Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 203 (2022) 334–341

Procedia Computer Science

www.elsevier.com/locate/procedia

The 12th International Conference on Sustainable Energy Information Technology (SEIT)
August 9-11, 2022, Niagara Falls, Canada

greenITS: a proposal to compute low-pollution routes

Hermenegilda Macià[a,*], Gregorio Díaz[a], Valentín Valero[a], Edelmira Valero[b], Enrique Brazález[a], Juan Boubeta-Puig[c]

[a] *Albacete Research Institute of Informatics of Albacete. Universidad de Castilla-La Mancha. Investigación 2, 02071 Albacete, Spain*
[b] *Higher Technical School of Industrial Engineers of Albacete. Universidad de Castilla-La Mancha. Building Infante Don Juan Manuel, Av. de España, 02071 Albacete, Spain*
[c] *Department of Computer Science and Engineering. University of Cadiz. Avda. de la Universidad de Cádiz 10, 11519, Puerto Real, Cádiz, Spain*

Abstract

A way to reduce carbon emissions in cities is through movement by bicycle or on foot. However, it sometimes means to pass through high-pollution zones and consequently breath low quality air. We then propose a green Intelligent Transportation System (ITS) for zero-emission mobility users, providing users with low-pollution routes to avoid the high-pollution zones. This proposal uses ITS to promote the use of alternative transportation to classical motor vehicles to reduce carbon emissions. This is based on Complex Event Processing (CEP) technology to gather and process real-time data, a Decision Support System designed as a Fuzzy Inference System (FIS) to make decisions about recommended transit zones, taking also into account the user experience level and specific weather data, and Colored Petri Nets (CPN) as a tool to compute the routes. This is therefore an all-in-one solution to provide green routes, with the benefits of each one of the technologies used.

Keywords: Intelligent Transportation System; Complex Event Processing; Colored Petri Nets; Fuzzy Inference System; Pollution;

1. Introduction

Encouraging citizens to move with zero emissions will contribute to the climate change mitigation. Particularly, cycling in cities is one of the most popular means of transportation and has the following well-known benefits: it reduces road traffic, carbon emissions and energy consumption, and often contributes to healthier lifestyles. It is well established that cyclists' health is affected by pollution [1, 2], therefore systems providing routes for cyclists should avoid high-polluted areas. Moreover, these systems should also consider the cyclist experience level and the wind. In this context, we can find in the literature some systems that provide routes for cyclists. In some cases, the routes are chosen to minimize risk and discomfort, as described in the work of Castells et al. [3], where authors used estimation models to evaluate both. Teslyuk et al. [4] implemented *LvisBicycleMap*, which provided an interactive city map

* Corresponding author. Tel.: +34-967 59 92 00 ; fax: +34-967 59 92 24.
E-mail address: hermenegilda.macia@uclm.es

10.1016/j.procs.2022.07.042

application to show contextual information about biking facilities and problematic areas reported by users. A step further was provided by Saelens at al. [5], who proposed a tool, called *HeyCitI*, to find the most healthy path in a city while avoiding areas with high pollution levels by using a heuristic search algorithm for finding the k shortest paths. In previous works, we analyzed the problem of controlling air pollution and road traffic in cities. In [6], we used Colored Petri Nets (CPNs) [7] to compute fast routes with the goal of reducing pollution levels in affected areas, and we combined Fuzzy Logic and Complex Event Processing (CEP) technologies for the same purpose in [8].

Although the scope of this proposal is close to these papers, they do not consider either the cyclist experience level or the wind condition. The methodology used in this work is also different to the methodologies used in [3, 4, 5]. Specifically, we propose an all-in-one solution: a green Intelligent Transportation System (greenITS) for mobility free of pollution emissions, providing low-pollution routes.

