SMART ACCESSIBILITY 2019

The Fourth International Conference on Universal Accessibility in the Internet of Things and Smart Environments

ISBN: 978-1-61208-691-0

February 24 – 28, 2019

Athens, Greece

SMART ACCESSIBILITY 2019 Editors

Lasse Berntzen, University College of Southeast Norway, Norway
Daniela Marghitu, Auburn University - Auburn, USA

SMART ACCESSIBILITY 2019

Forward

The Fourth International Conference on Universal Accessibility in the Internet of Things and Smart Environments (SMART ACCESIBILITY 2019) was held in Athens, Greece, February 24 - 28, 2019.

There are several similar definitions for universal accessibility, such as design for all, universal design, inclusive design, accessible design, and barrier free design. These and similar approaches are relevant to this conference. The focus will be on methods, tools, techniques and applications for human diversity, social inclusion and equality, enabling all people to have equal opportunities and to participate in the information society.

The accepted papers covered topics such as accessibility by design, digital inclusion, accessibility devices and applications. We believe that the SMART ACCESIBILITY 2019 contributions offered a large panel of solutions to key problems in areas of accessibility.

We take here the opportunity to warmly thank all the members of the SMART ACCESIBILITY 2019 technical program committee as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and efforts to contribute to the SMART ACCESIBILITY 2019. We truly believe that thanks to all these efforts, the final conference program consists of top quality contributions.

This event could also not have been a reality without the support of many individuals, organizations and sponsors. In addition, we also gratefully thank the members of the SMART ACCESIBILITY 2019 organizing committee for their help in handling the logistics and for their work that is making this professional meeting a success.

We hope the SMART ACCESIBILITY 2019 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the universal accessibility field.

We also hope that Athens provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

SMART ACCESIBILITY 2019 Chairs

Lukas Smirek, Stuttgart-Media University, Germany
Kristin Skeide Fuglerud, NR–Norsk Regnesentral, Norway

Christian Bühler, University of Dortmund, Germany
Daniela Marghitu, Auburn University - Auburn, USA
Simeon Keates, University of Greenwich, UK

SMART ACCESSBILITY 2019

COMMITTEE

SMART ACCESSIBILITY Advisory Committee

Lukas Smirek, Stuttgart-Media University, Germany
Kristin Skeide Fuglerud, NR–Norsk Regnesentral, Norway
Christian Bühler, University of Dortmund, Germany
Daniela Marghitu, Auburn University - Auburn, USA
Simeon Keates, University of Greenwich, UK

SMART ACCESSIBILITY 2019 Technical Program Committee

Mehdi Ammi, LIMSI-CNRS | University of Paris-Saclay, France
Margherita Antona, ICS-FORTH, Greece
Djamila Aouada, University of Luxembourg, Luxembourg
Lars Ballieu Christensen, Sensus ApS, Denmark
Anthony Lewis Brooks, Aalborg University Esbjerg, Denmark
Christian Bühler, University of Dortmund, Germany
Paloma Cáceres, Universidad Rey Juan Carlos, Madrid, Spain
Vítor Carvalho, IPCA & Algoritmi Research Centre, Portugal
Anat Caspi, Paul G. Allen School of Computer Science and Engineering - University of Washington, USA
William Chu, Tunghai University (THU), Taiwan
António Coelho, FEUP / INESC TEC, Portugal
Lynne Coventry, Northumbria University, Newcastle, UK
Mateus David Finco, Federal University of Paraíba, Brazil
Maria de los Angeles Cosio Leon, Universidad Autónoma de Baja California, Mexico
Carolina Del Valle Soto, Universidad Panamericana - Campus Guadalajara, Mexico
Antonio Diaz Tula, University of São Paulo, Brazil
Iyad Abu Doush, American University of Kuwait, Kuwait
Duarte Duque, Polytechnic Institute of Cavado and Ave, Portugal
Tamara Efrat, Bezalel Academy of Art and Design, Israel
Deborah Fels, Ryerson University, Canada
Stuart Ferguson, College of Health and Medicine - University of Tasmania, Australia
Esteban Guerrero, Umeå University, Sweden
Martin Heitmann, Berlin Institute of Technology, Germany
Susan Herring, Indiana University, Bloomington, USA
Beverley Hood, University of Edinburgh, UK
María Visitación Hurtado, University of Granada, Spain
Maria João Rodrigues, Universidade da Madeira, Portugal
Simeon Keates, University of Greenwich, UK
Adamantios Koumpis, Universität Passau, Germany
Georgios Kouroupetroglou, National and Kapodistrian University of Athens, Greece
Wenjuan Li, City University of Hong Kong, Hong Kong
Andre Luiz Brandao, Federal University of ABC, Brazil
Steven Mannheimer, School of Informatics and Computing - IUPUI, USA
Daniela Marghitu, Auburn University - Auburn, USA
Fernando Martin Sanchez, Instituto de Salud Carlos III, Spain
Paulo Martins, INESC TEC / University of Trás-os-Montes e Alto Douro, Portugal
Milton Mendes, Rural Federal University of the Semiarid - UFERSA, Brazil
Elisabeth Metais, Le Cnam, France

Copyright Information

Table of Contents

Predicting Uber Demand in NYC with Wavenet

Long Chen
Urban Big Data Center
Glasgow, UK
Email: long.chen@glasgow.ac.uk

Konstantinos Ampountolas
School of Engineering
University of Glasgow, UK
Email: Konstantinos.Ampountolas@glasgow.ac.uk

Piyushimita (Vonu) Thakuriah
Rutgers University
New Brunswick, NJ, USA
Email: p.thakuriah@rutgers.edu

***Abstract*—Uber demand prediction is at the core of intelligent transportation systems when developing a smart city. However, exploiting uber real time data to facilitate the demand prediction is a thorny problem since user demand usually unevenly distributed over time and space. We develop a Wavenet-based model to predict Uber demand on an hourly basis. In this paper, we present a multi-level Wavenet framework which is a one-dimensional convolutional neural network that includes two sub-networks which encode the source series and decode the predicting series, respectively. The two sub-networks are combined by stacking the decoder on top of the encoder, which in turn, preserves the temporal patterns of the time series. Experiments on large-scale real Uber demand dataset of NYC demonstrate that our model is highly competitive to the existing ones.**

Keywords*–*Anything; Something; Everything else.

I. INTRODUCTION

With the proliferation of Web 2.0, ride sharing applications, such as Uber, have become a popular way to search nearby sharing rides. Since the Uber demands changes over time at different regions that there is a gap of expectancy between the users and drivers. Various deep learning models have been proposed to facilitate time series data and achieved state-of-the-art performances in a great deal of real-wrold applications. For example, Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) typically employ a softmax activation function nodes to model correlations of series points. While generally effective, the models of this line, such as [16], largely rely on the hidden state of the previous nodes, which makes it hard to apply the parallel computing tricks within the series. Convolutional Neural Network (CNN) uses trainable receptive field unit to learn local shape patterns. However, most of these time-domain methods fail to consider external information of a time series, although some start to exploit it in a indirect manner.

Inspired by the recent success in a series of time series prediction, Wavenet [30], which employs a convolutional sequence embedding model, has demonstrated its ability of achieving highly competitive performance to the state-of-the-art, i.e., [8], [19]. The logic of Wavenet is to convert the embedding matrix as the "cache" of the previous interactions in a k-dimensional vector and deem the sequential pattens of the interaction as the signature of the "cache". Max pooling operation is then applied to learn the maximum value of the convolutional layer, so as to increase the scope of receptive field and tackle the issues of irregular input length. Fig. 1 depicts the key architecture of Wavenet. To this end, we present a new deep learning method for predicting the Uber demands

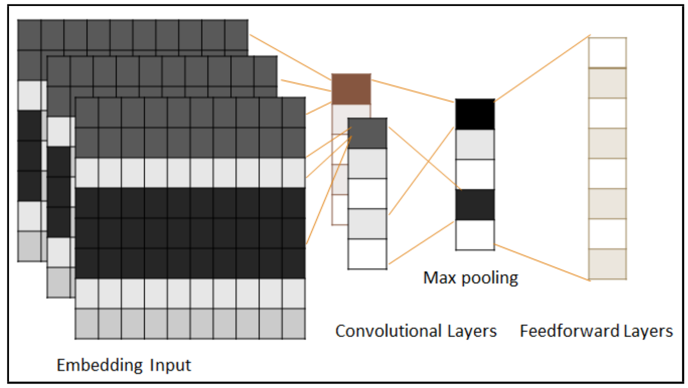


Figure 1. The structure of WaveNet, where different colours in embedding input denote $2 \times k$, $3 \times k$, and $4 \times k$ convolutional filter respectively.

at an hourly basis, which is a WaveNet-based neural network framework (UberNet) that exploits external features for time series analysis. Simply put, UberNet can utilize both WaveNet-based features (e.g., pickup demands) decomposition and the external features (days of week) to enhance the learning ability of deep neural networks. A max pooling layer is performed to "remember" the maximum value of the hidden layer, in order to magnify the power of receptive field.

The rest of this paper is organized as follows. We firstly defines the problem of Uber demand prediction with deep learning in Section II. Then, we introduce the related work in Section III. Section IV systematically presents the proposed framework for graph classification. The experimental results of prediction are reported in Section V. Finally, we present our conclusion and future work in Section VI.

II. PRELIMINARY

In this section, Wavenet is shortly recapitulated. Then, we formally introduce several concepts and notations. We introduce a novel neural network that is operated directly on the time series' waveform. The joint probability of a waveform, $x = x1, ..., x_T$, is given as the form of conditional probabilities as follows:

$$P(x) = \prod_{t=1}^{T} p(x_t | x_1, ..., x_{t-1})$$

Each datapoint x_t is therefore conditioned on the value of all previous timesteps. Similar to Wavenet, the core idea

 ISBN: 978-1-61208-692-7

of Ubernet is to embed the previous s timestamp as a $s \times k$ matrix I by taking the embedding query operation, as shown in Fig 1. Each row of the matrix mapped into the latent features of one timestamp. The embedding matrix is the “cache” of the s timestamp in the k dimensional vector. Intuitively, models of various CNNs that are successfully applied in series transformation can be customized to model the “cache” of an traffic demand sequence. However, there are two stark differences, which makes the use of Wavenet model counter-intuitive. First, the variable-length of the demand sequences in real-world entails a large number of “cache” of different sizes, whereas conventional CNN structures with receptive field filters usually fail. Second, the most effective filters for text cannot easily fit into the scenario of modelling demand sequence “caches”, since these filters (with respect to row-wise orientation) often fail to learn the representations of full-width embeddings.

Softmax Distribution One approach of learning the conditional distributions $p(x_t|x_1, ..., x_{t-1})$ over the individual timestep is to use a mixture density network. However, van den Oord et al. [2] showed that a softmax distribution gives a superior performance, even if the data is only partially continuous (as is the case for special events or holidays). A possible reason is that a categorical distribution tend to be more flexible which can sift out arbitrary distributions since it has no assumptions about the shape. We adopt the gated activation unit as the gated PixelCNN []

$$z = \tanh(W_{f,k} * x) \odot \mu(W_{g,k} * x) \quad (1)$$

where $*$ is the convolution operator, $\odot$ is an element-wise multiplication, μ represents a sigmoid function, k denotes the layer length, f and g are filter and gate, and W is a convolution filter, which will be detailed in the Section IV. In our preliminary experiments, we found that this non-linearity can greatly outperform the rectified linear activation function for modeling time series.

Given the normalized time series, we propose to use filters in [2], which traverse the full columns of the sequence “cache” by a single large filter. Specifically, the width of filters is usually equal to the width of the input “cache”. The height typically varies with respect to the sliding windows over timestamp at a time (Fig 1, embedding input). Filters of different sizes will generate variable-length feature maps after convolution (convolutional layer). To ensure that all maps have the same size, the max pooling is performed over each map, which selects only the largest number of each feature map, leading to a 1×1 map (Fig 1 Feedforward Layers). Lastly, the maps from these filters are concatenated which form a feature vector, which is then fed into a softmax layer that yields the probabilities of next timestamp. In our experiments, we will present results of UberNet with both horizontal and vertical convolutions.

From the above analysis of the convolutions in Wavenet, it is easy to see that there are many drawbacks with the current model. First, the max pooling operator cannot distinguish whether an feature that appear in the map is important or not. In addition, it is oblivious of the position where the max pooling occurs. The max pooling operator may be effective in the “cache” processing, which is actually detrimental for modeling long-range sequences. Furthermore, shallow network of WaveNet will only be suitable for only one hidden layer which often fail when it comes to long-range dependences. The last important drawback derives from the generative process of next timestamp, which we will discuss in Section 5.1.

III. RELATED WORK

Traffic prediction has become a popular topic, where various machine learning approaches have been proposed. In the seminal work of univariate forecasting model, FARIMA is employed to model and predict traffic condition. Min et al. [22] proposed a regression model that considers spatial and temporal interactions of road conditions. However, this model entails a heavy computational cost with a vast amount of parameters, and yet fail to consider several important features of a transportation. Later on, multivariate model of nonparametric regression [6] is designed to exploit the additional features of transportation to infer the traffic state, however, the performance gain was quite limited.

The recent advances in deep learning has provided new opportunities to resolve this problem. Ramdas et al. [28] gives an examination of traditional neural network approaches and they found that the training process of deep learning model is computationally prohibitive when comparing to the traditional ones. To address this issue, researchers have proposed the dropout mechanism [10], which aims to find a sparse representation that is frequently updated in real time. What is more, the approaches of this line still require a fine tuning of parameters, which is not applicable given a different dataset. Lastly, despite the good performance, these models mostly either entails a super-linear execution time [31] or transform the original time series into a constant size format, which may cause a memorization step with unnecessary cost. These problems grow exponentially bad when the length of the series increases.

IV. UBERNET

This section introduces our proposed framework there, which uses a novel probabilistic generative model that includes dilated convolutional blocks to create the receptive field. This model is different from the WaveNet models since: (1) the probability estimator can capture the transition process of all individual timestamp at once, instead of the last one in the series; (2) the convolutional layers are on the basis of the more efficient 1D dilated convolution rather than the standard 2D one.

The proposed UberNet framework consists of two core components, a decoder and an encoder. The former is a time series model of one-dimensional layers, which are then masked use dilation. The encoder process the output of decoder into a low-dimensional representation and is defined as a one dimensional convolutional layers as well but without masking. To learn the embedding bandwidth of the encoder and decoder, the decoder is stacked on the computational output of encoder.

Embedding Look-up Layer: Given a time series $t_0, ..., t_l$, the model retrieves each of the first s timestamps $t_0, ..., t_s$ via a look-up operation, and concatenate these timestamp embeddings together. Assuming the embedding dimension is 2k, where k will be set as the number of inner channels in the convolutional network. This results in a matrix of size. Note that unlike WaveNet that considers the input matrix as a 2D

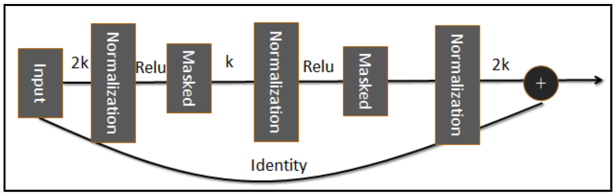


Figure 2. The structure of dilated residual blocks

"cache" during convolution operation, UberNet optimize the embedding via a 1D convolutional operation, which we will detail later.

Dilated layer It is clear in Fig 1 that the standard filter can only operate convolutional function within the receptive field by the depth of the neural network. This makes it difficult to handle long-range sequences. Hence, instead of employing standard convolutions, we designed the dilated convolution to create the generative model, where the dilated layer acts as the convolutional filter to a field which is broader than its original area via the dilation of a zeros matrix. This allow the model become more efficient as it requires fewer parameters. Therefore, a dilated filter is also known as the sparsed filter. Another advantage is that dilated layer does not change the spatial dimensions of the input so that the stacking will be much faster on the convolutional layers and residual structures.

By setting the width of convolutional filter as 3, we can see that the dilated convolutions allow for exponential increase in the size of receptive fields, while the same stacking structure for the standard convolution has only linear receptive fields. The dilated convolutional operation is more powerful to model long-range time series, and thus does not require the use of large filters or additional layers. Practically speaking, one need to carry out the structure of Fig 1 multiple times by stacking in order to further improve the capacity.

Masked Residual Increasing the depth of network layers can generate a higher-level condensed representations, however, it also easily causes some side effects, e.g., the vanishing gradient issue. To address this problem, residual learning mechanism is employed in cnn architectures. While residual learning has been proved to be effective in the field of computer vision,its effectiveness and feasibility in the context of traffic simulation is still largely unknown.

The logic of residual learning is that several convolutional layers can be stacked as a block, from which multiple blocks can communicate with each other through the skip connection scheme by passing signature feature of each block. The skip connection scheme can directly train the residual mapping instead of the conventional identity mapping scheme. This scheme not only maintain the input information but also increased the values of the propagated gradients, resolving the gradient vanishing issue.

Inspired by [24], we employ two residual modules as shown in Fig. 2. We encapsulate each dilated convolutional layer into a residual. The input layer and convolutional one are stacked through a skip connection (i.e., the identity line in Fig 2). Each block is represented as a pipeline structure of several layers, namely, normalization layer, activation (e.g., ReLU) layer, convolutional layer, and a skip connection in a specific manner. In this work we put the state-of-the-art normalization

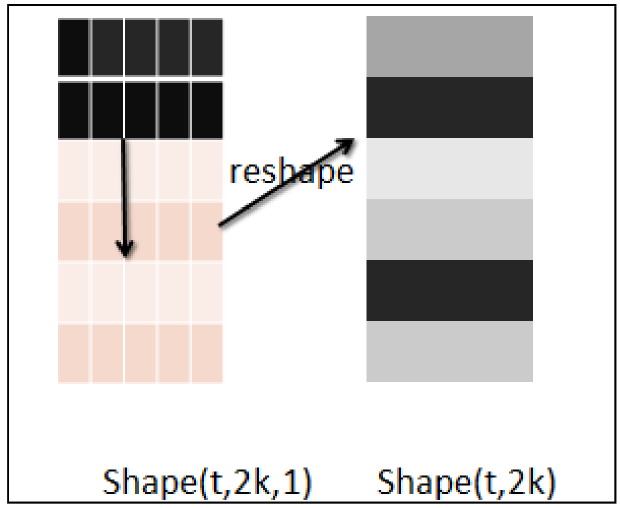


Figure 3. Transformation from the 2D dilated filter to the 1D dilated filter

layer before each activation layer, as it is well known that it has shown superior performance than batch normalization when it comes to sequence processing.

Final Layer The matrix in the last layer of the convolution structure (see Fig 2) has the same size as of the input embedding. However, the output should be a matrix that contains probability distributions of all timestamps in the output series, where the probability distribution is the expected one that generates the prediction results. We use one additional convolutional layer on top of the last convolutional layer in Fig 2. Following the procedure of one-dimensional transformation in Fig 3, we convert the expected prediction matrix as categorical distribution over timestamps, where each row vector is reshaped with the softmax operation.

