the bicycling infrastructure at the certain junctions in the city where the cargo bike logistic rider experiences more stress.**

***Index Terms*—bio-physical-data, EEG, Eye Tracking, data repository, behavioural data, bicycle data, mobility data, open data.**

I. Introduction

New bio-physical data may have a positive influence on different aspects of life. One topic that is often discussed is livability and sustainability of future cities [1] [2] . Knowledge about distraction factors in mobility applications could contribute to increase traffic safety and to adjust infrastructures to the demands of real users. A central problem here is that related data sets are often not available or at least not openly available on the Internet. In addition, the combination of behavioral data and mobility data is hard to find. One reason is that the behavioral data that is discussed in this contribution (EEG, Eye Tracking) is often gathered as part of scientific works under laboratory conditions. Another factor is that available mobility related data is often not openly available, but remains inside organizations because of data privacy or security concerns. To find a solution to this problem, the University of Oldenburg has developed a new data repository related to behavioral data. This includes an open data component to enable a complete open accessibility to the data sets, a visualization component to display the results in the form of geographic maps and Key Performance Indicators (KPIs) to make different data sets comparable. In this contribution, we focus on the aspect of the data portal. The development and the working steps of the implementation of the data management system (back-end) and the website (front-end) will be described in detail. The development and calculation of KPIs will be discussed in further publications. This contribution consists of the following individual sections. Section II is a survey of existing literature in the research field we are discussing in this contribution. It provides an overview about existing platforms related to behavioral data to identify the research gap. As we will see, existing solutions are not totally open, which means that the user has to register to get access to the data sources. Registration and reading user instructions takes time and could decrease user satisfaction. Existing website solutions do not seem to be very intuitive. It takes time to learn about functionalities and opportunities of the portals. In addition, mobility data in combination with behavioural data is not available so far. Mobility related distraction data could be a useful contribution to support practical and scientific applications. Section III gives an overview about the implementation of the behavioural data management system of the SmartHelm [23] research project. This includes the working steps of data integration, data processing, data visualization and publication. Section IV describes the functions and services of the behavioral data repository in detail. Section V are summarizing the preliminary results of the data management and data analysis in the SmartHelm project. The state of the art which is presented in the literature review section is compared with the functionalities of the behavioral data repository in Section VI. Section VII summarizes the results of the article and gives an outlook to the future development.

II. Related work

This next section describes the results of the literature survey carried out on the current state of the art of relevant platforms that provide human behavioural data. [3] lists out

 ISBN: 978-1-61208-962-1

different online based platforms for sharing and analyzing behavioural data. It describes in total 20 online platforms including data sharing, data visualization, offline or online analysis, and community cooperation. Among them, a few data repositories are short-listed in our work depending upon the type of data availability and relevance. Here, the factors considered for short-listing are if the type of data in the portals is bio-physical, sensory, mobility data and also the mode of accessibility of the data to the general public.

The ElectroEncephaloGraphy (EEG) / Event-Related potentials (ERP) based portal EEGbase [25] described in [4] - [6] is built as an open source web based data platform to store both raw and meta bio-physical data. This portal combines Semantic Web architectures such as Resource Description Framework (RDF) and Ontology Web Language (OWL) with common data structures. The developing methodology is based on the standalone relational data models. Although the main aim of the EEGbase portal is to provide open-source EEG/ERP data, users need to register and to create an account in order to access the data sources. This drawback is considered in our current SmartHelm Behavioural data repository, where end-users may directly access the data sources without any limitations. Another data platform referenced in our work is Neurobot [25] which is discussed in [7]. Neurobot is a web-based platform to publish neuroscience data for research and is developed in focus with Study data managers, Researchers, and Platform administrators. This platform does not only store and share the data, but also allows other researchers to upload data-sets and to contribute for further research. The key features of Neurobot [7] are data dictionary, user management, saved search, data query tool and API access. These additional features enhance the organization and increase the searching ability of the platform. Similar to EEGbase, the platform is not easily accessible because it requires an user account. Contributions [8] - [10] describe about the language independent object model Neuroscience Electrophysiology Objects (Neo). The model Neo [26] provides common representation of data acquired from electroencephalographic, intracellular or extracellular experimental recordings, or those that were generated from simulations. Unlike the platforms described in the previous papers, Neo is a standlone package developed as an open source software in Python programming language. The main functionality of Neo is to read and write different formats of electrophysiological data into a common representation, which would enhance interoperability and facilitate data-sharing. Although Neo is not exclusively a web or software based platform for data sharing, it indirectly acts as an open source component for building electrophysiological data sharing platforms. Another innovative EEG data sharing platform is IEEG-Portal [27] . It is a cloud based collaborative research platform developed for sharing, analyzing and visualizing data. [11] - [14] illustrate the methodology and technology behind the IEEG portal. One of the most significant features of IEEG portal is that it stores numerous large sized EEG data-sets based upon google web toolkit and Amazon S3 service. Due to the cloud feature, this