The most important contribution of our proposal is the combination of CEP [9], Fuzzy Inference System (FIS) [10], and CPNs. This is a novel idea with respect to the existing works on ITSs to compute low-pollution routes for cyclists. The novelty is also the inclusion of cyclist experience level and wind speed/direction, which are involved to compute the routes. Specifically, the proposal works as follows:

- CEP is a technology that allows us to gather and process the information from one or several data streams, so as to detect situations of interest in real time.
- FIS provides us with the capability to encode a knowledge using natural language terms, in most cases dealing with uncertainty, and applying a set of rules to provide some conclusions or actions from specific situations.
- CPNs are a powerful modeling tool that have a graphical representation and thus allow us to represent in an easy way the systems and analyze or simulate their behavior under different scenarios.

Figure 1 provides an overview of this proposal, which consists of the following steps:

(1a) Air Quality (AQ) and meteorological data are gathered from several sources, and these data are processed and correlated by the CEP engine, which is responsible for creating the complex events to feed the FIS.
(1b) The city map is modelled as a CPN.
(2) The FIS system issues a set of recommendations about the areas that should not be traversed.
(3) The domain expert interprets the FIS recommendations in three ways: open, pass-with-precaution or close.
(4) With the previous information from step (3), the CPN model is configured with the corresponding mobility restrictions.
(5) A cyclist asks for a fast route, taking into account her/his user level on cycling.
(6) The CPN model will provide this user with a low-pollution fast route.

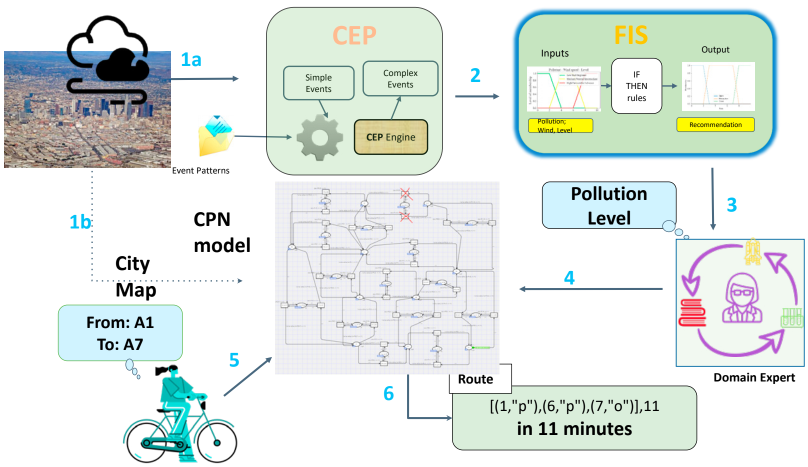


Fig. 1. Overview of our green ITS proposal

2. Scenario

In a previous work [6], we modeled Madrid city map (Spain) by using a CPN with the purpose of reducing air pollution, taking into account the emissions from road traffic. A similar approach is considered here for Albacete city (located in Spain, at latitude 38.99424 and longitude -1.85643), but considering its bike lane network[1] and the recently installed network of air quality sensors[2]. The expertise level of cyclist-users and wind rate and direction are also important parameters to be considered in the specific case of bicycle routes since the times to cover the different legs can vary significantly due to these parameters. Another important parameter that could be considered in the case of bicycles is that of altitudes, but Albacete is placed on a flat area, so we do not need to consider this parameter in this study.

Regarding the pollutant of interest for cyclists in Albacete, we have chosen PM_{10} (inhalable particulate matter with diameters equal or smaller than 10 micrometers). This pollutant is actually one of the most dangerous in Albacete, which is produced from diverse sources (construction, agriculture, fires, etc). Luengo et al. [11] studied commuters' exposure to ultrafine particles (UFP) for cyclists, and consequently the need to develop low emission zones for bikes.

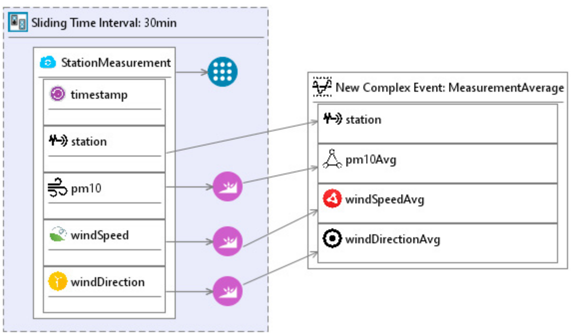
Although the present work is based on Albacete city, it can be easily adaptable to other contexts by including other parameters that might be relevant in those scenarios.