The aim of this softmax operation is to maximize the log-likelihood of the training data with respect to all hyperparameters. Fortunately, maximizing log-likelihood can be mathematically converted to minimizing the summation of the binary cross-entropy loss for each timestamp. For practical neural network with tens of millions of timestamps, the negative sampling strategy can be applied to avoid the calculation of the full softmax distributions. The performance operated by these negative sampling strategies is almost the same as the full softmax method given that the sampling size are properly tuned.

V. EXPERIMENT SETUP

In this section, we will show the statistic of the dataset, the parameter settings, and the experimental results.

A. *Datasets*

The Uber raw data on pickups is derived from a non-Uber FHV company. The trip information include day of trip, time of trip, pickup location, and driver for-hire license number.

We have chosen datasets which (a) have been made publicly available, so as to enable other researchers to reproduce our results, and (b) have key characteristics covering a large part of the design space (e.g., day of trip, pickup locations). A detailed statistic about this dataset over months is reported in Figure 4. It is clear that the number of pickups is increasing

TABLE I. The performance of pickups prediction with different sets of features(smape)

Learning Models	External Features	Pickup Features	All Features
ARIMA	235.34	243.29	235.22
Linear Regression	324.52	254.02	257.03
Prophet	320.43	**204.12**	**198.72**
LSTM	423.43	351.27	1421.30
UberNet	**203.42**	223.97	227.38

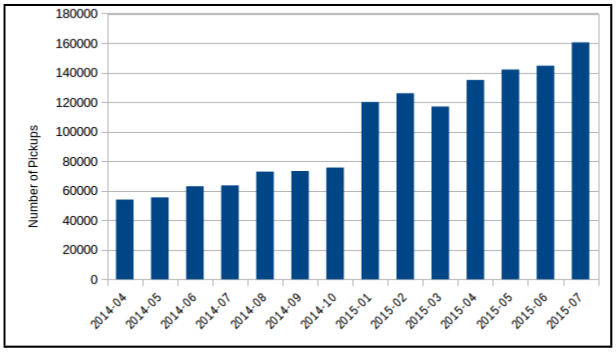


Figure 4. The statistic of pickups over months.

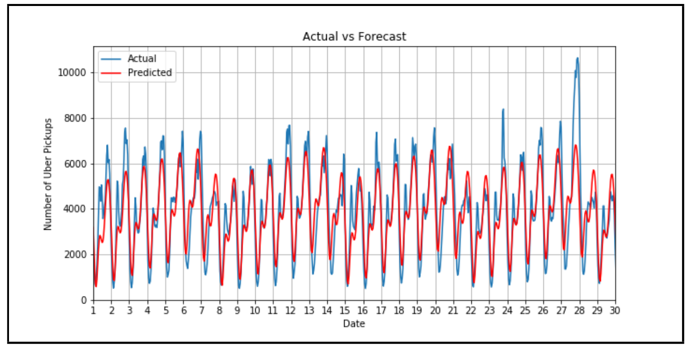


Figure 5. The performance of UberNet during June 2015.

over time. This is due to the fact that more and more people opt into Uber platform over this time period. Notice that we use the 2014 data as the training dataset and the 2015 data as the test dataset.

Approach

In our experiments, we compare the following four approaches:

1) the baseline approach Autoregressive Integrated Moving Average (ARIMA), where the "evolving variable" of interest is regressed on its own lagged (i.e., prior) values and the "regression error" is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past.
2) the off-the-shelf model Prophet, which is an additive regression model with a piecewise linear kernel. Prophet automatically detects changes in trends by selecting changepoints from the data. The monthly seasonal component is modelled using Fourier series, and a weekly seasonal component is captured using dummy variables.
3) the state-of-the-art Long Short Term Memory (LSTM), which is a type of generalized regression neural networks (GRNN) that aims to relax the constraints of the conventional NN architecture. LSTM used in our work is a simple neural network with four hidden layers (normalization and Relu) with the same number of neurons.
4) the UberNet, which is the approach that described in Section IV

B. Parameter Settings

For ARIMA model, we have the Adjusted Dickey-Fuller Test, where the model achieved the best performance when we set $p = 2$, $d = 1$, and $q = 1$. For Prophet model, we set all the parameters of the default values, the yearly seasonality is set as false. The results of varying models are reported in Table I. A rectifier activation function is used for the neurons in the hidden layer of LSTM. A softmax activation function is used on the output layer to turn the outputs into probability-like values. Logarithmic loss is used as the loss function and the efficient ADAM gradient descent algorithm is used to learn the weights.

Metric Symmetric mean absolute percentage error (SMAPE or sMAPE) is an accuracy measure based on percentage (or relative) errors. It is defined as $smape = \frac{abs(F-A)}{1/2(F+A)}$. This is one of the most commonly used metrics in time series analysis community, e.g., the Web Traffic Time Series Forecasting competition, since this metric is a relative value which ignores outliers and is invariant if the data is linearly rescaled.

C. Experimental Results

Table I shows the performance accuracy of pickup prediction through with different sets of features. It is clear that Prophet and UberNet outperform the other types of learning models. Furthermore, one can see that using both external features, namely, the weather and temperature conditions, works better than using pickup features or relational features alone, for all learning models. Last but not least, one can also observe that RNN approach, i.e., LSTM, fail to yield good results. The reason is probably that the dataset is too small for LSTM to unlock its power. One possible way to remedy this is to carry out hot encoding on all the features so that the feature space can be expanded for learning. We left this for future work.

To further demonstrate the effectiveness of our proposed model, Figure 5 show the performance of *UberNet* during the entire June of 2015. The blue line denotes the ground truth and the red line represent the prediction. It is easy to see that Ubernet can actually predict the pickups most of the time except some holidays. This is probably due to the overfitting issue of the deep learning architecture. UberNet can largely

 ISBN: 978-1-61208-692-7

resolve this issue but cannot eradicate it. We believe UberNet will achieve a even better result given more training data.

VI. Conclusion

The major contribution of this paper is threefold. First, we formally defined the problem of Uber pickup prediction as a machine learning task. Second, we identified several external features which can be used together with standard pickup features by learning algorithms to predict pickups in terms of their context. Third, we demonstrate that it is better to exploit both external features and pickup features through a WaveNet type deep neural network, namely, UberNet.

There are several interesting and promising directions in which this work could be extended. First, since users and Uber drivers can be represented as a graph structure, it will be interesting to learn the performance of some advanced graph regression algorithms, such as BB-Graph [2]. Second, UberNet in the current form relies on one of the simplest convolutional neural netowrk, which makes sense as a first step towards integrating the context information of Uber into learning model, but of course we could consider using more sophisticated neural network like RCNN. Lastly, For example, location of the user has not considered in this work yet , in future we would like to explore the geographical distribution to facilitate UberNet.

References

[1] Osvaldo Anacleto, Catriona Queen, and Casper J Albers. Multivariate forecasting of road traffic flows in the presence of heteroscedasticity and measurement errors. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(2):251–270, 2013.

[2] Merve Asiler and Adnan Yazıcı. Bb-graph: A new subgraph isomorphism algorithm for efficiently querying big graph databases. *arXiv preprint arXiv:1706.06654*, 2017.

[3] Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6(Jun):937–965, 2005.

[4] C Borgelt and M Berthold. Finding relevant substructures of molecules: Mining molecular fragments. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002)*, pages 51–58, 2002.

[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[6] Stephen Clark. Traffic prediction using multivariate nonparametric regression. *Journal of transportation engineering*, 129(2):161–168, 2003.

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016.

[8] Felix A Gers, Douglas Eck, and Jürgen Schmidhuber. Applying lstm to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri-01*, pages 193–200. 2002.

[9] Huahai He and Ambuj K Singh. Query language and access methods for graph databases. In *Managing and mining graph data*, pages 125–160. Springer, 2010.

[10] David P Helmbold and Philip M Long. On the inductive bias of dropout. *The Journal of Machine Learning Research*, 16(1):3403–3454, 2015.

[11] Jun Huan, Wei Wang, and Jan Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 549–552. IEEE, 2003.

[12] Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. An apriori-based algorithm for mining frequent substructures from graph data. *Principles of Data Mining and Knowledge Discovery*, pages 13–23, 2000.

[13] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[15] George Kuramochi, Michihiro andKarypis. Frequent subgraph discovery. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 313–320. IEEE, 2001.

[16] Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. Time-series extreme event forecasting with neural networks at uber. In *International Conference on Machine Learning*, number 34, pages 1–5, 2017.

[17] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[18] Lei Lin, Zhengbing He, and Srinivas Peeta. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. *Transportation Research Part C: Emerging Technologies*, 97:258–276, 2018.

[19] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

[20] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, Fei-Yue Wang, et al. Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intelligent Transportation Systems*, 16(2):865–873, 2015.

[21] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[22] Wanli Min and Laura Wynter. Real-time road traffic prediction with spatio-temporal correlations. *Transportation Research Part C: Emerging Technologies*, 19(4):606–616, 2011.

[23] Jennifer Neville and David Jensen. Iterative classification in relational data. In *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pages 13–20, 2000.

[24] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International Conference on Machine Learning*, pages 2014–2023, 2016.

[25] Siegfried Nijssen and Joost N Kok. A quickstart in frequent structure mining can make a difference. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 647–652. ACM, 2004.

[26] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[27] Nicholas Polson, Vadim Sokolov, et al. Bayesian analysis of traffic flow on interstate i-55: The lwr model. *The Annals of Applied Statistics*, 9(4):1864–1888, 2015.

[28] Aaditya Ramdas and Ryan J Tibshirani. Fast and flexible admm algorithms for trend filtering. *Journal of Computational and Graphical Statistics*, 25(3):839–858, 2016.

[29] John Rice and Erik Van Zwet. A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems*, 5(3):200–207, 2004.

[30] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *SSW*, page 125, 2016.

[31] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI*, pages 1655–1661, 2017.

Accessible Image Description Using Sample Example Cues

Dhruba Dahal
Romerike IT Consulting AS
Oslo, Norway
Email: dhruba.dahal03@gmail.com

Raju Shrestha
Oslo Metropolitan University
Oslo, Norway
Email: raju.shrestha@oslomet.no

***Abstract* - Image accessibility on the web is the main focus of this study. Due to the lack of proper image descriptions, it has been difficult to access intended information available on the informative images for the people with sight loss and who use assistive technologies, such as screen reader while surfing websites or web applications. This study defines the lack of effective solutions to author image description as an existing gap, and explores the possibility of helping to write a better image description with the use of different types of sample examples. Results suggest that it is effective to have a similar sample example description to write accessible image descriptions.**

Keywords - image description; accessibility; NCAM guidelines

I. INTRODUCTION

Tim Berners-Lee, the inventor of the World Wide Web, states that the power of the web is in its universality. Accessibility for everyone regardless of disability is an essential aspect. The statement reflects the significance of web accessibility which is about the fundamental design of web for all people regardless of their hardware, software, language, culture, location, and physical or mental ability and the fulfillment of this goal results in an accessible web with a diverse range of sight, hearing, movement, and cognitive ability [1].

This study emphasizes image accessibility on the web. The images on the web might be of several types: informative images, decorative images, functional images, images of text, complex images, groups of images, and image maps. Accessibility of these images simply means if the intended information given in it is accessible to the people including disabled people, such as visually impaired people. It is possible to make images accessible through the text description which is readable by assistive technologies such as a screen reader [2].

If we look at the real-world scenarios, there are massive number of images in the Internet which do not have text descriptions, and many of them that have are not appropriate and not good enough to convey necessary information [3]-[5]. This clearly indicates the lack of availability of descriptive summary of the images on the web intended for image accessibility. Literatures suggest that the main reason behind this may be the negligence of web authors, complexity of writing image description, and lack of time and motivation to read the accessibility guidelines having long text, it would be more effective if instead of traditional textual guidelines, a real-time guidance is provided.

In this work, we have investigated the possibility of encouraging and improving image description for better accessibility by providing example images with sample descriptions, which we call it as sample cues. The reset of the paper is organized as follows. Section II describes the related research. Section III presents the proposed sample cue-based method. Section IV presents experiments and results. Finally, we conclude the paper in Section V.

II. RELATED METHODS USED FOR IMAGE DESCRIPTION

In general, there are two broad categories of methods or authoring techniques used for describing an image. The first one is human powered authoring and the other one is computer algorithm-based authoring.

The system called VizWiz lets blind people take a picture, asks questions, and receive answers from distant workers almost in real-time [6]. TapTapSee, a mobile application developed particularly for blind and visually impaired users, takes a picture of any two- or three-dimensional object and tells the user audibly by identifying the objects within seconds [7]. Splendiani and Ribera [8] suggested to use a decision tree that may reduce ambiguity and enhance the relevance of alternative texts. Likewise, Morash et. al [9] compared two methods, Queried Image Description (QID) method and Free-Response Image Description (FRID) method, for novice Web workers to produce image descriptions for graph images based on National Centre for Accessible Media (NCAM) guidelines [10]. Although there are several human powered systems available, Wu et. al [11] claimed that all these systems so far are constrained by scalability, latency, cost, and privacy concerns.

On the other hand, Cundiff [12] developed a browser extension that adds descriptions to images on the web for blind people. After getting a user click on an image, the extension sends the image URL to the cloud sight API and gets the resulting description to the image. Similarly, Ramnath, Baker, and Vanderwende [13] introduced a system allowing smartphone users to generate captions for their photos. The system is based on a cloud service and the combined outcomes of the different modules result in a large set of candidate captions which are provided to the phone. Several computer algorithm-based solutions are available, which are intended for social media. Automatic alt-text (AAT) [11] is an example, which identifies objects, faces, and themes from photos and generate alternative text for screen reader users on Facebook. However, Morris et al. [14] found that currently available computer-generated captioning solutions are not robust enough to meet image accessibility requirements. They investigated accessibility of Twitter, which has traditionally been thought of as the most accessible social media platform for blind users, and found that image-based tweets are diverse, largely inaccessible.

III. PROPOSED SAMPLE CUE-BASED IMAGE DESCRIPTION

The literature suggested that the time-consuming accessibility guidelines are not so effective for having useful descriptions to the images uploaded on the web. Therefore, we have proposed a new sample example cue-based method to assist in writing image description to improve image accessibility. Similar example image(s) with description is provided as a sample cue in order to help writing a description for the given image. Description of these sample images are written by accessibility experts by following fourteen NCAM accessibility guidelines that have been developed based on several studies incorporating disabilities. The fourteen guidelines used are listed below.

NCAM image accessibility guidelines:

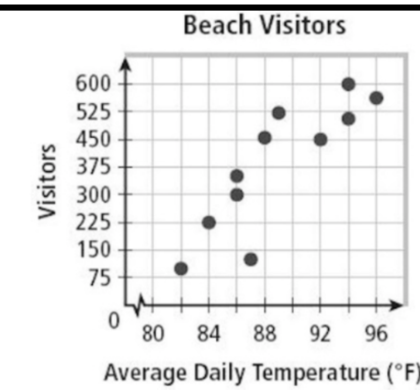
1. The description should be succinct.
2. Colors should not be specified unless it is significant.
3. The new concept or terms should not be introduced.
4. The description should be started with high level context and drilled down to details to enhance understanding.
5. The active verbs in the present tense should be used.
6. Spelling, grammar, and punctuation should be correct.
7. Symbols should be written out properly.
8. The description vocabulary should be added which adds meaning for example, "map" instead of an image.
9. The title and axis labels should be provided.
10. The image should be identified as a scatter plot and be focused on the change of concentration.
11. The central teaching point should be focused to determine if borders, region shapes, and bodies of water are important.
12. The description should be organized using number lists and pull the most important information in the beginning.
13. Physical appearance and actions should be explained rather than emotions and possible intentions.
14. The material should not be interpreted or analyzed, instead, the readers should be allowed to form their own opinions.

Among these fourteen guidelines, the first 8 guidelines are common to all types of images, while guidelines 9 and 10 are specific to graph images, guidelines 11 and 12 are specific to map images, and guidelines 13 and 14 are specific to natural images.

Modern Artificial Intelligence (AI) based algorithms have shown successful classification of images, even beating human intelligence. These algorithms can be used to find a similar example image for a given image to be described. Therefore, the proposed method could be a viable and effective solution for accessible image description.

IV. EXPERIMENTS AND RESULTS

In order to evaluate the effectiveness of the proposed method, this study conducts an online experiment to compare results of different sample cues on image descriptions. A custom web application software was developed for this. Sixty-five participants took part in the experiment who wrote text descriptions for given images with and without sample cues. We have limited our study to three different types of images: graph, map, and natural photos. The participants were asked to write descriptions, first without any sample example description (No cue), then by providing a random image with a description (Random cue), and finally by providing a similar image with a description (Similar cue). Figure 1 shows an example image description written by a participant for a graph image. Sample example images (cues) were selected randomly from the set of pre-classified images (graph, map and natural) with descriptions.

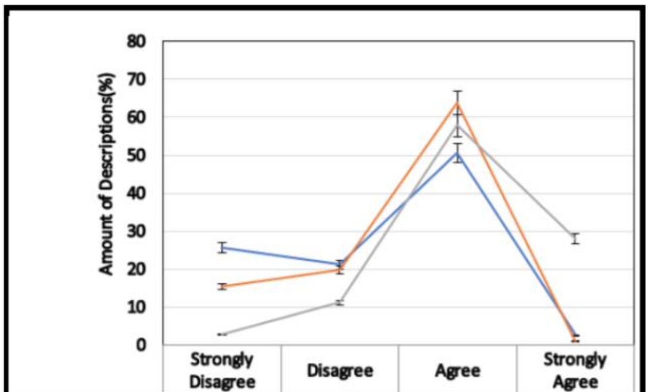


No cue	Random cue	Similar cue
Number of beach visitors with respect to the average daily temperature.	More the daily temperature more the beach visitors.	This graph is a scatter plot of daily visitors in a beach, which shows the increase of visitors in the beach with respect to increase in daily temperatures.

Figure 1 An example image description written by a participant while having no cue, a random cue, and a similar cue.

Effectiveness of the proposed method has been evaluated based on the compliance of the 500 image descriptions entered by the participants to the 14 NCAM guidelines as evaluated by the six experts who have a good knowledge on image accessibility.

As suggested by Allen and Seaman [15] and Boone and Boone [16], compliance of the image descriptions to the NCAM guidelines is measured in a Likert type rating scale from 1 to 4 (1 – strongly disagree, 2 – disagree, 3 – agree, and 4 – strongly agree). Figure 2 shows resulting compliance of the image descriptions (in percentage) to the overall 14 NCAM guidelines in three different cases with no cue, random cue and similar cue. The plots in the figure also shows standard errors in cases of all the four rating scales.