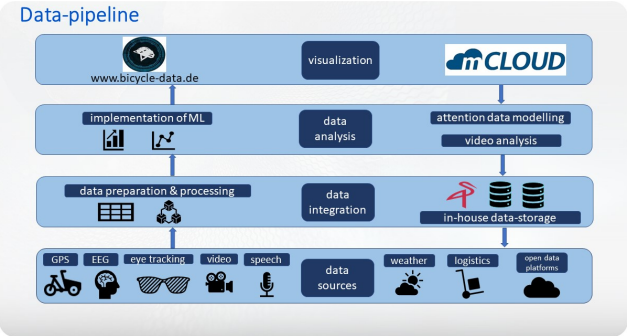

Fig. 1. Four layers architecture representing the data flow and their corresponding significance

platform facilitates the storage of data in terms of Terabytes. A disadvantage is mainly that the data privacy issue is not well explained. As the data is stored in the cloud, additional security and privacy measures need to be followed in order not to have any kind of data breach. Another drawback is that users need to have an individual account to access the data. Finally, the last paper in the literature review section is [15]. [15] it introduces an ontology based application Ontology of Heart Electrophysiology (OHE) portal for study of heart related biomedical activities. This ontology based tool provides a web based application for analyzing and to visualizing using the data acquired from heart electrophysiology data. This tool is an advancement in the bio-medicine sector as this type of ontology portals usually are not publicly available.

III. Methodology

The principle methodology to develop the SmartHelm open source based data portal is illustrated in this section. It explains the related data pipeline or data flow from data collection and processing to visualizing the results in various forms. Fig 1 shows the architecture behind the developed platform. In this section, each layer is explained in a separate paragraph.

In Fig 1, at the bottom is the source layer, where all relevant data sources are gathered and stored in the raw form in a project-owned storage server at the University of Oldenburg. The bio-physical data of the participants is critical from both ethical and legal perspectives and needs to be handled carefully. Due to data privacy, this database is not supposed to be stored in an external cloud server. Therefore, in the source layer, an open-source based simple storage solution (S3) client MinIO is deployed on the storage server to store the raw data on S3-based object storage. This MinIO client serves as a Data Lake to store the raw data. In principle, core data contains EEG sensors, eye-tracking, speech input commands, Global Positioning Systems (GPS) signals, and distraction marker data streams acquired from a group of subjects. Additional external mobility data sources for the data analysis are also integrated in the system depending upon the requirement. These include weather data [28], open

 ISBN: 978-1-61208-962-1

street map and open bicycle data [29], historical accident data from the city of Oldenburg which details the various types of accidents happening at different crossings, road-sections and narrow bicycle lines in Oldenburg. This open bicycle data is part of an open data portal which contains a comprehensive set of bicycle data such as bicycle counting, parking and near accident data. The data warehouse and the related website were developed as part of a student project Bicycle Data. On top of the source layer is the second layer known as an integration layer, which is further described in the following section.

The Integration-layer serves the purpose of aggregating all the heterogeneous data sources and storing the data in a data management system. The process of aggregation of the data sources is implemented in a number of steps, as described in the following. The primary step is pre-processing the raw data sources of the source layer. Individual data sources are pre-processed to view type, format, variation and important features lying among the different data sources. The output from the pre-processing step is selecting the detailed information individually from all the available data sources. The next step is designing the schema for the data management system. For conception of the schema, a detailed state-of-the-art study was carried out. Among these are the methods for data warehousing design [16], recent trends in the conceptual schema design [17], and Extraction Transformation Loading (ETL) methodologies & [18] and [19] are considered as reference design architectures. Therefore, as a benchmark from the reference methodologies, in this second step, a schema design is developed by considering the time as the central dimensional parameter in the data warehouse. Based on the central design parameter described in the previous step, the third step transformation script is developed according to the reference ETL methodologies [18] [19]. The transformation step constitutes each form of the heterogeneous data according to the schema structure. The pre-processed data sources are transformed depending on the steps in a data pipeline. That means that each and every heterogeneous data source is extracted from the Data Lake and then transformed separately to fit into the schema. After the transformation is completed, each data source is uploaded to fit the individual tables in the schema. Therefore, the last step in the integration layer is to automate the above three steps in the loop to populate the data warehouse with all the available new data sources and to store the data that would be acquired in future experiments (in real traffic situation).