3. greenITS: the event patterns

CEP is a powerful technology that allows analyzing and correlating large volumes of data for the purpose of detecting situations of interest (event patterns) in real time. This data analysis and prompt detection of event patterns is carried out by a specific software, called CEP engine. In this paper, specifically, we use the Esper CEP engine, which provides Esper EPL [12] as the implementation language of the patterns to be detected in real time.

However, the implementation of these patterns is beyond the reach of any domain expert. To alleviate this limitation, MEdit4CEP [13] is a graphical modeling tool that allows domain experts to intuitively model event patterns, which will then be transformed automatically to EPL code executable in a CEP engine. With this tool a pattern *MeasurementAverage* has been defined (Figure 2(a)), for which we obtain the Esper EPL code shown in Figure 2(b). This pattern receives events of type *StationMeasurement* as input. In these events (see Figure 2), we have a timestamp indicating the time at which the data were generated in the sensor, the station from which the data came from, the measurement of the PM_{10} pollutant, wind speed and wind direction. For each event *StationMeasurement*, pattern *MeasurementAverage* computes the PM_{10} average values for PM_{10} level, wind speed and wind direction in the last 30 minutes, producing a *MeasurementAverage* complex event, which contains these average values and the station identifier.

[1] http://www.albacete.es/es/ayuntamiento
[2] https://www.troposfera.es/albacete/red-albacete/panel-mapa-de-datos.html



(a) Model

```
@Name("MeasurementAverage")
insert into MeasurementAverage
select a1.station as station,
 avg(a1.pm10) as pm10Avg,
 avg(a1.windSpeed) as windSpeedAvg,
 avg(a1.windDirection) as windDirectionAvg
from pattern [(every a1 = StationMeasurement)].win:
                                    time(30 minutes)
```

(b) EPL code

Fig. 2. *MeasurementAverage* pattern

4. greenITS: the FIS

Fuzzy systems use concepts that allow some forms of uncertainty to be modeled. A Mamdani's FIS [10] is a Fuzzy Inference System based on rules. A set of linguistic variables are first identified for both the input and output (fuzzy sets [14]). Afterwards, a set of rules is defined. In our case, these rules are used to produce recommendations for not traversing certain areas. The Skfuzzy Python toolbox [15] was used for the implementation, using the `trampmf` membership function as fuzzy sets. Moreover, taking the default parameters of the Skfuzzy toolbox, namely: minimum as t-norm (AND operator); t maximum as t-conorm (OR operator); minimum (Implication operator); maximum of consequent fuzzy sets as aggregation method for combining rule consequents; and the defuzzification method used is the centroid (center of gravity).

For simplicity, this FIS only has three input variables: *pollutant*, *wind* and user-*level*, and a single output variable (*pass*). The values of these variables are normalized in a range from 0 to 10, where high values correspond to high values in the measurement.

The input variables, pollutant and wind, are obtained from the CEP engine, and they are normalized in the ranges indicated. In the case of wind level, considering the direction and the speed of the wind, a low value corresponds to a bad condition for the cyclists and a high level means favourable situation. Then, they are classified in the linguistic labels: Low, Medium and High for pollutant; and Bad, Neutral and Favourable for the wind.

The user-level is assigned taking into account the age, weight, fitness, experience and frequency, and it is classified in the linguistic labels as Beginner, Intermediate and Advance. The output variable *pass* is also defined in the range 0-10, but it is classified with the following linguistic labels: *open* (pass without restrictions), *precaution* (can pass, but with caution, either due to high pollution or wind) and *close* (not pass, due to high pollution or wind). Figure 3 shows the fuzzy sets of the input/output variables, where the three input variables have the same fuzzy sets. Table 1 contains the IF-THEN rules defined.