	Strongly Disagree	Disagree	Agree	Strongly Agree
No Cue	25.62	21.13	50.55	2.64
Random Cue	15.36	19.8	63.84	0.98
Similar Cue	2.89	11.22	57.89	27.98

Figure 2. Compliance of image descriptions with overall guidelines in 3 different cases.

From the figure, we see that, when no cues were provided, almost 53% of the image descriptions complied (which includes both ‘agree’ and ‘strongly agree’) to the overall guidelines. Compared to this, the compliance percentage increased significantly 12% when random cues were provided. The compliance percentage increased even more by 33% when similar cues were provided.

To determine statistical significance of these results, we conducted a Friedman test [17], which is a non-parametric alternative to the one-way ANOVA with repeated measures. This is useful to test for differences among groups when the dependent variable being measured is ordinal. It is suitable in our case since the intervals in the four Likert type rating scales used may not be equal. To examine where the differences occur, this study ran a separate Wilcoxon signed-rank test [17] on the related groups: no cue to random cue, no cue to similar cue, and random cue to similar cue. The table in Figure 2 shows the test results. The results show a significant effect of sample example cues on the quality of image description written by the users. Effect of random cues over no cue is small, whereas effect of similar cues over no cue and random cues is moderate.

V. CONCLUSIONS

This study investigated the effectiveness of providing real time sample example cues as an alternative to a set of guidelines for the users who have no or minimal knowledge about how to write an image description and the one who do not have enough time or do not want to read long guidelines before writing an image description.

The results demonstrate that similar example cue provides significant help than no example cue and random example cues in writing image descriptions in compliance with the NCAM guidelines to make them accessible.

As a future work, the study could be extended further with more images. Also, the effect of sample cues in different contexts and usability of the method by the real users with accessibility issues can be investigated.

REFERENCES

[1] S. L. Henry, “Introduction to Web Accessibility, 2005. [Online] available from: https://www.w3.org/WAI/fundamentals/accessibility-intro/, Retrieved: Jan. 2019.

[2] E. Eggert, and S. Abou-Zahra, “Web Accessibility Tutorials”, 2014. [Online] available from; https://www.w3.org/WAI/tutorials/images/, 2014, Retrieved: Jan. 2019.

[3] R. Bavani, A. Jaafar, and N. F. M. Yatim, “A study on web experience among visually impaired users in Malaysia”, Proceedings of the International Conference on User Science and Engineering (i-USEr), pp. 11-15, 2010.

[4] H. Francis, D. Al-Jumeily, and T. O. Lund, “A framework to support e-commerce development for people with visual impairment”, Proceedings of the 6th International Conference on Developments in eSystems Engineering, pp. 335-341, 2013.

[5] R. Goncalves, J. Martins, and F. Branco, “A review on the Portuguese enterprises web accessibility levels-a website accessibility high level improvement proposal”, Procedia Computer Science, vol. 27, pp. 176-185, 2014.

[6] J. P. Bigham et al., "VizWiz: Nearly Real-time Answers to Visual Questions", ACM User Interface Software and Technology Symposium (UIST), 2010.

[7] TapTapSee, “Assistive Technology for Blind and Visually Impaired”, 2014. [Online] available from: http://taptapseeapp.com/, Retrieved: Jan. 2019.

[8] B. Splendiani, M. Ribera, “How to textually describe images in medical academic publications”, Proceeding of the XV International Conference on Human Computer Interaction, vol. 67, pp. 1-3, 2014.

[9] V. S. Morash, Y. Siu, J. A. Miele, L. Hasty, and S. Landau, “Guiding novice web workers in making image descriptions using templates”, ACM Transactions on Accessible Computing (TACCESS), vol. 7, pp. 4-12, 2015.

[10] NCAM, “Guidelines for Describing STEM Images”, 2009. [Online] available from: http://ncam.wgbh.org/experience_learn/educational_media/stemdx/guidelines, Retrieved: Jan. 2019.

[11] S. Wu, J. Wieland, O. Farivar, and J. Schiller, “Automatic alt-text: Computer-generated image descriptions for blind users on a social network service”, CSCW, pp. 1180-1192, 2017.

[12] C. Cundiff, “Alt text bot”, 2015. [Online] available from: https://connectability.devpost.com/submissions/ 37785-alt-text-bot, Retrieved: Jan. 2019.

[13] K. Ramnath, S. Baker, L. Vanderwende, “AutoCaption: automatic caption generation for personal photos”, Proceedings of the IEEE Winter Conference on Applications of Computer Vision, pp. 1050-1057, 2014.

[14] M. R. Morris, A. Zolyomi, C. Yao, S.Bahram, J. P. Bigham, and S. K. Kane, “With most of it being pictures now, I rarely use it: Understanding Twitter’s evolving accessibility to blind users”, Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 5506-5516, 2016.

[15] E. I. Allen and C.A. Seaman, “Likert scales and data analyses”, Quality progress, vol. 7, pp. 40-64, 2007.

[16] H. N. Boone, and D. A. Boone, “Analyzing Likert data”, Journal of Extension vol. 50(2), pp. 1-5, 2012.

[17] Lund-Research, “Friedman test in SPSS statistics”. [Online] available from: https://statistics.laerd.com/spss-tutorials/ friedman-test-using-spss-statistics.php, 2013, Retrieved: Jan. 2019.

Preliminary Experiments on Driver Assistance System in Remote Control

-Effects of Communication Delays, Video Resolution and Frequency-

Kohei Kadowaki
Dept. of Applied Electronics
Tokyo University of Science
Tokyo, Japan
e-mail: 8118513@ed.tus.ac.jp

Naohisa Hashimoto
Robot Innovation Research Center
National Institute of AIST
Tsukuba, Japan
e-mail: naohisa-hashimoto@aist.go.jp

Simon Thompson /Yusuke Takinami
Robot Innovation Research Center
National Institute of AIST
Tsukuba, Japan

Shin Kato
Intelligent System Research Institute
National Institute of AIST
Tsukuba, Japan
e-mail: Shin-kato@aist.go.jp

Makoto Itami
Dept. of Applied Electronics
Tokyo University of Science
Tokyo, Japan
e-mail: itami@te.noda.tus.ac.jp

Abstract— **Automated vehicles are expected to solve traffic issues. We proposed a prototype of remote type automated vehicle system. However, the lack of information due to the limited number of cameras and signal delay makes the system difficult to control. To improve accurate remote control and compensate the delay, a method of trajectory prediction with changing fps or resolution is proposed. This demonstration paper introduces the automated vehicle system and explains the system configuration, communication and results. The experiment in this paper is measuring the delay at each fps and resolution.**

Keywords—Smart mobility; Automated Vehicles; Remote Control System; Vehicle to Infrastructure; Communication Network; Intelligent Transport System.

I. Introduction

There are researches in areas of intelligent vehicles in a roadway environment, and in particular in automated vehicles [1]. Introduction of automated vehicles for smart city is expected to solve traffic problems [2][3]. Figure 1 shows the concept image of the smart mobility [4]. The objectives of the mobility are the establishment of public acceptance, the clarification of business model, the establishment of social system and the establishment of automated driving technology [5][6]. However, automated vehicles without a driver is not allowed under current low in Japan. National police agency released the new guideline for remote type automated vehicle system for the real world experiments in 2017 [7]. In order to proceed them, a prototype of remote type automated vehicle system is proposed. In the system, the automated vehicle moves above a magnetic wire. The intensity of the signal from the wire is captured by the vehicle with algorithm of feedback control scheme. The scheme makes the vehicle able to follow the wire. However, if obstacles on the way, the vehicle must be remotely controlled to avoid the obstacles and go back to the magnetic wire again. The remote driver watches the live video from the front camera at the same time to operate the vehicle.

Remote type automated vehicle system has two main problems: the lack of information due to the limited number of cameras and signal delay. These problems makes operating vehicles difficult. Therefore, the assistance of driving is necessary for the remote driver to better understand the surrounding environment. By showing the trajectory prediction of the vehicle, the remote driver is assumed to better control the vehicle to avoid the obstacles [8]. The experiments in this paper are designed for the assistance system using video camera and OpenCV (ver.3.4.2). In the system, the video consists of still images that are captured consecutively and played back in quick succession. A frame is a single one of those images, and the frame rate is a measure of frequency: how often the video is updated with a new frame. Frame rate is measured in Frames Per Second (FPS). FPS is the number of frames of video in one second.

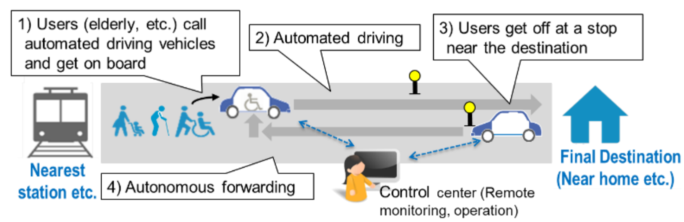


Figure 1. Concept image of the project.

1 pixel of an image has 8 bits, which shows depth of color and 3 channels, which shows 3 primary colors of light. The more pixels an image contains, the more transmission takes long time. Also, the more frames are sent, the more transmission takes long time. few frames and pixels make transmission quickly. This system is supposed to be used mainly in depopulated areas where the communication system is not well established. Communication delay is the trade-off between image resolution and frame rate. This paper shows limit value of image resolution and FPS to control cart correctly. Section 2 shows the configuration of the remote control system. Section 3 shows the experiments to quantify the correctness of prediction by using Global Positioning System (GPS).

II. Remote Type Automated Vehicle

This section shows how to remote control the cart and calculate future prediction.

A. System Configuration

Figure 2 shows the system configuration. The system consists of automated vehicles, a remote-control server, a monitors and communication tools. Wi-Fi or Long-Term Evolution (LTE) is used for wireless communication in this

 ISBN: 978-1-61208-692-7

system. If an operator changes the mode to take over an automated vehicle, the operator teleoperate the vehicle by using speed and steering controller. The whole program is composed of three functions running in parallel. User Datagram Protocol (UDP) sender function keeps sending the current steering angle and speed obtained from the game controller. UDP receiver function keeps listening to the message which is sent from the vehicle and updates the real time steering angle and speed. Show function captures the input of the controller and calculates the

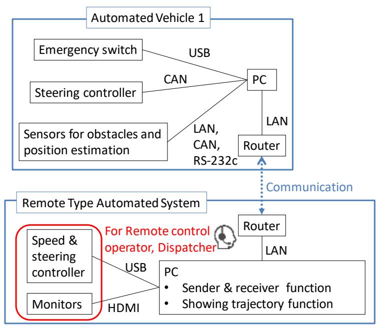


Figure 2. System Configuration

future path based the steering angle and speed of the vehicle then project to the image captured from the stream data.

B. *Contents of Communication*

Table 1 shows the contents of communication between the remote server and automated vehicles. Automated vehicles in the system communicate with the remote server by using shared memory. If the communication between the remote server and automated vehicles is unavailable for some reasons, automated vehicles stop.

TABLE I. CONTENTS OF COMMUNICATION (LEFT: FROM VEHICLE TO REMOTE SERVER, RIGHT: FROM REMOTE SERVER TO VEHICLE)

sm[n]	Contents (from vehicle to server)
10	Vehicle mode
11	Shift position
12	Vehicle speed
13	Vehicle steering angle
14	Brake status
15	Alive counter (not used)
16	Winker status
17	Obstacle information
18	Obstacle position X
19	Obstacle position Y
20	Flasher
21	Counter

sm[n]	Contents (from sever to vehicle)
500	Datacount
501	Target speed
502	Target steering angle
503	Shift position
504	Brake status
505	Driving mode
506	Winker status
507	Permission from oparator
508	Horn instruction

In the system, an occurrence of no communication in 2 seconds leads to communication failure and automated vehicles automatically stop.

C. *Trajectory Prediction*

The trajectory prediction is calculated based on the steering angle and velocity by applying the simplified vehicle dynamic model. Figure 3 shows the procedure of calculating the coordinates of 10 discrete points in the case of turning procedure and Figure 4 shows the motion of turning vehicle. Trajectory prediction in this paper are composed of three parts: left wheel, right wheel and middle between both wheels. Each trajectory is composed of 10 discrete points which are calculated with speed and steering angle. Once the trajectory is obtained, by projecting the trajectory in real world coordinates, the ideal path will be shown in the 2D image from the front camera, which will help the driver to better predict where the vehicle is going.

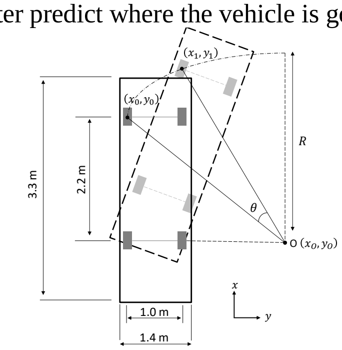


Figure 3. Motion model of vehicles

The program needs to project a path by taking account the operator's input and must be able to track any obstacle in the environment, as well as determine if the current trajectory may lead to a collision.

1. Input: Tread W, wheelbase L, current speed spd, current steering angle α
2. Output: 10 discrete points of trajectory (x_i, y_i), $i \in (0, 1, \cdots, 9)$
3. Step1: calculating turning radius R
4. $$R = \begin{cases} \sqrt{L^2 + \left(r + \frac{W}{2}\right)^2}, r = \frac{L}{\tan\alpha} & left \\ \frac{L}{\tan\alpha} & middle \\ r - \frac{W}{2} \quad , r = \frac{L}{\tan\alpha} & right \end{cases};$$
5. Step2: calculating the length S of the trajectory in 3 second.
6. $S = spd * 3;$
7. Step3:caluculating the central angle θ
8. $\theta = \frac{S}{R*10};$
9. Step4: calculating (x_i, y_i) for each point
10. If $(\alpha > 0)$ then
11. $$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} (x_0 - x_O)\cos(\theta * i) - (y_0 - y_O)\sin(\theta * i) + x_O \\ (x_0 - x_O)\sin(\theta * i) + (y_0 - y_O)\cos(\theta * i) + y_O \end{bmatrix};$$ $i \in (0, 1, \cdots, 9)$
12. If $(\alpha < 0)$ then
13. $$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} (x_0 - x_O)\cos(\theta * i) - (y_0 - y_O)\sin(\theta * i) + x_O \\ (x_0 - x_O)\sin(\theta * i) + (y_0 - y_O)\cos(\theta * i) + y_O \end{bmatrix};$$ $i \in (0, 1, \cdots, 9)$
14. end if

Figure 4. Trajectory prediction for turning

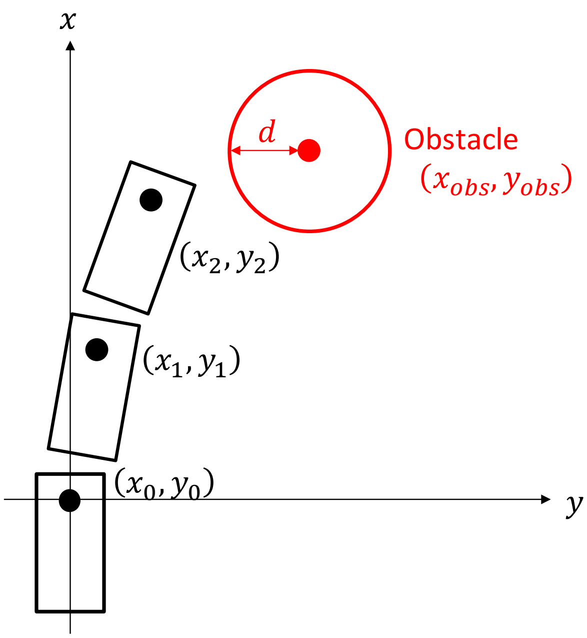


Figure 5. Relative distance model of vehicle and obstacle

The program is designed to display a project path to aid an operator when remotely maneuvering a vehicle through an obstacle course. Figure 5 shows the model of relative distance of vehicle and obstacle and the distance is calculated in

$$\sqrt{(x_i - x_{obs})^2 + (y_i - y_{obs})^2} \leq d. \qquad (1)$$

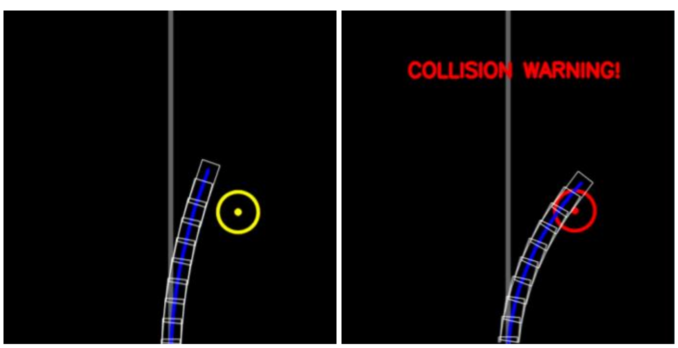


Figure 6. World coordinate (Left: Avoided obstacle, Right: Collided obstacle)

Figure 6 shows the result of collision detection from equation (1).

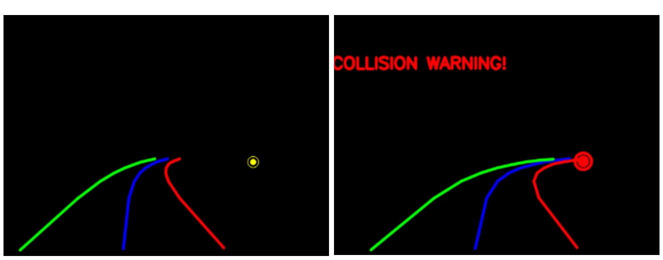


Figure 7. Image coordinate (Left: Avoided obstacle, Right: Collided obstacle)

The result of the transformation between world coordinate and image coordinate can be seen in Figure 7.

III. EXPERIMENTS

Figure 8 shows the automated vehicle which is controlled by an operator and the camera on the front of the vehicle. There are two laser range finders on front edge of the vehicle and GPS antenna on the top of the vehicle.

Figure 8. Automated vehicle (Left: Electronic vehicle, Right: Front camera)

Remote type automated system is on another vehicle which is shown in Figure 9. The system shows the trajectory which calculated by the speed and steering angle obtained from the automated vehicle on the monitor. During experiments, the data value of time, speed, steering angle, resolution and fps are logged.

Figure 9. The vehicle with remote type automated system

Experimental scene is shown in Figure 10. A slalom course was employed in the experiment. There are 10 obstacles which are putted in zigzag on the course. An operator needs to operate the vehicle to avoid the obstacles.

Figure 10. Test course

The experiment is to check whether low resolution of small fps makes the signal delay short. However, low resolution and small fps also makes teleoperation difficult. Figure 11 shows high and low resolution images from camera.