The data analysis layer can also be called the knowledge layer. From the variety of data stored in the data warehouse, different types of analysis can be performed such as attention modelling with EEG data, detection of Area of Interest (AOI) of the rider using the eye-tracking data and speech recognition system modelling. This layer serves the purpose of analysing the structured data stored in the Data Warehouse depending upon the goals of the research. Among these, some of the important methods are application of Machine Learning algorithms, implementation of Cross Industry Standard Process for Data Mining (CRISP-DM) [20], predictive analysis, and query based data analysis techniques. In the current paper, two types of analysis are illustrated. The first is the theoretical evaluation made on the feedback obtained after conducting interviews with experts and delivery professionals in cargo bike logistics and the second is analysing the data obtained from the experiments described above. As soon as attention related data from real traffic situations is available in a later stage of the SmartHelm project, the corresponding analysis with new Global Positioning Systems (GPS) routes will be carried out and published as a part of further publications.

The top most layer in the data flow is called the visualization layer. The results from our first visualizations are presented in Section 5. In general, this stage in the data flow and the graphical representation supports the understanding of the results and information obtained from the data analysis. The visualization layer, illustrated in Fig 2, consists of two functionalities a) publishing the raw data on an open data platform (i.e Data as a Service DaaS [21] [22]) and b) presentation of the results in various graphical forms on a dashboard. The implementation includes the development of a portal to download the raw data of the experiments and a dashboard for the visualization of the analysis results as part of the Bicycle Data website that was described earlier. To connect the data from the data warehouse to the respective visualization platform a Rest API service is developed.

IV. Functional components of the Data Repository

The next section describes the implementation and functional components developed based on the methodology described in the previous section. It focuses on the individual aspects of the portal i.e what kind of data will be made available on the portal, then how the portal design looks like and finally explaining in detail what are the functionalities it provides to the users.

A. Overview of the available data in the Repository

ElectroEncephaloGraphy – data : As mentioned in the previous section, the repository mainly consists of bio-physical data. Among these, the first data type is ElectroEncephaloGraphy (EEG) data which is acquired from a set of participants as part of scientific studies. In principle, raw EEG data consists of brain signal values obtained from 7 electrodes in the form of impedance. The frequency of the impedance values is 250 Hz and the data is filtered in terms of the participant, data and session acquired. Beside impedance values from 7 electrodes, other parameters such as timestamps, session and location of experiment and participant related information are included in the raw data.

 ISBN: 978-1-61208-962-1

Eyetracking – data : The second type of the bio-physical data as part of the repository is eye tracking data. It consists of eye gaze values obtained from an Augmented Reality Glass (Microsoft HoloLens 2). In total, there are six eye gaze values calculated in x, y and z directions from the origin and the current position respectively. These eye-gaze values measure the 3 dimensional distance from the object to the participant viewing position, which further facilitates in estimation of the point of vision of the participant during the course of the experiment. In comparison with the EEG data, the frequency of data recording for eye-tracking data is less (around 30 Hz).

Location – data : During the experiment, the Global Positioning Systems (GPS) coordinates of the participant are logged while riding a bike using an open-source application known as OwnTracks [30]. The OwnTracks application logs the location information with various position based measurements namely altitude, latitude, longitude coordinates, speed of the rider, radius around the region, and including relevant metadata information. Here, the measurement frequency of Global Positioning Systems (GPS) data is significantly lower than the above two data types, with a rough estimate is 5Hz frequency. This location information along with other behavioral data is also made available on the open data platform for the end-user to download. In principle, location data enables the scope of mobility point of view on the data, which means behavioural data can be integrated and analyzed on the basis of location data.