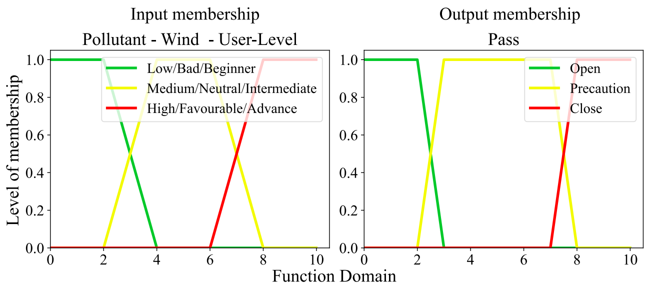


Fig. 3. Input/Output membership functions

Table 1. IF-THEN Rules for FIS

1. IF *pollutant* is `Low` AND *wind* is NOT `Bad` THEN *pass* is `Open`
2. IF *pollutant* is `Low` AND *wind* is `Bad` AND *level* is `Beginner` THEN *pass* is `Close`
3. IF *pollutant* is `Low` AND *wind* is `Bad` AND *level* is NOT `Beginner` THEN *pass* is `Precaution`
4. IF *pollutant* is `Medium` AND *wind* is `Bad` AND *level* is `Advance` THEN *pass* is `Precaution`
5. IF *pollutant* is `Medium` AND *wind* is `Bad` AND *level* is NOT `Advance` THEN *pass* is `Close`
6. IF *pollutant* is `Medium` AND *wind* is NOT `Bad` THEN *pass* is `Precaution`
7. IF *pollutant* is `High` THEN *pass* is `Close`

As an illustration, Figure 4 shows the surfaces obtained from the application of the rules considering the user-levels 2 (Beginner), 5 (Intermediate) and 9 (Advance). For instance, taking as input 4 for pollutant, 2 for wind and 2 for user-level, the output is *close* (8.76). In contrast, for a user-level of 9, while the same input values for pollutant and wind, the output is *pass with precaution* (5). An input of 8 for pollutant, regardless of wind and level, produces an output

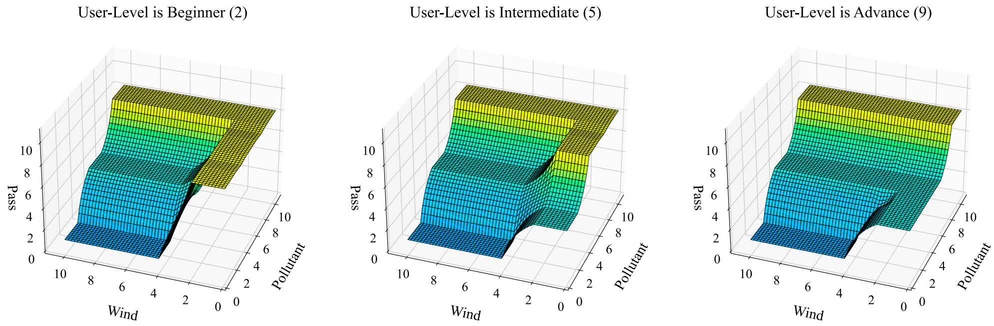


Fig. 4. Fuzzy space.

of *close* (8.76). Besides, the interpretation of the output of the FIS will consist of two parameters: the first one is the crisp value, and the second value is a string with the associated linguistic label of the variable pass (according to the domain expert).

5. greenITS: CPN for the routes

A Petri Net (PN) [16] is a mathematical formalism for system modeling with a graphical representation of the model as a bipartite directed graph with two types of node: places (drawn as circles) and transitions (drawn as rectangles). Places usually represent system states, in our case, user-routes or information data. Transitions are the actions or events that produce changes in the system state. In our case, they represent the crossing from one zone in the city to another one. Places in PNs are annotated with tokens (marking), which indicate the state of the component or element modeled by the place. Places and transitions can be connected by arcs, either place-transition (PT) or transition-place (TP) arcs. A firing rule determines the conditions under which transitions are fired (executed) in order to change the current marking. The firing of a transition removes a number of tokens equal to the weight of the corresponding PT-arc from each precondition place and writes on its postcondition places as many tokens as indicated by the corresponding TP-arcs.