Figure 11. Camera images (Left: 1001×667, Right: 100×67)

In this experiment, there are 2 patterns of resolutions, 2patterns of frequency of frames, and 2 patterns of communication delay before sending value of velocity and steering angle from operator to cart. This system can change the timing of sending input data, thus this system can add the communication delay intentionally in addition to the constant delay. First, The GPS data of the target course is taken by driving the cart. Second, the subject control the cart to drive on the course by remote control system while taking GPS real data. In this experiment, the quantity of correctness of prediction in each frame rate and image resolution is shown as the difference of the GPS target data and real data. Each experiment was done 10 times.

IV. Conclusion

This paper introduces the driving assistance system for remote control and explained the system configuration, contents of communication, results and the plan of experiments in the real field, as a short paper. The results of the experiments have not been concluded yet. The experiment will be continued, analyzed and shared the experimental results after the completion of the project as a future work.

Acknowledgment

The study has been supported by Ministry of Economy, Trade and Industry in Japan and Ministry of Land, Infrastructure and Transport in Japan.

References

[1] F. Granda *et al*., “Deterministic Propagation Modeling for Intelligent Vehicle Communication in Smart Cities,” Sensors, Vol.18, no.7, pp.1-28, July 2018, doi:10.3390/s18072133.

[2] J. L. Zambrano-Martinez, C. T. Calafate, D. Soler, J. C. Cano, and P. Manzoni, “Modeling and Characterization of Traffic Flows in Urban Environments,” Sensors, vol.18, no.7, pp.1-19, July 2018, doi:10.3390/s18072020.

[3] M. D. Cia *et al*., “Using Smart City Data in 5G Self-Organizing Networks,” IEEE IoT J., vol.5, no.2, pp.645-654, Apr. 2018, doi:10.1109/JIOT.2017.2752761.

[4] N.Hashimoto *et al*., “Introduction of Prototype of Remote Type Automated Vehicle System by using Communication between Operator and Vehicles in Real Environment”, Proceedings of ITST 2018, 2018

[5] A. K. Tripathy, K. Pradyumna, N. K. Ray, S. P. Mohanty, “iTour: The future of Smart Tourism: An Iot Framework for the Independent Mobility of Tourists in Smart Cities,” IEEE Consumer Electronics Magazine, vol.7, no.3, pp.32-37, Apr. 2018, doi:10.1109 / MCE. 2018.2797758.

[6] I. Docherty, G. Marsden, J. Anable, “The governance of smart mobility,” Transporttation Research Part A: Policy and Practice, vol.115, pp.114-125, Sep. 2018, doi: 10.1016/j.tra.2017.09.012.

[7] National Police Agency in Japan. *Automated driving System 2.0: A Vision for Safety*. [Online]. Available from: https://www.npa.go.jp/bureau/traffic /selfdriving/index.html, retrieved: Dec. 2018

[8] Ruike Ren, Wei Wang, Jinze Liu, Yan Li, Li Wang, “Teleoperation of unmanned ground vehicle based on 3D trajectory prediction,” IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC 2016), Oct. 2016, pp.790-794, doi:10.1109/IMCEC.2016.7867318

Virtualized Sensor System: an Access Unification and Software-defined Sensors

Naoki Aoyama
Graduate School of Engineering,
Nagoya Institute of Technology,
Nagoya-shi, 466-8555 Japan
Email: naoki@moss.elcom.nitech.ac.jp

Yonghwan Kim
Graduate School of Engineering,
Nagoya Institute of Technology,
Nagoya-shi, 466-8555 Japan
Email: kim@nitech.ac.jp

Yoshiaki Katayama
Graduate School of Engineering,
Nagoya Institute of Technology,
Nagoya-shi, 466-8555 Japan
Email: katayama@nitech.ac.jp

***Abstract*—As the growth and the spread of Internet of Things (IoT), various context-aware applications, which determine their behaviors based on the recognized context, are widely studied and developed. To develop such applications, an application programmer has to understand every sensor's specification, e.g., access method (i.e., how to get a current value), for handling all the necessary sensors. Moreover, even some sensors are the same types, e.g., temperature sensors, they may have different units of values, hence, the unification of each sensor's unit may be required. These make some inexperienced programmers hard to develop context-aware applications. In this paper, we present a novel middleware named *Virtualized Sensor System (VSS)*, which provides some features for application programmers as follows: (i) provides unified methods to access every sensor in the system without any knowledge of their specifications, and (ii) can create new *software-defined sensors* by composing of some hardware sensors. To help to create a new *software-defined sensor*, we present a new markup language, *Virtual Sensor Markup Language (VSML)*. Our proposed system *VSS* and markup language *VSML* can make the developments of the context-aware applications using various sensors easy even if an application programmer is inexperienced in them.**

Keywords–Sensor Middleware; Sensor Virtualization; Software-defined Sensor; Support for Application Developments; Internet of Things.

I. INTRODUCTION

Background and contribution of this study, and related works are introduced in this section.

A. Background and Contribution

As the growth and the spread of Internet of Things (IoT) [1]-[3], various context-aware applications, which determine its behavior based on the recognized context, are widely studied and developed [4]. Context-aware applications can provide more flexible services to users based on their current context (e.g., their locations, activity history, or their environments), hence, more context-aware applications, especially using various sensors, are expected to emerge. However, the emergence of new various sensors may make many programmers difficult to develop applications, because they should understand every sensor's specification to develop a context-aware application using these new sensors. This implies that to make sensors easy to use regardless of the specifications of new sensors is an important issue. And as context-aware applications become more multifaceted, some specific sensors which do not exist but may be useful for some limited applications can be demanded. For example, there can be application programmers who want to use a water vapor capacity sensor which can be

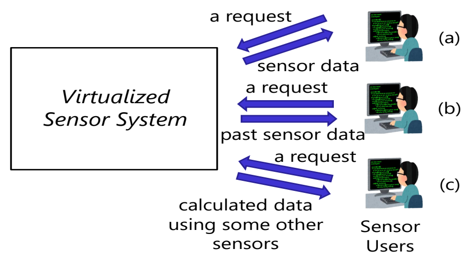


Figure 1. Proposed system in this paper

calculated using one thermometer (i.e., temperature sensor) and one hygrometer (i.e., humidity sensor) and there also can be some other programmers who want to use a new motion sensor consisting of several motion sensors which are referred in a particular order. There are several problems for application programmers to develop applications using various sensors. Here, we mainly focus on the following three issues: (a) how to acquire data from sensors of various specifications, (b) how to refer to past data of sensors, and (c) how to use data from sensors which do not actually exist but can be defined using some other sensors. To solve the three issues mentioned above, in this paper, we present a new middleware named *Virtualized Sensor System (VSS)* which provides three helpful features for sensor users as follows: (a) a function for unifying access methods and data formats, (b) a database for managing all the past data from sensors, and (c) a function for creating software-defined sensors. From these features of our proposed system, sensor users can easily access all sensors using unified methods, even their past data, regardless of the actual existence of them (see Figure 1).

B. Related Works

A system for unifying interfaces of IoT devices using SQL query is proposed [6]. In [6], Unified IoT Device Query (UDQ) system allows application programmers to use an unified interface for using sensors. UDQ system is capable of unifying interfaces of only hardware sensors. However, to use some calculated data using some hardware sensors, an application programmer has to implement some methods for its calculation and accession for all the necessary sensors.

A system to manage network-connected sensors by applying Simple Network Management Protocol (SNMP) technology is also proposed [7]. The data generated by network-connected sensors which are managed by SNMP are stored in virtual database called Management Information Base (MIB).

 ISBN: 978-1-61208-692-7

SNMP agents have MIBs, and they output the data in response to requirements from SNMP managers. SNMP managers are able to access MIBs that SNMP agents have by using commands defined in SNMP. However, only hardware sensors can be managed in the system proposed in [7], and the system does not allow users to define new sensors' data which are calculated by data that hardware sensors generate.

A middleware named *Global Sensor Networks (GSN)* [8] for processing sensor data in sensor networks is proposed. *GSN* enables users of *GSN* to define new sensors through an XML-based language. A new sensor defined by a user of *GSN* is called a virtual sensor. Application programmers can acquire data from new sensor networks consisting of defined virtual sensors in the same way. However, the XML-based language which is used in *GSN* makes a user of *GSN* difficult to define a virtual sensor which is referred in a particular order because a user of *GSN* cannot specify conditions which determine the order directly. *VSS* provides a way to express a sensor which is referred in a particular order to users of *VSS*, and users of *VSS* can define such sensors easier than users of *GSN* (i.e., users of *VSS* can direcly define such sensors). For example, a new temperature sensor that notifies to users when the following event occurs can be defined by users of *VSS* easier: an event that sensor's data generated by a hardware temperature sensor drops below 20 degrees only after sensor's data generated by the same sensor becomes greater than 20 degrees.

A network simulator for Wireless Sensor Network (WSN) named *Configurable Multi-Layer WSN (CML-WSN)* is proposed in [9]. Each node in WSN has a capability for sensing and ad-hoc network is constructed based on each node's communication range. One of the main problems of ad-hoc networks is that there is no infrastructure, so the routes change dynamically. This may cause a decrease in quality of services, much power consumption, or security problems. As a result, communication problems, e.g., packet loss, can be occurred. To address the problems, *CML-WSN* allows users to create network topologies, configure devices, inject packets, and change network settings. However, different points between *CML-WSN* and *VSS* are the features which are provided to sensor users, and *VSS* provides methods to users of *VSS* that make developments of applications easy.

C. Organization of this paper

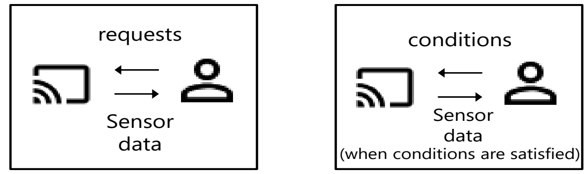
The rest of the paper is structed as follows. We introduce some preliminaries to help to understand our work in Section II. Section III presents our proposed system *VSS* and its prototype system is presented in Section IV. A conclusion and future works are given in Section V.

II. Preliminaries

Preliminaries to help to understand our work are introduced in this section.

A. Sensors

In this paper, we call a device generating any digital data which can be processed by any computational entities a sensor, this means that sensors considered in this paper have wider meaning than typical sensors. For example, we can consider OpenWeatherMap [5], which is a web-based service providing weather information, as a sensor even no hardware sensor exists. We call any devices or services generating any digital data an *actual sensor*.



(a) Pull (b) Push

Figure 2. Pull and Push data acquisition

B. Two acquisition types: Pull and Push

Pull: A pull acquisition is an acquisition method to acquire sensor data immediately whenever it is requested.

Push: A push acquisition is an acquisition method to acquire sensor data by notification from sensors when one or more preset conditions by the sensor user are satisfied.

These imply that a pull and a push acquisition can be expressed as an active and a passive acquisition respectively. These two acquisition types are depicted in Figure 2.

C. Software-defined Sensor

In this paper, we call a sensor which is created by composing of *actual sensors* a *software-defined sensor*. For example, a water vapor capacity sensor which can be calculated using one temperature sensor and one humidity sensor, and a motion sensor consisting of several motion sensors, which are referred in a particular order are both *software-defined sensors*. Unified methods provided to sensor users as an API enable sensor users to use *actual sensors* and *software-defined sensors* in the same way. We call a user who defines a *software-defined sensor* in order to provide its data to sensor users a system manager. We present an XML-based language named *Virtual Sensor Markup Language (VSML)* which enables system managers to define new *software-defined sensors*. We call a file written in *VSML* a virtual sensor definition file. We assume that there are processes which interpret virtual sensor definition files and create requested *software-defined sensors* at all times.

III. Proposed System: *VSS*

A design and an implementation of proposed system *VSS* are introduced in this section.

A. Overview of VSS

An overview of *VSS* is shown in Figure 3. *VSS* has four main layers in order to solve requests from sensor users illustrated in Figure 1. Firstly, sensor users can use virtual sensors in *VSS* using two acquisitions (Pull and Push) regardless of the specifications of the sensors due to *Unifying Data Acquisition Layer*. Secondly, *VSS* enables sensor users to use unified data formats due to *Unifying Data Format Layer*. Thirdly, *VSS* enables sensor users to use all the past data due to *Data Management Layer*. Finally, *VSS* enables sensor users to use *software-defined sensors* defined by system managers due to *Creating Software-Defined Sensors (S.D.S) Layer*.

To help to understand the features of our proposed system, we give an example scenario here. Assume that there are two different temperature sensors denoted by t_1 and t_2 respectively and two different humidity sensors denoted by h_1 and h_2. And we also assume a sensor user who considers to implement a context-aware application using these all sensors and two additional special sensors as follows: (a) a sensor outputs the

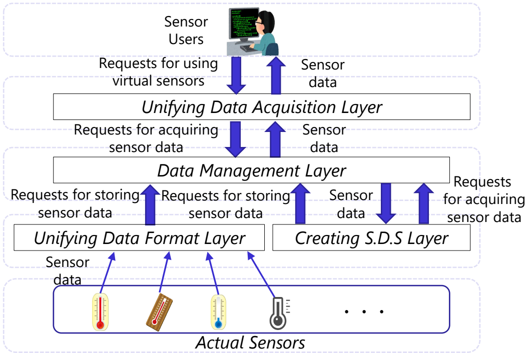


Figure 3. Architecture of our proposed system *VSS*

average value of two temperature values which are generated by two different temperature sensors a year ago respectively, and (b) a sensor of the same function as the sensor (a) using two different humidity sensors. Let t_3 (resp. h_3) be the former one (resp. the latter one). Note that the temperature sensors t_1 and t_2 are *actual sensors* but t_3 is a *software-defined sensor* in this case. If the sensor user arranges all these 6 sensors, the user must understand the specifications of 4 *actual sensors*, and has to maintain database for storing sensor data from at least 1 year ago. The sensor user easily uses all sensors introduced above using the API provided by *VSS* even the user never knows these sensors' specifications. Particularly, *Unifying Data Format Layer* unifies all sensors' data formats and units. Note that the data format and the unit depend on the specification of each sensor. Thus, the sensor user does not need to know the specifications of 4 *actual sensors*. *Data Management Layer* maintains all sensors' data, hence the sensor user can access the sensor data a year ago. *Creating S.D.S Layer* allows users to create any *software-defined sensors* freely, so the sensor user can create two *software-defined sensors* in this case. Finally, *Unifying Data Format Layer* provides many access methods for virtual sensors in the system as an API. This layer allows the sensor user to acquire sensors' data easily. These all layers are illustrated in Figure 3. As the example scenario introduced above, our proposed system *VSS* offers many helpful layers for sensor users who want to develop applications (especially, context-aware applications) using various sensors. The detail of each layer will be explained in the next subsection.

B. Layers which construct VSS

VSS consists of four layers as shown in Figure 3. Each layer in Figure 3 consists of one or more modules, and each module in any layers is a function, which consists of the following three components: (a) input data, (b) steps of an execution, and (c) output data. Note that any implementations of each layer in Figure 3 is encapsulated from any modules in another layer.

1) Unifying Data Format Layer: Modules in *Unifying Data Format Layer* receives sensor data from *actual sensors*, and converts the format of the sensor data into the predetermined unified format.

2) Data Management Layer: *Data Management Layer* manages all sensor data received from drivers in *Unifying Data Format Layer*. Each sensor data received in *Data Management Layer* is stored in its local database with its received time as its timestamp. *Data Management Layer* realizes the following two acquisition types: (a) returns sensor data which is maintained in its local database when it requested, and (b) returns sensor data which is maintained in its local database when its received data is changed.

3) Unifying Data Acquisition Layer: Modules in *Unifying Data Acquisition Layer* receive requests from sensor users and realize the two types of acquisition (Pull and Push).

Pull: A module for pull acquisitions requests the most recent sensor data before the specific time which is requested by a sensor user to *Data Management Layer*, and forwards it to the sensor user.

Push: A module for push acquisitions receives a sensor's name (consisting of a sensor's type, a unit of output data, and a datetime), a condition, and an event handler in advance, and when this layer detects satisfactions of conditions required by sensor users, the module notifies sensor users by calling event handlers specified by them.

4) Creating S.D.S Layer: *Creating S.D.S Layer* creates *software-defined sensors* defined by virtual sensor definition files, and these sensors can be used as *actual sensors*. *Creating S.D.S Layer* maintains virtual sensor definition files and interpret them. *Creating S.D.S Layer* bring sensor data which are necessary for calculations from *Data Management Layer*, calculate data using them, and send calculated sensor data back to *Data Management Layer*. System managers can define two types of *software-defined sensors*, combination-type and sequence-type, using *VSML*. When a system manager sends a virtual sensor definition file to *VSS*, *Creating S.D.S Layer* creates a *software-defined sensor* based on a type which is written in the file. Two types of specifications for system managers to define *software-defined sensors* using *VSML* are presented in the following paragraphs.

Combination type: Modules for combination-type *software-defined sensors* have three roles. Firstly, these modules bring sensor data which are necessary for calculations from *Data Management Layer*. Secondly, these modules calculate data using current sensors' values. Finally, these modules send calculated data to *Data Management Layer*. System managers define a combination-type *software-defined sensor* by specifying the following three items: (a) a description of one or more method names of pull acquisition (i.e., which sensor data are used for calculations), (b) a formula for calculations (i.e., how to calculate data specified in (a)), (c) a description of metadata consisting of the following three items: (i) a sensor's type (e.g., temperature), (ii) a unit of output data, and (iii) a data type which implies that defined sensor data consists of only a datetime, or consists of a value and a datetime.

Sequence type: Modules for sequence-type *software-defined sensors* have three roles. Firstly, these modules maintain each sensor's state in its local storage. Secondly, these modules transit the state if necessary referring received sensor data in its chronological order. Finally, this state transition determines the output data and its output timing, and these modules send the determined output data by the state transition to *Data Management Layer*. In this paper, a state transition table is used for maintaining a history of sensor data to create sequence-type *software-defined sensors*. System managers define a sequence-type *software-defined sensor* by specifying the following three items: (a) a description of one or more method names of push acquisition (i.e., which conditions are used for state transitions), (b) a description of a state transition table (i.e., how to determine output data using conditions specified in (a)), (c) a description of metadata consisting of the following three items: (i) a sensor's type (e.g., temperature), (ii) a unit of output data, and (iii) a data type which implies that defined sensor data consists of only a datetime, or consists of a value and a datetime.

C. Implementation of VSS

In this subsection, we explain the implementation of each module in each layer, presented in the previous subsection, in our proposed system *VSS*.

1) Unifying Data Format Layer: Unifying Data Format Layer maintains sensor drivers. Each driver is corresponding to each *actual sensor* in one-to-one. When a driver receives sensor data from its corresponding *actual sensor*, the driver converts the format of the sensor data into the predetermined unified format. Each driver sends sensor data and its corresponding metadata to *Data Management Layer*. A metadata consists of a sensor's name, a sensor's type, a unit of output data, and a data type which implies that the output data may change or not. A metadata of each *actual sensor* is generated and sent by its corresponding driver.