Additional – data – sources : In addition to the above mentioned data sources, some another potentially relevant data sources are also prepared to be integrated in the data management platform. These are primarily mobility data such traffic volume data collected from various streets in the city of Oldenburg, which consists of counting information on the number of different types of transport vehicles crossing a particular traffic junction on an hourly basis. In addition to that, other types of behavioral data such as speech events, user interaction events and attention hot-spots data will also be integrated in the database.

B. Smarthelm API for data integration

In the methodology, integration layer describes the building of the data warehouse through integrating all the available data sources. The main functional component developed to access the data in the data warehouse is the API which is required to connect the data warehouse and the visualization component where the end-users have access to download and view the results. The API is developed based on the RestAPI technology using an open source Python web framework known as FastAPI. In contrast to other API framework technologies, one of the main advantages of FastAPI is the generation of automatic API docs depending upon the design of the download platform. The SmartHelm API doc shown in Fig 2 illustrates different kinds of accessible data parts

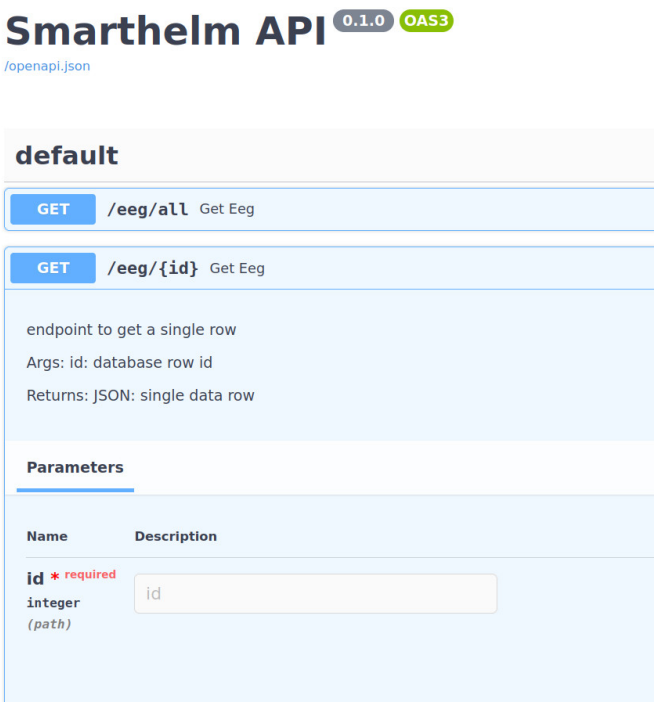

Fig. 2. An example from SmartAPI doc showing some of the available endpoints provided by the API where each endpoint serves the purpose of integrating the platform directly from the back-end

represented as end-points. Here, each end-point indicates one data element which can be accessed to pull the specific data queried by the end-user. The SmartHelm API doc Interface(Fig 2) in general provides the information about all kinds of data elements that are available for integration. The key functional utility of SmartHelm-API underlies in linking the API for other external projects and can be used as an open source service to integrate the behavioural data repository with the back-end. For example, if an external project has the requirement of accessing only the EEG data, the API facilitates a gateway to pull the specific EEG data. This end-points format can be obtained from the API doc. Fig 2 clearly shows a sample end-point on how to obtain the EEG data for a time interval.

C. SmartHelm Data platform User Interface

The key concept behind the SmartHelm Data Platform User Interface (UI) is providing intuitive user experience throughout the portal. Fig 3 and Fig 4 illustrate a quick insight into the data portal. The design is made using a simplistic approach with a wider range focus and more intuitive for the general public. Along with the research community, other stakeholders such as politicians, municipality officials, city traffic planners, and the general public can easily understand what kinds of data are openly available to access and their functionality. In addition, the other driving force for developing the open source data platform is to enable data availability in the form

 ISBN: 978-1-61208-962-1

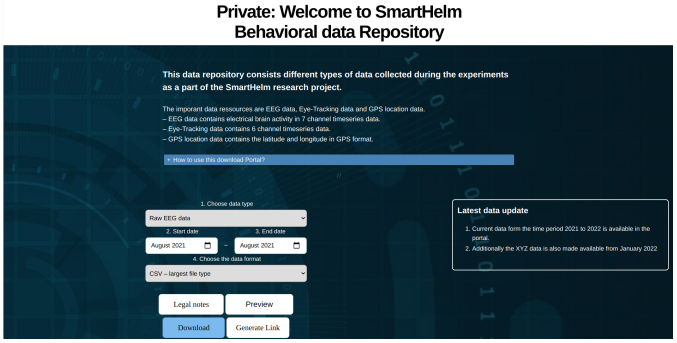

Fig. 3. This full size image from the front-end of SmartHelm Behavioural data Repository presents a complete overview of the portal which shows the description text and below is the data selection menu along with various functionality buttons.