A well-known extension of Petri Nets is CPN [7], which extend the basic model with data information on the tokens. We use CPN Tools [17] for editing, simulating, and analyzing Colored Petri nets, so we use the notation of this specific tool [18]. Places have an associated color set (colset), which specifies the set of token colors (a data type) allowed at this place. In our case, we only have two color sets: `ZoB` and `I4` (see Figure 5, place color sets are indicated next to the places, in their right bottom part). In addition, transitions can have guards that restrict their firing. These are Boolean expressions constructed by using the variables, constants, operators and functions of the model (guards are indicated above the transitions, see Figure 5). Arc inscriptions in CPNs are extended to color set expressions, which are constructed using variables, constants, operators and functions. The arc expressions must evaluate to a color or multiset of colors in the color set of the attached place (see the labels in the arcs in Figure 5). The specific declarations in CPN Tools for the color sets used in this study are the following:

```
colset INTR=product INT*STRING;
colset INTlistR=list INTR;
colset I=int with 0..10;
colset ZoB=product I*INT*INTlistR*INT;
colset I4=product I*I*I*INT;
```

ZoB values consist of 4 fields. The first is an integer (from 0 to 10) indicating the level of experience of the cyclist, the second is the zone destination (integer), the third is a list representing a route, where each leg is represented by the zone number and a string, the latter indicating if the zone is open (”o”) or it must be crossed with precaution (”p”). Finally, the fourth field indicates the total time required (in minutes) for this route. Color set *I4* also contains 4 integer fields, where the three first are in the scale 0..10. These contain the following information regarding the movements between zones. For a movement between zones A1 to A6 the first field is the PM_{10} level in this area, the second field is the wind level from A1 to A6, the third field is the wind level from A6 to A1, and the fourth is the estimated time for this movement for a cyclist with expertise level equal to 5 and assuming a wind level of 5 (neutral). Furthermore, the specific guards used in this case are the following:

```
fun noReturnB(i:INT,yy:ZoB)= not( mem (#3yy) (i,"o")) andalso not( mem (#3yy) (i,"p"))
fun isDestB(xx:ZoB)=  (hd(rev (#3 xx)) = (#2 xx,"o")) orelse (hd(rev (#3 xx)) = (#2 xx,"p"))
fun isOpenB(i:INT,yy:ZoB)= noReturnB(i,yy) andalso not(isDestB(yy))
fun passBi(i:INT,yy:ZoB,inf:I4)= isOpenB(i,yy) andalso #1 fis(#1 inf,#2 inf,#1 yy)<7
fun passBd(i:INT,yy:ZoB,inf:I4)= isOpenB(i,yy) andalso #1 fis(#1 inf,#3 inf,#1 yy)<7
```

where guard *noReturnB* is used to avoid coming back in the routes; function *isDestB* is used to check whether we have reached our destination or not; function *isOpenB* combines both functions *noReturnB* and *isDestB* to check is a leg is open. Finally, both functions *passBi* and *passBd* use the FIS information to determine if we can traverse a zone (the crisp value returned by the FIS must be less than 7). The arc inscriptions are the following:

```
fun tBi(xx:ZoB,inf:I4)= floor((real (#4 inf)*(1.0 -(real (#2 inf)-5.0) / 10.0)) / (0.5+ real(#1 xx ) /10.0))
fun tBd(xx:ZoB,inf:I4)= floor((real (#4 inf)*(1.0 -(real (#3 inf) -5.0)/ 10.0)) / (0.5+ real(#1 xx ) /10.0))
fun traverseSectionBi(i:INT,xx:ZoB,inf:I4)= (#1 xx, #2 xx, (#3 xx)^^[(i,fis(#1 inf,#2 inf,#1 xx) )], #4 xx
                                                                                              + tBi(xx,inf))
fun traverseSectionBd(i:INT,xx:ZoB,inf:I4)= (#1 xx, #2 xx, (#3 xx)^^[(i,fis(#1 inf,#3 inf,#1 xx) )], #4 xx
                                                                                              + tBd(xx,inf))
fun mini(y:ZoB,z:ZoB)=if (#4y < #4z) then 1‘y else 1‘z
```