2) Data Management Layer: Data Management Layer maintains a local database for storing sensor data and metadata. It stores all the sensor data received from *Unifying Data Format Layer* and *Creating S.D.S Layer* to its local database with its timestamp (i.e., receipt time of sensor data), and it sends corresponding sensor data to a module in *Unified Data Acquisition Layer* or *Creating S.D.S Layer* when it requested.

3) Unifying Data Acquisition Layer: Unifying Data Acquisition Layer consists of two modules for the two acquisition types introduced in Section III.B. A module for pull acquisitions provides a method for using pull acquisition illustrated in Section III.B.3) to sensor users. A module for push acquisitions consists of two components, a module for judging conditions, and an Event-Condition-Action (ECA) rule database. The ECA rule is a rule for an event-driven action. An ECA rule is generally represented by the following syntax: On *Event* If *Condition* Do *Action*, which means that when the *Event* occurs, if a system satisfies the *Condition*, then the system executes the *Action*. A module for judging conditions receives ECA rules specified by sensor users. An ECA rule received by a sensor user consists of the following three components: (a) a sensor's name for an *Event*, (b) an expression which includes the sensor's name specified in the *Event* for a *Condition*, and (c) an event handler which is called when the *Condition* is satisfied for an *Action*. For example, when a sensor user requests to call an event handler when the output value from one sensor T becomes greater than 20, a module for judging conditions meets the requirement if the user assigns $T > 20$ as a *Condition*. An algorithm for push acquisitions is executed as the following steps.

(Step1) When a sensor user sends a sensor's name, a condition, and an event handler to the module for judging conditions, a module for judging conditions registers the received sensor's name as an Event, the received condition as a Condition, and the received event handler as an Action. This set of an Event, a Condition, and an Action is registered as an ECA rule. The module for judging conditions stores every ECA rule to its database. (Step2) Whenever each sensor's name registered as an Event outputs a data with its timestamp, the module for judging conditions acquires it from *Data Management Layer*. (Step3) When the module for judging conditions acquires new data from a sensor registered as an event, it checks the conditions referring the corresponding ECA rules in ECA rule database. If the conditions are satisfied, the module for judging conditions proceeds the next step, Step4. (Step4) The module for judging conditions determines each event handler corresponding the ECA rules, and call it.

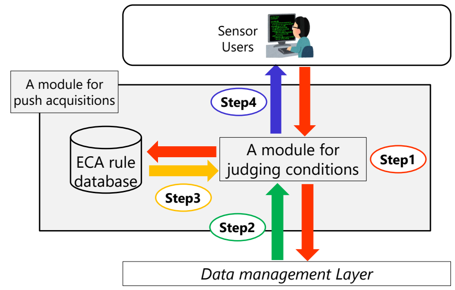


Figure 4. A module for push acquisitions

Diagrams which illustrate how steps of proposed algorithms proceed are depicted in Figure 4, 5, and 6. One or more arrows of the same color as the frame color of the ellipse surrounding each step ID illustrated inside it represent the corresponding dataflow described in an algorithm. A diagram for verifying the different steps of the above algorithm is illustrated in Figure 4.

4) Creating S.D.S Layer: Creating S.D.S Layer consists of the following four components: (a) virtual sensor definition files, (b) a module for interpreting virtual sensor definition files, (c) a module for combination-type *software-defined sensors*, and (d) a module for sequence-type *software-defined sensors*. Specifications of the module (a) are presented in subsection III.B, and the module (b) can be implemented with an XML parser, e.g., Document Object Model (DOM), which is available as an API. And two implementations which realize two modules, (c) and (d), are presented in the following paragraphs.

Combination type: A module for combination-type *software-defined sensors* consists of the following two components: (a) a module for interpreting virtual sensor definition files, and (b) a module for exchanging data with *Data Management Layer* and calculating data using values which are got from *Data Management Layer* based on those definition files. The module (b) consists of for corresponding one combination-type *software-defined sensor*. For example, we assume that two sensors A and B exist in the system. In this case, a system manager can create a new sensor C as a *software-defined sensor* which returns the average value of the output values of two sensors A and B. The formula to calculate the output value of C is maintained by the above module (b). An algorithm for a combination-type *software-defined sensor* is executed as the following steps.

(Step1) A system manager sends a virtual sensor definition file (shortly, definition file) to *VSS*. The definition file includes (a) a list of methods for acquiring the values of the sensors which are necessary for calculating a new value, (b) a formula for a calculation, and (c) metadata of the required sensors. These are stored in the module for calculations. (Step2) The module for calculations gathers all the sensor values using

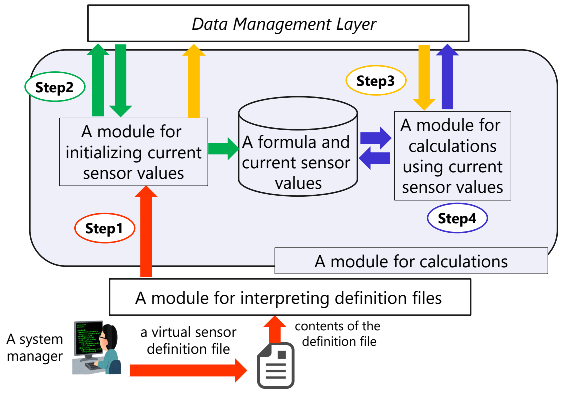


Figure 5. A module for calculations

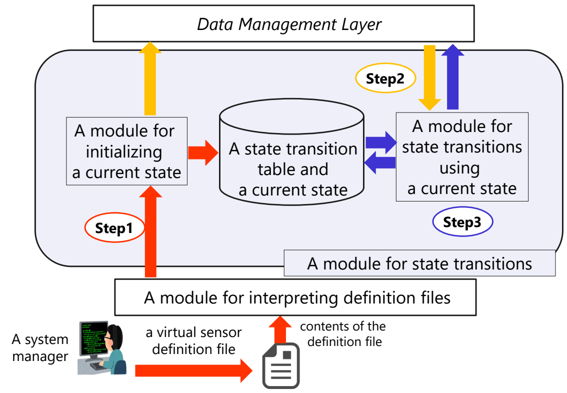


Figure 6. A module for state transitions

methods in the definition file, and calculate the value using the gathered values referring the formula in the definition files. This value is stored in this module temporarily. (Step3) The module for calculations sends a request to send the corresponding sensor values whenever they are updated to *Data Management Layer*. This implies that a push acquisition from *Data Management Layer* is realized by this request. Whenever the module for calculations receives a new sensor data, the module for calculations enqueues it to its local maintaining queue. (Step4) If the local queue is not empty, the module for calculations dequeues one sensor value and calculates a new value again referring the stored formula. This new value is compared with the value stored temporarily in Step3, and the newly calculated value is sent to *Data Management Layer* as a new updated value if it is different with the temporary value. Otherwise, the new value is discarded. This process will be repeated until its local queue becomes empty.

A diagram for verifying the different steps of the above algorithm is illustrated in Figure 5.

Sequence type: A module for sequence-type *software-defined sensors* consists of the following two components: (a) a module for interpreting virtual sensor definition files, and (b) a module for exchanging data with *Data Management Layer* and transiting its current state based on those definition files. The module (b) consists of each module for corresponding one sequence-type *software-defined sensor* defined by system managers. For example, when a system manager defines a *software-defined sensor* T which sends sensor data to *Data Management Layer* based on state transitions shown in TABLE I, a module for state transitions maintains a state transition table and a current state of the table. The value ϵ in TABLE I represents that no data is sent to *Data Management Layer* even if the condition is satisfied.

TABLE I. AN EXAMPLE OF STATE TRANSITION TABLE

Source	Destination	Condition	Output
0	1	$T > 20$	ϵ
1	0	$T < 20$	"return20"
1	2	$T > 21$	ϵ
2	1	$T < 21$	"return21"

Note that every state transition in a module for state transitions implies that the module for state transitions transits the current state which is represented with a source to the state which is represented with a destination defined by a system manager. An algorithm for a sequence-type *software-defined sensor* is executed as the following steps.

(Step1) A system manager sends a virtual sensor definition file (shortly, definition file) to *VSS*. The definition file includes (a) a list of methods for acquiring the conditions of the sensors which are necessary for transiting a current state, (b) a state transition table consisting of the following four items: (i) a state which represents a source, (ii) a state which represents a destination, (iii) sensor data which is sent to *Data Management Layer* when the transition from the source to the destination occurs, and (iv) a method name of a push acquisition, (c) an initial state name of the state transition table specified in (a), and (d) metadata of the required sensors. These are stored in the module for state transitions. The initial state is set as a current state of the state transition table. (Step2) The module for state transitions sends a request to send the corresponding conditions whenever they are satisfied to *Data Management Layer*. This implies that a push acquisition from *Data Management Layer* is realized by this request. Whenever the module for state transitions receives a new sensor data, the module for state transitions enqueues the condition which is satisfied in this case to its local maintaining queue. (Step3) If the local queue is not empty, the module for state transitions dequeues one condition. The module for state transitions updates the current state with the state which represents destination referring the stored state transition table and the current state. If the state transition occurs by the condition and data which is specified by a system manager is a string of letters except for a special symbol ϵ referring the stored state transition table, the module for state transitions sends the data to *Data Management Layer*. This process will be repeated until its local queue becomes empty.

A diagram for verifying the different steps of the above algorithm is illustrated in Figure 6.

IV. PROTOTYPE SYSTEM

We implement the prototype system of our proposed system to verify its operation.

A. Environment of the prototype system

The prototype system has three kinds of sensors: a temperature sensor (BD1020HFV), a temperature and humidity sensor (DHT11), and three motion sensors (Grove-PIR Motion Sensor) as shown in Figure 7.

In our prototype system, all the layers of *VSS* are implemented using Java, and we use the file system of Windows 10, NTFS, for the database of *Data Management Layer*. WatchService [10], which is provided by a basic library of

 ISBN: 978-1-61208-692-7

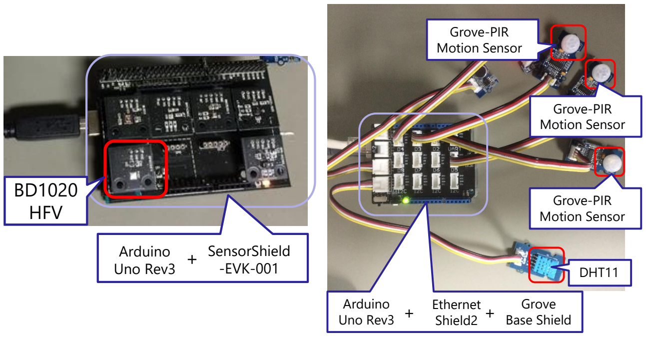


Figure 7. Hardware sensors used in the prototype system

Java, is used for detecting changes of sensor data in *Data Management Layer*. The environment of the prototype system is shown in TABLE II.

TABLE II. EXPERIMENTAL ENVIRONMENT

CPU	Intel Core i5 6500 3.20GHz
OS	Windows 10
RAM	DDR4 8 GB
JRE	1.8.0_172-b11

As an example of the *software-defined sensor*, we introduce a new *software-defined sensor* which varies its output value by the sequence of the output of three different motion sensors (see Figure 8).

```
<VirtualSensor name="multimotion" type="sequence">
  <states>
    <statename>0</statename>
    <statename>1</statename>
    <statename>2</statename>
  </states>
  <initialstate>
    <statename>0</statename>
  </initialstate>
  <functions>
    <function>
      <source>0</source>
      <condition>watchMotionUpdate("MOTIONa",
          currenttime, "change")</condition>
      <output>a</output>
      <destination>1</destination>
    </function>
    <function>
      <source>1</source>
      <condition>watchMotionUpdate("MOTIONb",
          currenttime, "change")</condition>
      <output>ab</output>
      <destination>2</destination>
    </function>
    <function>
      <source>2</source>
      <condition>watchMotionUpdate("MOTIONc",
          currenttime, "change")</condition>
      <output>abc</output>
      <destination>0</destination>
    </function>
  </functions>
  <description>
    <sensortype>motion</sensortype>
    <datatype>string</datatype>
    <unit>null</unit>
  </description>
</VirtualSensor>
```

Figure 8. A definition of a new motion sensor

B. Latency for push acquisitions

We confirm the average latency of push acquisitions, and as a result, we found that the average latency is 3.65ms in the case of prototype system. We expect to estimate latencies which are confirmed in other environments of implementations of *VSS*.

V. CONCLUSION AND FUTURE WORKS

In this paper, we proposed *VSS* which provides unified methods to access any sensor regardless of their specifications, and can create new *software-defined sensors* by composing of one or more *actual sensors*.

To verify the operations of our proposed system *VSS*, we implemented the prototype system, and we found our system is operated well and push acquisitions have practically low latency in the case of our prototype system. However, some problems can be considered if our system is implemented in the distributed manner, e.g., a system consisting of two or more computational entities which are connected by networks. Especially, there are several important issues which have to be discussed in an implementation of *Data Management Layer*. For example, we may consider the following issues: (a) a fault-tolerance; a system guarantees its availability even if some nodes in the system fail, and (b) load balancing; to ensure a reliability of the system, a load of each node should be balanced as possible. Finally, we also consider that how to implement a sequence-type *software-defined sensor* can consistently process the message from every sensor in the actual order in which it is sent.

REFERENCES

[1] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A Literature Review," Journal of Computer and Communications, volume 3, pp. 164-173, 2015.

[2] S. Pote, "Internet of Things Applications, Challenges and New Technologies," International Conference on Advances in Computer Technology and Management (ICACTM), At Pimpri Chinchwad, Pune, pp. 45-51, Feb 2018.

[3] Z. K. A. Mohammeda, and E. S. A. Ahmedb, "Internet of Things Applications, Challenges and Related Future Technologies," World Scientific News, Vol. 67, pp. 126-148, 2017.

[4] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," International Journal of Ad Hoc and Ubiquitous Computing, Vol. 2, Issue 4, pp. 263-277, June 2007.

[5] OpenWeatherMap, URL: https://openweathermap.org/ [retrieved: January, 2019].

[6] S. Kinoshita, Y. Kuga, and O. Nakamura, "Unifying Interfaces of IoT Devices Using SQL Query (in Japanese)," Information Processing Society of Japan Multimedia, Distributed, Cooperative, and Mobile(DICOMO) Symposium, pp. 1036-1041, 2016.

[7] H. Hui-Ping, X. Shi-De, and M. Xiang-Yin, "Applying SNMP Technology to Manage the Sensors in Internet of Things," The Open Cybernetics & Systemics Journal, pp. 1019-1024, 2015.

[8] K. Aberer, M. Hauswirth, and A. Salehi, "The Global Sensor Networks middleware for efficient and flexible deployment and interconnection of sensor networks," Swiss Federal Institute of Technology, Lausanne (EPFL), Tech. Rep., pp. 1-21, 2006.

[9] C. Del-Valle-Soto, F. Lezama, J. Rodriguez, C. Mex-Perera, and E. M. de Cote, "CML-WSN: A configurable multi-layer wireless sensor network simulator," Applications for Future Internet: International Summit, pp. 91-102, AFI 2016, Puebla, Mexico, May 25-28, 2016.

[10] WatchService, URL:https://docs.oracle.com/javase/7/docs/api/java/nio/file/WatchService.html [retrieved: January, 2019].

 ISBN: 978-1-61208-692-7

WADcher: a Unified Web Accessibility Assessment Framework

Aggeliki Spyrou[1], Nikolaos Kaklanis[1], Dimitrios Tzovaras[1], Yehya Mohamad[2], Henrike Gappa[2], Carlos A Velasco[2], Martin Breidenbach[2], Serena Caria[3], Fabio Paternò[3], Francesca Pulina[3]
[1]Information Technologies Institute, Centre for Research and Technology Hellas, 6th Klm. Charilaou-Thermi Road, Thessaloniki, Greece
email: {aspyrous, nkak, Dimitrios.Tzovaras}@iti.gr
[2]Fraunhofer Institute for applied Information Technology (FIT), Schloss Birlinghoven, 53757 Sankt Augustin, Germany
email: {yehya.mohamad, henrike.gappa, carlos.velasco, martin.breidenbach} @fit.fraunhofer.de
[3] CNR -ISTI, Human Interfaces in Information Systems Laboratory, 56124 Pisa, Italy
email: {serena.caria, fabio.paterno, francesca.pulina}@isti.cnr.it

Abstract— **This paper presents the architecture of Web Accessibility Directive Decision Support Environment (WADcher), a unified web accessibility assessment framework for large-scale assessment of compliance against web accessibility recommendations and legislations. It aims to provide customized support for various user groups like web commissioners, web developers and accessibility experts. To achieve this, the WADcher platform offers tools and interfaces like an Observatory for visualization of the assessment results, a Decision Support Environment for expert reviews and Application Programming Interfaces (APIs) to existing tools.**

Keywords- web accessibility; WCAG; WAD; web accessibility standards; EARL.

I. INTRODUCTION

The web is a fundamental resource related with many aspects of daily life, e.g., work, leisure, education, commerce and social networking. Beyond this, access to services offered through the web provided by public authorities is important for all citizens to stay informed, interact with authorities, fulfill obligations and apply for benefits. To ensure access for disabled, elderly and other disadvantaged persons to websites and mobile applications of public authorities and essential web resources of the private sector, the EU Web Accessibility Directive (WAD) on the “accessibility of the websites and mobile applications of public sector bodies” came into force on 26/10/2016 [11]. The WAD’s accessibility requirements describe what must be achieved to enable the user to be able to perceive, operate, interpret and understand a website, a mobile application and related content. WAD aims to ensure that the websites and mobile applications of the public sector are made accessible on the basis of internationally acknowledged accessibility requirements.

When aiming for web accessibility, it is inevitable to establish procedures to analyse how well the web can be used by people with different disabilities, i.e., assess the accessibility status of a website. Having an automated way to assess the accessibility of web pages opens the way to perform large-scale analysis of web accessibility [5][6]. Large-scale accessibility evaluations of the Web are not yet well established. This may be due to the dependency of computational resources for large-scale analysis [1][7]. Current assessment tools have a number of limitations, among others, they cannot detect all issues, manual checking with accessibility experts is still needed [9]. Moreover, the majority of current automatic accessibility assessment tools does not produce results in a standardized machine-readable format as Evaluation and Report Language (EARL) [12].