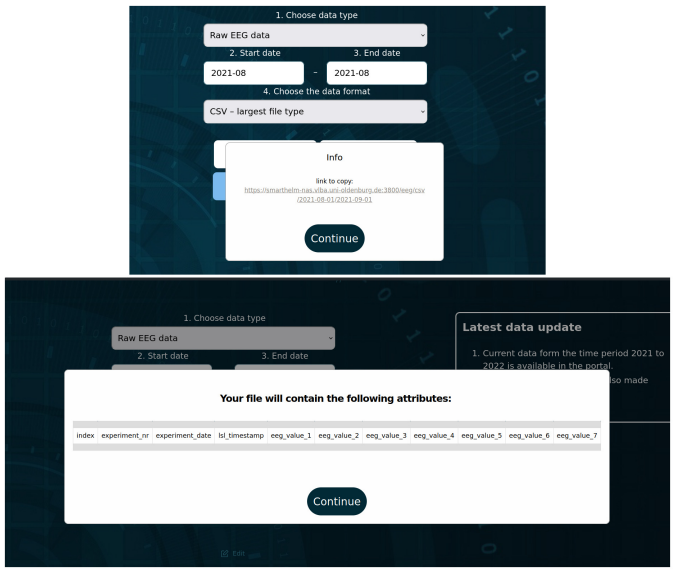

Fig. 4. The Fig 4 displays functionality of various buttons below the selection menu such as generate link on the top and preview of the selected data to download.

of Data as a Service (DaaS). The UI design in Fig 3 reflects the sole purpose of the platform and the steps taken to ensure the goal of providing Data as a Service to the end-user. In the following, the content and usage of the UI are described in detail. The header menu explains briefly to the end-user which different types of data sources are currently available for download. After the header menu, a quick 'how-to' steps menu is integrated to make it more intuitive for the end-user to understand the various steps involved from the selection of the data until downloading the data in the desired data. Below 'how-to' steps menu is the data downloading section. In this section, different types of drop-down boxes are available for the user different categories such as type of data, selection of date interval, type of the data format to be downloaded and so on.

V. Challenges in the development

The next section describes the different findings during the development of the open source platform which constitutes a blend of data collected from sensors and mobility systems.

The previous sections described technical difficulties for the implementation. Various challenges regarding data privacy and security problems have to be addressed before the technical implementation of the above represented high-level system architecture can take place. In this section, all the necessary aspects related to ethical and legal issues are described, which are taken into consideration before modelling the system. Ethical and legal issues play an vital role in the SmartHelm project. Permanent surveillance of the participants during experiments is not the intention of the project and needs to be avoided to increase the acceptability of the riders in the future. There is no kind of real-time data transfer during the experiments which guarantees that misuse by the intermediate user of the gathered bio-physical data (EEG, Eye-Tracking, etc.) can be ruled out. In the methodology section, we explained how the raw data is stored in the grass root level using the S3-storage(Data-Lake) technique, but before that step the data transfer from data collection to landing into our Data Lake is also a challenging step. Here, certain measures are evident for a secure data transfer between the two components. To safeguard this critical problem, dedicated user policies are created on the Data Lake server to facilitate the data collection supervisors to upload the data directly on the S3-storage-server, which bridges the data transfer gap without any external channel. Here, the policies are confined in such a way that the users who are uploading the data into the server have minimal access to the server, which means they do not have rights to delete or edit the existing data, but only upload new data sets. The next challenging factor specifically about the data is time synchronization between different data streams. As explained, the data is heterogeneous data and obtained from different sensors and mobile applications. There does not exist by default by an synchronization in time-steps for all kinds of data-streams. Maintaining an uniformity is a crucial factor for further processing.