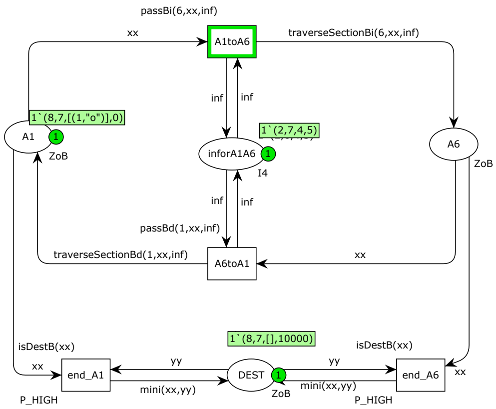


Fig. 5. CPN part modeling the movements from zone 1 to 6 and viceversa.

Functions *tBi* and *tBd* are used to compute the times to cross from one zone to another, with the following expression, assuming t is the estimated time to cross between zones when wind is $w = 5$ and cyclist's level is $l = 5$:

$$tb_{t,w,l} = \text{floor}\left(t \cdot \frac{1 - \frac{w-5}{10}}{0.5 + \frac{l}{10}}\right)$$

where, for instance, $tb_{5,5,5} = 5$, $tb_{5,0,5} = 7$, $tb_{5,10,5} = 2$, $tb_{5,5,0} = 10$, and $tb_{5,5,10} = 3$.

Traverse function *traverseSectionBd* is used to update the information stored at *ZoB* places with the new traversed zone. For instance, in Figure 5, token `(8,7,[(1,"o")],0)` in place A1 specifies a cyclist that starts in zone 1 with final destination zone 7 and her experience is 8 and token `(2,7,4,5)` specifies the conditions to travel between zones 1 and 6, where 2 is the PM_{10} level, 7 is the wind level when traversing from zone 1 to zone 6, 4 is the wind level in the opposite direction, i.e., from zone 6 to 1, and the last value 5 is the time to travel between these two zones under regular conditions (wind level 5 and expertise level 5). Transition *A1toA6* is enabled because guard *passBi* is satisfied. After firing this transition, this token in A1 will be removed and a new token in A6 will be produced with values `(8,7,[(1,"o"),(6,"o")],3)` by executing the function *traverseSectionBd*, which specifies that the cyclist is now in zone 6 and has taken 3 minutes to travel from zone 1 to zone 6. Place *DEST* is used to determine when a cyclist has reached the final destination i. In this case, the transition *end_Ai* would be fired due to its high priority and guard *isDestB* is satisfied. The token `((8,7,[],10000))` will be replaced with a new token specifying the followed route and the total travel time using function *mini*.

5.1. Synthetic results

We provide readers with a route example to analyze the simulation results. We present the results for a cyclist to travel from zone 1 to zone 7, according to the city map of Figure 6 corresponding to Albacete city. In this scenario,