To push forward efforts in making web sites accessible European-wide, monitoring of the accessibility status of particularly public web sites by the EU has become part of WAD. For this purpose, the European funded research project WADcher aims to develop an integrated platform to verify the compliance of websites and mobile applications of public sector bodies with the accessibility requirements set out in the WAD, and subsequent Commission implementing decisions on monitoring methodologies [13] and model accessibility statements [14]. WADcher will provide APIs to automatic web accessibility assessment tools, so any third-party tool can be connected to WADcher by just using the corresponding APIs. The WADcher platform will provide support to conduct automatic assessment using any tool that is registered and supports the published WADcher APIs. Additionally, WADcher will provide a workbench to support manual checking conducted by accessibility experts and the production of customized reports for different target user groups in a standardized machine-readable EARL format. The WADcher system is built on proven state-of-the-art accessibility assessment tools and will provide an easy-to-use, cost-effective, harmonised accessibility periodic monitoring platform that is transparent, transferable, comparable and reproducible to meet the European standard EN 301 549 V1.1.2 (2015-04) [15] as specified in the WAD. WADcher will provide a management layer that supports assessment and tracking of the accessibility requirements set out in the WAD.

The Directive encourages both Member States and the Commission to promote use of such compatible tools. This paper elaborates on the following aspects: the state of the art of tools for accessibility validation, also considering studies conducted in WADcher to gather user requirements, the main use cases identified for the pilots of WADcher and the general design and architecture.

The rest of this paper is organized as follows. Section II describes the carried-out State of the Art Analysis. Section III describes the user requirements collected from users in specific countries. Section IV addresses WADcher architecture approach. Section V provides details about the use cases. The acknowledgement and conclusions close the article.

II. STATE OF THE ART ANALYSIS

In order to review the state of art in WADcher among the many tools available to developers for assessing the accessibility of websites, various open tools as well as well-known commercial solutions for web accessibility services were examined. As a first source of information the list of tools provided by the W3C Consortium was considered, focusing first on those that check compliance against the WCAG 2.0 guidelines and support at least the English language (so, we have excluded tools like Hera FFX [1] and Vamolà [8], for example). Then, we looked at those supporting analysis of at least both (X) HTML and CSS, and covered most guidelines (thus we discarded those considering for example only colors) and that are freely accessible, available either online or as desktop applications. On the commercial side, we examined various tools such as those offered by imergo [10], Deque Systems [16], Siteimprove [17] and Make Sense [18]. The analysis conducted considers also the free Deque plugins for Chrome and Firefox – aXe plugins [19] – and the Siteimprove Accessibility checker for Chrome [20].

In addition to supporting WCAG 1.0 and WCAG 2.0, some of the considered tools allow web designers and developers to check the accessibility against additional national guidelines: MAUVE [1][3], for example, enables the user to select the Stanca Act for the assessment; Asqatasun the French guidelines (RGAA); AChecker [22] and imergo the guidelines of the German government (BITV); Siteimprove extends the available guidelines to the Japanese Industry Standard (JIS).

When analyzing the tools' features, various technical aspects were considered: firstly, the audit depth, that is the assessment scope (if tools can assess single web pages and/or entire websites, and also password protected pages); secondly, the type of input needed (e.g., URL, file upload, directly entered code). We have also checked if they allow the customization of accessibility audits, for example if the developer can specify the guidelines or their priority level for the assessment.

Moreover, focus on the presentation of test results was given. First, we identified the report type: if it is mainly code-oriented, statistical, or graphical (or a mix of these). Then, we looked at the differentiation they make between different types of problems: errors, warnings, potential problems and information that usually require additional manual check. In the end, we observed both the summarizing overview and the full report of the assessment that they provide, and the kind of filtering that it is possible to apply to the results in order to get specific information.

Among free accessibility assessment tools, Asqatasun, TAW Web [23], the A11Y Machine [24], WaaT [2] allow the web designer/developer to evaluate more than a single web page at a time (Total Validator [25] allow to evaluate entire websites in the Pro version). The assessment scope of the commercial services is extended to more than one web page at a time, except for Deque's Amaze [26]. The extension of MAUVE to evaluation of multiple pages at the same time is under development.

Customization of accessibility audits is supported by the majority of the free tools (9 out of 12): AChecker, Asqatasun, Cynthia Says, imergo, MAUVE, TAW Web, the A11Y Machine, Total Validator, and WaaT. However, there are different levels of customization: some of them allow the selection of the preferred guidelines only (among those supported) while others allow also the selection of the priority level of the guidelines.

Regarding the presentation of the assessment results, a summarizing overview is mostly supported, which provides the total number of issues encountered, divided into the different kinds of problems found during assessment (errors, warnings, info/potential problems,), and some kind of filtering is also possible at this stage. Moreover, some of the tools – such as MAUVE, which is code oriented – provide the entire source code with errors and warnings highlighted, and each problem has a link to the corresponding W3C success criteria and techniques description.

The majority of the tools do not provide visualization of the errors/warnings on rendered web page, since they are mostly code-oriented. WAVE is the only free tool that provides this kind of report, even if it provides also the pinpointing of problems in the source code. Regarding commercial solutions, Deque's aXe browser extensions allows the developer to highlight issues on the running web page; Siteimprove extension provides this kind of pinpointing too. Thanks to its customised rendering engine, imergo® provides a detailed analysis of the accessibility issues of a website as perceived by the user in a browser.

The A11Y Machine, Siteimprove Intelligence Platform and WorldSpace Comply [27] provide, in the full report, a dashboard that keeps tracks of the accessibility improvements, allowing the monitoring of the accessibility status over time. As regards other more specific information provided in the full reports, only the A11Y Machine provides the pinpointing of problems in the DOM.

All the tools analyzed provide some sort of filtering of the results: among them, we found that Asqatasun is the only one that provides the filtering of reports by Web content type. An interesting feature offered by Siteimprove – both the Intelligence Platform and the browser extensions – is the possibility to filter the results taking into account the role of the user (Editor, Webmaster, and Developer). Deque Systems, instead, offers different services, each one tailored on the accessibility competence of the user: for example, Amaze is thought for developers who are not accessibility experts, while comply is addressed to accessibility managers and experts.

From our analysis, we can identify three main findings. Firstly, reports provided are mainly code-oriented: the visualization of fails and warnings on the rendered web page is a feature that could be useful for not accessibility experts and commissioners, who are not primarily interested into details related to the code to be fixed. Another feature that is not frequently supported is the locating of accessibility issues in the DOM that, on the other hand, is an information that could be useful for accessibility experts and developers. Secondly, other useful features that are not currently supported are related to the filtering of the results. In the summarizing overviews, it is always provided filtering by status of the issues and by priority level of the guidelines, but the possibility of filtering issues by web content type is never given to users. This kind of filtering could be provided also in the full report, as Asqatasun does. It would allow web developers and accessibility experts to better manage their work of fixing errors, by considering problems related to an aspect at a time. The last missing feature that we can point out is the possibility to filter the full report by the frequencies of the errors detected (e.g., single or common issues); as for the web content type filtering, it could be useful especially for the accessibility experts' audience.

WADcher aims to provide developers, designers, accessibility experts and policy makers an integrated and personalised web accessibility support environment by also addressing the aforementioned issues of existing tools.

III. User Requirements Analysis

The design of the WADcher framework is based on both analysis of existing tools, as presented in previous section, and the needs of target users. In order to identify and analyse the needs of the various target users involved, we carried out three main requirements elicitation activities: questionnaires, interviews and a workshop. In particular, for the questionnaires, 148 people across five countries (Ireland, Italy, Greece, Austria, and Germany) filled in the provided online questionnaires. As for the interviews, one interview was done in Germany and it involved a web content author/ accessibility expert; other two interviews were conducted in Italy and involved a web content editor and a web editorial staff member. In addition, a workshop was organized in which about forty people composed of mainly web developers, content writers and accessibility experts participated.

A first general consideration is that people involved in our elicitation activities have limited knowledge of accessibility assessment tools, and they usually encounter difficulties in considering both accessibility and evaluating it in their projects. Regarding web commissioners, if they do not take into account accessibility is because of lacking knowledge and competences, and lack of time and resources. Web developers and accessibility experts, instead, cannot work towards ensuring that websites and application meet the accessibility standards because of commissioner's imposition, lack of time, knowledge and for budget limitations.

When working to ensure accessibility with the automatic assessment tools, the main difficulties regard the inclusion of dynamic content in the evaluation process, the restrictions imposed by Content Management Systems, and some limitations of such tools, such as: the detection of several false positives; they do not perform a thorough check (often not all guidelines are covered); the limited guidance on how to fix the detected issues, and more and clearer information (explanation, suggestions, examples) about the errors is desirable.

Based on their knowledge of available accessibility assessment tools, they recommended the features they would expect from them. Among those, the following were indicated: adaptation of the report to the technical level of the user, by providing one more technical report addressed to developers, and one addressed to commissioners and content authors; analysis of dynamic content, suggestion for JavaScript frameworks and libraries; giving a percentage measure of the overall accessibility level reached.

IV. WADcher Framework Architecture

The core components of the WADcher framework were designed by taking into account the requirements presented in the previous section and they can be split into four logical layers (See Figure 1.).

The **knowledge layer** includes all the knowledge/data needed for the proper functioning of the large scale components (e.g., list of web sites to be evaluated, guidelines and policies to be considered in web accessibility assessment, rule sets, etc.) as well as the knowledge/results that will be extracted (e.g., ontologies [28] that describe the guidelines of WCAG 2.x and ARIA standards in a semantic manner, etc.).

The **service layer** includes all the core software components of the large-scale cloud-based architecture. This layer includes an **Evaluation manager** component that interacts with the registered tools (Imergo, MAUVE and WaaT) and the WADcher knowledge base, it also contains the **Test sample definer** that creates automatically test samples based on methodologies like, e.g., UWEM [3], which are limited either by the number of maximum pages or the focus on a specific technical aspect (e.g., certain web

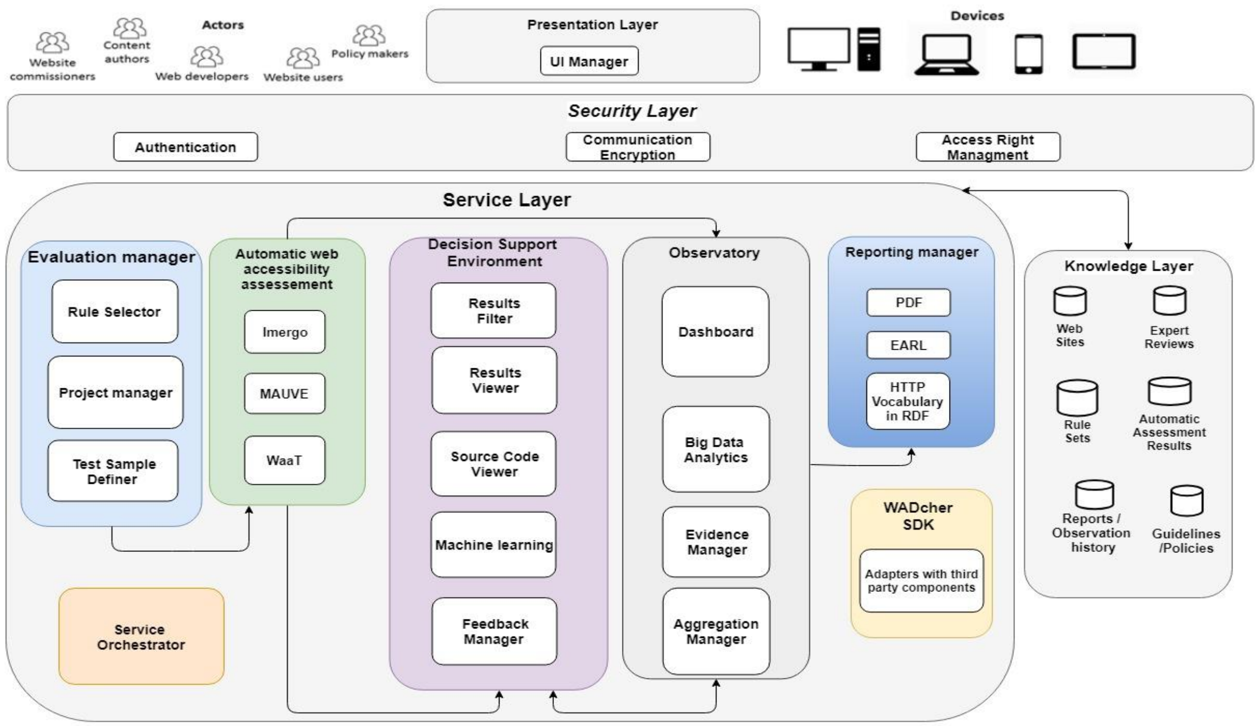


Figure 1. WADcher's architecture overview

content types). The **Rule Selector** allows the user to select the rule set or subset of rules to be used for the validation of the testing sample. Through the **Project manager component,** the user defines the properties needed for a project which can be run multiple times having receptive versions. By that, the monitoring of a website at different times became possible.

Then, the results from Evaluation manager are provided to the **Automatic Web Accessibility Assessment** component, which contains all the available tools (imergo, MAUVE and WaaT) and performs the evaluation which produces an EARL report that is provided to the Decision Support Environment and the Observatory modules.

The **Decision Support Environment (DSE)** aims at empowering accessibility experts who conduct expert reviews and web commissioner to govern the accessibility policy, statements, etc. in order to achieve this it contains components like, the **Results viewer** component which presents the results to the user and provides assistance in solving detected issues, through solutions, examples and informations about the technology related to the issue. The **Results filter** provides filtering and grouping options (e.g., errors, cannot tells, web content type) for the results presented in Results viewer and Source code viewer, in addition the **Source code viewer** component provides access to the source code of the evaluated URL and enables the user to find issues inside the source code as well as on a rendered preview**.** Furthermore, the **Feedback manager** enables accessibility experts to provide valuable feedback to the Decision Support Environment, which is considered for the refinement of rules. The goal is to make the DSE as autonomous as possible with the use of **Machine learning** algorithms that enable the automatic refinement of the rules by continuous usage.

The DSE communicates with an **Observatory** module running also on a cloud environment, contains a **Dashboard** component that presents to the user the different projects according to the WADcher Observatory data model. The responsibility for performing periodic assessments on large sets of monitored web sites and triggering notifications when the accessibility level alters between different times assumed by the **Evidence manager**, moreover the **Aggregation manager** is responsible for aggregating different types of assessment results (i.e., results coming from automatic, semi-automatic and manual tests performed by accessibility experts). In addition, a **Big data analytics** component aggregates all the testing results and, through advanced visualization mechanisms, allows the efficient monitoring of the quality of large volumes of web content.

In order to generate the final report, the data from the Observatory are send to a **Reporting manager** component, which provides the final report in various types. The supported report types are:

i. A PDF report containing all the errors and cannot tells of the evaluated web page(s).
ii. An EARL-based report of the detected errors and cannot tells.
iii. A PDF version of the EARL-based report, containing all the information of the EARL-based report in human readable format.
iv. A report containing all the HTTP content transferred between WADcher and the server that hosts the evaluated web application, using the HTTP Vocabulary in RDF.

The **Service Orchestrator** is responsible for the management of the notifications between all WADcher's components.

Furthermore, a **WADcher SDK** provides a web-based API to enable access to the services of WADcher to other external applications.

The **security layer** is responsible for the security, authentication and access rights management.

The **presentation layer** includes all the user interfaces that enable user interaction with the large-scale services and through the UI manager component, presents the results in a unified matter regardless of the device used.

V. INDICATIVE USE CASES

As WADcher aims at providing support for different target groups, those have different requirements and approaches towards furthering accessibility in the web. As has been elicited in the user requirements phase of WADcher, web commissioners mainly require a high-level assessment of their web pages while accessibility experts and web developers need to have the option to drill down to the source code level in order to check for minuscule problems and solve those. Therefore, the usage patterns of these three groups diverge heavily from each other and it is important to understand each approach in order to provide a holistic solution for all users.

Generally, WADcher is designed so that the users, especially those with little knowledge on accessibility assessment, do not have to worry about the report handling between the external assessment tools and WADcher, thus these steps are omitted in the use cases.

As the web commissioners are the focal group who is responsible not only for producing reports but also for triggering assessments and improvements their Use Case shall be explored first.

A. Web Commissioners

The user experience for the web commissioners starts with the Observatory, which represents a summarized representation of all the reports that the web commissioner has issued for resources that belong to him (i.e., web pages he/she is responsible for). Along with the reports, some general statistics regarding the accessibility of the checked sites is also provided. Within the Observatory, the web commissioner has the option to either look at existing or new reports that have been automatically compiled or returned by web developers or accessibility experts. The primary task of the web commissioners is the issuing of new automatic checks. For that, first, the address of the sites to check has to be entered, then, the commissioner selects an automatic assessment tool out of a predefined list. Then, the rules or rulesets (e.g., WCAG 2.0 Level AA) upon which the site should be checked are selected and eventually the assessment process is started. Upon completion of the automatic assessment, the web commissioner is informed and can access the compiled report through the Observatory. These reports can be forwarded to accessibility experts for further investigation or to web developers for improvements both are provided in pre-defined lists in order to assure competent handling.

As the EU Commission under the WAD requires an annual reporting, there is also the option to compile higher-level reports in accordance to national or EU requirements (EU 2018/1523).

B. Accessibility Experts

As most automatic assessment tools are not be able to test every possible accessibility issue in an automatic manner, the returned reports contain issues marked as "cannot tell", which means that the tool is not sure whether there is a problem or not. These need manual checking through accessibility experts.

The use case for the accessibility experts usually begins with a notification that a web commissioner has submitted an automatic assessment report to them and requests for manual checking by the expert. The accessibility expert then logs in to the Observatory and there receives the report for a first quick check on the identified errors and on the rules and guidelines that the sites have been checked for. For a more elaborate assessment, the accessibility expert is then able to drill down to a source code level and make annotations and comments directly to the code. After completing the manual assessment, the expert can send the supplemented report, along with a notification, back to the web commissioner for further action.

C. Web Developers

The main task for web developers is the correction of identified accessibility issues. Therefore, a developer gets notified when a commissioner has submitted a report for improvement. As with the other roles, the developer then logs in to the Observatory to have a quick assessment of the identified issues as well as the rules and guidelines that were checked for. However, the central information for this group is provided on the source code level, which the web developer can drill down to and assess automatically identified errors or the annotations made by the accessibility expert. To support the improvement of sites, some guidelines and best practices are provided alongside the errors. After completing the changes, there is the option to rerun the existing report and forward the results to the web commissioner or to simply return a notification informing about the changes made.

Besides these general use cases, all WADcher users are able to issue automatic assessment requests of web pages through the Observatory on their own, for example if a web developer wants to check the accessibility of a newly developed site or an accessibility expert deems a site non-accessible.

VI. Conclusion And Future Work

Digitalization is a rapidly moving on process affecting nearly all areas of our everyday life. Thus, it is of utmost importance that the result of this process will not lead to exclusion of user groups, such as disabled users or the elderly. For this purpose, the WAD directive became into force. However, its implementation requires effective tools for large-scale testing and monitoring of web applications also to ensure a harmonized evaluation and reporting approach throughout Europe. So far, such tools are not available and therefore will be in focus of the WADcher project. Intermediate results of this project are presented in this paper and will be summarized in the following.