VI. Advantages in comparison with other portals

First and foremost, the main advantage of SmartHelm behavioural data repository is the unique characteristics in comparison with the other bio-physical data platforms. From the Section 2 through literature study, it is evident that compatibility, end-user serviceability and degree of simplicity of the older bio-physical data platforms are considerably behind in some key aspects than the SmartHelm data portal. In this context, primarily comes user accessibility to the platform in SmartHelm data repository. All external visitors do not essentially require any kind of user-agreement before downloading the data as well as a separate user account is not needed to be registered on the portal. These is one among the two key entities which differentiate the SmartHelm download portal from other human bio-physical data portals. In addition, a few other components which enhance the portals' wider

 ISBN: 978-1-61208-962-1

scope of application are as follows. First comes the degree of intuitiveness for the user, which is explained in the previous section. Unlike other portals, the required information for the end-user is served without any heterogeneity. The next factor would be the user-acceptance, which means, to increase the user acceptance, the portal is designed in such a way that the end user has the complete control to be served by themselves by interacting with the portal. For example, the advantage of choosing the specific time interval in order to access the data represents the kind of options that increase the probability of user-acceptance. To achieve this, depending upon on the requirement, a user survey will also be conducted and the feedback will be considered to enhance the usability of the portal. The third additional factor would be the preview of the data before download. Although it is not a standalone functionality or a modern plugin, this kind of functionality is not found in other portals and it facilitates the user to obtain an overview of the data.

VII. Conclusion

The availability of mobility data integrated with behavioural data is a rare combination and very few research works acquire this kind of data. In addition, accessing such kind of sensible data through an open source medium would be more versatile. The SmartHelm behavioral data repository introduced in this work scientifically illustrated an open source platform consisting of an unique combination of mobility and bio-physical data. The problem statement i.e., the research gap described in the introduction section is that there exist few platforms which provide bio-physical data to the public, but still there are certain demerits which are clearly explained and the alternative solution is provided with the data portal developed in this work. The behavioural and mobility data integrated will be made accessible to the general public through the SmartHelm data portal as well as the mCLOUD / Mobilithek portal, which is provided by the German Ministry of Transport and Digital Infrastructure (BMDV). Moreover, some other benefits include the fact that city infrastructure planning of municipalities can be supported by these data sets during the improvement of road infrastructures or bike lanes. In the context to the cargo bike riders, data-driven decision-making probably increases work efficiency. Therefore, as an end-note to brief the further steps in the project, a visualization service is well planned in order to present the data analysis results in various graphical, visual forms such as heat-maps, digital city map, etc., which fulfills the main goal of the project to assist cargo bike riders by reducing the mental and visual stress in everyday parcel delivery. From a wide range perspective, this open accessible data to the public and publishing the results obtained from the overall project will, in return, encourage other major cities and municipalities in Germany to switch cargo bike logistics, which would become one of the major contributions to green logistics.

Acknowledgments

SmartHelm is funded by the German Federal Ministry of Transport and Digital Infrastructure (BMDV) as part of the mFUND program (project number 19F2105B) with a funding amount of around 1.48 Mio. Euro. As part of mFUND, the BMDV supports research & development projects in the field of data based and digital mobility innovations. Part of the project funding is the promotion of networking between the stakeholders in politics, business, administration and research as well as the publication of open data on the mCLOUD portal (which will be superseded by Mobilithek in 2022/23).

References

[1] E. Taniguchi, "Concepts of city logistics for sustainable and liveable cities." Procedia-Social and Behavioral Sciences no. 151, 2014, pp 310-317

[2] E. Taniguchi, & R. G. Tompson (eds.), "City Logistics 3: Towards Sustainable and Liveable Cities." John Wiley & Sons, 2018.

[3] Y. Chen, W. Zhong-yi, Y. Gang, and H. Lan, "An overview of online based platforms for sharing and analyzing electrophysiology data from big data perspective." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 7, no. 4 2017: e1206.

[4] P. Ježek, and M. Roman, "EEG/ERP Portal–Semantic Web Extension: Generating Ontology from Object Oriented Model." In 2010 Second WRI Global Congress on Intelligent Systems, vol. 3, pp. 392-395. IEEE, 2010.

[5] P. Ježek, and M Roman. "Database of EEG/ERP experiments." In HEALTHINF 2010-Proceedings of the Third International Conference on Health Informatics, pp. 222-227, 2010.

[6] P Brüha, and M Roman. "Portal for research in electrophysiology—Data integration with neuroscience information framework." In 2012 5th International Conference on BioMedical Engineering and Informatics, pp. 1099-1103. IEEE, 2012.

[7] R. Badia et al., "INCF Program on Standards for data sharing: new perspectives on workflows and data management for the analysis of electrophysiological data." In Techn. Report https://www.incf. org/about-us/history/incf-scientific-workshops, 2015.