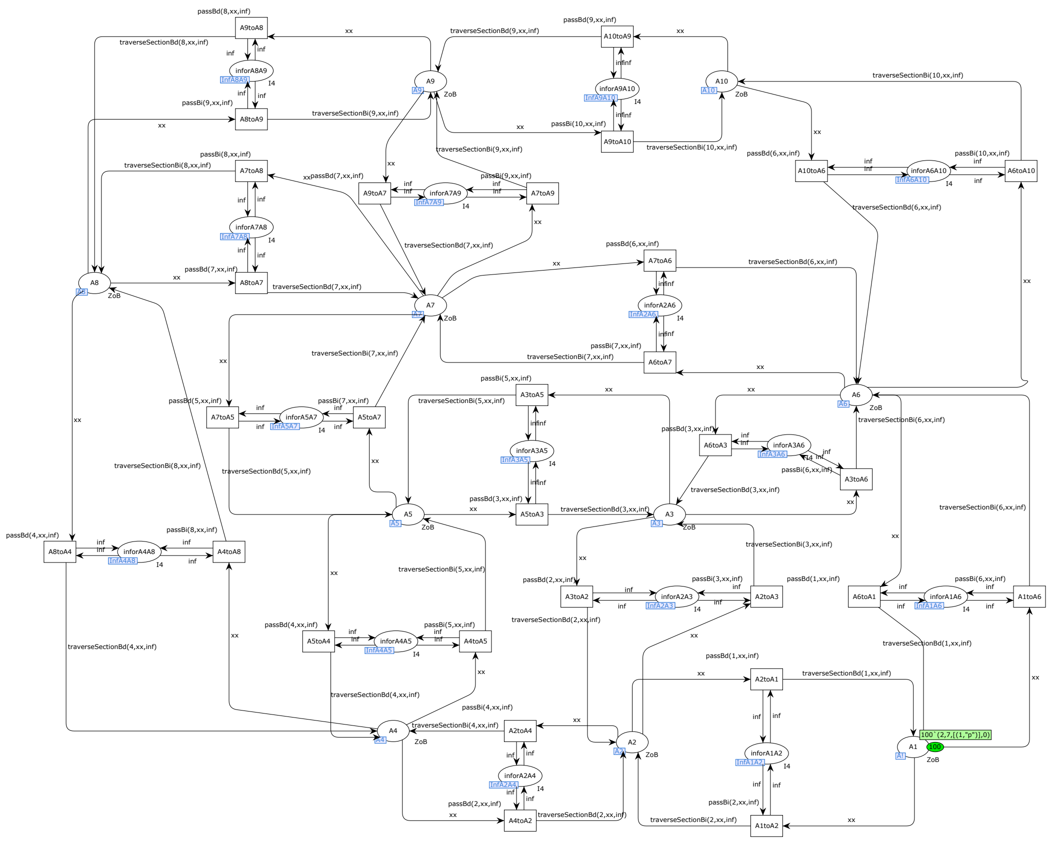


Fig. 6. city map

we consider a standard travel time between zones of 5 min and different values for the cyclist expertise, pollution and wind levels to analyze how they affect the routes and times obtained. Considering the following initial token at place *A1*: `(e,7,[(1,r)],0)`, we obtain the following results depending on the values of *e*, *r* and restriction levels:

- Case 1: r = "*o*", wind level 5 and pollution 2 for all areas leads to the route `[(1,"o"),(6,"o"),(7,"o")]` and the following times: 14 min for $e = 2$, 10 min for $e = 5$, and 6 min for $e = 9$.
- Case 2: With r = "*p*", bad wind conditions (level 1) and pollution 2 there is no route available for $e = 2$ and we obtain the route `[(1,"p"),(6,"p"),(7,"p")]` with times: 14 min for $e = 5$, and 9 min for $e = 9$.
- Case 3: r = "*o*", wind level 9 (good conditions) and pollution 2 for all areas leads to the route `[(1,"o"),(6,"o"),(7,"o")]` and the following times: 8 min for $e = 2$, 6 min for $e = 5$, and 4 min for $e = 9$.
- Case 4: for r = "*o*", $e = 5$ and wind level 5 (neutral) for all areas, pollution level 2 except for the crossing from zone 6 to 7 where the level is 9, we obtain the route `[(1,"o"),(6,"o"),(10,"o"),(9,"o"),(7,"o")]` and travel time 20 min.

6. Conclusions and future work

The main aim of this work is the computation of low-pollution routes for cyclists. To accomplish this objective, we have proposed an all-in-one solution combining three different technologies CEP, FIS and CPNs. CEP is used to gather and correlate air quality and meteorological information from different sources by matching event patterns. The FIS technology helps users to make decisions about traversing zones depending on his/her expertise and the values provided by the event patterns. Finally, the CPNs provide users with the recommended routes based on the FIS outputs.

The benefits provided by this solution are the following:

1. It provides a context-aware system considering certain aspects such as the atmospheric, pollution condition and the user expertise as cyclist.
2. It can be adapted considering other specific city maps, new restrictions or information on the city, weather conditions, users or vehicles.
3. It provides users with customized routes according to their interests.

As future work, we plan to apply this solution to other major cities in order to analyze a high variety of pollutants. In addition, other information on city infrastructures will be analyzed. For instance, this proposal could be combined with public transportation infrastructures or with the use of other vehicles, such as electric bikes or scooters, so as to reduce travel times. It is also desirable to enhance this proposal by developing mobile apps and combining it with existing navigation systems.