The core contribution of WADcher includes an integrated platform, which will offer tools and interfaces as described below to support its main stakeholders (web commissioner, web developers and expert reviewers):

[1] An Observatory with dashboards for data visualization of aggregated assessment results enabling different stakeholders such as web commissioners, policy makers and the general public to monitor changes in the accessibility level of websites.

[2] A Decision Support Environment for expert reviewers to detect barriers that need further manual investigation and guidance to Web developers in repairing accessibility errors

[3] Infrastructure that provides also APIs to existing automatic assessment tools and other external tools like CMSs for checking webpages during design and content authoring to make them “born accessible”.

To evaluate the usefulness of WADcher developments and satisfaction of main stakeholders, we also plan to empirically validate the results of the proposed accessibility infrastructures with the relevant communities in four European countries (Austria, Greece, Italy and Ireland).

Acknowledgment

This work is supported by the EU funded project WADcher (H2020 - 780206).

References

[1] A. G. Schiavone and F. Paternò. “An extensible environment for guideline-based accessibility evaluation of dynamic Web applications,” Universal access in the information society, vol. 14, no. 1, pp. 111-132, 2015.

[2] C. A Velasco, D. Denev, D. Stegemann, and Y. Mohamad, “A web compliance engineering framework to support the development of accessible rich internet applications,” ACM. 978-1-60558-153-8 doi: 10.1145/1368044.1368054

[3] C. Bühler, H. Heck, O. Perlick, A. Nietzio, and N. Ulltveit-Moe, “Interpreting results from large scale automatic evaluation of web accessibility,” International Conference on Computers for Handicapped Persons, pp. 184-191, July 2006.

[4] F. Paternò and A. G. Schiavone, "The role of tool support in public policies and accessibility," ACM Interactions, vol. 22, no. 3, pp. 60-63, 2015.

[5] J. L. Fuertes, R. González, E. Gutiérrez, and L. Martínez, “Hera-FFX: a Firefox add-on for semi-automatic web accessibility evaluation,” International Cross-Disciplinary Conference on Web Accessibililty, p. 26-35, 2009. doi: 10.1145/1535654.1535661

[6] P. Ackermann, E. Vlachogiannis, and C. A. Velasco, “Developing Advanced Accessibility Conformance Tools for the Ubiquitous Web,” Procedia Computer Science, vol. 67, pp. 452-457, ISSN 1877-0509, 2015. doi:10.1016/j.procs.2015.11.086

[7] R. Lopes, D. Gomes, and L. Carriço, “Web not for all: a large scale study of web accessibility,” International Cross Disciplinary Confer-ence on Web Accessibility (W4A) ,p. 10, ACM, April 2010.

[8] S. Mirri, L. A. Muratori, M. Roccetti, and P. Salomoni, “Metrics for accessi-bility on the Vamolà project,” International Cross-Disciplinary Conference on Web Accessibililty (W4A) ,pp. 142-145, ACM, April 2009.

[9] T. Oikonomou et al. , “Waat: personalised web accessibility evaluation tool”. Procs. International Cross-Disciplinary Conference on Web Accessibility ,p. 19, ACM, 2011.

[10] Y. Mohamad, C. A. Velasco, N. Kaklanis, D. Tzovaras, and F. Paternò, “A Holistic Decision Support Environment for Web Accessibility,” Springer 2018, pp. 3-7, ISBN 978-3-319-94276-6, doi: 10.1007/978-3-319-94277-3_1

[11] http://data.europa.eu/eli/dir/2016/2102/oj

[12] https://www.w3.org/TR/EARL10-Schema/

[13] “Commission implementing decision on establishing a monitoring methodology and arrangements for reporting under the WAD” at https://ec.europa.eu/info/law/better-regulation/initiatives/ares-2018-2604213/feedback/add_da

[14] “Commission implementing decision on establishing a model accessibility statement under the WAD”, at https://ec.europa.eu/info/law/better-regulation/initiatives/ares-2018-2604172_en

[15] http://www.etsi.org/deliver/etsi_en/301500_301599/301549/01.01.02_60/en_301549v010102p.pdf Available from : 2019.01.10

[16] https://www.deque.com/ Available from : 2019.01.10

[17] https://siteimprove.com/en/ Available from : 2019.01.10

[18] http://mk-sense.com/ Available from : 2019.01.10

[19] https://www.deque.com/axe/ Available from : 2019.01.10

[20] https://support.siteimprove.com/hc/en-gb/articles/115002413812-Siteimprove-Accessibility-Checker-Chrome-Extension Available from : 2019.01.10

[21] https://www.tawdis.net/ Available from : 2019.01.10

[22] https://achecker.ca/checker/index.php Available from: 2019.01.10

[23] https://www.tawdis.net/ Available from : 2019.01.10

[24] https://www.npmjs.com/package/the-a11y-machine Available from : 2019.01.10

[25] https://www.totalvalidator.com/index.html Available from : 2019.01.10

[26] https://www.deque.com/tools/amaze/ Available from : 2019.01.10

[27] https://www.deque.com/tools/worldspace-comply/ Available from : 2019.01.10

[28] http://www.accessible-eu.org/index.php/ontology.html Available from : 2019.01.10

Even Smarter Data: Using Crowdsourcing to Improve Accessibility in Real-Time

A Smart Data Framework to Provide Live Updates for Public Transport Accessibility

Paloma Cáceres, Carlos E. Cuesta, Almudena Sierra-Alonso, Belén Vela, José María Cavero, Miguel A. Garrido
Escuela Técnica Superior de Ingeniería Informática
Universidad Rey Juan Carlos, URJC
Móstoles (Madrid), Spain
e-mail: {paloma.caceres,carlos.cuesta,almudena.sierra,belen.vela,josemaria.cavero,miguel.garrido}@urjc.es

***Abstract*—Smart cities initiatives introduce the challenge of a more inclusive society that provides equal opportunities. Nevertheless, there are not many systems or apps designed to generate routes in the public transport for people with special needs. For instance, Google Maps is often used to calculate routes and find points of interest. However, it does not provide detailed information regarding specific facilities such as accessibility in transit. Even worse, it is not easy to find, to download or even to process specific and detailed accessibility information, in order to develop services for users. In this paper, we propose a smart data framework able to manage accessibility data in public transport and provide such user-level services. Moreover, this framework is also able to deal with real-time information by gathering notifications about incidents in the transport network. These data are obtained in a crowdsourcing process, and provides live updates for information, so that they can be used to recalculate accessible routes. In order to illustrate the process, we describe a case study with which we validate the framework, consisting of a smart app to generate accessible subway routes and to notify accessibility incidents.**

Keywords - Public transport; smart city; accessibility; incident management; crowdsourcing; linked open datasets.

I. INTRODUCTION

Smart cities appear as the main approach to tackle one of the greatest challenges in the first half of the century - urban mobility. This is a complex problem with many facets and nuances, but our strategy is always the same: to get better, more detailed data about many issues, and to use it in a intelligent way, to provide better focused answers to these problems.

The source of any problems in urban mobility is the flow of people and vehicles trying to use the same spaces at the same time. One improvement is to stress the importance of using the public transport - however, this choice would be much more efficient if people was able to avoid colliding with other people when traversing the cities. They could use their own smart devices to do that, by finding specific routes within the urban space.

A software system which provides maps and calculates routes in public transport networks is not a novelty anymore - however, most of the current routing systems still do not consider accessibility. Even the most popular among them, Google Maps, does not provide accessible routes. Our previous research [1]-[3] intends to use semantic information to fill this niche.

Building upon this infrastructure, our platform is able to evolve beyond its original constraints, and then to consider higher accessibility levels. Our initial architecture had to deal with mostly non-volatile data, i.e., data with a small rate change. When our system had to compute a route, it usually uses very stable data (structure of the network, predefined timetables), and even their alterations have usually a certain durability (planned works, changes in the infrastructure or the rolling stock). However, other sources of change are much less predictable: collapses, floods, accidents, even the traffic flow. These are usually easy to perceive once they happen - but many citizens are unaware of those *incidents* until they bump into them.

Thus, our system must include incident management to be complete. This would allow our smart data to include live information and to provide an even more intelligent response in real-time, which takes the current situation into account. This is particularly important when considering accessibility and the relevant groups of interest. This issue effectively offers an advantage over other approaches, such as those discussed in the related works section.

This approach uses a *crowdsourcing* strategy: users themselves are who provide live information about incidents as they find them. This information is sent through mobile devices, and stored in our routing platform, probably from many coincidental sources. These new data are collapsed and integrated with pre-existing accessibility information - so semantic annotations are also used for their processing. In summary, their combination can be conceived as *real-time accessibility data*.

This integration presents several challenges: their specific meanings and vocabulary; the incident lifecycle management; or the handling of simultaneous sources, and even their scale. Our evolved architecture must deal with all this, and this paper presents our solution.

The paper is structured as follows: Section 2 presents some related works. Section 3 describes our CoMobility and Access@City projects, which are the context of this work. Section 4 presents our smart data framework, describing our proposed client-server, scaling architecture, a semantic vocabulary to annotate events and the workflows between client and server sides. Section 5 describes a case study. Our conclusions and future works are presented in Section 6.

 ISBN: 978-1-61208-692-7

II. RELATED WORKS

There already are several software applications which inform about specific aspects of the public transport domain. In some cases, they include accessible wayfinding information or/and accessibility features, or elements for people with special needs or disabilities. Other software provides data about public transport using a crowdsourcing approach. In this section, we discuss some of the more representative of them.

Among those related to accessibility, Landmark Ontology for Hiking [4] is focused on elderly people and helps them to walk less by using wheelchairs. It formally represents landmarks for hiking. Wheelmate [5] and/or Wheel Map [6] provide some information about accessible places for people riding a wheelchair. Access Map [7] provides accessible routes to people with mobility needs. All of these just consider mobility-related disabilities. On the other hand, “Ciudades Patrimonio de la Humanidad” [8] has a web application which provides accessible routes to people with special mobility needs, including blind or hearing-impaired people. However, none of these applications are customizable.

Among those using a crowdsourcing approach, some applications use crowdsourcing to improve the experience of using the public transport and to provide real-time information about the public transport status. For example, Tiramisu Transit [9] provides data, such as how full the bus is, or whether there is some wheelchair space left. Moovit [11] is able to plan routes and indicate when to get off, or the status of the service. Swiftly [12] works with transport agencies, rather than the general public. It provides more accurate vehicle arrival data for these transport agencies, so that they can give better information to their users. The OneBusAway [10] project consists of a set of tools to improve the user experience in the public transport, by achieving on-time performance of buses and other transit systems, decreasing wait times, increasing feelings of safety or even increasing transit trips per week. OneBusAway provides several feedback mechanisms that allow users to make comments about these tools.

Other initiatives to use crowdsourcing are the BUSUP [13] and CIVITAS [14] projects. The first one allows users to book crowdsourced buses on demand and the second one intends to achieve a cleaner, better transport in Europe. A CIVITAS subproject is dedicated to mobility strategies for vulnerable groups.

OpenTripPlanner (OTP) is another project to provide services for passenger information and transport network analysis. It computes routes combining transit, pedestrian, bicycle and car segments traversing networks built from OpenStreetMap [15] and GTFS [16] data. OTP also takes (transport) accessibility into account.

As we see, the number of systems intending to improve the user experience in the public transport are increasing. However, most of them (with some exceptions, such as Tiramisu Transit or CIVITAS) do not take the information about accessibility elements into account; but these elements are necessary for users with special needs, to also improve their own experience in the public transport. Even the mentioned systems [9][14], though providing accessibility features, do not allow the users to inform about the actual state of these elements: whether a lift is operative, if there are works that prevent or hinder the access of (for instance) blind people, etc.

To the best of our knowledge, currently there are no software applications which analyze the status of the public transport network to estimate the availability of accessibility features, while also using crowdsourcing to update their data.

III. THE CONTEXT: COMOBILITY AND ACCESS@CITY PROJECTS

This work is developed in the context of two research projects. The first one, called CoMobility [18], defines a multimodal architecture based on linked open data for sustainable mobility. Its main goals are to improve citizens’ mobility and to optimize their trips by combining public transport and car sharing. The second one, called Access@City [19], is a coordinated project that defines a technological framework in which to process, manage and use open data concerning public transport with the goal of promoting its accessibility. One of its subprojects is Multiply@City [20]. Figure 1 provides a general depiction of this latter project. Multiply@City focus on processing and harmonizing public transport accessibility data in a semantic manner by means an ontology, considering that data are provided by different sources and have different formats (as represented on the top of Figure 1). The accessibility data are obtained from open data by means of Web scraping and here, they can also be updated via crowdsourcing techniques (from smartphones, as represented on the bottom of Figure 1). The left blue square represents the engine for extracting and processing the data from different sources, both from public transport companies and from crowdsourcing.

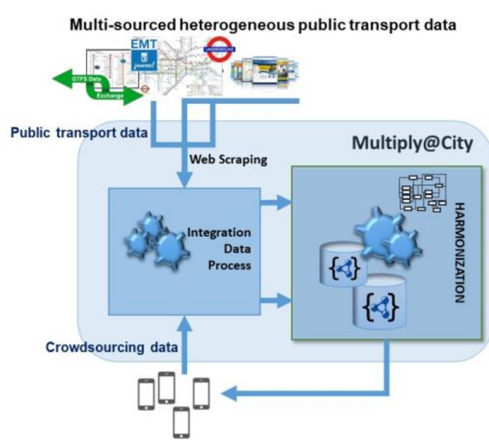


Figure 1 Multiply@City project architecture.

The right blue square represents the engine to semantically annotated previous data, following our MAnto vocabulary (a vocabulary to identify the different infrastructure elements of public transport like lines or stops, and their corresponding accessibility features like lifts, escalators, etc.). Next, semantic data are stored into a semantic repository: one collection of data is for the infrastructure elements of the public transport and the other is for the incidents about the accessibility features of the infrastructure.

The Regional Consortium for Public Transport in Madrid (CRTM) [21], Madrid public bus company (EMT Madrid) [22], and the Spanish Society for Blind People (ONCE) [23] have expressed their interest in the results of our CoMobility and Access@City projects.

IV. Proposed Smart Data Framework

The intent of our proposal is to improve accessibility information about public transport in order to support new social accessibility services, such as calculating public transport routes which are accessible for all. These routes are based on stable data about the infrastructure of the transport network (i.e., stations, lines and stops) and on continuously updated data about accessibility features (i.e., a lift or an escalator does not work). To consider these last features, we had to develop a solution which also includes information about the current state of the public transport network, using a crowdsourcing approach.

This proposal applies a client-server architectural style, with features akin to the microservices approach. On the client side, our software is able to update the accessibility information and to calculate a route based on the currently available accessibility features, taking into account the user's accessibility needs. On the server side, we have developed and deployed a datastore which holds the relevant information about the transport network and its accessibility features, listens and processes different notifications from our client software and returns any data which is requested by the same client.

A. The Proposed Architecture

As we mentioned above, our framework implements a client-server architecture, with some additional features.

In the client side, we have developed a smart device software (still a prototype), called MMA4A, which offers two different alternatives to users: (i) to request an accessible route considering their specific needs; (ii) to collect events or incidents about unavailable accessibility features. To compute the route (i), it is previously necessary to obtain the infrastructure data about the public transport network, and to request the current state of this infrastructure – provided by the server. To communicate the events collected by the users (ii), it is also necessary to send these incidents to the server, in order to update the current accessibility features in the network. Therefore, it is required to develop a specific server architecture that holds the infrastructure data and its accessibility status, updated with a crowdsourcing approach. This sequence of activities and their data exchanges are shown in Figure 2.

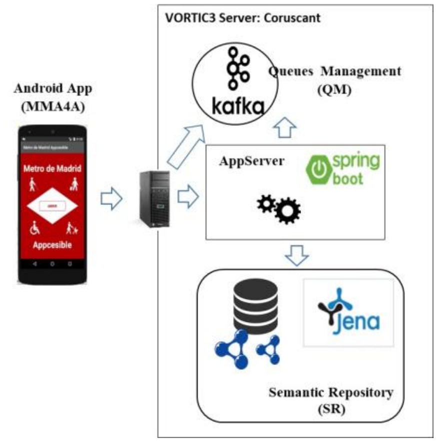


Figure 2 Proposed architecture (smart client-semantic server).

Concerning (i), the server must send the infrastructure data of the transport network, such as stations, lines and stops; and the last updated data about their accessibility features. Our implementation uses Spring Boot [24] as an applications server (AppServer) and a Jena Semantic Repository (SR) [25]. In previous works, [1][2], we developed a semantic data set about the infrastructure, which is now stored in the Jena SR.

Regarding (ii), the server should simultaneously listen to the (potentially many) notifications of incidents or events about the accessibility features that, at this moment, are not available for the public transport network. For this reason, we have implemented an Apache Kakfa server [26] as a Queuing Manager (QM). The QM gathers the different events notified from the crowd of sources, i.e., the users through their smart devices. This information, once processed, must be stored in the semantic repository, so the AppServer also manages and controls the communication between the QM and the SR.

To sum up, we have implemented the following elements in the server (Coruscant, a HP multicore high-performance server, with a direct high-speed Internet connection): a Spring Boot as the application server (AppServer), an Apache Kakfa server as a QM and a Jena SR. Then, the Jena repository maintains two separate data collections: one of them provides the (mostly static) data about station, lines and stops, called the *infrastructure collection* (i) and the other one provides the (dynamic) data of the network state about the accessibility features, called the *events collection* (ii).

B. Semantic Annotation of Data obtained from Crowdsourcing

When a user notifies an event or incident about an accessibility feature by means of the MMA4A app, the corresponding data are to be stored to the *events collection* of our SR. These accessibility features are part of the infrastructure of the public transport and must be fitting with the user's needs. For example, a person with a mobility disability or temporary issue needs a lift to access to the public transport network. For this reason, we need to know which accessibility features are associated to each station or stop place.

To identify the specific accessibility features of any public transport network, we have carefully studied the Identification of Fixed Objects in Public Transport (IFOPT [27]) reference datamodel. It is a standard, which defines a model for the main *fixed objects* related to the access to Public Transport (e.g., stop points, stop areas, stations, connection links, entrances, etc.). IFOPT extends a previous model, i.e., the European Reference Data Model for Public Transport Information (Transmodel [28]). Transmodel is the European reference data model in the field, which provides a model of both public transport concepts and data structures that may be useful when building information systems related to the different kinds of public transport. It does not, however, provide any information about accessibility. IFOPT includes specific structures with which to describe accessibility data concerning the equipment of vehicles, stops and access areas. In this work, we have studied the different users' needs and the corresponding accessibility features able to satisfy them. In fact, we developed a proposal to annotate data of the bus network in a semantic way based on IFOPT in previous work [3]. Now, we have based on it to identify the users' needs. TABLE I resumes the elements and their correspondence with these users' needs.