[8] S. Garcia et al., "Neo: an object model for handling electrophysiology data in multiple formats." Frontiers in neuroinformatics 8 2014: 10.

[9] J. Grewe, T. Wachtler, and J. Benda, "A bottom-up approach to data annotation in neurophysiology." Frontiers in Neuroinformatics, 5 2011: 16.

[10] A. Sobolev et al., "Integrated platform and API for electrophysiological data." Frontiers in neuroinformatics 8 2014: p.32.

[11] JB. Wagenaar et al., "A multimodal platform for cloud-based collaborative research." In 2013 6th international IEEE/EMBS conference on neural engineering (NER), pp. 1386-1389. IEEE, 2013.

[12] L G. Kini, A. Davis Kathryn, and B. Wagenaar Joost, "Data integration: Combined imaging and electrophysiology data in the cloud." Neuroimage, 124 2016: pp. 1175-1181.

[13] M R. Bower et al., "Metadata and annotations for multi-scale electrophysiological data." In 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 2811-2814. IEEE, 2009.

[14] B H.Brinkmann et al., "Multiscale electrophysiology format: an open-source electrophysiology format using data compression, encryption, and cyclic redundancy check." In 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 7083-7086. IEEE, 2009.

[15] B. Gonçalves, Z. Veruska, G. Giancarlo, and G G.P.F. José, "An ontology-based application in heart electrophysiology: Representation, reasoning and visualization on the web." In Proceedings of the 2009 ACM symposium on Applied Computing, pp. 816-820. 2009.

[16] A. Bauer, and G. Holger. Data-Warehouse-System: Architecture, Development, Application, dpunkt. verlag, 2013.

[17] P. Vassiliadis, A. Simitsis, L. Liu, and M. Özsu, Encyclopedia of Database Systems. New York: Springer, https://doi.org/10.1007/978-1-4899-7993-3 2009, pp.1095-1101.

 ISBN: 978-1-61208-962-1

[18] P. Digra, P. Abrol, and P. Lehana, Design and Development Of Distributed Image Acquisition and Recording System for Network Based Appli-cations. In International Journal of Scientific and Technical Advancements 4, no. 3, 2018, pp.1-8.
[19] A. Dearmer, Complete Guide to Database Schema Design https://www.xplenty.com/blog/complete-guide-to-database-schema-design-guide/ Published:16th of July 2021. Last Access: 29th of November 2021.
[20] R Wirth, and H. Jochen, "CRISP-DM: Towards a standard process model for data mining." In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining, vol. 1, pp. 29-40. 2000.
[21] Z. Zheng, Z. Jieming, and R. Lyu. Michael. "Service-generated big data and big data-as-a-service: an overview." In 2013 IEEE international congress on Big Data, pp. 403-410. IEEE, 2013.
[22] X Wang, T. Yang Laurence, L Huazhong, and M. Jamal Deen. "A big data-as-a-service framework: State-of-the-art and perspectives." IEEE Transactions on Big Data 4, no. 3 2017: pp. 325-340.
[23] Research project SmartHelm official website, funded by German Federal Ministry for Digital and Transportation; 2022; accessed on 15.March. 2022; website url = https://smart-helm.com/
[24] EEGbase, RRID:nif-0000-08190;year 2022; accessed on 12.March.2022; URL = http://eegdatabase.kiv.zcu.cz/homepage;jsessionid=10r3hyplemadg?0
[25] International Neuroinformatics Coordinating Facility (INCF); 2022; accessed on February 2022; URL = https://www.incf.org/form/neurobot-user-account
[26] Neo, 2010-2022, accessed February 2022; URL = https://neo.readthedocs.io/en/stable/io.html
[27] IEEG portal, 2022; accessed on February 2022; URL = https://www.ieeg.org/
[28] Visual Crossing Corporation, Visual crossing weather API; year 2022; accessed on March 2022; URl= https://www.visualcrossing.com/weather-api
[29] Project Group Bicyle Data, VLBA, University of Oldeburg; year 2021; accessed on February 2022; URL= https://www.bicycle-data.de/bicycles-data/
[30] Owntracks MQTT broker application, year 2022, accessed on Febraury 2022, URL= https://owntracks.org/booklet/tech/json/

 ISBN: 978-1-61208-962-1