Acknowledgements

This research was supported by the Spanish Ministry of Science, Innovation and Universities and the European Union FEDER funds under Grants RTI2018-093608-B-C32 and RTI2018-093608-B-C33, the JCCM project cofinanced with European Union FEDER funds, ref. SBPLY/17/180501/000276/1 and SBPLY/17/180501/000276/2, the UCLM group research under Grant ref. 2021-GRIN-30993 and the Research Plan from the University of Cadiz and Grupo Energético de Puerto Real S.A. under Grant IRTP03_UCA.

References

[1] Amy Thai, Ian McKendry, and Michael Brauer. Particulate matter exposure along designated bicycle routes in Vancouver, British Columbia. *Science of The Total Environment*, 405(1):26–35, 2008.
[2] Piers MacNaughton, Steven Melly, Jose Vallarino, Gary Adamkiewicz, and John D. Spengler. Impact of bicycle route type on exposure to traffic-related air pollution. *Science of The Total Environment*, 490:37–43, 2014.
[3] David Castells-Graells, Christopher Salahub, and Evangelos Pournaras. On cycling risk and discomfort: urban safety mapping and bike route recommendations. *Computing*, pages 1259–1274, 2020.
[4] Vasyl Teslyuk, Vitalij Shevchyk, Michal Greguš ml, Taras Teslyuk, and Oleksandra Mishchuk. The recommendation system for cyclists LvivBicycleMap. *Procedia Computer Science*, 155:105–112, 2019. The 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019),The 14th International Conference on Future Networks and Communications (FNC-2019),The 9th International Conference on Sustainable Energy Information Technology.
[5] Saelens M., Kinoo Y., and Weyns D. HeyCitI: Healthy Cycling in a City using Self-Adaptive Internet-of-Things. *Proceedings - 2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion, ACSOS-C 2020*, page 226 – 227, 2020.
[6] Gregorio Díaz, Hermenegilda Macià, Valentín Valero, Juan Boubeta-Puig, and Fernando Cuartero. An Intelligent Transportation System to Control Air Pollution and Road Traffic in Cities Integrating CEP and Colored Petri Nets. *Neural Computing and Applications*, 32(2):405–426, 2020.
[7] Kurt Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, Vol. 2*. Springer-Verlag, London, UK, 1995.
[8] Hermenegilda Macià, Gregorio Díaz, Juan Boubeta-Puig, Edelmira Valero, and Valentín Valero. Combining Fuzzy Logic and CEP Technology to Improve Air Quality in Cities. In *Computational Science - ICCS 2019 - 19th International Conference, Faro, Portugal, June 12-14, 2019, Proceedings, Part V*, volume 11540 of *Lecture Notes in Computer Science*, pages 559–565. Springer, 2019.
[9] David Luckham. *Event Processing for Business: Organizing the Real-Time Enterprise*. John Wiley & Sons, New Jersey, USA, 2012.
[10] E.H. Mamdani and S. Assilian. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 7(1):1 – 13, 1975.
[11] Javier Luengo-Oroz and Stefan Reis. Assessment of cyclists' exposure to ultrafine particles along alternative commuting routes in Edinburgh. *Atmospheric Pollution Research*, 10(4):1148–1158, 2019.
[12] Esper. http://www.espertech.com/esper/. (Accessed, March 2022).
[13] Juan Boubeta-Puig, Guadalupe Ortiz, and Inmaculada Medina-Bulo. MEdit4CEP: A model-driven solution for real-time decision making in SOA 2.0. *Knowledge-Based Systems*, 89:97–112, 2015.
[14] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965.
[15] Travis; Peterson Oliphant and Pearu Jones. SciPy - Skfuzzy, 2020.
[16] James Lyle Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1981.
[17] CPN Tools. http://www.cpntools.org/. (Accessed, March 2022).
[18] Kurt Jensen and Lars M. Kristensen. *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Springer Publishing Company, 1st edition, 2009.