TABLE I RELATIONSHIP BETWEEN FEATURES AND USERS' NEEDS

Accessibility features of public transport (based on IFOPT)	Accessibility needs of users (based on IFOPT)			
	Auditory and visual	Mobility	Phobia to lifts	Phobia to escalators
Lift	✓	✓	×	✓
Escalator	✓	×	✓	×
Ramp	✓	✓	✓	✓
Stairs	✓	×	✓	✓

The previous table summarizes the events or incidents about accessibility features we consider in this work. Relative to this, we have also analyzed both Transmodel and IFOPT standards to identify the specific characteristics of such events, in order to include them. Based on both models, we have developed a specific *vocabulary* as a RDF Schema [29], which comprises the necessary information to annotate, in a semantic way, these events about accessibility, obtained via crowdsourcing. Figure 3 shows this terminology and their relationships.

Using this vocabulary, we can register whether a lift does not work (*hasLift FALSE*) in a specific *StopPlace* (i.e., station) of a specific *ofLine* (line of transport). This event has also an associated opening date (*openDate*). When the incident is solved, we can close it indicating the final date (*closeDate*) and then assigning a TRUE value to both *hasLift* (the original feature) and *closedEvent*.

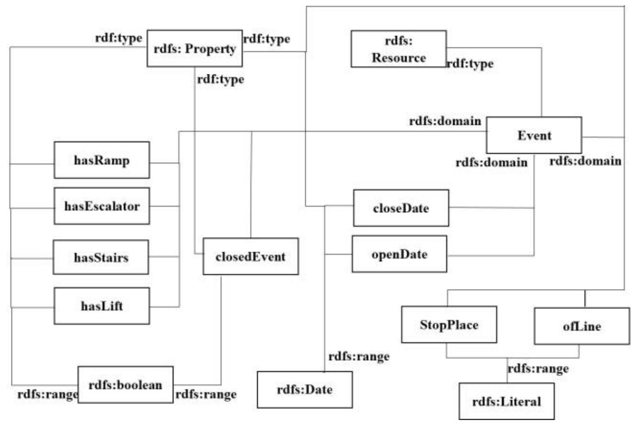


Figure 3 Events vocabulary in RDF Schema.

In summary, the event data about accessibility features, processed from many sources (*crowd*), is semantically annotated based on the vocabulary and it is then stored in the Jena semantic repository, in the events collection.

C. The workflows between client and server side

As mentioned before, MMA4A is a software for smart devices, able to compute a specific public transport route based on the users' needs and to notify the events or failures related to accessibility features of the network. We want to underline that the MMA4A app is still a prototype concerning the accessibility of the user interface (it does not include a fully accessible UI).

To calculate the route, the software needs to work with updated data (both about the network infrastructure and the state of the accessibility features). To provide an updated bootstrap, we have decided that when the app starts up, it requests to download this information to the server.

Also, from the client perspective, we have identified the different actors and use cases (see Figure 4).

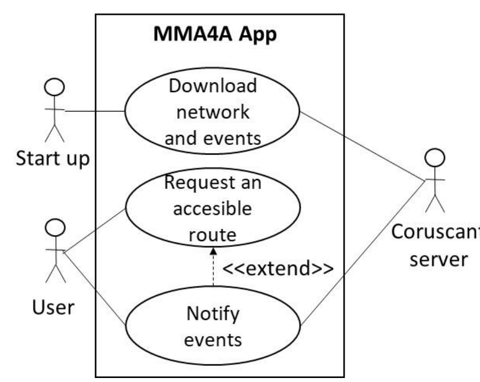


Figure 4 Use case diagram for MMA4A.

Now, we describe the workflows between client and server in more detail (see Figure 5).

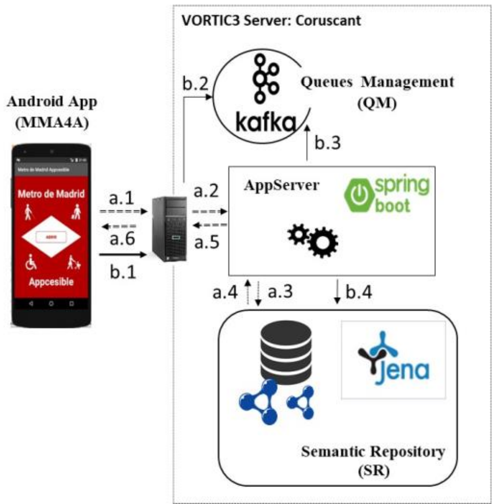


Figure 5 Workflows of the proposed architecture.

These are the workflows corresponding to the relationships between the Coruscant server actor and the associated use cases (*Download network and events*; and *Notify events*, respectively): (a) When the MMA4A starts up, it requests to download the data about the public transport network and the accessibility events stored in the Coruscant server. These two downloading requests are sent to the RESTful AppServer API; consequently, it generates the respective RDF/XML [30] files from the Jena Semantic Repository to return them to the app; (b) When a user notifies an event or failure about an accessibility feature, the app sends the event to the QM in the server side. Then, the AppServer consumes this information from the QM and next, it writes this incident in the SR as a RDF triple. Figure 5 shows (a) and (b) workflows between client and server side.

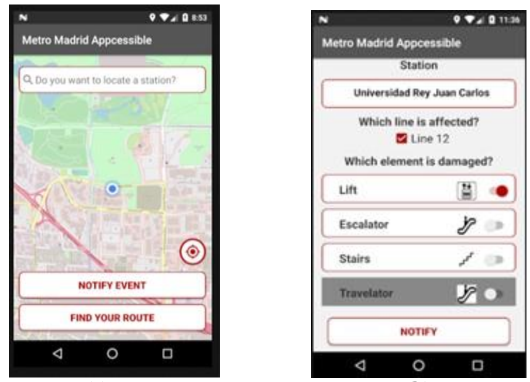
V. THE CASE STUDY

In the following section, we validate our smart data framework against real data from the subway in the city of Madrid, Spain (Metro Madrid [31]), used by means of the prototype MMA4A app. Specifically, we want to validate the incident management ("crowdsourcing") facility.

Before using this semantic dataset, it has been validated by means of SPARQL queries. These were defined to verify the correctness of data. It is relevant to highlight that this information was the outcome of an integration process from several sources of public transport data.

To validate our proposal, we have defined the following process. First, the user selects the "*Notify an event*" option; second, the user indicates what accessibility feature does not work in a specific station; third, we validate the RDF code generated for this event; fourth, a different user, who has a specific user need, request a route; this route is composed by stations where some accessibility features do not work, including one related to the previously notified event.

Next, we show this process for a specific case study in a detailed way. First, a user (UserA) selects to notify an event (*Notify events* use case). Figure 6a depicts the user interface where the user can select this. Second, the user notifies that the *lift* of the Universidad Rey Juan Carlos *station* in Metro Line12 does not work (Figure 6b).



(a) (b)

Figure 6 App user interface: (a) "*Notify an event*" and (b) "*Which accessibility elements do not work*".

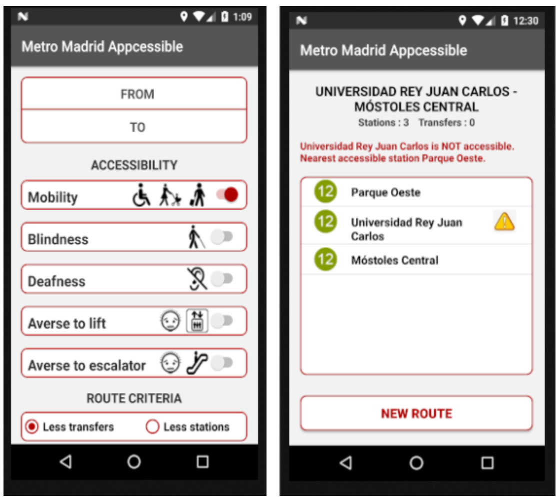
Third, when the UserA selects the NOTIFY bottom, the app generates an RDF event notification to send to the server. Figure 7 shows the RDF code describing the event generated before, about the lift accessibility feature in the Universidad Rey Juan Carlos station. This information is stored in the Jena SR by the AppServer component.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:mao="http://com.vortic3.MANTO#"
   xmlns:foaf="http://xmlns.com/foaf/0.1/" >
  <rdf:Description rdf:about="http://com.vortic3.MANTO#Universidad Rey Juan Carlos-Linea12MetroSur-lift">
   <rdf:type rdf:resource="http://com.vortic3.MANTO#Event"/>
   <mao:ofLine>https://www.metromadrid.es//es/viaja_en_metro/red_de_metro/lineas_y_horarios/linea12.html</mao:ofLine>
   <mao:hasLift>FALSE</mao:hasLift>
   <mao:openDate>2018-06-22T13:40:36.686</mao:openDate>
   <mao:closedEvent>FALSE</mao:closedEvent>
   <mao:StopPlace>https://www.metromadrid.es//es/viaja_en_metro/red_de_metro/estaciones/UniversidadReyJuanCarlos.html</mao:StopPlace>
  </rdf:Description>
</rdf:RDF>
```

Figure 7 RDF code about the event: Lift failure at the Universidad Rey Juan Carlos station in Line12

Fourth, another user (i.e., UserB) will request a route. Previously, when the MMA4A was started up in UserB's smart device (e.g., after a reboot), the app requests the infrastructure data and the stored events to the server. Regarding the calculation of the route, there are several possibilities: for instance, it is possible for users to obtain an accessible route in the Metro by means of a specific algorithm, considering the different users' needs. Perhaps no accessible route exists, but if one does, then this app will be able to calculate it. The user interface in Figure 8a shows the information that the app requires in order to establish the route. First, the origin and the destination of the trip; second, the kind of user's special needs (defined by groups: people with special mobility needs, blind people, hearing-impaired people); and third, a decision about the route (minimizing commutes or stations). In our case study, UserB indicates a mobility need. Then, the app computes the route, taking these choices into account.

It is important to stress that the algorithm provides a route, only after examining that the start, destination and any transfer stations have the required accessibility features. In our case study, we have requested a route starting at the Universidad Rey Juan Carlos station. Then, as previously indicated, the app notifies a lift failure in this station (see Figure 8b) and offers as an alternative the closest accessible stations (in our case study, the closest accessible stations are Parque Oeste and Mostoles Central as also shown in Figure 8b).



(a) (b)

Figure 8 App user interface: (a) Request a route and (b) Notify an event to the user.

Figure 9 provides a partial image of the Metro Madrid map, to show that the closest stations offered to the user are the best alternative.

Figure 9 Partial image of the network of Metro Madrid.

The app also asks to any other user who passes by this station whether the lift works now. Figure 10 shows this alert message.

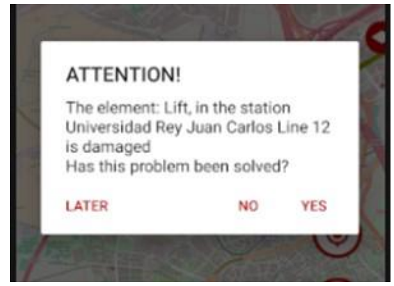


Figure 10 About Metro Madrid. Alert message of MMA4A.

We have carried out many other tests according to the different user needs identified in this solution (auditory and visual disabilities, mobility disability, phobia to lifts and phobia to escalators). We have also verified that the routes provided by MMA4A are correct for these kinds of users.

In summary, our experience with this app proves that data in the semantic repository can be used to generate the routes for users with special needs, while the incidents are registered and considered while elaborating those routes. The experiment described here shows that the behavior of the system is correct at this moment. We cannot guarantee that event notifications were always true.

You can download both the data infrastructure and the data events (RDF files) whose structure we describe in this paper in the following links: http://coruscant.my.to:8080/download/metro.xml, http://coruscant.my.to:8080/download/events.xml.

VI. CONCLUSION AND FUTURE WORK

One of the major challenges of smart cities initiatives is to achieve an inclusive society for all citizens, including those with special needs, such as mobility issues. If this challenge is to be met, more thorough information about the means of transport and their accessibility features is required, to be able to arrange and provide accessible routes for everybody, including those special users.

There already are several web applications and tools which provide information and services for transport users. We have studied some of them and to the best of our knowledge, currently no software application takes fully into account the current status of the public transport network with respect to the accessibility features; in particular, updating data with a crowdsourcing strategy, to compute accessible routes for special needs users.

In order to solve this issue, we have proposed a *smart data architecture* to manage both data sources about the infrastructure of the public transport.

Our proposal considers accessibility features when it computes a route. But it goes beyond that, by taking live information into account, such as incidents that may occur at any time. Thus, our dataset provides a smarter response in real-time. These live data, coming from crowdsourcing, are collapsed and integrated with existing accessibility information, providing a *real-time accessibility dataset*.

We provide a Kafka Queuing Manager to gather users' notifications about the state of accessibility features in the public transport, and a Jena Semantic Repository to store both these data and the infrastructure data about the transport network. We have also developed a smart app (still a prototype) for public transport users, which is able to compute accessible routes taking the user's needs into account, and also to notify incidents or events in the accessibility features of the network. In order to semantically annotate the data before storing them, we have also defined a specific vocabulary as a domain-specific RDF Schema.

Regarding future work in this area, we intend to include more public transport information from other sources and means of transport, and to integrate them with the current architecture. For this purpose, it will be necessary to semantically harmonize them using our MAnto ontology, which has already been used to semantically annotate the data stored in the infrastructure collection of our Jena repository. This will make possible to provide fully open datasets for different public transport networks. Moreover, these datasets must be published in an open platform, providing free access to accessibility and special needs data. We have also worked in gathering information about accessible pedestrian routes in the city, obtained via crowdsourcing techniques, capturing the geographical information and accessibility features of these routes. This information will be also incorporated as smart data into the Multiply@City platform.

ACKNOWLEDGMENT

This work is supported by the Multiply@City and the Access@City projects (TIN2016-78103-C2-1-R), funded by the Spanish Ministry of Science, Innovation and University. We also would like to thank Iván Pérez Muñoz for his assistance in the development process.

REFERENCES

[1] P. Cáceres, A. Sierra-Alonso, C. E. Cuesta, J. M. Cavero, and B. Vela, "Enhancing Accessibility Information in Google Maps: Adding new pieces of information to GTFS to improve accessibility" In Proceedings on the Second International Conference on Universal Accessibility in the Internet of Things and Smart Environments (SMART ACCESSIBILITY 2017), pp. 6-9, Nice (France), 2017.

[2] P. Cáceres, A. Sierra-Alonso, B. Vela, J. M. Cavero, and C. E. Cuesta, "Towards Smart Public Transport Data: A Specific Process to Generate Datasets Containing Public Transport Accessibility Information" In Proceedings on the Third International Conference on Universal Accessibility in the Internet of Things and Smart Environments (SMART ACCESSIBILITY 2018), pp. 66-71, Rome (Italy), 2018.

[3] P. Cáceres, A. Sierra-Alonso, C. E. Cuesta, B. Vela, and J. M. Cavero, "Modelling and Linking Accessibility Data in the Public Bus Network", Journal of Universal Computer Science, vol. 21, no. 6, pp. 777-795, 2015.

[4] T. Sarjakoski et al., "Landmarks and a Hiking Ontology to Support Wayfinding in a National Park During Different Seasons". In M. Raubal, D. M. Mark, and A. U. Frank (Eds.), Cognitive and Linguistic Aspects of Geographic Space, Lecture Notes in Geoinformation and Cartography. Springer, 2013. DOI:10.1007/978-3-642-34359-9.

[5] Wheelmate,http://myhealthapps.net/app/details/13/wheelmate, [retrieved: Dec. 2018].

[6] Wheelmap, https://wheelmap.org/, [retrieved: Dec. 2018].

[7] Accessmap, https://www.accessmap.io/, [retrieved: Dec. 2018].

[8] Ciudades Patrimonio de la Humanidad, http://www.ciudadespatrimonio.org/accesibilidad/?idioma=en, [retrieved: Dec. 2018].

[9] J. Zimmerman et al., "Field Trial of Tiramisu: Crowd-Sourcing Bus Arrival Times to Spur Codesign". In Proc. CHI 2011, New York, pp. 1677–1686, 2011.

[10] B. Ferris, K. Watkins, and A. Borning. "OneBusAway: Results fromProviding Real-Time Arrival Information for Public Transit". In Proc. CHI 2010, Atlanta, Georgia, April 10–15, 2010.

[11] Moovit, https://moovit.com/, [retrieved: Dec. 2018].

[12] Swiftly, https://www.goswift.ly/, [retrieved: Dec. 2018].

[13] BUSUP, https://www.busup.com/, [retrieved: Dec. 2018].

[14] CIVITAS, http://civitas.eu/, [retrieved: Dec. 2018].

[15] OpenTRipPlanner, http://www.opentripplanner.org/, [retrieved: Dec. 2018].

[16] OpenStreetMap, https://www.openstreetmap.org, [retrieved: Dec. 2018].

[17] GTFS, Google Transit Feed Specification, https://developers.google.com/transit/gtfs/, [retrieved: Dec. 2018].

[18] C. E. Cuesta, P. Cáceres, B. Vela, and J. M. Cavero, "CoMobility: A Mobile Platform for Transport Sharing", ERCIM News, no. 93, pp. 22-23, 2013.

[19] Access@City project (TIN2016-78103-C2-1-R), funded by the Spanish Ministry of Science, Innov. & University.

[20] Multiply@City project (TIN2016-78103-C2-1-R), funded by the Spanish Ministry of Science, Innov. & University.

[21] Regional Consortium for Public Transports of Madrid, http://datos.crtm.es/, [retrieved: Dec. 2018].

[22] EMT Madrid home page, http://www.emtmadrid.es/Index.aspx?lang=en-GB , [retrieved: Dec. 2018].

[23] Spanish Society of Blind People (ONCE, Organización Nacional de Ciegos de España) http://www.once.es/new/otras-webs/english, [retrieved: Dec. 2018].

[24] Spring Boot, http://spring.io/projects/spring-boot, [retrieved: Dec. 2018].

[25] Apache Jena, https://jena.apache.org/index.html, [retrieved: Dec. 2018].

[26] Apache Kafka. https://kafka.apache.org/intro, [retrieved: Dec. 2018].

[27] CEN/TC 278. Intelligent transport systems - Public transport - Identification of Fixed Objects In Public Transport (IFOPT), EN 28701:2012.

[28] Transmodel, standardization, E. C. (2016). Transmodel, Road Transport and Traffic Telematics. Public Transport. Ref. Data Model, EN 12896, http://www.transmodel.org/en/cadre1.html, [retrieved: Dec. 2018].

[29] RDF Schema, https://www.w3.org/TR/rdf-schema/, [retrieved: Dec. 2018].

[30] RDF/XML, https://www.w3.org/TR/rdf-syntax-grammar/, [retrieved: Dec. 2018].

[31] Metro Madrid home page, http://www.metromadrid.es/en/, [retrieved: Dec. 2018].

